Yun Qing Shi
Hyoung-Joong Kim
Fernando Perez-Gonzalez (Eds.)

LNCS 7128

Digital-Forensics and Watermarking

10th International Workshop, IWDW 2011
Atlantic City, NJ, USA, October 2011
Revised Selected Papers

Springer

Lecture Notes in Computer Science 7128

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

Yun Qing Shi Hyoung Joong Kim
Fernando Perez-Gonzalez (Eds.)

Digital-Forensics and Watermarking

10th International Workshop, IWDW 2011
Atlantic City, NJ, USA, October 23-26, 2011
Revised Selected Papers

Springer

Volume Editors

Yun Qing Shi
New Jersey Institute of Technology, NJIT
Newark, NJ 07102, USA
E-mail: shi@njit.edu

Hyoung Joong Kim
Korea University
Graduate School of Information Security, CIST
Seoul 136-701, South-Korea
E-mail: khi-@korea.ac.kr

Fernando Perez-Gonzalez
Galician Research and Development Center
in Advanced Telecommunications, GRADIANT
3610 Vigo, Spain
E-mail: fperezg@unm.edu

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-32204-4 e-ISBN 978-3-642-32205-1
DOI 10.1007/978-3-642-32205-1

Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2012943038

CR Subject Classification (1998): E.3, D.4.6, K.6.5, I.3-4, H.4, H.3

LNCS Sublibrary: SL 4 – Security and Cryptology

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

The International Workshop on Digital-Forensics and Watermarking 2011 (IWDW11), the 10$^{\text{th}}$ IWDW, hosted by the New Jersey Institute of Technology (NJIT), was held in the ACH (Atlantic City Hilton) Hotel, Atlantic City, New Jersey, USA, during October 23–26, 2011. IWDW11, following the tradition of IWDW, aimed to provide a technical program covering the state-of-the-art theoretical and practical developments in the field of digital watermarking, steganography and steganalysis, forensics and anti-forensics, and other multimedia-related security issues. With 59 submissions from 13 different countries and areas, the technical committee selected 37 papers (27 oral and 10 poster presentations) for publication, one paper for the best student paper award, and one for the best paper award. Besides these papers, the workshop featured an opening talk delivered by the Senior Vice President of NJIT Donald Sebastian; two invited lectures entitled "Modern Trends in Steganography and Steganalysis" and "Photo Forensics – There Is More to a Picture than Meets the Eye" presented, respectively, by Jessica Fridrich and Nasir Memon; and one two-hour open discussion among all participants.

First of all, we would like to thank all the authors, reviewers, lecturers, and participants for their valuable contributions to the success of IWDW11. Our sincere gratitude also goes to all the members of the Technical Program Committee, international publicity liaisons and our local volunteers for their careful and hard work in the wonderful organization of this workshop. We appreciate the generous support from the New Jersey Institute of Technology, Korea Institute of Information Security and Cryptography (KIISC), and MarkAny. Finally, we hope that you will enjoy reading this volume and that it will provide inspiration and opportunities for your future research.

December 2011

Yun Qing Shi
Hyoung Joong Kim
Fernando Perez-Gonzalez

Organization

General Chairs

Donald H. Sebastian	New Jersey Institute of Technology, USA
Heung Youl Youm	KIISC, Korea

Technical Program Chairs

Yun Q. Shi	New Jersey Institute of Technology, USA
H.J. Kim	Korea University, Korea
Fernando Pérez-González	University of Vigo, Spain

International Publicity Liaisons

Alex Kot	Nanyang Technological University, Singapore
Jiwu Huang	Sun Yat-sen University, China
Anthony TS Ho	University of Surrey, UK
Ton Kalker	Huawei, USA

Technical Program Committee

Wen Chen	Dialogic, USA
Lee-Ming Cheng	City University of Hong Kong, Hong Kong, SAR China
Jana Dittmann	University of Magdeburg, Germany
Jean-Luc Dugelay	Eurecom, France
Miroslav Goljan	Binghamton University, USA
Fangjun Huang	Sun Yat-sen University, China
Mohan Kankanhalli	National University of Singapore, Singapore
Xiangui Kang	Sun Yat-sen University, China
Darko Kirovski	Microsoft, USA
C.-C. Jay Kuo	University of South California, USA
Heung Kyu Lee	KAIST, Korea
Chang-Tsun Li	University of Warwick, UK
Zheming Lu	Zhejiang University, China
Nasir Memon	NYU-Poly, USA
Jiangqun Ni	Sun Yat-sen University, China
Zhicheng Ni	LSI, USA
Ioannis Pitas	University of Thessaloniki, Greece
Alessandro Piva	University of Florence, Italy
Youg-Man Ro	KAIST, Korea
Kouichi Sakurai	Kyushu University, Japan

Hans Georg Schaathun	Alesund U. College, Norway
Sviatoslay Voloshynovskiy	University of Geneva, Switzerland
Andreas Westfeld	HTW Dresden, Germany
Ed Wong	NYU-Poly, USA
Min Wu	University of Maryland, USA
Shijun Xiang	Jinan University, China
Weiqi Yan	Queen's University Belfast, UK
Hongbin Zhang	Beijing U. Tech, China
Xinpeng Zhang	Shanghai University, China
Yao Zhao	Beijing Jiao Tung University, China
Dekun Zou	Dialogic, USA

Table of Contents

Invited Lectures (Abstracts)

Session 1: Steganography and Steganalysis

Session 2: Watermarking

Session 3: Visual Cryptography

Session 4: Forensics

Session 5: Anti-Forensics

Session 6: Fingerprinting, Privacy, Security

Modern Trends in Steganography and Steganalysis

Jessica Fridrich

Department of Electrical and Computer Engineering
SUNY Binghamton
Binghamton, NY, USA

Abstract. Only recently, researchers working in steganography realized how much the assumptions made about the cover source and the availability of information to Alice, Bob, and the Warden influence some of the most fundamental aspects, including the way a steganographic system is built and broken and how much information can be securely embedded in a given object. While for simple artificial sources the problem of embedding messages undetectably has been resolved, it remains vastly open for empirical covers, examples of which are digital media objects, such as digital images, video, and audio. The fact that empirical media is fundamentally incognizable brings serious complications but also gives researchers plenty of opportunities to uncover very interesting and sometimes quite surprising results. An example is the square root law of imperfect steganography that states that the size of secure payload in empirical objects increases only with the square root of the cover size. Since steganographic methods designed to be undetectable with respect to a given model are usually easy to attack by going outside of the model, modern steganography works with complex models of covers in which the embedding distortion is minimized, hoping that it will be difficult for the Warden to work "outside of the model." The problem of cover source model is equally important in steganalysis. However, while working with complex models in steganography is feasible, learning a relationship between cover and stego objects in a high-dimensional model space can be quite challenging due to the rapidly increasing complexity of classifier training, lack of training data, and loss of robustness. In my talk, I will provide a retrospective view of the field, point out some of the recent achievements as well as bottlenecks of future development in source model building and machine learning.

Y.Q. Shi, H.J. Kim, and F. Perez-Gonzalez (Eds.): IWDW 2011, LNCS 7128, p. 1, 2012.

Photo Forensics – There Is More to a Picture than Meets the Eye

Nasir Memon

Polytechnic Institute of NYU
Six MetroTech Center, Brooklyn, NY

Abstract. Given an image or a video clip can you tell which camera it was taken from? Can you tell if it was manipulated? Given a camera or even a picture, can you find from the Internet all other pictures taken from the same camera? Forensics professionals all over the world are increasingly encountering such questions. Given the ease by which digital images can be created, altered, and manipulated with no obvious traces, digital image forensics has emerged as a research field with important implications for ensuring digital image credibility. This talk will provide an overview of recent developments in the field, focusing on three problems. First, collecting image evidence and reconstructing them from fragments, with or without missing pieces. This involves sophisticated file carving technology. Second, attributing the image to a source, be it a camera, a scanner, or a graphically generated picture. The process entails associating the image with a class of sources with common characteristics (device model) or matching the image to an individual source device, for example a specific camera. Third, attesting to the integrity of image data. This involves image forgery detection to determine whether an image has undergone modification or processing after being initially captured.

Y.Q. Shi, H.J. Kim, and F. Perez-Gonzalez (Eds.): IWDW 2011, LNCS 7128, p. 2, 2012.

An Improved Matrix Encoding Scheme for JPEG Steganography

Vasily Sachnev and Hyoung Joong Kim

Department of Information, Communication and Electronics Engineering,
Catholic University of Korea, Bucheon, 420-743, Korea
Graduate School of Information Security and Management,
Korea University, Seoul 136-701, Korea
bassvasys@hotmail.com, khj@korea.ac.kr

Abstract. This paper presents an efficient JPEG steganography method based on improved matrix encoding. Compared to the original matrix encoding (ME), the proposed improved matrix encoding uses two intersected ME blocks of the DCT coefficients as a single combined block. We propose a way to get a join solution of the two intersected ME, such that the intersected area does not affect the result. Due to intersection the improved matrix encoding may use a matrix encoding scheme with higher embedding rate. In order to survive steganalysis we hides data to DCT coefficients which cause the lowest distortion after modification. We used the original bitmap image for computing the distortion and getting the modified JPEG image. The proposed insert-remove strategy modifies the input stream of the DCT coefficients by inserting or removing coefficients 1 or -1. Any insertion and removing results the different solutions for the improved matrix encoding. Among all possible solutions the proposed method chooses solution with the lowest distortion. Such method significantly increases the number of possible solutions and, as a result, decreases the total distortion after data hiding. The experiments include the steganalysis of the proposed improved matrix encoding with and without using the insert-remove strategy. The experiment results shows that the proposed methods has lower detectability of the steganalysis compared to the existing steganographic methods.

Keywords: Matrix encoding, steganography, undetectable data hiding.

1 Introduction

The extreme growth of the communication technologies (i.e., internet, mobile communication) keeps attention on many aspects of information security. The important information has to be protected from any threats and malicious actions. Hence, the steganography can be a very efficient tool for achieving high level of security. One of the most important purposes of information security is to hide existence of the secret information. Here, the secret message has to be hidden to a cover signal (i.e., image, sound, or text). The modified image with hidden data has to be statistically undetectable from unmodified images. Such

Y.Q. Shi, H.J. Kim, and F. Perez-Gonzalez (Eds.): IWDW 2011, LNCS 7128, pp. 3–15, 2012.

approach enables an undetectable communication by sending both unmodified and modified images.

In our paper we will talk about JPEG steganography. One of the first steganography method for JPEG images was JSteg [1]. This method embeds data by changing the LSB values of the quantized DCT coefficients. However, this method can be easily detected by estimating the shape of the histogram of the modified DCT coefficients. Provos [15] divides the DCT coefficients into two disjoint subsets, hides data to the first subset, and compensate histagram's change by modifying the second subset. Methods presented in [2] and [12] used a similar approach. On the other hand, Solanki et. al. [21] utilized the robust watermarking scheme for steganography purposes. They embed data to image in the spatial domain by using a technique robust against JPEG compression. Their scheme provides less degradation of the features of DCT coefficients, and, as a result, less detectability.

Another way to survive against steganalysis is reducing the number of modified coefficients. Traditionally, one DCT coefficient has been used for hiding one bit of data. Westfeld [22] suggested to use a matrix encoding technique for hiding data to DCT coefficients. The matrix encoding technique is based on the Hamming code. His scheme hides more than one bit by changing at most one coefficient in a block. As a result, the matrix encoding allows hiding data with higher embedding rate.

Fridrich et. al [7] uses the concept of the "minimal distortion" to improve the security, i.e. hiding data to coefficients which cause less distortion. The proposed Perturbed Quantization (PQ) steganography utilizes the wet paper coding for hiding data. Note that the proposed method requires the original bitmap image for improving the performance of data hiding.

Later Kim et. al [11] improved the performance of the matrix encoding by modifying coefficients with less distortion impact. In fact, the proposed modified matrix encoding method (MME) changes more coefficients compared to the matrix encoding. They show that the distortion after modifying one coefficient can be higher than that after modifying two coefficients. Thus, the data hiding by modifying one or two coefficients per block may have less total distortion, that causes less detectability for steganalysis. Similar to PQ MME requires the original bitmap image for data hiding.

Schönfeld and Winkler [20] found a way to hide data using more powerful error correction code (ECC). They used structured BCH code [3] for data embedding. Later Zhang et. al [24] significantly imroved the data hiding based on BCH. Their method can easily find the flips for the BCH and acheives better detectability compared to existing methods.

Most of the above-mentioned steganographic methods uses the non overlapped blocks of the DCT coefficients for hiding portioned messages. In the proposed method one block of DCT coefficients unifies two intersected standard matrix encoding blocks. Thus, the proposed method combines the block wise embedding and new idea based on intersection. Block wise embedding divides the stream of the DCT coefficients and hidden message into the separate blocks and solves the equations for hiding data for each block individually. Using two intersected blocks increases the embedding rate by utilizing the intersected area two times. In the proposed method, the block of the DCT coefficients can be modified by inserting new nonzero coefficients 1 or -1, or removing coefficients 1 or -1. Such modification is carried out carefully and sophisticatedly in order to reduce distortion.

This paper is organized as follows. Section 2 explains the details of the existing matrix encoding schemes. Section 3 presents the proposed improved matrix encoding. In the Section 4 we propose the insert-remove strategy. The encoder and decoder are presented in the section 5. Section 6 provides the experimental results. Section 7 concludes the paper.

2 Matrix Encoding

Matrix encoding is data hiding method based on Hamming code. The (n, m, t) matrix encoding technique can hide n bits of data into $m = 2^n - 1$ binary coefficients by flipping at most t coefficients. For $t = 1$, the matrix encoding scheme becomes $(n, 2^n - 1, 1)$, $n = 2, 3, ..., k$.

Each DCT coefficients presents one binary coefficient computed as follows:

$$b_i = \begin{cases} c_i \ mod\ 2 & \text{if } c_i > 0, \\ c_i - 1 \ mod\ 2 & \text{if } c_i < 0 \end{cases} \tag{1}$$

where b_i is the corresponding bit of the nonzero DCT coefficient c_i; $b = \{b_1, b_2, b_3, ..., b_N\}$ is the stream of computed binary coefficients; N is the number of nonzero DCT coefficients.

The computed stream of binary coefficients is divides into the blocks of n coefficients. Matrix encoding hides binary message $\mathbf{m}$ ($|\mathbf{m}| = m$) to each stream $v = v_1, v_2, v_3, ..., v_n$ by modifying one coefficient in the position j such that:

$$\mathbf{m} = H \cdot r \tag{2}$$

where H is the parity-check matrix; r is the modified block of binary coefficients.

The position j is computed as follows:

$$j = (xor[(H \cdot v)_2, \mathbf{m}])_{10}, \tag{3}$$

where $xor[A, B]$ is the bitwise xor operation for the binary streams A and B ($|A| = |B|$); operations $(D)_2$ and $(E)_{10}$ convert the decimal number D into binary stream and binary stream E into decimal number, correspondingly.

Flipping the coefficient at j-th position converts the stream v to r. The flipped coefficient C is computed as follows:

$$C = \begin{cases} c \pm 1 & \text{if } c \geq 2 \ \&\ c \leq -2, \\ 2 & \text{if } c = 1, \\ -2 & \text{if } c = -1 \end{cases} \tag{4}$$

where, c is the original DCT coefficient.

Note that the matrix encoding does not provides the choice for data hiding and modifies the coefficient in the j-th position computed using Equation (3). However, this coefficient may cause significant distortion after modification. The total effect of the hiding data to the image may produce the unacceptable distortion to survive steganalysis.

Later Kim at al. [11] presented the modified matrix encoding (MME). Presented scheme considers the distortion effect and provides a choice for data hiding. Their method modifies two or three coefficients instead of one for original matrix encoding. They found that the total distortion due to modification of two or three coefficients may cause lower distortion compared to that of modifying a single coefficient for original matrix encoding. Flipping two and three coefficients increases the number of possible solutions for hiding any message m from 1 to $n/2$ and $(n/2)^2$, respectively. Such as big choice decreases the total distortion and, as a result, decreases the detectability of the steganalysis.

The set of solutions for MME is computed as follows:

$$(j)_2 = xor[(j_{21})_2, (j_{22})_2], \tag{5}$$

or

$$(j)_2 = xor[(j_{31})_2, (j_{32})_2, (j_{33})_2], \tag{6}$$

where $J'' = (j_{21}, j_{22})$, or $J''' = (j_{31}, j_{32}, j_{33})$, are the sets of solutions for modifying two J'' and three J''' coefficients.

Unlike the prior steganographic methods the MME uses the original bitmap image to get the modified JPEG image. Here, the bitmap image is used for computing the flipped distortion for each DCT coefficient. Thus, among the solutions J'' and J''' MME chooses solution with the lowest total distortion.

3 Improved Matrix Encoding

Matrix encoding and modified matrix encoding use the same standard block of the n coefficients for hiding m bits of data. We proposed the improved matrix encoding where two blocks of n coefficients unified into one block with intersected area I (see the Figure 1). In the presented example (a_1, a_2, a_3,...,a_{11}) is the combined block of the DCT coefficients; (v'_1, v'_2,...,v'_7) and (v''_1, v''_2,...,v''_7) are the corresponding binary coefficients for the blocks n_1 and n_2, respectively. By using the different I the proposed scheme enables different sizes of the combined block. Such scheme is more flexible compared to original matrix encoding and has higher embedding rate.

Here, one combined block hides two different messages m. In general, the proposed method requires to find the solution for two matrix encoding blocks n_1 and n_2 for hiding message $m = \{m_1, m_2\}$ together, like.

$$\begin{cases} m_1 = H \cdot r_1 \\ m_2 = H \cdot r_2 \end{cases} \tag{7}$$

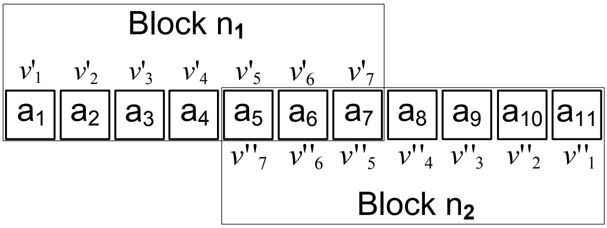

Block n1
v'1 v'2 v'3 v'4 v'5 v'6 v'7
a1 a2 a3 a4 a5 a6 a7 a8 a9 a10 a11
v''7 v''6 v''5 v''4 v''3 v''2 v''1
Block n2

Fig. 1. Two intersected blocks of the improved matrix encoding

where r_1 and r_2 are the streams of binary coefficients presented in Figure 1; H is the parity-check matrix from Equation 2.

Note that, hiding message m_1 to block n_1 modifies the block n_2 and vice versa. We utilized the modification of MME to get the proper flips to solve (7).

The proposed modification of MME unifies the solutions from J'' and J''' such that the flip positions cover only non intersected area (i.e., $j = J'', J''' \ni I$) for blocks n_1 and n_2. Such solutions for block n_1 do not affect the block n_2 and vice versa. Thus, we can get solutions for both blocks separately.

However, even if some flip positions j for the block n_1 belong to the intersected area I, we can consider the effect of those j to get solutions for the block n_2. Note that the matrix encoding has the following relationship between solutions for flipping one, two and three coefficients as: $(j)_2 = xor[(j_{21})_2, (j_{22})_2] = xor[(j_{31})_2, (j_{32})_2, (j_{33})_2]$, where j is the flip position for original matrix encoding. Assume j^I and j^{II} are the flip positions for the block n_1 and n_2 according to the Equation (7), $j^I \in I$, $j^{II} \ni I$. Note that the modifying of coefficient j^I changes the solution for block n_2. Using the Equation (5) we can get a new flip position j^{II}_{new} for the block n_2 as follows:

$$(j^{II}_{new})_2 = xor[(j^{II})_2, (j^I)_2], \tag{8}$$

where $j^{II}_{new} \ni I$.

The solution $(j^I; j^{II}_{new})$ satisfies the Equation (7). According to the Equations (5) and (6) the solution for block n_1 may have one, two or three flip positions. Thus, one, two or three flip positions may belong to the intersected area I ($J_I = j \in I$). In this case the new flip position for block n_2 is computed as follows:

for $|J_I| = 2$

$$(j^{II}_{new})_2 = xor[(j^{II})_2, xor[J_I(1)_2, J_I(2)_2]], \tag{9}$$

where $j^{II}_{new} \ni I$,

and for $|J_I| = 3$

$$(j^{II}_{new})_2 = xor[(j^{II})_2, xor[J_I(1)_2, xor[J_I(2)_2, J_I(3)_2]]], \tag{10}$$

where $j^{II}_{new} \ni I$,

The solution for block n_2 is $(j^{II}_{new}, J_I(1), J_I(2))$ or $(j^{II}_{new}, J_I(1), J_I(2), J_I(3))$ for $|J_I| = 2$ or $|J_I| = 3$, respectively. The join solution for blocks n_1 and n_2 are (j^I, j^{II}), (j^I, j^{II}_{new}), $(j^I_{21}, j^I_{22}, j^{II}_{new})$, or $(j^I_{31}, j^I_{32}, j^I_{33}, j^{II}_{new})$, for $|J_I| = 0, 1, 2, 3$. Similar, the block n_2 can be used for getting the join solutions for improved matrix encoding. Here, the solution for block n_2 are j^{II}, $(j^{II}_{21}, j^{II}_{22})$, and $(j^{II}_{31}; j^{II}_{32}; j^{II}_{33})$. The resulted join solutions are (j^I, j^{II}), $(j^{II}; j^I_{new})$, $(j^{II}_{21}, j^{II}_{22}, j^I_{new})$, or $(j^{II}_{31}, j^{II}_{32}, j^{II}_{33}, j^I_{new})$, for $|J_I| = 0, 1, 2, 3$.

The proposed scheme enables different size of the intersected area I. It is clear that the larger I frequently results situations where $j^{II}_{new} \in I$ or $j^I_{new} \in I$. Such situation means that the calculated new flip position is belonged to the intersected area I and can not be used as a flip for the new solution. Thus, the larger I the lower number of possible solutions according to the Equations (9) and (10). Hence, we have to find an appropriate size I to control the number of possible solutions.

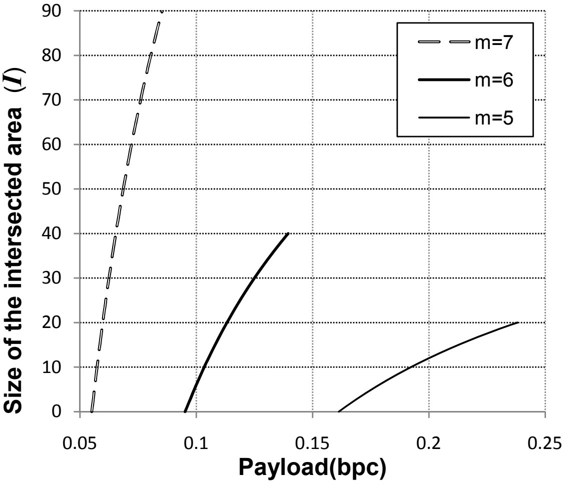


Fig. 2. The size of the intersected area I vs. payload(bpc) for different matrix encoding schemes

Due to intersection the proposed method hides two messages m to the block of $2 \cdot n - I$ coefficients. However, the matrix encoding hides the same message to the $2 \cdot n$ coefficients. Note that the lower block's size the higher embedding rate of the method. Hence, we estimate the embedding rate of the improved matrix encoding.

The embedding rate for the proposed method is computed as follows:

$$e = \frac{2 \cdot m}{2 \cdot (2^m - 1) - I}, \tag{11}$$

where I is the size of the intersected area.

According to the Equation (11) the larger size I the higher embedding rate e. However, the larger I the lower the number of possible solutions for the improved matrix encoding. Thus, we have to define the proper I such that the improved matrix encoding provides enough solutions to survive steganalysis. We tested the improved matrix encoding for different I and payloads. The results are presented in Figure 2. Presented I shows the theoretical limits for the improved matrix encoding according to the Equation (11). For example, if the necessary payload has the size 0.1 bit per coefficient (bpc), the proper I is 6. Note that using the proper I form Figure 2 guaranties the maximum efficiency for the improved matrix encoding.

However, maximum embedding rate does not guarantee the low detectability of the steganalysis. In general, steganalysis estimates the artificial changes in the specific features of the tested images. If the feature's degradation is large, the steganalysis classifies the tested image as a stego. Thus, in order to survive steganalysis the steganographic methods have to distort the image's features as low as possible. By using the large I the improved matrix encoding maximizes the embedding rate (i.e., increases the number of hidden bits per one flip), but reduces the number of possible solutions for hiding data. As a result, the proposed method may fail to find the solution with low distortion and, finally, the steganalysis may succeed to detect stego image. Thus, choosing the proper size of the intersected area I is the trade off between high embedding rate (i.e., large I) and large number of possible solutions (i.e., low I). We tested several I and found that the $I = \lfloor 0.5 \cdot n \rfloor$ is the most appropriate size of the intersected area I. Later we used this I in our experiments.

The original matrix encoding uses only portion of the DCT coefficients for data hiding. Hence, we present a new method which can utilize almost all DCT coefficients for the data hiding. The proposed method is based on using two different schemes together. Two schemes uses the different block size n_1^p and n_2^p, and have different payloads m_2^p and m_2^p. This method divides the stream of DCT coefficient $(c_1, c_2, ..., c_N)$ and the message M into two parts and hides data to each part separately. The optimal number of the blocks (k_1 and k_2) for the both schemes can be computed as follows:

The relation between the numbers of blocks for the scheme 1 and 2 is presented as follows:

$$\begin{cases} n_1^p \cdot k_1' + n_2^p \cdot k_2' = N \\ m_1^p \cdot k_1' + m_2^p \cdot k_2' = |M|, \end{cases} \tag{12}$$

where N is the number of DCT coefficients.

The computed k_1' and k_2' are the non integer. Thus, we have to choose the nearest integers $k_1 = \lfloor k_1' \rfloor \pm 1$ and $k_1 = \lfloor k_1' \rfloor \pm 1$ such that:

$$\begin{cases} n_1^p \cdot k_1 + n_2^p \cdot k_2 \leq N \\ m_1^p \cdot k_1 + m_2^p \cdot k_2 \geq |M|, \end{cases} \tag{13}$$

4 Insert-Remove Strategy

The performance of the improved matrix encoding can be significantly increased by using a new insert-remove strategy. The proposed strategy is based on fact that the input stream of the DCT coefficients can be modified before data hiding by inserting or removing coefficients 1 and -1. Data hiding to modified stream of DCT coefficients may result lower distortion and, as a result, lower detectability of the steganalysis.

The proposed insert-remove strategy uses the stream of non rounded quantized DCT coefficients a_q computed as follows:

$$a' = DCT(B), \quad a_q = \frac{a'}{Q}, \quad a_r = round(a_q), \tag{14}$$

where B is the 8 by 8 block of the image pixels; a' is the block of non-quantized non-rounded DCT coefficients; a_q is the block of quantized non-rounded DCT coefficients; a_r is the block of quantized rounded DCT coefficients; Q is the quantization matrix according to the quality factor Q_f.

According to the proposed insert-remove strategy the stream a of quantized non rounded coefficients builded from the blocks a_q is divided into the three sets: modifiable $c_m = a \in (-\infty; -1.5) \cup (1.5; \infty)$, removable $c_R = a \in [-1.5; -0.5) \cup (0.5; 1.5]$, and insertable $c_I = a \in [-0.5; -0.25) \cup (0.25; 0.5]$. The set c unifies modifiable, insertable and removable sets (i.e., $c = c_m \cup c_R \cup c_I$). The set $C = c_m \cup c_R$ contains all nonzero rounded DCT coefficients. According to the Equation 1 only nonzero rounded DCT coefficients (i.e., set C) have the corresponding informative bits and are used for data hiding. The proposed improved matrix encoding uses the blocks of n^p nonzero DCT coefficients from the set C for data hiding. In general, set C is the subset of the unified set of coefficients c. Thus, each block c_b unifies the n^p coefficients form the set C and some insertable coefficients from the set c (i.e., $c_b = c'_m \cup c'_R \cup c'_I$, where $C' = c'_m \cup c'_R$ is the block of n^p non zero DCT coefficients from the set C). Inserting or removing of any coefficients from c'_I and c'_R produces a new block C' with a new solution for data hiding. As a result, the proposed insert-remove strategy significantly increases the number of possible solutions and helps to find the most appropriate solution with the lowest distortion.

In the proposed improved matrix encoding we used the method for computing distortion similar to MME [11]. The distortion for each DCT coefficient is computed as follows:

$$D = E^2 \cdot Q^2, \tag{15}$$

$$E = \begin{cases} 0.5 - |C - \lfloor C \rfloor|, & \text{if } C \in c_m, \\ 1.5 - |C|, & \text{if } C \in c_R. \end{cases}$$

The distortion due to inserting or removing D_{IR} is computed as follows:

$$D_{IR} = |0.5 - |C||^2 \cdot Q^2, \text{if } C \in c_R \cup c_I. \tag{16}$$

where Q is the corresponding quantization coefficient of the quantization table.

The resulted distortion for the block of DCT coefficients is computed as follows:

$$D_b = \sum_{i=1}^{l} D_i + D_{IR} \tag{17}$$

where l is number of flipped coefficients.

5 Encoder and Decoder

The encoder of the proposed steganographic method based on improved matrix encoding and insert-remove strategy is organized as follows:

For the given bitmap image Im, payload P, quality factor Q_f and secret key K process:

1) Divide image Im into non-overlapped 8 by 8 blocks of pixels and process DCT, quantization and rounding as presented in (14). Remove DC coefficients. Obtain blocks a', a_q, a_r, and streams of DCT coefficients a. Permute stream a using K and any pseudo-random generator. Obtain the stream c from the permuted stream a.
2) Define sets: modifiable c_m, insertable c_I and removable c_R.
3) Define the schemes 1 and 2, and the number of the blocks k_1, k_2 using (12) and (13). Divide the payload P into two parts according to the guidelines in the Section 3.
4) Define the i-th block of the DCT coefficients $c_{b_i} = c'_{m_i} \cup c'_{R_i} \cup c'_{I_i}$, where c'_{m_i}, c'_{R_i}, and c'_{I_i} are the modifiable, removable and insertable subsets for the current block. Start from the first block $i = 1$. If $i = k_1 + 1$ switch to the scheme 2.
5) Define the block of non-zero rounded DCT coefficients $C'_i = c'_{m_i} \cup c'_{R_i}$.
6) Get the solutions for the block C'_i using improved matrix encoding. Compute the distortion D for each solution using Equation (17). Choose solution J_m with the lowest distortion D_m and store it.
7) Modify the block C'_i by inserting or removing coefficients from the sets c'_{R_i}, and c'_{I_i}. Obtain a new block: r) after removing $C'_i = c'_{m_i} \cup c''_{R_i}$, where $c''_{R_i} = c'_{R_i} - c'_{R_i}(j)$ is the modified removable set, $c'_{R_i}(p)$ is the removed coefficient; i) after inserting $C'_i = c'_{m_i} \cup c'_{R_i} \cup c'_{I_i}(q)$, where $c'_{I_i}(q) = \pm 1$ is the inserted coefficient. p and q is the current position for insertion and removing.
8) Repeat steps 5 - 6 for all insertable and removable coefficients from c'_{R_i}, and c'_{I_i}.
9) Choose solution among J_m with the lowest distortion D_m. According to the best solution modify one, two ,three or four coefficients (see explanation in the Section 3) and, if necessary, insert or remove coefficient in the block c'_{b_i}.
10) Process all $k_1 + k_2$ blocks using steps 4 - 9. Obtain the modified stream $c' = \{c'_{b_1}, c'_{b_2}, ..., c'_{b_{k_1+k_2}}\}$

11) Recover the original sequence of the DCT coefficients a' from the modified stream c' using the secret key K and utilized pseudo-random generator. Add DC coefficients, round the coefficients a' and obtain the modified JPEG image Im'.

The decoder of the proposed steganographic method is organized as follows:

For the given modified JPEG image Im', quality factor Q_f, secret key K, and respected size of the payload $p = |P|$ process:

1) Read the DCT coefficients from the JPEG file. Permute them using the secret key K and utilized pseudo-random generator. Remove the DC coefficients. Obtain the stream of nonzero DCT coefficients C.
2) Using the 12 and 13 define the scheme 1 and 2, and the number of blocks k_1 and k_2. Here, $N = |C|$.
3) Divide C into the blocks according to the k_1 and k_2.
4) Decode data from each block using (7).

The steganographic method based only on improved matrix encoding skips the steps 7 and 8.

6 Experimental Results

The experiments included the hiding different payloads to the set of bitmap images using the proposed improved matrix encoding with and without insert-remove strategy. The set of modified and original compressed images was analyzed by powerful steganalysis algorithm proposed by Pevny and Fridrich [13], [14]. Their method uses 274 different features of the DCT coefficients and deeply investigates artificial changes in the tested images. The union of the 274 features from the unmodified and modified images were used for making the model in the support vector machine. The parameters of SVM are different for different sets of tested images (in average $C = 40000$, and $\varepsilon = 0.0004$).

A set of 4000 test images (768 × 1024) distributed by CorelDraw and taken from several different cameras was used in our experiments. Experiments were carried out for 5 different payloads (0.05, 0.1, 0.15, 0.20 and 0.25 bits per coefficient (bpc)) and quality factor 75. We used the model adapted to quality factor 75 and tested five sets of the test images for five types of payload. Each training set had 1500 cover and 1500 stego images. The rest 1000 images were used for testing. The result shows the error probabilities of the steganalysis for each set of the stego images (see Figure 3).

The error probability is computed as follows:

$$e = \frac{1}{2}(P_a + P_b), \tag{18}$$

where P_a is the probability of misdetection (i.e., the unmodified image is classified as modified) and P_b is the probability of misclassification (i.e., the modified image is classified as unmodified). In our experiments we tested both proposed methods: 1) based only on improved matrix encoding; and 2) improved matrix encoding combined with the proposed insert-remove strategy. The proposed

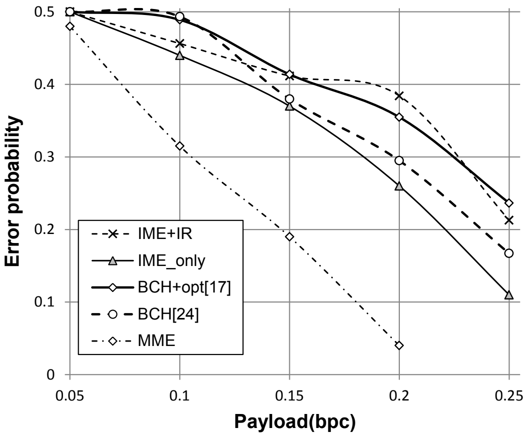


Fig. 3. Error probability vs. payload (bpc) for quality factor 75

methods achieve high error probability for all tested payloads. For payloads up to 10% both methods have detectability close to 50 percent. That means the steganalysis cannot distinguish the unmodified images from modified. Note that methods based on matrix encoding (MME, proposed improved matrix encoding IME, and IME plus insert-remove strategy IME+IR) have slightly higher detectability compared to methods based on BCH (see Figure 3). This probability is almost equal to that of the coin tossing. For higher payloads around 15 to 20 percents the proposed methods shows much better performance compared with MME. Significant improvement over the MME is justified on the fact of using ME schemes with higher embedding rate. Hence, the proposed method with insert-remove strategy shows the significant improvement over the method with improved matrix encoding only (see Figure 3). For payload of 25 percent, both methods shows 0.11 and 0.211 of the error probability, respectively. The error probabilities are significantly better than those of the MME [11]. BCH based steganography [17] and [24] shows slightly better performance for capacities (up to 0.1), and considerably lower detectability for higher capacities (except 0.25 bpc). For example, proposed IME+IR shows 0.03 lower detectability for capacity 0.2 bpc in terms of error probability points compared to [17].

7 Conclusion

In this paper we present an efficient data hiding technique for steganography. Compared with MME, the proposed modification of matrix encoding (i.e., improved matrix encoding) achieves different size of the combined block, and, as a results, enables to choose the ME schemes with higher embedding rate. As a result, the proposed improved matrix encoding significantly outperforms the MME in terms of points of the error probabilities. Using two different embedding schemes (see Equations (12) and (13)) enables to use almost all available DCT coefficients. The proposed strategy based on inserting and removing coefficients 1 or -1 increases the number of possible solutions and significantly decreases the total distortion. The experimental results show that the insert-remove strategy improves performance significantly. The combination of the IME and the proposed insert-remove strategy achieves high error probability against powerful steganalysis. This paper also shows that the idea of using two overlapped blocks can improve the performance of steganography further.

Acknowledgment. This work was in part supported by Catholic University of Korea, National Research Foundation of Korea (project 2011-0013695), ITRC and BK21 Project, Korea University and IT R&D program (Development of anonymity-based u-knowledge security technology, 2007-S001-01).

References

1. Upham, D.: `http://www.funet.fi/pub/crypt/stegangraphy/jpeg-jsteg-v4.diff.gz`
2. Eggers, J., Bauml, R., Girod, B.: A communications approach to steganography. In: Proc. of EI SPIE, San Jose, CA, vol. 4675, pp. 26–37 (2002)
3. Chien, R.T.: Cyclic decoding produce for the Bose-Chaudhuri-Hocquenghem codes. IEEE Transactions on Information Theory 11, 549–557 (1965)
4. Fridrich, J.: Minimizing the embedding impact in steganography. In: Proc. of ACM Multimedia and Security Workshop, Geneva, Switzerland, September 26-27, pp. 2–10 (2006)
5. 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)
6. Fridrich, J., Filler, T.: Practical methods for minimizing embedding impact in steganography. In: Proc. EI SPIE, San Jose, CA, vol. 6505, pp. 2–3 (2007)
7. Fridrich, J., Goljan, M., Soukal, D.: Perturbed quantization steganography using wet paper codes. In: Proc. of ACM Workshop on Multimedia and Security, Magdeburg, Germany, September 20-21, pp. 4–15 (2004)
8. Fridrich, J., Pevny, T., Kodovsky, J.: Statistically undetectable JPEG steganography: dead ends, challenges, and opportunities. In: Proc. of ACM Workshop on Multimedia and Security, Dallas, Texas, September 20-21, pp. 3–15 (2007)
9. Fridrich, J., Goljan, M., Soukal, D.: Perturbet quantization steganography. ACM Multimedia and Security Journal 11(2), 98–107 (2005)
10. Fridrich, J., Goljan, M., Soukal, D.: Wet paper coding with improved embedding efficiency. IEEE Transactions on Information Security and Forensics 1(1), 102–110 (2005)
11. Kim, Y.H., Duric, Z., Richards, D.: Modified Matrix Encoding Technique for Minimal Distortion Steganography. In: Camenisch, J.L., Collberg, C.S., Johnson, N.F., Sallee, P. (eds.) IH 2006. LNCS, vol. 4437, pp. 314–327. Springer, Heidelberg (2007)
12. Noda, H., Niimi, M., Kawaguchi, E.: Application of QIM with dead zone for histogram preserving JPEG steeganography. In: Proc. of ICIP, Genova, Italy (2005)
13. Pevny, T., Fridrich, J.: Multiclass blind steganalysis for JPEG images. In: Proc. of SPIE, San Jose, CA, January 16-19, vol. 6072, pp. 257–269 (2006)

14. Pevny, T., Fridrich, J.: Merging Markov and DCT features for multi-class JPEG steganalysis. In: Proc. of SPIE, San Jose, CA, vol. 6505, pp. 3–4 (2007)
15. Provos, N.: Defending against statistical steganalysis. In: Proc. of 10th USENIX Security Symposium, pp. 24–24 (2001)
16. Sachnev, V., Kim, H.J., Zhang, R., Choi, Y.S.: A novel approach for JPEG steganography. In: Proc. of 7nd International Workshop on Digital Watermarking 2008, Busan, Korea, November 10-12, pp. 216–226 (2008)
17. Sachnev, V., Kim, H.J., Zhang, R.: Less detectable JPEG steganography method based on heuristic optimization and BCH syndrome coding. In: Proc. of ACM Workshop on Multimedia and Security, Princeton, NJ, September 7 - 8, pp. 131–139 (2009)
18. 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)
19. Schnfeld, D., Winkler, A.: Embedding with syndrome coding based on BCH codes. In: Proc. of ACM Workshop on Multimedia and Security, pp. 214–223 (2006)
20. Schönfeld, D., Winkler, A.: Reducing the Complexity of Syndrome Coding for Embedding. In: Furon, T., Cayre, F., Doërr, G., Bas, P. (eds.) IH 2007. LNCS, vol. 4567, pp. 145–158. Springer, Heidelberg (2008)
21. Solanki, K., Sarkar, A., Manjunath, B.S.: YASS: Yet Another Steganographic Scheme That Resists Blind Steganalysis. In: Furon, T., Cayre, F., Doërr, G., Bas, P. (eds.) IH 2007. LNCS, vol. 4567, pp. 16–31. Springer, Heidelberg (2008)
22. 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)
23. Zhao, Z., Wu, F., Yu, S., Zhou, J.: A lookup table based fast algorithm for finding roots of quadratic or cubic polynomials in the $GF(2^m)$. Journal of Huazhong University of Science and Technology (Nature Science Edition) 33(1) (2005)
24. Zhang, R., Sachnev, V., Kim, H.J.: Fast BCH Syndrome Coding for Steganography. In: Katzenbeisser, S., Sadeghi, A.-R. (eds.) IH 2009. LNCS, vol. 5806, pp. 48–58. Springer, Heidelberg (2009)

Steganalysis of LSB Matching Revisited for Consecutive Pixels Using B-Spline Functions

Shunquan Tan*

School of Computer Science and Software Engineering
Shenzhen University, Shenzhen, China, 518060

Abstract. Least significant bit matching revisited steganography (LSBMR) is a significant improvement of the well-known least significant bit matching algorithm. In this paper, we point out that LSBMR for consecutive pixels and its descendants, including the edge adaptive image steganography based on LSBMR, introduces intrinsic statistical imbalance in secret data embedding process, which results in the imbalance of the power of the additive stegonoise. This intrinsic imbalance can be used to construct a dimensionless discriminator using B-spline smoothing. Experimental results show that the proposed steganalytic method is a reliable detector against LSBMR for consecutive pixels and the edge adaptive image steganography based on LSBMR when block size is 1. An embedding rate estimator based on B-spline functions which can roughly estimate the embedding rate is proposed as well.

Keywords: LSB matching for consecutive pixels, B-spline smoothing, steganalysis, steganography, media security.

1 Introduction

Least significant bit matching steganography[11], also known as ±1 steganography, is a tough target for steganalyzers. In [3], Harmsen and Pearlman pointed out that LSB matching is equivalent to a lowpass filtering of the histogram of the cover image, which can be quantified by a decrease in the center of mass (COM) of the histogram characteristic function (HCF). Using this property, the authors proposed a reliable discriminator for RGB color images. In [5], Ker indicated that the HCF COM method in [3] is ineffective on gray-scale images and introduced two ways of applying the HCF COM method: calibrating using a down-sampled image and computing the adjacency histogram instead of the usual histogram. Experiments show that Ker's methods are reliable for gray-scale images subject to harsh JPEG compression and can get fair results for uncompressed gray-scale images when embedding rate is 100%. In [7], Li et al. further improved the two detectors proposed in [5] and apply them on the difference image, which is defined as the difference of the adjacent pixels of an image (Referred as Li-1D

* This work is supported by National Natural Science Foundation of China (Project 61103174), Foundation for Distinguished Young Talents in Higher Education of Guangdong (Grant No. LYM10116), and SZU R/D Fund (Grant No. 201048).

Y.Q. Shi, H.J. Kim, and F. Perez-Gonzalez (Eds.): IWDW 2011, LNCS 7128, pp. 16–29, 2012.

in this paper). The new detectors outperform Ker's methods and achieve acceptable accuracy at an embedding rate of 50%. Other notable works include the steganalytic method proposed by Huang et al.[4] based on the alteration rate of the number of neighborhood pixel values whose efficiency is comparable to Li-1D, the steganalytic method proposed by Zhang et al. for JPEG decompressed images[17], and the method proposed by Wong et al.[14] for estimating the embedding rate of ±1 steganography which has only shown success in certain image dataset. Besides the specific detectors, some universal steganalytic algorithms such as [16] and [2] can also be used to attack LSB matching steganography with a relatively high detecting rate. However, recent studies show that no detectors for LSB matching have yet proven universally reliable and their performances heavily depend on types of images[1].

The least significant bit matching revisited algorithm (LSBMR)[9] proposed by Mielikainen is a significant improvement of the LSB matching steganography. Using a pair of pixels as an embedding unit, the scheme reduces the expected number of modifications per pixel from 0.5 to 0.375 when the embedding rate is 1, which means that it can show better resistance than LSB matching for steganalysis while the payload holds. LSBMR has triggered a great deal of further researches on pair-wise LSB matching[15,6]. One of the recent important achievements in this field is the edge adaptive image steganography based on LSB matching revisited (EALSBMR for short in this paper) proposed by Luo et al.[8]. EALSBMR selects the embedding regions according to the size of secret message and the difference between two consecutive pixels in the cover image. Using this way, edge regions are adaptively used to embed secret bits and thus the security is significantly enhanced. Regardless of the progress of pair-wise LSB matching steganography, most of the researchers only view LSBMR and its descendants as variants of LSB matching. So according to our best knowledge, there is still no report on target steganalysis against LSBMR and its descendants.

There are different pixel pair selection schemes for LSBMR. One of which adopted by EALSBMR divides the cover image into non-overlapping embedding units with every two consecutive pixels. In this paper, we point out that LSBMR for consecutive pixels (LSBMRCP for short) introduces intrinsic statistical imbalance which can be used to attack it and its descendants, including EALSBMR. The paper is organized as follows. Section 2 gives a brief overview of LSBMRCP and EALSBMR. Section 3 shows the details of the target steganalyzer proposed by the author. Section 4 presents experimental results. Finally, concluding remarks and future work are given in Section 5.

2 Overview of LSBMR for Consecutive Pixels and EALSBMR

LSBMRCP firstly divides a cover image of size of $m \times n$ into a serial $\mathbf{I}$ of non-overlapping embedding units with every two consecutive pixels (x_i, x_{i+1}), where $i = 1, 3, \ldots, mn-1$, assuming n is an even number. The secret message is also divided into a serial $\mathbf{M}$ of two consecutive message bits (m_i, m_{i+1}). After message embedding, an embedding unit (x_i, x_{i+1}) is modified as (x'_i, x'_{i+1}) in the stego image. The value of the ith message bit m_i is equal to the LSB of x'_i, and the value of the $i+1$th message bit m_{i+1} is equal to the value of a binary function of x'_i and x'_{i+1}:

$$f(x'_i, x'_{i+1}) = \mathrm{LSB}(\lfloor x'_i/2 \rfloor + x'_{i+1}) \tag{1}$$

Eq. (1) has the following properties:

$$f(x_i - 1, x_{i+1}) \neq f(x_i + 1, x_{i+1}), \qquad \forall x_i, x_{i+1} \in \mathbf{Z} \tag{2}$$
$$f(x_i, x_{i+1}) \neq f(x_i, x_{i+1} + 1), \qquad \forall x_i, x_{i+1} \in \mathbf{Z} \tag{3}$$

The properties (2) and (3) guarantee that both an increase and a decrease of x_i or x_{i+1} by one will change the value of Eq. (1). Therefore by applying ±1 operation to x_i or x_{i+1}, $f(x_i, x_{i+1})$ can be set to the desired value. Each bit pair of **M** is embedded in a given pixel pair of **I** chosen in the order determined by the same pseudo-random sequence generator as in LSB matching [11]. The embedding algorithm for a pixel pair is presented in Fig. 1. Saturated pixels, i.e., pixels that have either a minimal or maximal gray-scale value are bypassed in the embedding process.

```
if $m_i = \mathrm{LSB}(x_i)$ then                              ▷ $x_i$ remains untouched
    if $m_{i+1} \neq f(x_i, x_{i+1})$ then
        $x'_{i+1} = x_{i+1} \pm 1$                             ▷ $x_{i+1}$ is modified
    else
        $x'_{i+1} = x_{i+1}$                                   ▷ $x_{i+1}$ remains untouched
    end if
    $x'_i = x_i$
else                                                           ▷ $x_i$ is modified
    if $m_{i+1} = f(x_i - 1, x_{i+1})$ then
        $x'_i = x_i - 1$
    else
        $x'_i = x_i + 1$
    end if
    $x'_{i+1} = x_{i+1}$                                       ▷ $x_{i+1}$ remains untouched
end if
```

Fig. 1. LSBMR embedding algorithm for a pixel pair

EALSBMR is a region adaptive spatial domain LSB steganography. It uses the absolute difference between two adjacent pixels as the criterion for region selection, and adopt LSBMRCP as the data hiding algorithm. The threshold T used in region selection for a given secret message M can be determined as follows. Let V be the set of consecutive pixels, $EU(t)$ be the set of pixel pairs whose absolute differences are greater than or equal to a parameter t:

$$EU(t) = \{(x_i, x_{i+1}) \big| \, |x_i - x_{i+1}| \geq t, \forall (x_i, x_{i+1}) \in V\} \tag{4}$$

Then the threshold T can be calculated by:

$$T = \underset{t}{\operatorname{argmax}}\{2 \times |EU(t)| \geq |M|\} \tag{5}$$

where $t \in \{0, 1, \ldots, 31\}$. $|EU(t)|$ denotes the number of the pixel pairs in $EU(t)$, and $|M|$ denotes the number of the bits in M. Secret bits are embedded in the pixel pairs of $EU(T)$ using LSBMR algorithm. In order to construct V, the cover image of size of $m \times n$ is first divided into non-overlapping blocks of $Bz \times Bz$ pixels. The block size Bz is an embedding parameter and can be set to 1, 4, 8 or 12. When $Bz > 1$, rotation with a random degree in the range of $\{0, 90, 180, 270\}$ is applied on each block to improve the security. The resulting image is further divided into non-overlapping embedding units with every two consecutive pixels via raster scanning, which compose of the set V. It can be noted that when $Bz = 1$, the rotation operation is bypassed and the consecutive pixels are directly picked up from the cover image.

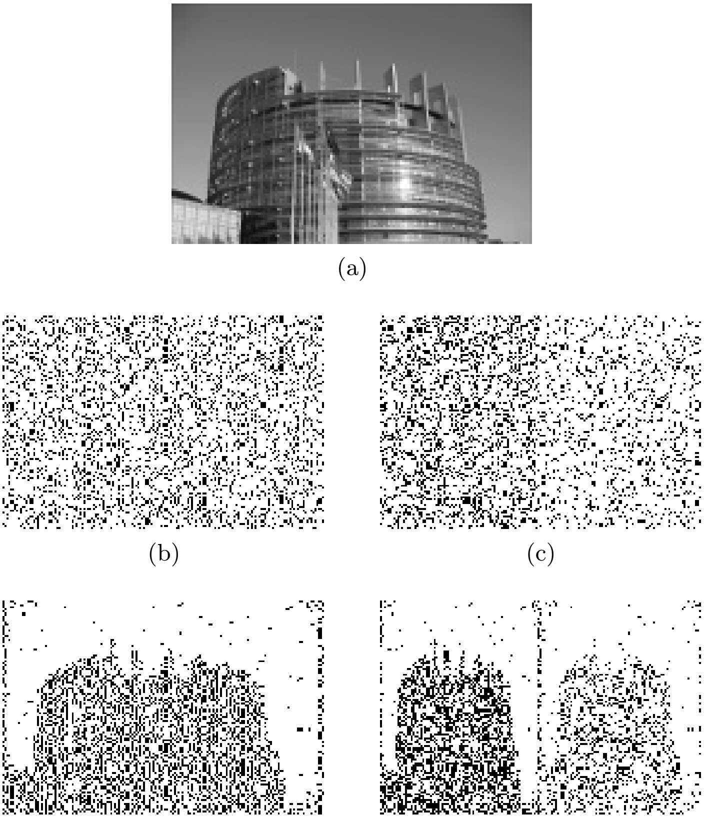
3 Steganalyzing the LSBMR Algorithm for Consecutive Pixels

3.1 Effect of LSBMRCP Embedding

Refer back to the embedding algorithm mentioned in Fig. 1. The probability that x_i, the first pixel of the pixel pair (x_i, x_{i+1}) get modified is 0.5, since it will get altered as long as m_i is not equal to the LSB of x_i. However, the probability that the second pixel x_{i+1} get modified is 0.25 (half of the probability that x_i is modified), since it only get altered when the two prior conditions $m_i = \text{LSB}(x_i)$ (with probability 0.5) and $m_{i+1} \neq f(x_i, x_{i+1})$ (with probability 0.5) are true, as described in lines 1- 6 of the algorithm.

An illustration of the imbalance is shown in Fig. 2. Fig. 2b shows the modification matrix D between the cover image Fig. 2a and its corresponding LSBMRCP stego image (embedding rate: 50%). It seems that The modified pixels are uniformly scattered around the pixel plane. In order to demonstrate the imbalance introduced by LSBMRCP, the serial of the embedding units $\{(x_i, x_{i+1})\}$ is divided into two non-intersect sub-serial $\{x_i\}$ and $\{x_{i+1}\}$. Denote the corresponding values of D for $\{x_i\}$ and $\{x_{i+1}\}$ by D_1 and D_2. Then a pseudo difference image Fig. 2c, i.e., a re-arranged version of Fig. 2b is constructed, whose left half part is a $m \times \frac{n}{2}$ array row-wise reshaped from D_1, and right half part is the one reshaped from D_2. It is clear from Fig. 2c that there are much more modified pixels in $\{x_i\}$ than in $\{x_{i+1}\}$ for a given embedding rate. Fig. 2d and Fig. 2e represent the corresponding modification matrix and its re-arranged version between fig. 2a and its EALSBMR stego image (embedding rate: 50%, $T = 2$, $Bz = 1$). The same imbalance can also be found for this edge adaptive scheme when $Bz = 1$.

Harmsen and Pearlman pointed out that secret data embedding process can be considered as an external force which corrupts the image[3]. That is to say, the



(a)

(b)

(c)

(d)

(e)

Fig. 2. (a) Cover image. (b) Modifications between the cover image and the corresponding LSBMRCP stego image. The black dots denote the pixels of the cover image get modified during the embedding process in the corresponding positions, while the white dots denote the pixels untouched. (c) The re-arranged version of Fig. 2b. (d) Modifications between the cover image and the EALSBMR stego image. (e) The re-arranged version of Fig. 2d.

embedding process can be modeled as the addition of additive noise (stegonoise) to the cover image. The more pixels get modified, the more the power of the additive stegonoise is added to the cover image. In this paper, The power of the stegonoise is referred to as the intensity factor of the embedding process applied to a pixel series. It can be concluded that the intensity factor in $\{x_i\}$ should be larger than that in $\{x_{i+1}\}$.

3.2 Intensity Factor Estimation Using B-Spline Functions

Let $\{y_i\}, i = 0, 1, \ldots, n,\ y_i \in \mathbf{Z}$ be a serial of pixels in a cover image, and $\{y_i'\}, i = 0, 1, \ldots, n, y_i' \in \mathbf{Z}$ be a serial of corresponding pixels in the stego image generated from that cover image by LSBMRCP. From the deduction above, we know that each pixel in $\{y'_i\}$ is the cover image's pixel y_i plus a stegonoise:

$$y'_i = y_i + \varepsilon_i, i = 0, 1, \ldots, n, \ \varepsilon_i \in \mathbf{Z} \tag{6}$$

The intensity factor of the embedding process for a given stegonoise series $\{\varepsilon_i\}$ is defined as its $\mathscr{L}^2$ norm:

$$\mathcal{IF} \triangleq \|\{\varepsilon_i\}\| = (\sum_{i=0}^{n}(\varepsilon_i^2))^{\frac{1}{2}} \tag{7}$$

Most of the time, what an attacker can get are some suspected stego images. Accurate estimation of intensity factor depends on the good estimation of the stegonoise series $\{\varepsilon_i\}$, and at last from (6) we can see that it depends on the performance of the calibration process, which attempts to estimate the original pixel series $\{y_i\}$ from the suspected stego one. Suppose $\{y_i\}$ is an equally-spaced sample series of an unknown function $y(t)$ on $[0, 1]$. Without loss of generality, let the approximation of $y(t)$ be a polynomial spline $g(t)$ of order m in the Sobolev space $W_2^{(m-1)}$:

$$y_i = y(t_i) \approx g(t_i), i = 0, 1, \ldots, n, t_0 = 0, t_n = 1, t_{i+1} - t_i = \frac{1}{n} \tag{8}$$

$g(t)$ is a real-valued function with absolutely continuous m-1st derivative and square integrable m-th derivative and can be constructed from a weighted sum of shifted B-splines and is uniquely characterized by the discrete sequence of spline coefficients $\{g_k\}$ [12]:

$$g(t) = \sum_{k=0}^{n} g_k \beta^m(x - k) \tag{9}$$

Where $\beta^m(x)$ is the symmetrical B-spline of order m:

$$\beta^m(x) = \sum_{j=0}^{m+1} \frac{(-1)^j}{m!} \binom{m+1}{j} (x + \frac{m+1}{2} - j)^m \mu(x + \frac{m+1}{2} - j) \tag{10}$$

and where $\mu(x)$ is the unit step function:

$$\mu(x) = \begin{cases} 0, x < 0 \\ 1, x \geq 0 \end{cases} \tag{11}$$

$g(t)$ precisely interpolates $\{y'_i\}$ if the impact of $\{\varepsilon_i\}$ is neglected. Otherwise $g(t_i)$ can be regarded as a smoothing representation of $\{y'_i\}$. The extent of smoothing is controlled by the following two functionals:

$$J(g) = \int_0^1 (g^{(m)}(u))^2 \mathrm{d}u, \tag{12}$$

$$R(g) = \frac{1}{n} \sum_{i=0}^{n} (g(t_i) - y'_i)^2, \quad t_i \in [0, 1] \tag{13}$$

The function $g(t)$ to be constructed shall:

(A) minimize $R(g) + \lambda J(g)$, for a given $\lambda, 0 < \lambda < \infty$,
(B) minimize $J(g)$ with the constraint $R(g) \leq S$, for a given S, $0 \leq S < \infty$

There exists a unique λ which depends on the selection of intuitive parameter S so that the solution to (A) is also the solution to (B). When $m = 2$, the solution of (A) and (B) for a given λ is a cubic B-spline [10,12]. S, being $1/n$ times the apparent residual sum of squares after smoothing is performed, establishes a sort of compromise between the desire for an approximation that is reasonably close to the data and the requirement of a function that is sufficiently smooth. Wahba gives a theoretical upper bound for S [13]:

$$S \leq \sigma^2\{1 - k[1 + o(1)]\} \tag{14}$$

where σ^2 is the variance of $\{\varepsilon_i\}$, $o(1) \to 0$ as $n \to \infty$. k and η can be determined using the following equations when $m = 2$:

$$k = \eta \left(\frac{\|y^{(4)}\|^2}{\sigma^2}\right)^{\frac{1}{9}} n^{-\frac{8}{9}} \tag{15}$$

$$\begin{aligned} \eta &= \left(\frac{1}{8\pi}\int_0^{\infty} \frac{du}{(1+u^4)^2}\right)^{\frac{8}{9}} \cdot \frac{37}{3} \\ &= \left(\frac{3\sqrt{2}}{128}\right)^{\frac{8}{9}} \cdot \frac{37}{3} \end{aligned} \tag{16}$$

From equations (14)- (16) we can see that the theoretical upper bound for S depends on the variance of stegonoise σ^2, the length of the pixel serial of a suspected image n and the $\mathscr{L}_2$-norm of the fourth derivative of $y(t)$:

$$\|y^{(4)}\| = \Big(\int_0^1 (y^{(4)}(u))^2 du\Big)^{\frac{1}{2}} \tag{17}$$

Unfortunately, although σ^2 can be estimated for a given image, there is usually not a computable way to get the value of $\|y^{(4)}\|$. But in the context of this paper, the special form of the unknown function $y(t)$ is trival and the power of the stegonoise $\{\varepsilon_i\}$ is relatively small. So a quartic B-spline $y_b(t)$ which interpolates the sample series $\{y_i'\}$ can be used to approximately represent $y(t)$. A quartic B-spline is a piecewise polynomial of order 4 and after the fourth derivative it turns to a piecewise constant function $\sum_{i=1}^{n} c_i \chi_{[t_{i-1},t_i]}(t)$ where c_i is a constant, $\chi_{[t_{i-1},t_i]}(t)$ is an indicator function on interval $[t_{i-1}, t_i]$. Its $\mathscr{L}_2$-norm is easy to calculate:

$$\|y^{(4)}\| \approx \|y_b^{(4)}\| = \Big(\frac{1}{n}\sum_{i=1}^{n} c_i^2\Big)^{\frac{1}{2}} \tag{18}$$

3.3 Steganalytic Feature and Estimation of Embedding Rate

Given a target image, let $\mathcal{IF}_1$ denotes the intensity factor of the sub-serial $\{x_i\}$ of the embedding units, $\mathcal{IF}_2$ denotes the intensity factor of $\{x_{i+1}\}$. Our observation is that the power of the noise introduced during the image capture and post-processing procedure is usually distributed evenly over the spatial domain. Thus it can be concluded that $\mathcal{IF}_1 \approx \mathcal{IF}_2$ is held for a normal cover image. On the other hand, as the deduction in Sect. 3.1 indicated, $\mathcal{IF}_1 > \mathcal{IF}_2$ is held for a stego image generated by LSBMRCP. In view of the variation between the magnitudes of corresponding values, $\mathcal{IF}_1/\mathcal{IF}_2$ is used as a dimensionless discriminator for the presence of LSBMRCP steganography. The following assertion is the core of the proposed steganalytic algorithm:

$$\mathcal{IF}_1/\mathcal{IF}_2 \approx 1 \qquad for\ a\ cover\ image, \tag{19}$$

$$\mathcal{IF}_1/\mathcal{IF}_2 > 1 \qquad for\ a\ LSBMRCP\ stego\ image. \tag{20}$$

Fig. 3 shows the values of $\mathcal{IF}_1/\mathcal{IF}_2$ for 200 gray-scale images randomly selected from NJIT image dataset before and after data embedding, in which the symbols '○' and '×' stand for that of the cover images and the stego images. The stego images in left half and right half are generated by LSBMRCP and EALSBMR with Bz=1 respectively. It can be seen that the values of $\mathcal{IF}_1/\mathcal{IF}_2$ for most of the cover images concentrate around 1. However, the corresponding values of $\mathcal{IF}_1/\mathcal{IF}_2$ for the stego images spread above the horizontal line at 1. We can clearly distinguish a majority of the cover and stego images.

Eq. (14) shows that when calculating $\mathcal{IF}_1$ and $\mathcal{IF}_2$, the theoretical upper bound of the intuitive parameter S monotonically increases in a linear fashion with increasing variance of stegonoise, which in turns also increases along with the increment of embedding rate. Denote the corresponding variances of stegonoise of the suspected stego image, $\{x_i\}$ and $\{x_{i+1}\}$ by σ^2, σ_1^2 and σ_2^2,

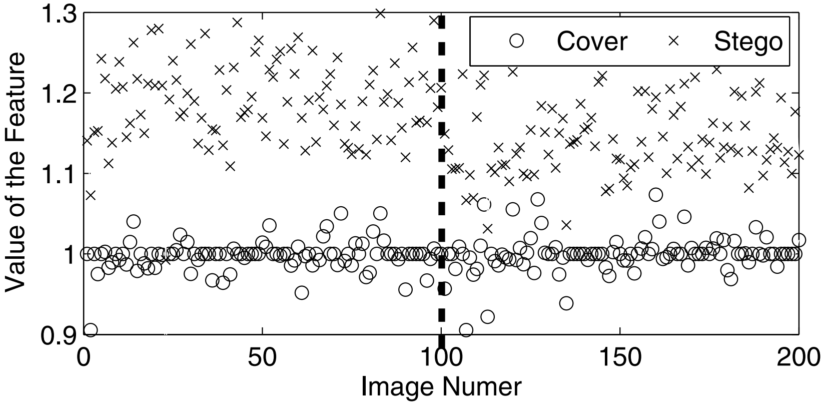


Fig. 3. Steganalytic features of 200 cover images and the corresponding LSBMRCP stego images (Left half, with 50% embedding rate), and EALSBMR stego images (Right half, with 50% embedding rate, Bz=1).

respectively. The monotonic relationship between them is clear and for a given embedding rate $\sigma_1^2 > \sigma^2 > \sigma_2^2$. In theory, once the embedding rate of a given LSBMRCP stego image is known, σ^2, σ_1^2, σ_2^2 and the corresponding upper bound for S, which denoted by S_c can be calculated from it respectively. Conversely, if S_c is determined, σ^2, σ_1^2 and σ_2^2, and finally the embedding rate of the suspected stego image can also be computed from it respectively.

The result smoothing cubic B-spline based on S_c represents the original cover image. S_c acts as a critical point in the smoothing/denoising process. The result spline based on a S less than S_c still contains stegonoise and $\mathcal{IF}_1 > \mathcal{IF}_2$ is held. On the other hand, the spline based on a S larger than S_c is a smoothed version of the recovered cover image, so that $\mathcal{IF}_1 \approx \mathcal{IF}_2$. Given a LSBMR stego image, we can calculate the value of $\mathcal{IF}_1/\mathcal{IF}_2$ using a progressively increasing S which starts from 0. The critical point S_c lies in the interval in which the value of $\mathcal{IF}_1/\mathcal{IF}_2$ falls from larger than 1 to approximately equal to 1. The variance of the stegonoise σ^2 can be reversely calculated from S_c using equations (14)-(16). Then the estimation of embedding rate can be determined since there is a monotonic relationship between σ^2 and it. Fig. 4 shows the value of $\mathcal{IF}_1/\mathcal{IF}_2$

as a function of the intuitive parameter S for a LSBMRCP stego image Fig. 4a. The value of S_c, the corresponding upper bound of S which is calculated using equations (14)- (18), is 0.186. It is clear to see that in Fig. 4c the value of the curve falls from larger than 1 to approximately equal to 1 around $S_c = 0.186$.

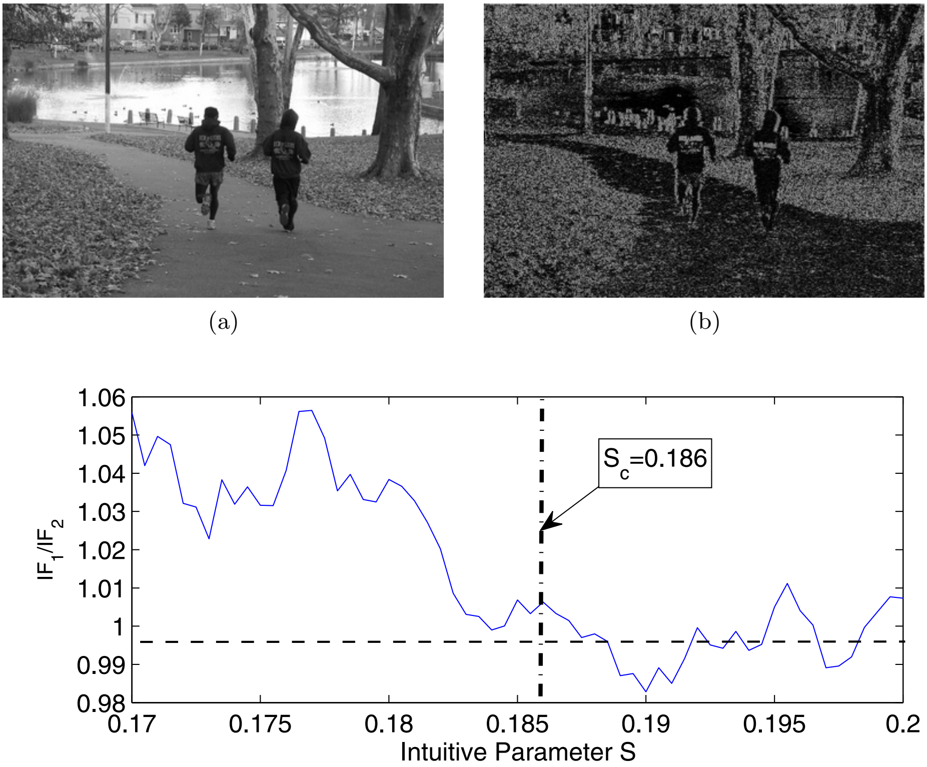
3.4 Steganalytic Algorithm

The details of the proposed steganalytic algorithm is shown as follows:

Step 1: The preprocessing procedure. The suspected stego image of size of $m \times n$ is firstly divided into a serial $\mathbf{I}$ of non-overlapping embedding units with every two consecutive pixels (x_i, x_{i+1}). Then $\mathbf{I}$ is further divided into two non-intersect sub-serial $\{x_i\}$ and $\{x_{i+1}\}$.

Step 2: Estimation of the stego noise series. Assume that each pixel in $\{x_i\}$ and $\{x_{i+1}\}$ is the summation of the corresponding original pixel in the unknown cover image and a stegonoise. Using the B-spline smoothing based calibration process mentioned in Sect. 3.2, two real-valued functions g_1 and g_2 are constructed for $\{x_i\}$ and $\{x_{i+1}\}$, respectively. $\{g_i\}$ and $\{g_{i+1}\}$ are the serials of the corresponding sample values of g_1 and g_2, and they can be regarded as the reconstructed pixel series in the unknown cover image for $\{x_i\}$ and $\{x_{i+1}\}$. Then two stego noise series $\{\varepsilon_i\}$ and $\{\varepsilon_{i+1}\}$ are finally obtained using Eq. (6).

Step 3: Removal of the potentially corrupted data. It is no doubt that g_1 and g_2 are just approximation of the unknown cover image. Therefore $\{\varepsilon_i\}$ and $\{\varepsilon_{i+1}\}$, which are computed from the subtraction of corresponding pixels in the suspected stego image and the approximation of the original cover image, may contains some potentially corrupted data, especially for those images with highly texture regions. In order to remove them, the items in the stego noise series are firstly sorted by value. And then the sorted series is divided into three quartiles:



(a) (b)

(c)

Fig. 4. (a) LSBMRCP stego image (embedding rate: 50%). (b) Pseudo image of the re-sampled points generated from the fourth derivative of the quartic interpolating B-spline $y_b(t)$. (c) Value of $\mathcal{IF}_1/\mathcal{IF}_2$ as a function of S for the stego image Fig. 4a.

- Lower quartile (denoted by Q_1), the lowest 25 percent data.
- Second quartile (denoted by Q_2), the median 50 percent data.
- Upper quartile (denoted by Q_3), the highest 25 percent data.

The data in Q_1 and Q_3 is abandoned in order to filter out the potentially corrupted data from a stego noise series. The second quartile of $\{\varepsilon_i\}$ and $\{\varepsilon_{i+1}\}$ are designated as $\{\varepsilon_i^{'}\}$ and $\{\varepsilon_{i+1}^{'}\}$, respectively.

Step 4: Construction of the classifier. $\mathcal{IF}_1$ and $\mathcal{IF}_2$ are computed using Eq. (7) from $\{\varepsilon_i^{'}\}$ and $\{\varepsilon_{i+1}^{'}\}$, respectively. If $\mathcal{IF}_1/\mathcal{IF}_2 > 1 + \theta$, then the suspicion is correct and that target image is indeed a stego one. Otherwise the target image is an innocent cover image. θ is a predefined positive parameter.

Step 5: Estimation of embedding rate for an already-known stego image. Set the initial value of S to zero, and progressively increase its value by 0.001. The very first S which makes $\mathcal{IF}_1/\mathcal{IF}_2 \leq 1 + \theta$ is the critical point S_c. Then the estimation of the embedding rate can be calculated from S_c as mentioned in Chap. 3.3.

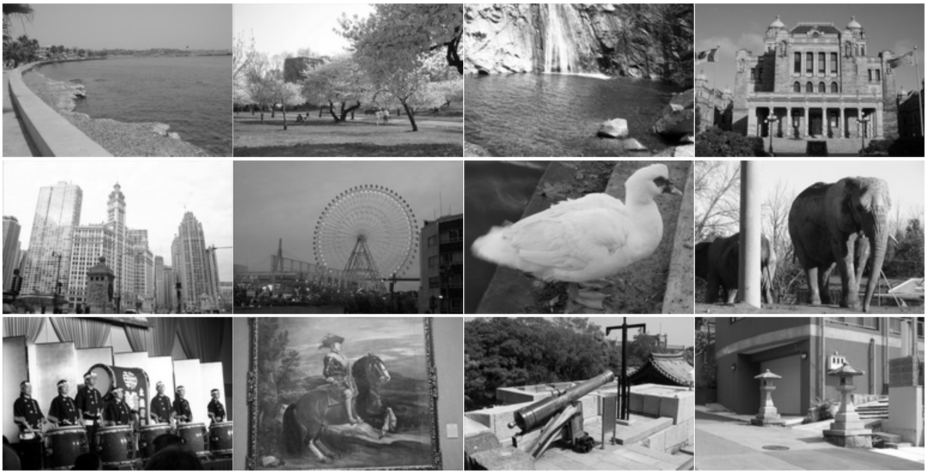


Fig. 5. Sample images from the image database

4 Experimental Results

To evaluate the proposed steganalytic method in spatial domain gray-scale images, two image datasets are used: NJIT dataset including 3680 uncompressed color images with a size of either 512 × 768 or 768 × 512, which were taken with different kinds of camera, and our dataset including 1320 uncompressed images with good quality. In all, there are 5000 original uncompressed color images for testing. The image database contains all kinds of images: natural scene, architecture, animals, indoor, outdoor, etc. Fig. 5 gives some sample images. All the images have been converted to gray-scale before the experiments. Stego images are generated using LSBMRCP and EALSBMR with $Bz = 1$.

We compare our method against Li-1D[7], since the experimental results of the previous works[7,8] show that Li-1D outperforms other LSBM steganalytic methods. In Fig. 6, receiver operating characteristic (ROC) curves of our proposed method and Li-1D are given for the set of LSBMRCP and EALSBMR stego images. It can be seen that Li-1D can only get fair result even when embedding rate is high for detecting LSBMRCP, while completely fails against EALSBMR with $Bz = 1$. On the contrary, our proposed method obtains satisfactory results for detecting stego images generated by LSBMRCP and EALSBMR method with $Bz = 1$. For a given false positive rate, the true positive rate of EALSBMR $Bz = 1$ is lower than LSBMRCP. This is because as an edge adaptive scheme, EALSBMR is apt to hide secret bits in the image regions with high texture which may depress the performance of the calibration process based on B-spline smoothing mentioned in Sect. 3.2.

Furthermore, the embedding rate for an already-known stego image can be estimated. Fig. 7 illustrated the estimated value of the embedding rate for LSBM-RCP stego images with different embedding rates. The stego images are arranged

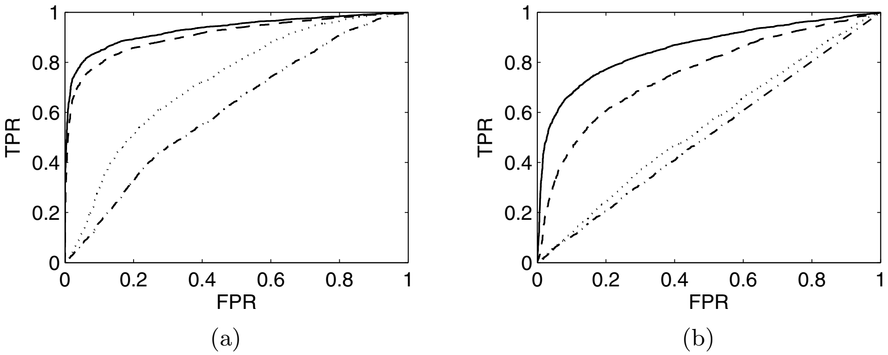


Fig. 6. Comparisons of ROC curves. The different curves stand for: our proposed method against LSBMRCP (solid), and EALSBMR ($Bz = 1$) (dashed); Li-1D against LSBMRCP (dotted), and EALSBMR ($Bz = 1$) (dash-dot). (a) 50% embedding rate. (b) 25% embedding rate.

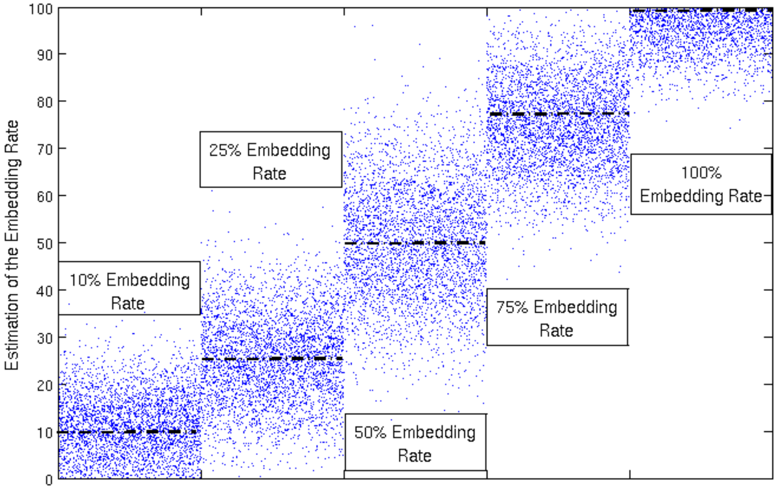


Fig. 7. Estimated value of the embedding rate for LSBMRCP stego images with embedding rate of 10%, 25%, 50%, 75% and 100%

according to their real embedding rate, from left to right 10%, 25%, 50%, 75% and 100%, respectively. It can be seen that the estimated embedding rates spread around the corresponding real embedding rates (denoted by dashed horizontal lines). In the current stage, we can only roughly estimate the embedding rate of a given stego image. The average estimation error is still relatively high. This is because a quartic B-spline is used to approximately represent the real image function $y(t)$ in order to effectively calculate the $\mathscr{L}_2$-norm of its fourth derivative as mentioned in Sect. 3.3, which unavoidably introduce additional distortion.

5 Conclusion and Future Work

In this paper, we point out that LSBMR for consecutive pixels and its descendants, including EALSBMR introduce intrinsic statistical imbalance in data embedding process, which results in the imbalance of the intensity factor, i.e. the power of the additive stegonoise, added to the pixel pairs. According to the deduction, a dimensionless discriminator based on intensity factor estimation using B-spline smoothing is constructed. Experimental results show that the proposed method is a reliable detector against LSBMR for consecutive pixels and EALSBMR with Bz=1. As an extension of the proposed method, we also put forward an embedding rate estimator based on B-spline functions which can roughly estimate the embedding rate. However, in the current stage, our method can not attack EALSBMR with $Bz > 1$. This is because that the reliability of the proposed method depends on the correct partition of embedding units with two consecutive pixels, while the rotation operation of EALSBMR with $Bz > 1$ prevents us to do so. How to construct efficient detector for EALSBMR with $Bz > 1$, it is our work for further study.

Acknowledgment. The author would like to thank Prof. Yun Q. Shi at New Jersey Institute of Technology, New Jersey, USA, for providing us the test image set.

References

1. Cancelli, G., Doerr, G., Barni, M., Cox, I.: A comparative study of ±1 steganalyzers. In: 2008 IEEE 10th Workshop on Multimedia Signal Processing 2008, pp. 791–796 (2008)
2. Goljan, M., Fridrich, J., Holotyak, T.: New blind steganalysis and its implications. In: Proc. SPIE on Security, Steganography, and Watermarking of Multimedia Contents, San Jose, CA, USA, vol. 6072, pp. 1–13 (2006)
3. Harmsen, J.J., Pearlman, W.A.: Steganalysis of additive noise modelable information hiding. In: Proc. SPIE on Security and Watermarking of Multimedia Contents, Santa Clara, CA, USA, vol. 5020, pp. 131–142 (2003)
4. Huang, F., Li, B., Huang, J.: Attack LSB matching steganography by counting alteration rate of the number of neighbourhood gray levels. In: 14th IEEE International Conference on Image Processing, ICIP 2007, pp. 401–404 (2007)
5. Ker, A.D.: Steganalysis of LSB matching in grayscale images. IEEE Signal Processing Letters 12(6), 441–444 (2005)
6. Li, X., Yang, B., Cheng, D., Zeng, T.: A generalization of LSB matching. IEEE Signal Processing Letters 16(2), 69–72 (2009)
7. Li, X., Zheng, T., Yang, B.: Detecting LSB matching by applying calibration technique for difference image. In: Proc. 10th ACM Workshop on Multimedia and Security, Oxford, U.K, pp. 133–138 (2008)
8. Luo, W., Huang, F., Huang, J.: Edge adaptive image steganography based on LSB matching revisited. IEEE Trans. Inf. Forensics Security 5(2), 201–214 (2010)
9. Mielikainen, J.: LSB matching revisited. IEEE Signal Processing Letters 13(5), 285–287 (2006)

10. Reinsch, C.H.: Smoothing by spline functions. Numerische Mathematik 10, 177–183 (1967)
11. Sharp, T.: An Implementation of Key-Based Digital Signal Steganography. In: Moskowitz, I.S. (ed.) IH 2001. LNCS, vol. 2137, pp. 13–26. Springer, Heidelberg (2001)
12. Unser, M.: B-spline signal processing: part I–theory. IEEE Trans. Signal Process. 41(2), 821–833 (1993)
13. Wahba, G.: Smoothing noisy data with spline functions. Numerische Mathematik 24, 383–393 (1975)
14. Wong, P.W., Chen, H., Tang, Z.: On steganalysis of plus-minus one embedding of continuous-tone images. In: Proc. SPIE on Security, Steganography, and Watermarking of Multimedia Contents, San Jose, CA, USA, vol. 5681, pp. 643–652 (2005)
15. Xu, H., Wang, J., Kim, H.J.: Near-optimal solution to pair-wise LSB matching via an immune programming strategy. Information Sciences 180(8), 1201–1217 (2010)
16. Xuan, G., Shi, Y.Q., Gao, J., Zou, D., Yang, C., Zhang, Z., Chai, P., Chen, C.-H., Chen, W.: Steganalysis Based on Multiple Features Formed by Statistical Moments of Wavelet Characteristic Functions. In: Barni, M., Herrera-Joancomartí, J., Katzenbeisser, S., Pérez-González, F. (eds.) IH 2005. LNCS, vol. 3727, pp. 262–277. Springer, Heidelberg (2005)
17. Zhang, J., Zhang, D.: Detection of LSB matching steganography in decompressed images. IEEE Signal Processing Letters 17(2), 141–144 (2010)

A Drift Compensation Algorithm for H.264/AVC Video Robust Watermarking Scheme

Xinghao Jiang[1,2,3], Tanfeng Sun [1,3,*], Yue Zhou[1], and Yun Q. Shi[2]

[1] School of Information Security Engineer,
Shanghai Jiaotong University, Shanghai, 200240, China
{xhjiang,tfsun}@sjtu.edu.cn
[2] Department of Electrical and Computer Engineering,
New Jersey Institute of Technology, Newark, NJ 07102, USA
shi@njit.edu
[3] Key Lab. of Shanghai Information Security Management and Technology Research,
Shanghai 200240, China

Abstract. A novel drift compensation algorithm for robust H.264/AVC video watermarking scheme is proposed. The drift compensation algorithm is implemented to reduce the visual distortion, which includes the reduction of the alteration of reference blocks caused by watermarking process. In our method, motion vector residuals of macroblocks in P frame are used as payloads of watermark. Discrete Cosine Transform (DCT) is performed on the motion vector residual group to utilize the energy compact property so that robustness against lossy compression attack can be obtained. According to the experimental results, our algorithm can obtain excellent imperceptibility and can significantly diminish the distortion influence. High rate lossy compression attack can be resisted effectively and an average of 80% accuracy rate of watermark detection can be achieved.

Keywords: Video Watermark, H.264/AVC, Drift Compensation, Motion Vector.

1 Introduction

With the rapid development of multimedia industry, digital videos are becoming popular in our life and on the Internet. The requirement of copyright protection becomes a great challenge, digital watermarking technologies have been considered as one of the most effective solution for this problem. Video is usually encoded in compressed format, so it is the more practical way to embed the watermark in compression domain.

Many researchers are following this filed and a variety of H.264/AVC video watermarking schemes have been proposed. Langelaar and Lagendijk [1] first proposed the differential energy watermark algorithm which utilized DC coefficients

[*] Corresponding author.

Y.Q. Shi, H.J. Kim, and F. Perez-Gonzalez (Eds.): IWDW 2011, LNCS 7128, pp. 30–41, 2012.

of DCT (Discrete Cosine Transform). Wu and Wang [2] presented a watermarking algorithm that embedded the watermark in I-frames. That scheme survived H.264 compression attacks with more than a 40:1 compression ratio in I-frames; however, it requires decompressing the video in order to embed the watermark. Another H.264 watermarking method proposed by Noorkami and Mersereau [3] embedded a readable watermark in the quantized AC coefficients, but its robustness against common watermarking attacks is not satisfactory. They also presented robust watermarking schemes in [4], but the original (uncompressed) video was required for calculating the parameter of visual model. All the algorithms above do not take the distortion drift into consideration, so that the optimal effect of imperceptibility can hardly be achieved by these algorithms all the time.

In this paper, we propose a drift compensation algorithm as well as a robust watermarking method for H.264/AVC video. Experimental results indicate that the proposed drift compensation algorithm significantly reduces the visual distortion influence of watermark and the watermarking scheme is robust to lossy compression attack.

The rest of this paper is organized as follows: In Section 2, some important features of H.264/AVC are briefly introduced. Section 3 presents our watermarking scheme. Section 4 shows the results and analysis. In Section 5, the conclusion and some future direction are presented.

2 Relevant Features for H.264/AVC Video Compression

In this section, two relevant features for H.264/AVC video compression will be introduced. Tree structured block size is the basis of macroblock selection scheme in our watermarking scheme and the distortion drift is one of the problems we aim to settle.

2.1 Tree Structured Block Size

H.264/AVC supports block sizes ranging from 16x16 to 4x4 luminance samples with many options. The luminance component of each MB (macroblock) (16x16 samples) may be split up in 4 ways shown as on the left of Fig. 1.

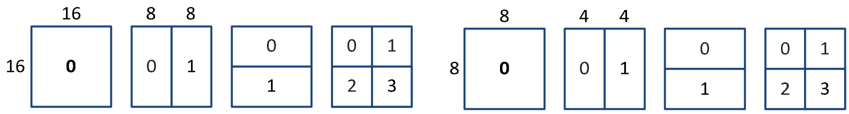


Fig. 1. Macroblock partitions: 16x16, 16x8, 8x16, 8x8 and sub-partitions: 8x8, 4x8, 8x4, 4x4

Each of the sub-divided regions is a MB partition. If the 8x8 mode is chosen, each of the four 8x8 MB partitions within the MB may be split in a further 4 ways as shown on the right of Fig. 1.

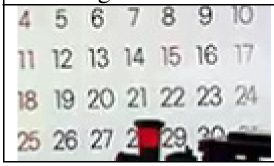
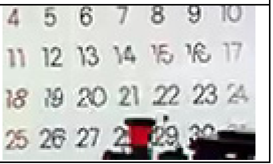
2.2 Distortion Drift

In motion compensation, there are three kinds of motion vectors named MV, MVP and MVD. Apparently, MV is the abbreviation of motion vector, which is a two-dimensional vector used for inter prediction that provides an offset from the coordinates in the decoded picture to those in a reference picture. MVP means motion vector prediction, which represents the motion vector in the reference picture. MVD stands for motion vector residual, i.e. the difference between MP and MVP. Relationship among these vectors is as follow:

$$MVD = MV - MVP \tag{1}$$

Altering the value of motion vector residual will make the embedded MB change, which means certain block in frame becomes different. In addition, this distortion is then propagated to the adjacent MBs and the succeeding frames due to motion prediction compensation, even though these MBs and frames have not been watermarked. This is called distortion drift and methods adopted to prevent such distortion happening is called distortion compensation.

Table 1. Comparison between the original frame and the frame with distortion drift

Original Frame	Distortion Drift
	

For instance, there are two 16x16 MBs named A and B. A is the reference block of B in the process of median prediction, which means MVP_B=MV_A. Increase of MVD_A (step ① in Fig. 2) leads to MV_A (step ②) and MVP_B (step ③) increasing given that MVP_A is not modified. Even with MVD_B unchanged, MV_B will finally increase (step ④). Such influence will cause severe distortion drift. In such case, decrease of MVD_B can compensate the increase of MVP_B and finally ensure MV_B unchanged.

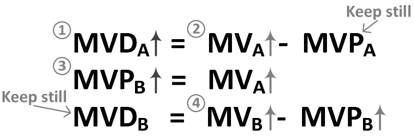


Fig. 2. Example of distortion drift

In literature [5,6], drift compensation was conducted to prevent such distortion propagation, but this method required a partial reconstruction of some pixels. Some other watermarking systems [7] avoids using drift compensation by attaching the watermark embedder with the video encoder. This approach however increases the computation burden significantly.

3 Proposed Video Watermarking Scheme

3.1 Watermarks Generation

In our scheme, one binary image is taken as the content of watermark. To improve the security of our algorithm and the robustness of embedded watermark, pseudorandom permutation with a secret key is performed on the watermark image before embedding.

Assume that T^2 is the original watermark image with binary values. Use the given secret key K to perform Arnold transform on T^2. Thinking of the T^2 as the 2-dimensional image space, Arnold's transform is the transformation $\Gamma : T^2 \rightarrow T^2$ given by:

$$\Gamma\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} \bmod K \tag{2}$$

K is the chosen secret key, and pixel value at (x, y) is replaced by value at (x', y') calculated from the above formula.

Finally, the binary image is converted into binary sequence with non-statistical properties so that it is difficult to be detected.

3.2 Drift Compensation Algorithm

As shown on the left of Fig. 3, the black block 'W' indicates the watermarked MB and all grey blocks are influenced by distortion drift. The number inside the block means the order of impact. '1' blocks are impact by 'W' blocks and spread distortion to '2' blocks. For the right of Fig. 3, 'C' blocks are the blocks being compensated so that no further impact happens on the adjacent blocks.

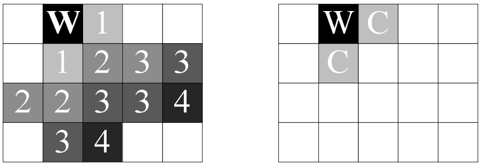


Fig. 3. Distortion drift and drift compensation

For more general circumstance, drift compensation are performed as follow:

Step 1: Store information (MB number and amount of modification) of previous MB if it has MVD changed because of watermark embedding;

Step 2: If the reference block of current MB has been watermarked according to the information recorded in Step 1, then:

i. If the current MB is not selected to embed watermark, just perform reverse modification which has been performed on the reference block;

ii. If the current MB is selected to embed watermark, do the embedding modification and store information of current MB. However, these two modifications on current MB and reference MB need to be accumulated. Drift compensation will be conducted together next time;

However, the reference block can occasionally be altered once the modification of MVD is done, which would cause worse drift distortion. In the median prediction one of the three MBs with median motion vector value is selected to be reference MB. The reason of such change is that modification on MVD changes MV, cause impact on the relationship of the three MBs and make the reference MB altered.

To solve such problem, more works need to be done before Step 2:

- If median prediction isn't performed, no change will happen and just go to Step 2.
- If median prediction is performed, retrieve three MVs and increase/decrease them according to the modification information stored. Thus the actual median value is there and reference block can be determined.
- If reference block has no change, go to Step 2;
- If change happens, new modification is calculated by MV of new reference block subtracting MV of original reference block. Then go to Step 2.

The drift compensation approach is illustrated in Fig. 4:

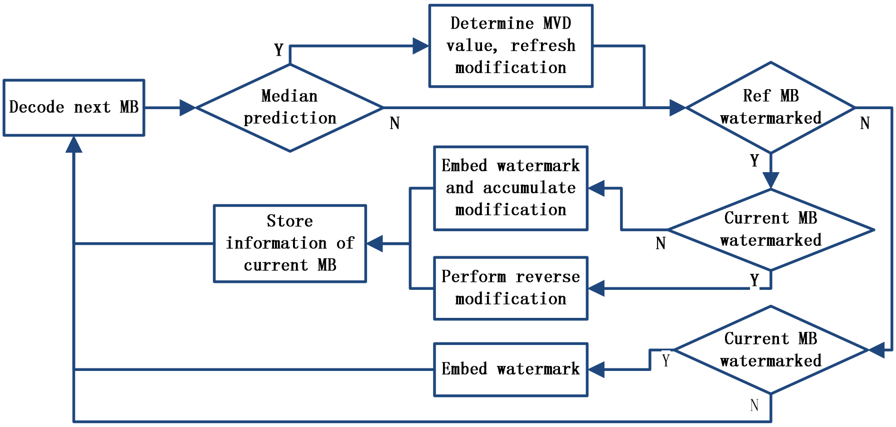

Decode next MB
Median prediction
Y
N
Determine MVD value, refresh modification
Ref MB watermarked
N
Y
Embed watermark and accumulate modification
Current MB watermarked
N
Y
Store information of current MB
Perform reverse modification
Embed watermark
Current MB watermarked
Y
N

Fig. 4. Drift compensation approach

3.3 Macroblock Selection

As shown in Fig. 5, in our scheme, MBs containing 8x8 sub-MBs are selected to embed watermark, because only these 8x8 sub-MBs may be split into more sub-partitions, which helps to limit the visual influence to the most extent. As a result, the right bottom sub-partition inside the right bottom sub-MB is used.

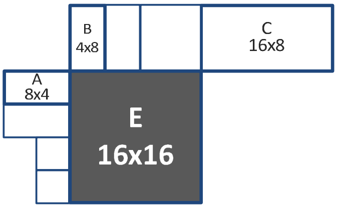


Fig. 5. Current block and adjacent block with sub-partitions

Actually, we have another reason to choose the right bottom sub-MB/sub-partition to embed watermark. As mentioned above, changes on one MB may cause distortion drift once this MB is referred by other MBs. But the right bottom sub-MB/sub-partition is the least possible one to be referred by adjacent MBs. In the media prediction, we suppose E is the current MB/sub-MB/sub-partition. The uppermost block is A, the leftmost one is B and C is on the up right side. Then the right bottom sub-MB/sub-partition is the least possible to be taken as reference block.

3.4 Anti-recompression Analysis

For those watermarking schemes based on motion vector proposed before, watermarks can barely survive normal video operations, such as scaling and rotation, lossy recompression and etc. Parity of motion vectors was usually the payload of watermark and the parity contains only one bit difference. Such difference will be vanished after being re-encoded, which means watermark cannot survive after some normal video operations, let alone some intentional attack. DCT domain based schemes, however, are usually robust to such operations due to the 'energy compaction' property of DCT. The change on one MV due to attacks spreads into several adjacent MVs so that the impact is weakened. Finally, the watermark can still survive.

In our scheme, MVDs are collected and DCT are then performed. DC coefficients and two adjacent AC coefficients containing the 'energy' of these MVDs are modified to embed watermark. When confronted with some unintentional/intentional attack such as lossy compression attack, value of MVDs will undoubtedly change, but the distribution of 'energy' won't change much so relationship between two 'energy' keep stable. That is the reason why our scheme is robust to lossy compression attack.

3.5 Watermarking Embedding Scheme

The watermark embedding approach is illustrated in Fig. 6:

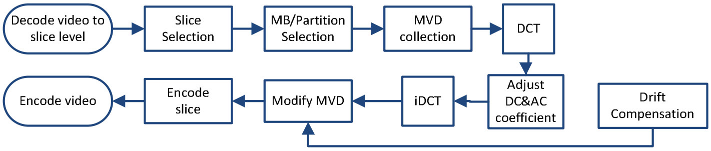


Fig. 6. Watermark embedding approach

The watermark embedding approach is described as follow:

1. The videos are decoded until a slice level is reached;
2. If the I/B slice being decoded in the video sequences, skip to next slice. If the current slice is the P slice, decode the slice;
3. Decode MB syntax of P slice. Only P MBs containing four 8x8 sub-MBs are selected. Find the right bottom sub-MB (8x8) and read motion vectors of the right bottom sub-partition if sub-partition exists. The horizontal motion vector residual of grey partition is used to embed watermark;

4. Take a line as a unit and start search from the right bottom corner of the slice. If we succeed to collect 8 P MBs, continue to search the next line until we gather 32 MBs (4 lines are needed in all);
5. Divide these 32 MBs into two groups(16 MBs each), modules horizontal MVDs of the right-bottom sub-partition in each MB with 10;
6. Convert MVDs into 4x4 matrix and perform DCT transform;

$$MVD'_{k_1k_2} = \sum_{n_2=0}^{3}\sum_{n_1=0}^{3} MVD_{n_1n_2} \cos\left[\frac{\pi}{3}\left(n_1+\frac{1}{2}\right)k_1\right]\cos\left[\frac{\pi}{3}\left(n_2+\frac{1}{2}\right)k_2\right] \quad (3)$$

$$k_1 = 0,\ldots,3,\quad k_2 = 0,\ldots,3$$

7. According to the watermark bit to be embedded, modify the value of DC coefficients and AC coefficients;

$$\begin{cases} MVD''_A - MVD''_B > T, \ if \ \ watermark=1 \\ MVD''_A - MVD''_B \le T, \ if \ \ watermark=0 \end{cases} \quad (4)$$

where MVD''_A and MVD''_B are the sum of DC coefficients and 2 AC coefficients adjacent to DC coefficients. T is the threshold.

8. Conduct IDCT(inverse DCT) transform, encode current P slice and make change into the original MVDs;

$$MVD''''_k = MVD - MVD \bmod 10 + MVD'''_k,\ \text{k=0,...,31} \quad (5)$$

where MVD'''_k is the result of MVD''_k after IDCT. MVD is original horizontal motion vectors residual.

9. Loop from the step 2 until we reach the end of video.

3.6 Watermarking Detection Scheme

The watermark detection approach is illustrated in Fig. 7:

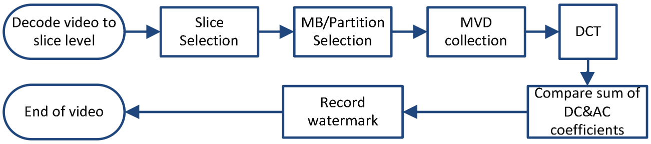


Fig. 7. Watermark detection approach

The watermark detection scheme is described as follow:

1. The suspected videos are decoded until slice level is reached;
2. If the I/B slice being decoded in the video sequences, skip to next slice. If the current slice is the P slice, decode the slice;

3. Decode MB syntax of P slice. Search for P MBs containing 8x8 sub-MBs, then we choose these MBs. Read motion vectors of each sub-MB. Read horizontal MVDs of the right bottom sub-MB.
4. Take lines as a unit and start from the right bottom corner. If we succeed to collect 8 MBs, continue to search the next line until 32 MBs gathered.
5. Divide these 32 MBs into two groups(16 MBs each), modules horizontal MVDs of the right-bottom sub-partition in each MB with 10. Convert them into 4x4 matrix and do DCT transform.
6. Make comparison of the sum of DC & AC coefficients of two groups as fellow:

$$\begin{cases} watermark=1, \ if \ \ MVD_A^{''}-MVD_B^{''}>T \\ watermark=0, \ if \ \ MVD_A^{''}-MVD_B^{''}\le T \end{cases} \tag{6}$$

7. Loop from the step 2 until we reach the end of video.

4 Experimental Results

In this experiment, the H.264/AVC codec JM8.6 [8] is used to test CIF (universal standard test sequence, 352x288) named ‘Bus’, ‘Flower’, ‘Mobile’ and ‘Stefan’. Both of them are encoded with a frame rate of 20 fps. Coding type is set as IPPP. ‘Elecard Stream Tools’, a video quality analysis software, is used in our experiment.

4.1 Watermark Imperceptibility

4.1.1 Experiments Compared with Original Video

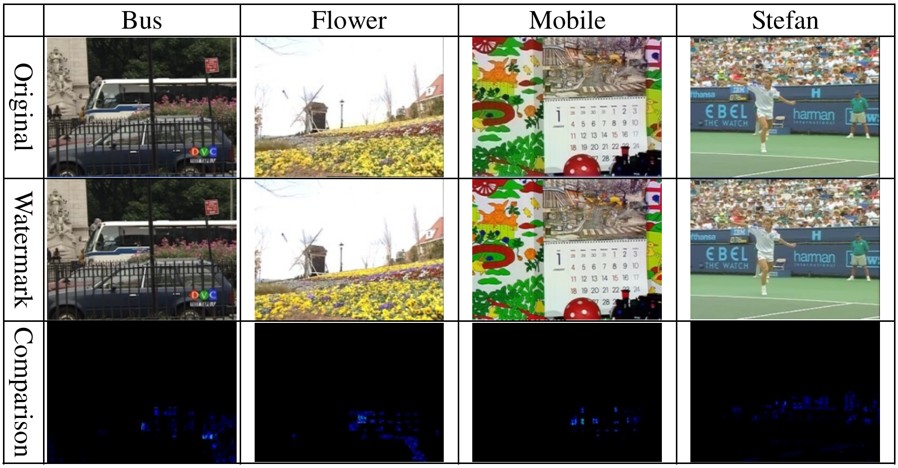
To evaluate the impact of watermark precisely, we introduce a new objective evaluation indicator: SSIM (Structural Similarity). It is a new indicator measuring the similarity of two video frames. That the value is close to ‘1’ indicates the high similarity of the two videos. As shown in Table 2, most of SSIM values being more than 0.98 illustrates that there is almost no impact on video similarity before and after watermark embedded.

Table 2. Results of video quality test under different QP (Quantization Parameter)

SSIM	QP=24	QP=26	QP=28	QP=30	QP=32
Bus	0.9925	0.9918	0.9958	0.9833	0.9822
Flower	0.9949	0.9934	0. 9912	0. 9921	0. 9931
Mobile	0.9905	0.9911	0.9913	0.9926	0.9898
Stefan	0.9873	0.9914	0.9907	0.9861	0.9889

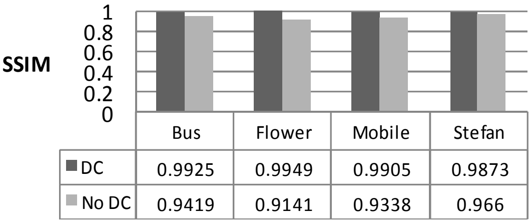
In Table 3, the grey area in ‘Comparison’ row means the difference between original video and watermarked video. As shown below, the embedded watermark does not affect the subjective video quality of the reconstructed image.

Table 3. Comparison between original video and video with watermark



4.1.2 Experiments Compared with No Drift Compensation Scheme

Drift compensation algorithm is adopted to avoid drift distortion and to improve the quality of watermarked video. We find that SSIM value drops a lot if drift compensation is not carried out as shown in Fig. 8. The decline of SSIM value is 0.08 at most, which means a great decrease in video quality. So scheme using drift compensation gains a better imperceptibility.



	Bus	Flower	Mobile	Stefan
■ DC	0.9925	0.9949	0.9905	0.9873
■ No DC	0.9419	0.9141	0.9338	0.966

Fig. 8. SSIM comparison between schemes with DC and without DC (DC stands for drift compensation)

4.1.3 Experiments Compared with Other H.264/AVC Scheme

To analyze the imperceptibility of our scheme, we compare our scheme with Sun’s algorithm in [9].

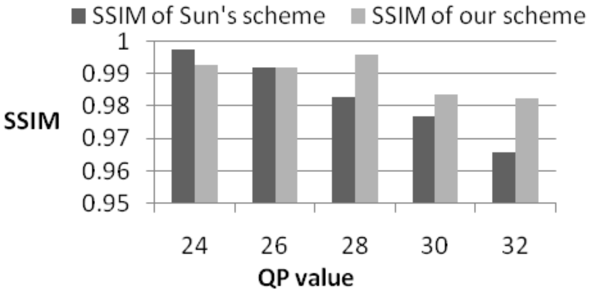


Fig. 9. Comparison of imperceptibility performance

In this experiment, we use sequence 'Bus' with QP ranging from 24 to 32. From the results shown in Fig. 9, we can conclude that our scheme has a better imperceptibility performance than Sun's algorithm with QP value ranging from 26 to 32 while Sun's algorithm gains better video quality with QP=24. The proposed scheme is embedded watermarks in VLC domain so that it performs well in high bit-rate condition while when confronted with quality descent, the influence of watermark becomes bigger.

4.2 Robustness against Lossy Compression Attack

In this experiment, all four sequences are used. After watermark being embedded, these videos are decoded and re-encoded with different QP value and some other parameters so that the recompression is performed.

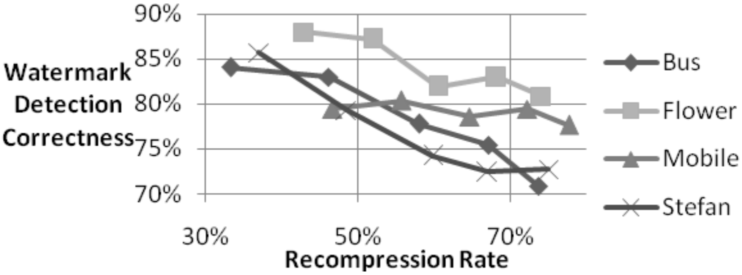
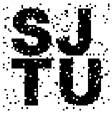
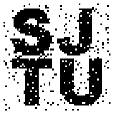
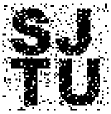
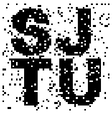
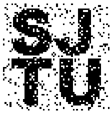


Fig. 10. Watermark correctness rate after lossy compression

Fig. 10 demonstrates the robustness of our watermark scheme against lossy compression attack. Under a lossy compression rate less than 50%, the correctness rate can keep above 79% while at a low recompression rate 40%, the best correctness is 88%. An extreme lossy compression rate 80% is tested and the correctness of watermark detection can still keep above 70% under such circumstance. Conclusion can be drawn that our scheme has excellent robustness against lossy compression attack.

Table 4. Watermark image detected from sequence named "flower" after lossy compression attack

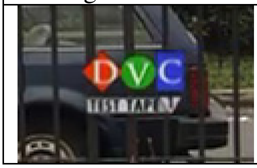
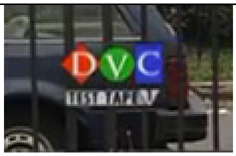
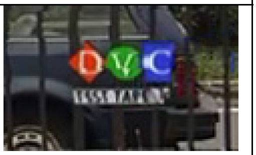
QP	Original	24	26	28	30	32
Image						

Images shown in Table 4 are watermark images detected from one of the test sequence named "flower" after lossy compression attack. As illustrated by Table 4, the watermark image changes more or less after the attack. With the increase of QP value, the compression rate becomes higher and the loss of watermark information gets more. However, the pattern can still be easily recognized even with an error rate of 20%-30%.

4.3 Watermark Capacity Analysis

The capacity of a watermarking scheme decides the application scope of the algorithm. In our proposed scheme, each P frame is embedded two bits watermark and little visual decrease can be notice. But if we try to embed 3 bits in each frame, the imperceptibility of our scheme will slightly decrease occasionally.

Table 5. Comparison of different watermark capacity

Original	2 bits embedded	3 bits embedded
		

As illustrated in Table 5, with two bits embedded, the video quality has no visible decrease compared with the original video frame while given that 3 bits are embedded in one frame, some distortion will be found.

5 Conclusion

In this paper, a novel drift compensation algorithm for robust H.264/AVC video watermarking scheme is proposed. The drift compensation algorithm is adopted to improve the imperceptibility of watermarking. MB selection scheme lowers the influence on video quality and reduces the possibility of drift distortion. The 'energy compaction' property of DCT is utilized to embed information into motion vector residual and good robustness against lossy compression attack is achieved. The experimental results indicate that the video quality is almost the same as that of the original because of the drift compensation algorithm being implemented. Even with a high recompression rate of 80%, the correctness of watermark detection can still reach 78%. So this scheme has excellent robustness against lossy compression attack. Our future work will focus on investigating the performances of various motion vector residuals used as payloads so that we can determine the optimal solutions for the proposed scheme.

Acknowledgments. This work was supported by the National Natural Science Foundation of China (No. 60802057, 61071153), Sponsored by Shanghai Rising-Star Program (10QA1403700), Program for New Century Excellent Talents in University (NCET-10-0569).

References

1. Langelaar, G., Lagendijk, R.: Optimal differential energy watermarking of DCT encoded images and video. IEEE Trans. Signal Process. 10, 148–158 (2001)
2. Wu, G.Z., Wang, Y.J.: Robust watermark embedding/detection algorithm for H.264. J. Electron. Imag. 14, 13013–13019 (2005)
3. Noorkami, M., Mersereau, R.M.: Compressed-domain video watermarking for H.264. In: Proc. IEEE Int. Conf. Image Processing, pp. 890–893 (2005)
4. Noorkami, M., Mersereau, R.M.: A framework for robust watermarking of H.264-encoded video with controllable detection performance. IEEE Trans. Inf. Forensics Security 2, 14–23 (2007)
5. Alattar, A.M., Lin, E.T., Celik, M.U.: Digital Watermarking of Low Bit-Rate Advanced Simple Profile MPEG-4 Compressed Video. IEEE Transactions on Circuits and Systems for Video Technology 13, 787–800 (2003)
6. Zeng, X., Chen, Z., Chen, H., et al.: Drift Compensation in Compressed Video Reversible Watermarking. In: WRI World Congress on Computer Science and Information Engineering, pp. 271-275 (2009)
7. Sakazawa, S., Takishima, Y., Nakajima, Y.: H.264 native video watermarking method. In: IEEE International Symposium on Circuits and Systems, pp. 1439–1442 (2006)
8. Suhring, K.: H.264/AVC Joint Model 8.6 (JM-8.6) Reference Software, `http://iphome.hhi.de/suehring/tml/`
9. Sun, T.F., Jiang, X.H., Lin, Z.G., et al.: An H.264/AVC Video Watermarking Scheme in VLC Domain for Content Authentication. China Communications 7, 30–36 (2010)

A High Performance Multi-layer Reversible Data Hiding Scheme Using Two-Step Embedding

Junxiang Wang[1,2], Jiangqun Ni[1,⋆], and Jinwei Pan[1]

[1] School of Information Science and Technology, Sun Yat-Sen University
Guangzhou 510006, P.R. China
[2] School of Mechanical & Electronic Engineering, Jingdezhen Ceramic Institute
Jingdezhen 333403, P.R. China
issjqni@mail.sysu.edu.cn

Abstract. In this paper, we present a new histogram shifting based multi-layer reversible data hiding scheme. By incorporating a flexible framework of two-step embedding (TSE), the proposed scheme can solve the problem of communicating and adoption of optimal pairs of peak and zero points and work in both pixel difference and predictive error domain for high performance reversible data hiding. A modified location map, which indicates only the actual overflow/underflow pixels, is constructed to facilitate the compression of location map. Compared with similar schemes, experimental results demonstrate the superior performance of the proposed scheme in the terms of embedding capacity and stego-image quality.

Keywords: Reversible data hiding, Multi-layer embedding, Two-step embedding, Location map, Histogram shifting, Pixel difference.

1 Introduction

In recent years, data hiding techniques have found wide applications in copyright protection and content authentication of digital multimedia. The inherent defect of the conventional data hiding schemes is that they can usually not completely recover the original image after the image has been modified for data hiding. For some specific scenarios, such as military, medical and legal applications, even the slight distortion in images is not tolerated. Therefore, reversible data hiding techniques are developed, which enable the decoder to not only extract the secret data as traditional schemes, but also perfectly reconstruct the original cover image without any distortion.

Many reversible data hiding schemes have been reported in literatures since Barton proposed his first reversible data hiding scheme [1] in 1997. In general, the existing reversible data hiding schemes can be classified into three categories: i.e., lossless compression [2]-[4], difference expansion (DE) [5]-[11], and histogram-shifting (HS) [12]-[19]. The schemes [2], [3] devised by Fridrich *et al.*

[⋆] Corresponding author.

Y.Q. Shi, H.J. Kim, and F. Perez-Gonzalez (Eds.): IWDW 2011, LNCS 7128, pp. 42–56, 2012.

belonged to the first category, which losslessly compressed the LSB planes to create spare space for data embedding. Later, Tian developed a high capacity reversible data hiding technique referred as difference expansion (DE) [5]. The proposed scheme explored the relevance of coefficients in Harr wavelet transform domain to implement DE operation, and then hided the secret data in the expanded vacant bits. Tian's scheme has been extended recently in [6]-[11].

Histogram shifting (HS) based reversible data hiding was first proposed by Ni *et al.* [12] in 2006, which selected a pair of peak and zero points in histogram and then shifted the bins between the two points by 1 towards zero point for reversible data embedding. The HS based reversible data hiding scheme has found wide applications for its high stego-image quality. The embedding capacity, however, is usually limited due to a flat histogram. Meanwhile, most HS related schemes are required to transmit extra side information, e.g. pairs of peak and zero points, therefore are non-blind in nature. By exploring the generalized Gaussian distribution of wavelet coefficients, Xuan *et al.* [14] and Wu [15] implemented HS in the domain of integer digital wavelet transform (IDWT), and obtained high embedding capacity. Tsai *et al.* in [17] designed a HS based scheme on the predictive errors and also obtained significant performance improvements. To meet the blind requirements, Hwang *et al.* [13] utilized the highest frequency bin in the pixel histogram as flag instead of the peak point. And Tai *et al.* [18] designed a synchronization mechanism by selecting fixed pairs of peak and zero points, which were not guaranteed to be the optimal ones. Therefore, the performance of the scheme was somewhat scarified.

In this paper, we propose a two-step embedding (TSE) technique to improve the HS based schemes, which not only meets the blind requirement but also guarantees the flexible selection of optimal pairs of peak and zero points for high performance reversible data hiding. Moreover, TSE is also utilized for location map reduction to further increase the embedding capacity.

The rest of the paper is organized as follows. The proposed scheme and related technical issues are described in Section II. The experiment results and analysis are given in Section III. Finally, the conclusions are summarized in Section IV.

2 The Proposed Scheme

In this section, the two-step embedding technique and construction of improved location map are presented first, then followed by the description of the proposed HS based multilayer reversible data hiding scheme which includes the embedding and extraction process.

2.1 Histogram Shifting on Pixel Differences

The HS based approach on pixel differences is employed in the paper and briefly reviewed as follows. For an N-pixel 8-bit grayscale cover image I with a pixel value x_i, where x_i denotes the grayscale value of i^{th} pixel, $0 \leq i \leq N-1$.

Scan the image I in an inverse S-order and compute the pixel difference d_i as follows:

$$d_i = |x_i - x_{i+1}|, (0 \le i \le N-2) \tag{1}$$

Based on the histogram of the generated pixel differences, determine the optimal peak point P and zero point Z. For HS based reversible data hiding, 1 bit secret message is embedded when a peak point is encountered. Therefore the capacity is computed by

$$capacity = h(P) \tag{2}$$

where $h(\bullet)$ denotes the frequency of bin P in the histogram. It is noted that since only the pixels with values between P and Z would generate distortion by one during embedding, the distortion caused by the embedding process is evaluate by

$$\begin{aligned} MSE &= \frac{1}{size}\left(\sum_{i\in U(P,Z)} h(i)\times(\varDelta i)^2 + \sum_{(j=P),(b='1')} (\varDelta j)^2\right) \\ &\quad ,(\varDelta i = 1, \varDelta j = 1) \\ &= \frac{1}{size}\left(\sum_{i\in U(P,Z)} h(i) + \sum_{(j=P),(b='1')} (1)^2\right) \end{aligned} \tag{3}$$

where *MSE* is the Mean Square Error between the stego and cover image, *size* and b denote the size of cover image and secret bit, respectively.

According to (2) and (3), to achieve a largest embedding capacity with a better stego-image quality, the optimal peak and zero point pair is determined as the highest frequency point P in the histogram and the closest zero frequency point Z to P as mentioned in literature [20].

Based on the chosen P and Z, the single layer HS operation is then performed on pixel difference d_i to generate the marked difference d'_i as follows.

HS shifts the histogram bins between P and Z towards the Z direction to create a vacant position near P. Scan the image of pixel difference in the same inverse S-order, and 1-bit message b is embedded whenever P is encountered. If $b = 0$, P keeps unchanged; otherwise, P is changed to the neighboring vacant bin. Assume P is on the left of Z, the HS is performed as follows.

$$d'_i = \begin{cases} d_i + 1, if\ d_i \in U(P,Z) \\ d_i + 1, if\ d_i = P\ and\ b = '1'; \ (0 \le i \le N-2) \\ d_i, \quad\ otherwise \end{cases} \tag{4}$$

where $U(P,Z)$ denotes the open set between P and Z.

The stego-image y_i is then generated according to the marked difference d'_i, i.e.,

$$y_i = \begin{cases} x_{i+1} + d'_i, if\ x_i > x_{i+1} \\ x_{i+1} - d'_i, if\ x_i < x_{i+1} \end{cases} (0 \le i \le N-2) \tag{5}$$

At the receiving end, based on the side information P and Z, the secret data extraction and image restoration are implemented in the inverse order as embedding phase, i.e.,

$$d'_i = |y_i - x_{i+1}|, (0 \leq i \leq N-2) \tag{6}$$

$$\begin{aligned} b &= \begin{cases} 0, if\ d'_i = P \\ 1, if\ d'_i = P+1 \end{cases}, (0 \leq i \leq N-2) \\ d_i &= \begin{cases} d'_i - 1, if\ d'_i \in U(P,Z] \\ d'_i, \quad otherwise \end{cases}, (0 \leq i \leq N-2) \end{aligned} \tag{7}$$

where $U(P,Z]$ denotes the interval between values P and Z except P.

The original image can be restored via (8),

$$x_i = \begin{cases} x_{i+1} + d_i, if\ y_i > x_{i+1} \\ x_{i+1} - d_i, if\ y_i < x_{i+1} \end{cases}, (0 \leq i \leq N-2) \tag{8}$$

It is noted that the original image should be losslessly recovered pixel by pixel via performing (6)-(8) repeatedly until all the pixels have been processed. Namely to recover the marked pixel y_i, the original pixel x_{i+1} should be recovered first and then the marked difference d'_i for y_i could be obtained by (6). Later, y_i could be restored to x_i by (7)-(8).

The process described by (1)-(8) uses only a single pair of peak and zero points and represents a single layer embedding and extraction. When the payload of secret data is increased, the strategy of multi-layer embedding can be employed, which repeatedly implements the HS embedding based on the resulting marked differences image and utilizes only one pair of optimal peak and zero points for each embedding layer as shown in Fig.2. Namely, for k^{th} layer embedding, the formula (4) is implemented on the marked differences image in $(k-1)^{th}$ layer, denoted as $d_i^{(k-1)}(0 \leq i \leq N-2)$, to generate the marked differences image $d_i^k(0 \leq i \leq N-2)$ in k^{th} layer. Based on the final marked difference d_i^m in m^{th} layer, namely d'_i, the stego-image is generated via (5). Similarly, the process of extraction and restoration for multi-layer embedding is performed pixel by pixel in the reverse order.

The framework of HS based multi-layer embedding can simplify the selection of optimal P and Z to a great extent in each embedding layer and thus leads to relatively large capacity with better stego-image quality. The recipient, however, should be given the peak and zero point pair of each level via additional channel for secret data extraction and image restoration. To tackle the issue of transmitting those extra side information, a two-step embedding scheme is developed in subsequent subsection.

2.2 Two-Step Embedding Scheme

The proposed two-step embedding (TSE) strategy provides a synchronization mechanism to communicate side information for HS based multi-layer embedding. We describe the implementation of TSE for single layer embedding as shown in Fig.1 and then extend to the multi-layer embedding as shown in Fig.2. First, the original image is mapped into pixel differences according to (1) and a pair of optimal peak and zero points (P, Z) in difference histogram is determined via (2)-(3). Then the differences image is partitioned into two non-overlapped areas as shown in Fig.1, namely A1 and A2, respectively. We implement the first step embedding to hide parts of secret data into A1 by using HS as mentioned in section II-A, and then generate the stego-pixels in A1 via (5). Next, the peak and zero points (P, Z) are accommodated in the least significant bits (LSBs) of the selected marked pixels in A1, which are determined with key K, and the replaced LSBs of the stego pixels in A1 are recorded and concatenated to the remaining secret data. Finally, the HS based second step embedding is performed to hide the rest secret data and generate the stego-image in A2.

At the decoder, the stego-image is identically partitioned as the embedding side. With the same key K, the peak and zero points are extracted from their stored LSBs of A1. Based on the retrieved side information, e.g. peak/zero points, the pixels in A2 are iteratively restored as described in section II-A and the secret data and the replaced LSBs hidden in A2 are extracted. After the replaced LSBs in A1 are recovered with the extracted LSBs, the same extraction process is performed to extract the secret data and restore the pixels in A1. Thus the complete secret data is extracted and whole cover image is recovered.

When the payload of secret data is lager, the approach of multi-layer embedding is employed as shown in Fig.2. For HS based m-layer embedding incorporating TSE, the embedding of each layer except the final m^{th} layer is implemented in the way as described in Section II-A. The aforementioned TSE method is then utilized to complete the final m^{th} layer embedding and hide all the side information, e.g. the pairs of peak and zero points for each layer, in A1 of m^{th} layer. At the decoder, the extraction and restoration are iteratively performed pixel by pixel in the reverse order as the embedding process.

Note that A1 should be large enough to accommodate the complete side information for each layer. Consider that the absolute value of one pixel difference for a 8-bit grayscale image is represented by only 9 bits, 18 bits are required to represent the pair of peak and zero points for each layer. The requirement for A1 is easily met for practical application $(m \leq 5)$.

2.3 Improved TSE for Location Map Reduction

HS based embedding on the pixel differences may lead to overflow and underflow, which means the resulting stego-pixels may be not in the range $[0, 255]$ for a 8-bit grayscale image. To tackle the issue, histogram-narrowing technique (HN) is usually adopted [18]. Note that each layer HS operation leads to a maximum distortion for one pixel by one unit, the accumulated distortion between an original pixel and the stego-pixel is not more than m units for m-layer embedding. The HN operation is described by

$$x'_i = \begin{cases} x_i + m, \, if \;\; I(i) \in [0, m-1] \\ x_i - m, \, if \;\; I(i) \in [255 - m + 1, 255] \\ x_i, \qquad otherwise \end{cases}, (0 \leq i \leq N-2) \tag{9}$$

where x'_i denotes the narrowed pixel.

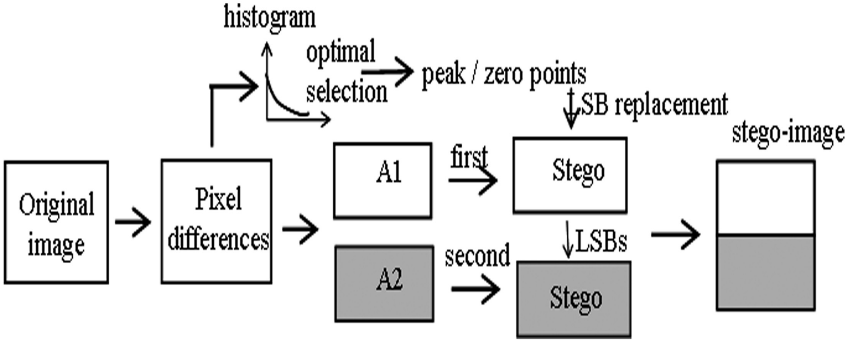


Fig. 1. Framework of two-step embedding for single layer embedding

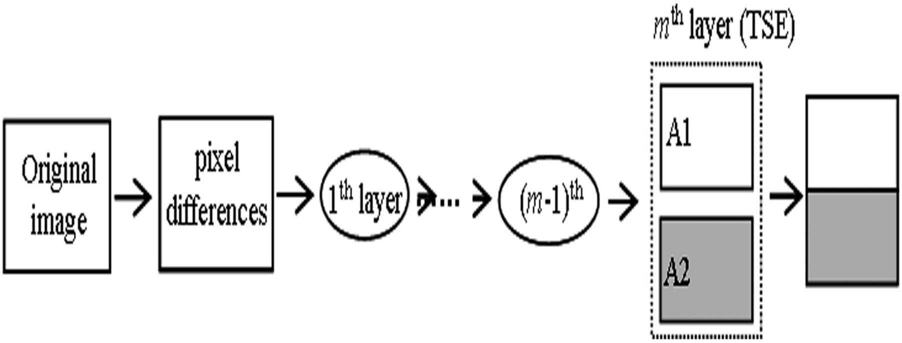


Fig. 2. Framework of two-step embedding for multi-layer embedding

To distinguish the source of the overlapping pixels after HN operation, the location map is introduced, which equals to the size of the cover image. For a narrowed pixel, we assign $'0'$ in the location map; otherwise, we assign $'1'$. The location map is then losslessly compressed and embedded into the cover image together with the secret data. Note that the HN operation is usually employed as a preprocessing step and makes all the potentially overflowed/underflowed pixels (POPs) in the range $[0, m-1] \bigcup [255-m+1, 255]$ narrowed.

In view of the fact that not all the POPs actually overflowed/underflowed during multi-layer embedding, we proceed to exchange the order of HN operation and multilayer embedding and identify only the actually overflowed/underflowed pixels (AOPs) after all the POPs have been processed. For m-layer embedding, we classify the pixels of cover image with gray value in the range $[0, m-1] \bigcup [255-m+1, 255]$ as POPs. The remaining pixels are denoted as R_POPs. The cover image is then converted to pixel differences and HS based embedding is implemented for the first $(m-1)$ layers. The identification of AOPs

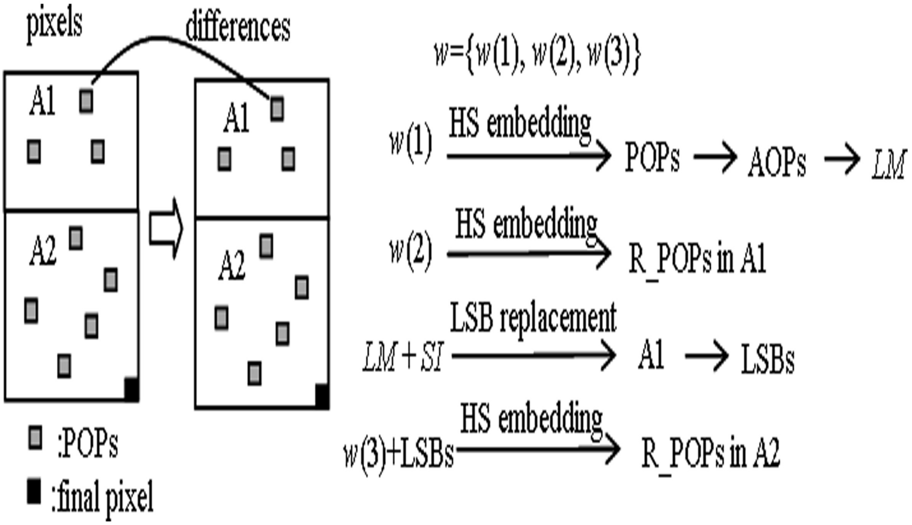


Fig. 3. The sketch map of ITSE for location map reduction

and two-step embedding are performed in the m^{th} layer as shown in Fig. 2. To simplify the description, we take the single layer embedding ($m = 1$) as example as shown in Fig.3. The secret message w is partitioned into 3 parts, i.e. $w = \{w(1), w(2), w(3)\}$. Perform the HS among the pixel differences in the position of POPs to embed $w(1)$ and generate the corresponding marked pixels via (4)-(5). With the marked pixels in POPs, the AOPs are then identified to generate the improved location map. The histogram narrowing (HN) is also performed for all AOPs via (9). An improved two-step embedding (ITSE) is then performed in the position of R_POPs. Let *LM* and *SI* denote the compressed location map and side information (peak and zero points for each layer), respectively. The first step embedding hides message $w(2)$ into the pixel differences in the position of R_POPs in A1 and then reconstructs the corresponding marked pixels from marked differences. With LSB replacement, the *LM* and *SI* are then embedded into the LSBs of stego-pixels determined by key K in A1. The replaced LSBs and the remaining message $w(3)$ are hided into pixel differences in the position of R_POPs in A2 for the second step embedding. Thus the final stego-image is obtained. Note that the last pixel in the original image, i.e. $x(N-1)$, keeps unchanged during m layers embedding. The detailed data extraction and image restoration process will be described in section II-E.

2.4 Embedding Process

In this subsection, for an N-pixel 8-bit grayscale cover image I with i^{th} pixel value denoted as x_i, where $0 \leq i \leq N-1$, and the message w, the proposed scheme on pixel differences is described as follows.

1) Calculate the pixel differences according to the cover image I.
2) Determine the embedding layer m based on the message w and the histogram of pixel differences.
3) Identify the pixels in cover image with gray value in the range $[0, m-1] \bigcup [255-m+1, 255]$ as POPs, and R_POPs for the remaining pixels.
4) Partition the message w into m parts, i.e. $w = \{w_1, w_2..., w_m\}$, Initialize layer $k = 1$, and then perform HS based multi-layer embedding under the framework of TSE.
    - 4.1) For k^{th} layer embedding, we select a pair of optimal peak and zero points, denoted as P_k and Z_k respectively, and then hide w_k in an inverse S-order by using HS method as described in section II-A.
    - 4.2) If $k \neq m$, let $k = k + 1$ and repeat the step 4.1 for the next layer embedding. Otherwise, go to step 4.3.
    - 4.3) Perform the final layer embedding with improved TSE (ITSE) considering location map reduction. Partition the message w_m into three parts, denoted as $w_m = \{w_m(1), w_m(2), w_m(3)\}$, and let $SI = \{(P_k, Z_k)|1 \leq k \leq m\}$. Embed the message w_m, compressed location map LM denoting AOPs, side information SI and the total number of embedding layer m into the m^{th} layer using ITSE as mentioned in section II-C. Thus the stego-image $Y = \{y_i|0 \leq i \leq N-1\}$ is generated.

Note that the histogram narrowing and image partition operation are performed during the ITSE period in step 4.3.

2.5 Extraction Process

In this process, we extract the embedded message and recover the marked image Y to its original version without communicating auxiliary information.

1) Divide the stego-image into A1 and A2 as did in embedding side, and collect the auxiliary information, e.g. $LM + SI + m$, from the LSBs of the marked pixels in A1 determined with key K.
2) Decompress LM and generate the location map to indicate the actually overflowed/underflowed pixels (AOPs). Perform the inverse HN operation on the AOPs.
3) For m-layer embedding, the stego-pixels in $Y = \{y_i\}_{i=0,...,N-1}$ are losslessly recovered and secret data are extracted pixel by pixel in the inverse S-order from the final pixel y_{N-1} until all the pixels are processed.
    - 3.1) The final pixel in stego-image is kept unchanged in embedding, i.e., $y_{N-1} = x_{N-1}$, and set $i = N-2$.
    - 3.2) Recover the marked pixel y_i and extract the secret data hidden at position i during m-layer embedding. With marked pixel y_i, the corresponding marked difference d_i' in m-th layer, is obtained via (6). To recover the difference at position i in each layer, the formula (7) and the extracted SI are applied repeatedly until the original difference d_i is obtained. The original pixel x_i is then recovered via (8).

Denote the secret data hidden in y_i by m-layer embedding as $B_i = \{b_i^1, b_i^1 ... b_i^m\}$, where b_i^k represents the secret bit hidden in y_i during the k^{th} layer embedding. Note that secret bit array B_i for y_i is defined for algorithm description. In case no secret bit is embedded at position i during k^{th} layer embedding, b_i^k is null and the size of B_i is decreased.

3.3) Let $i = i - 1$. If $i \geq 0$, repeat the step 3.2 and restore the next stego-pixel y_{i-1} by using recovered pixel x_i. Otherwise, the process of reversible data extraction is completed and goes to step 4.

4) Reconstruction of the secret data w. With the secret array $B_i (0 \leq i \leq N-2)$ for each pixels y_i, we then proceed to reconstruct the secret data embedded during m-layer embedding, i.e., $w = \{w_1, w_2, ..., w_m\}$. According to II-C, the embedding for the final layer (m^{th} layer) is quite different from those in previous layers, therefore the secret data w_m embedded in the final layer is reconstructed individually.

4.1) Construct the secret data w_k hidden in the k-th layer ($k \neq m$) by

$$w_k = \{b_i^k | 0 \leq i \leq N-2\} (1 \leq k \leq m-1) \tag{10}$$

4.2) To construct w_m, we first identify the positions of POPs and R_POPs according to the recovered $x_i (0 \leq i \leq N-2)$. We then partition $b_i^m (0 \leq i \leq N-2)$ into three parts, i.e., $w_m(1)$, $w_m(2)$ and $w_m(3)$, which are hidden in the position of POPs, R_POPs of A1 and R_POPs of A2 in the m^{th} layer embedding, respectively. Finally we concatenate the three parts of secret data to form $w_m = \{w_m(1), w_m(2), w_m(3)\}$.

4.3) Combine the secret data $w_i (i = 1, 2, ..., m)$ embedded in each layer to form the final extracted data $w = \{w_1, w_2, ..., w_m\}$.

2.6 Extension to Prediction Errors

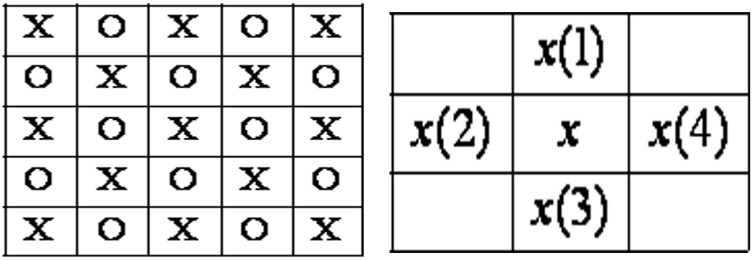
To take advantage of the correlation among neighboring pixels in natural images, a good prediction model is usually applied. The resulting predictive errors usually have a much shaper histogram than that of pixel differences. By extending the TSE framework to the domain of prediction errors, significant performance improvement can be expected.

An efficient prediction model proposed in [11] is incorporated with improved two-step embedding (ITSE). To implement the prediction, the image is divided into two sets, denoted as "round" and "cross" sets, as shown in Fig.4. Then the adjacent four pixels in different set are exploited to predict the current pixel x by

$$\tilde{x} = \frac{\lfloor x(1) + x(2) + x(3) + x(4) \rfloor}{4} \tag{11}$$

and the predictive error is calculated by

$$d = (x - \tilde{x}) \tag{12}$$



X	O	X	O	X
O	X	O	X	O
X	O	X	O	X
O	X	O	X	O
X	O	X	O	X

(a)

	$x(1)$	
$x(2)$	x	$x(4)$
	$x(3)$	

(b)

Fig. 4. The sketch for the prediction

$$X^i \Rightarrow \tilde{O}^i = \left[X^i(1)+X^i(2)+X^i(3)+X^i(4)\right]/4 \Rightarrow d_0^i = (O^i - \tilde{O}^i) \Rightarrow (d_0^i)' \Rightarrow$$

$$O^{i+1} = \tilde{O}^i + (d_0^i)' \Rightarrow O^{i+1}$$

(a)

$$X^i \Rightarrow \tilde{O}^i = \left[X^i(1)+X^i(2)+X^i(3)+X^i(4)\right]/4 \Rightarrow (d_0^i)' = O^{i+1} - \tilde{O}^i \Rightarrow d_0^i \Rightarrow$$

$$O^i = \tilde{O}^i + d_0^i \Rightarrow O^i$$

(b)

Fig. 5. The sketch for predictive error based $(i+1)^{th}$ layer embedding process (a) and extraction process (b)

It is noted that the prediction scheme leads to that the embedding process is slightly different from the previous one. The process of prediction and embedding is performed in turn for two sets. Let Z^i, d_Z^i and $(d_Z^i)'(Z = 'X'$ or $'O')$ be the marked pixel after i^{th} layer embedding, the predictive error for Z^i and the marked error, respectively. And Z^0 is the cover pixel.

The $(i+1)^{th}$ layer embedding and extraction process is performed as shown in Fig. 5(a), where we predict each pixel O^i in "round" set with $\tilde{O}^i$ by using its 4 neighboring pixels in "cross" set and embed the secret data in the predictive error d_O^i through HS operation and generate $(d_O^i)'$.Thus the marked pixels in the "round" set can be computed by $O^{i+1} = \tilde{O}^i + (d_o^i)'$, which is denoted as $(O^i \Longrightarrow O^{i+1})$. Then by using marked O^{i+1} in "round" set to predict the pixel X^i in "cross" set, the same process is applied for $(i+1)^{th}$ layer embedding of "cross" set, which is denoted as $(X^i \Longrightarrow X^{i+1})$. Until then the $(i+1)^{th}$ layer embedding is completed. The process of $(i+1)^{th}$ layer image restoration and data extraction for "round" set ("corss" set is the same) is implemented as shown in Fig. 5(b). When a large payload of secret data should be embedded, multi-layer embedding ($m \geq 2$) is needed by implementing the process shown in Fig.5 alternatively for the two sets in each layer.

3 Experimental Results and Analysis

To evaluate the proposed scheme, we test six $256 \times 256 \times 8$ -bit gray images with different texture characteristics, i.e., Lena, Peppers, Baboon, F16, Goldhill and Boat, and use run-length coding (RLC) to losslessly compress the location map. In addition, we maintain the PSNR of stego-image to be greater than 38 dB in our simulation for a high visual quality of stego-image.

3.1 The Efficiency for Location Map Reduction

To justify the efficiency of the improved location map, a parameter *E_Map* is introduced to evaluate the percentage of location map reduction which is defined as follows,

$$E\_Map = \frac{|LM_t - LM_i|}{LM_i} \times 100\% \tag{13}$$

where LM_t and LM_i denote the size of traditional and improved location map, respectively.

Table I gives a comparison between the traditional location map proposed in [18] and our improved one as mentioned in section II-C under different embedding layers, which demonstrates the feasibility of the proposed scheme for location map reduction. It is observed that, for the images, e.g. Lena, Goldhill and F16, the grayscale values of most of their pixels are not in the range of potentially overflow/underflow (POPs) during embedding, therefore the reduction of location map size with the improved scheme is limited. However, for other test images, the location map size reduction is significant due to a relatively large portion of their pixels are in the range of POPs. In addition, with the embedding layer increasing, the potential for pixel overflow is increased and the proposed scheme is more efficient.

3.2 Comparison between the TSE in Pixel Differences, in Predictive Errors and Other Schemes

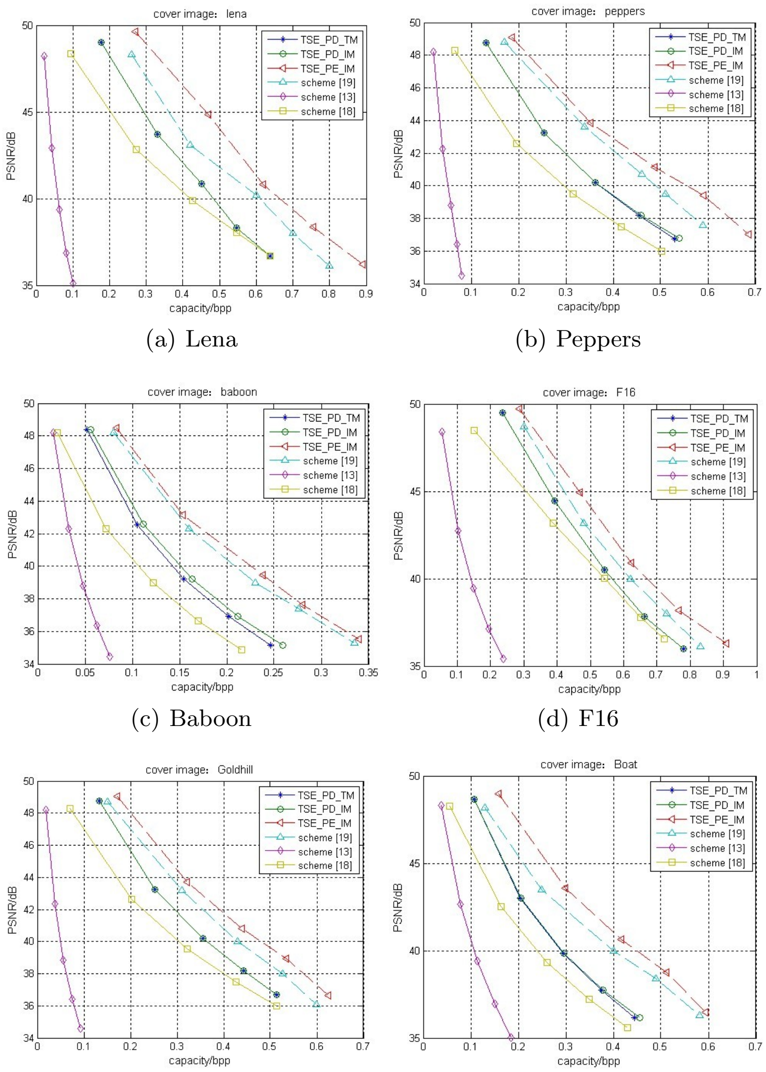
We implement the proposed TSE based reversible data hiding on the pixel differences and predictive errors, which are described in section II. Fig. 6 gives the performance comparisons between TSE with different settings and other schemes, where TSE_PD_TM, TSE_PD_IM and TSE_PE_IM represent the proposed TSE scheme in pixel differences with traditional and improved location

Table 1. Location map size comparison with different embedding layers

Cover image (512×512)	chosen schemes	the embedding layer m				
		1	2	3	4	5
Lena	Traditional location map [18]	20	20	20	20	20
	Improved location map	20	20	20	20	20
	Overflow/Underflow	N	N	N	N	N
	E_Map	–	–	–	–	–
Peppers	Traditional location map [18]	100	240	740	2160	4380
	Improved location map	100	220	620	1320	2100
	Overflow/Underflow	Y	Y	Y	Y	Y
	E_Map	0%	9.1%	19.4%	63.6%	108.6%
Baboon	Traditional location map [18]	1120	1760	2480	3080	3520
	Improved location map	160	240	280	360	340
	Overflow/Underflow	Y	Y	Y	Y	Y
	E_Map	600%	633.3%	785.7%	755.6%	935.3%
F16	Traditional location map [18]	20	20	20	20	20
	Improved location map	20	20	20	20	20
	Overflow/Underflow	N	N	N	N	N
	E_Map	–	–	–	–	–
Goldhill	Traditional location map [18]	20	20	20	20	20
	Improved location map	20	20	20	20	20
	Overflow/Underflow	N	N	N	N	N
	E_Map	–	–	–	–	–
Boat	Traditional location map [18]	200	480	800	1320	2980
	Improved location map	180	320	460	620	780
	Overflow/Underflow	Y	Y	Y	Y	Y
	E_Map	11.1%	50%	73.9%	112.9%	282.1%

map and TSE in predictive errors, respectively. It is observed in Fig.6 that the performance of TSE in predictive errors is significantly better than that of TSE in pixel differences, which indicates that well utilization of the image redundancy could lead to a better performance. Fig. 6 also shows that, for test images, e.g. Peppers, Baboon and Boat, with relatively large portion of POPs, the performance improvements on the location map reduction are gradually verified with the increase of embedding layer m. In addition, for other test images, e.g. Lena, F16 and Goldhill, the performance curves are identical due to no potentially overflow/underflow (POPs) in the images during embedding.

We then compare our TSE based scheme with other similar schemes [13] and [18], which adopted the HS based reversible data hiding and met the blind requirements. As shown in Fig. 6, both the TSE scheme in pixel differences and predictive errors outperforms the other two schemes with distinct margins. The scheme in [13] is implemented on pixel domain and doesn't take into account the correlation between neighboring pixels. Consequently the generated histogram is flat, which explains the inferior performance of scheme in [13]. The scheme in [18], however, works on the pixel differences and utilizes a binary tree structure



(a) Lena

(b) Peppers

(c) Baboon

(d) F16

(e) Goldhill

(f) Boat

Fig. 6. Performance comparison between the proposed TSE and other schemes

to communicate the side information of pairs of peak and zero points. Although the blind requirement is met, the adoption of fixed peak and zero point pairs leads to relatively poor performance for the scheme. Finally, when compared with a high performance scheme [19] reported recently which used interpolation errors, our TSE scheme also consistently outperforms it as shown in Fig.6.

4 Conclusion

In this paper, a new two-step embedding scheme for HS based multi-layer reversible data hiding is proposed. Different from the conventional HS based scheme, the proposed scheme exploits TSE to solve the problem of communicating side information. The TSE framework also ensures the adoption of optimal peak and zero point pair in each layer for high performance reversible data hiding. In addition, an improved location map, which indicates only the actual overflow/underflow pixels, is constructed to facilitate the compression of location map and further increase the embedding capacity. Extensive simulations are carried out, which demonstrates that the proposed scheme could not only meet the blind requirement but also achieve a high capacity with better stego-image quality.

Acknowledgments. This work is supported by National Natural Science Foundation of China under contract 60970145.

References

1. Barton, J.M.: Method and Apparatus for Embedding Authentication Information within Digital Data. U.S. Patent 5646997 (1997)
2. Fridrich, J., Goljan, M., Du, R.: Invertible Authentication. In: Proc. SPIE Security and Watermarking of Multimedia Contents III, San Jose, CA, pp. 197–208 (2001)
3. Fridrich, J., Goljan, M., Du, R.: Lossless Data Embedding for All Image Formats. In: Proc:SPIE Security Watermarking Multimedia Contents IV, San Jose, CA, pp. 185–196 (2002)
4. Celik, M.U., Sharma, G., Tekalp, A.M., Saber, E.: Lossless Generalized-LSB Data Embedding. IEEE Trans. Image Process 14(2), 253–266 (2005)
5. Tian, J.: Reversible Watermarking Using A Difference Expansion. IEEE Trans. Circuits Syst. Video Technol. 13(8), 890–896 (2003)
6. Kamstra, L., Heijmans, A.M.: Reversible Data Embedding into Images Using Wavelet Techniques and Sorting. IEEE Trans. Image Process 14(12), 2082–2090 (2005)
7. Kim, H.J., Sachnev, V., Shi, Y.Q., Nam, J., Choo, H.G.: A Novel Difference Expansion Transform for Reversible Data Embedding. IEEE Trans. Inf. Forensic Security 3(3), 456–465 (2008)
8. Hu, Y., Lee, H.K., Li, J.: DE-based Reversible Data Hiding with Improved Overflow Location Map. IEEE Trans. Circuits Syst. Video Technol. 19(2), 250–260 (2009)
9. Lee, S., Yoo, C.D., Kalker, T.: Reversible Image Watermarking Based on Integer-To-Integer Wavelet Transform. IEEE Trans. Inf. Forensics Security 2(3), 321–330 (2007)
10. Thodi, D.M., Rodriguez, J.J.: Expansion Embedding Techniques for Reversible Watermarking. IEEE Trans. Image Process 16(3), 721–730 (2007)
11. Sachnev, V., Kim, H.J., Nam, J., Suresh, S., Shi, Y.Q.: Reversible Watermarking Algorithm Using Sorting and Prediction. IEEE Trans. Circuits Syst. Video Technol. 19(7), 989–999 (2009)
12. Ni, Z., Shi, Y.Q., Ansari, N., Su, W.: Reversible Data Hiding. IEEE Trans. Circuits Syst. Video Technol. 16(3), 354–362 (2006)

13. Hwang, J., Kim, J., Choi, J.: A Reversible Watermarking Based on Histogram Shifting. In: Shi, Y.Q., Jeon, B. (eds.) IWDW 2006. LNCS, vol. 4283, pp. 348–361. Springer, Heidelberg (2006)
14. 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)
15. Wu, X.: Reversible Semi-fragile Watermarking Based on Histogram Shifting of Integer Wavelet Coefficients. In: Proc. Digital Ecosystems and Technologies, Cairns, Australia, pp. 501–505 (2007)
16. Lin, C.C., Tai, W.L., Chang, C.C.: Multilevel Reversible Data Hiding Based on Histogram Modification of Difference Images. Pattern Recognition 41(12), 2582–3591 (2008)
17. Tsai, P., Hu, Y.C., Yeh, H.L.: Reversible Image Hiding Scheme Using Predictive Coding and Histogram Shifting. Signal Processing 89(6), 1129–1143 (2009)
18. Tai, W.L., Yeh, C.M., Chang, C.C.: Reversible Data Hiding Based on Histogram Modification of Pixel Differences. IEEE Trans. Circuits Syst. Video Technol. 19(6), 906–910 (2009)
19. Luo, L., Chen, Z., Chen, M., Zeng, X., Xiong, Z.: Reversible Image Watermarking Using Interpolation Technique. IEEE Trans. Inf. Forensics Security 5(1), 187–193 (2010)
20. Yang, B., Schmucker, M., Niu, X., Busch, C., Sun, S.: Approaching Optimal Value Expansion for Reversible Watermarking. In: Proceedings of ACM Multimedia and Security Workshop, New York, USA, pp. 95–101 (2005)

A New Watermarking Method with Obfuscated Quasi-Chirp Transform

Kazuo Ohzeki[1], YuanYu Wei[1], Yutaka Hirakawa[1], and Kiyotsugu Sato[2]

[1] ISE, Shibaura Institute of Technology, 3-7-5 Toyosu, Koutouku, Tokyo 135-8548 Japan
[2] IE, College of Industrial Technology, 1-27-1 Nishikoya, Amagasaki, Hyogo, 661-0047 Japan
{ohzeki,hirakawa}@sic.shibaura-it.ac.jp,
m710101@shibaura-it.ac.jp, kiyo@cit.sangitan.ac.jp

Abstract. Watermark detection software is obfuscated using a table to hide embedding and detection algorithms. As the table size is limited, the block size is also limited for watermarking. To address this situation, a new quasi-chirp transform is developed to improve embedding efficiency. The quasi-chirp transform is different from the conventional DCT or Fourier transform. It contains multiple frequency components in a single basis of the transform. It disperses image data rather than compressing it, as the DCT does. The dispersed data increases the range for embedding watermarks. The chirp transform is able to embed even on a flat area of an image. Using this chirp transform, embedding and detection experiments for image data with small block sizes were carried out. A high SNR and robust watermark with an evaluated obfuscation were obtained.

Keywords: orthogonal transform, quantization, obfuscation, embedding, detection.

1 Introduction

Obfuscation of watermarking embedding software and detection software is important to improve security. Even for private watermarking, it is necessary to disclose detection software outside the owner's management area when needed for copyright detection events, such as copyright dispute problems. By hiding detection software at all times using obfuscation, analysis of the detection software is prevented. This watermarking system can also be as changeable as fingerprinting, since there are many parameters to the construction of our transform. By realizing the obfuscation of watermark embedding and detection software, we can use the watermarking system for a long time without making any changes to the framework system, the only requirement being to set parameters for individual users. These effects are still valid for a private watermarking system.

There are several different methods of software obfuscation, such as Collberg's basic and many other methods [1], the recent Drape's summary [2], and Barak's impossibility proof [3-4]. Many of Collberg's examples make software programs difficult to read for humans, but the converted programs can be analyzed and can be

Y.Q. Shi, H.J. Kim, and F. Perez-Gonzalez (Eds.): IWDW 2011, LNCS 7128, pp. 57–71, 2012.

recognized with some degree of effort because the methods are not theoretically guaranteed. On the other hand, Barak et al. proved the impossibility of obfuscation, and presented their findings in a paper [3]. Barak's method of proving impossibility is very difficult and we will not discuss how to relate the proof to our obfuscation method here. However, we will present several different points in the assumptions between Barak's method [3] and our proposed obfuscation method.

Our obfuscation method is realized using a table function with only an input and an output interface. This method ultimately hides all clues and traces of program execution. The method collects all calculated results in advance and stores them in a Read Only Memory (ROM) or constant array in a software program. In our system, in the obfuscation fields, an array method is implemented that uses an array variable to make the software more complicated, instead of using a single variable [2]. Therefore, we would like to call this a ROM method [5] in this paper instead of an array method. One of the differences between the proof of the impossibility of obfuscation and our method is that the proof is carried out on a continuous domain and utilizes infinite elements available in real numbers, while our method is carried out on a discrete domain with only a finite number of elements available. A second difference is that the proof states that the obfuscation is a compiler and the obfuscated program is still a program to calculate something, while our method is not a language-based compiler but a dictionary, and it collects all the results. Another point is that the proof is based on a virtual black-box compiler forming an oracle machine, while our method is based on an actual white-box ROM.

Chow et al. presented a white-box obfuscation method using a kind of table function [6]. This was for the obfuscation of encryption/decryption in the Data Encryption Standard (DES) method. The method converted part of an affine transform in processing key data into a table. It was reported that, as the obfuscated part was limited to the affine transform, there were still many other parts left that were not obfuscated. Also, the calculation workload was huge and the program was slow [6]. Our method is fast in implementation because accessing ROM or array data can be done in a single step as a minimum, and at most in several steps in the case of a multi-stage ROM construction. In both cases, the execution time is nearly zero, and thus is different from Chow's method. We will continue to use the word "function" for multiple output data, though it should really be called mapping. This is because a function produces only a single value, not a set of values. However, what we are talking about is multi-valued functions or vector functions. Chow's method was extended by Wyseur to become more like white-box cryptography. He classified the obfuscation methods into strong deterministic ones or weak probabilistic ones. The problem with conventional obfuscation definitions is that they have a number of infinite operations such as polynomial, non-polynomial, or a point function on continuous variables. Our policy is to treat the definition of obfuscation in a finite space.

This paper develops a new orthogonal transform, which has a special frequency characteristic. The paper also reports on experiments on watermark embedding and detection under JPEG attacks. The results outperform conventional DCT methods. Related work concerning obfuscation and watermarking with orthogonal transforms is described in Ch. 2. In Ch. 3., new quasi-chirp transform with block sizes of four and eight blocks are constructed for watermarking with obfuscation. Experiments and evaluation are carried out in Ch. 4. Finally, conclusions and further studies are described in Ch. 5.

2 Obfuscation Method

Objects of obfuscation are separated into two categories, these being data and programs. For data obfuscation, encryption keys, passwords, personal information and secret constants are listed [7]. For software program obfuscation, it is difficult to treat all kinds of software in a single framework [2][5] with any degree of generality. Recent papers on obfuscation concern analyses to discover malware and viruses that are obfuscated to hide themselves [8]. The papers describe sound practical applications using classical obfuscation methods [9]. Approaches to software obfuscation can be divided into two ways of thinking: the general all-purpose and the specific restricted types.

Obfuscation methods for general all-purpose software include many kinds of functions, from a simple constant or a linear with a single variable, to complex functions, such as the Bessel function with modifications. For a simple function, any obfuscation can be disabled through simple analysis. For example, a linear function with a single variable, such as y=ax+b, can be estimated by testing two pairs of inputs and outputs if we know that it is a linear function. However, if we are not aware that it is really a linear function in advance, analysis usually starts from a lower dimensional polynomial, moving to a higher one, beginning with a single variable and going on to multi-variables within an allowable time. Relatively speaking, we can easily hit the linear function “y=ax+b” in the process of a full search. Considering the above, obfuscation of a simple function is meaningless because it is understood by inspecting input and output relations in a short time period. Therefore, target functions for obfuscation should be restricted to a subset with more complexity than the contemporary numerical calculation technology.

A point function is used as an example of the target function for obfuscation [2]. However, the point function is not an appropriate for obfuscation because the function is a kind of constant function on a continuous domain. It is easy to break an obfuscated constant function. There are several differences between our obfuscation type and Barak’s type. One is that we are targeting complex functions with nonlinear quantizing operations. The second is that our type does not have a compiling process, while Barak’s type is defined as a compiler, and the converted program still works to process input data to produce output data. The third difference is that our type is a discrete finite procedure, while Barak’s is a proof of a continuous function on a continuous variable domain utilizing an infinite number of elements.

3 Watermarking with an Obfuscated Chirp Transform

We propose to obfuscate watermarking detection software. Because it contains one of the orthogonal transforms and a non-linear quantization, an analysis of this obfuscated table requires a set of multi-dimensional non-linear equations of unknown variables.

The basic proposed system is shown in Fig.1. From original image G, the owner of the image produces embedded image Gw in a secret region. Detection software is obfuscated and disclosed to the open public region when needed to detect the watermark. The obfuscated detector is only disclosed when an authentication inquiry occurs. The proof of authentication of an embedded watermark is still difficult, but these watermarks will contribute to finding the proportion of misused or tampered media from socially distributed images. A social observer monitors the distributed media and reports the ratios of correctly watermarked and tampered images to the owner.

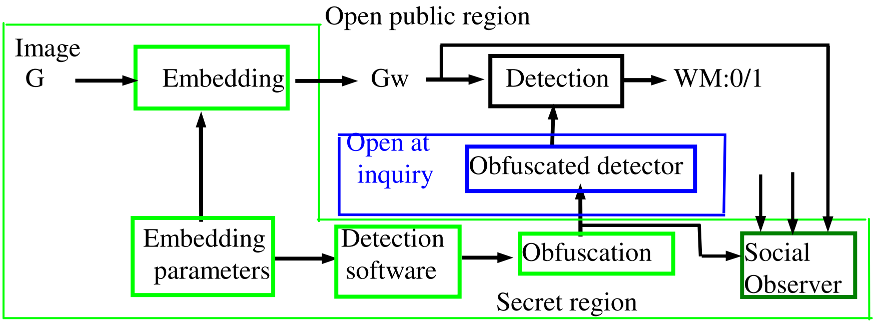


Fig. 1. The basic proposed watermarking system.

There are several types of watermarking system. These include the Private Watermarking (Non-Blind) system, which requires the original image and a key, and the Semi-Private (Semi-Blind) system, which requires a key and secret information [10][11][12]. The proposed method uses information from the original image and is classified as the Semi-Blind type. A Non-Blind method has been described in recent papers. This uses an orthogonal transform of the Naturalness Preserving Transform (NPT) [13]. This NPT is rather close to the conventional symmetrical Hadamard, DCT or Fourier transform, and is quite different from our proposed asymmetrical quasi-chirp transform.

The basic watermarking embedding that we are proposing is composed of processing methods to transform image block data and to quantize it with a specified width, as shown in Fig. 2. An orthonormal chirp transform is used. The most important parts of the operations are the transformation and quantization. The operations of the orthogonal transforms are carried out by first multiplying the coefficient values and input image luminance values, and then by summing them. To convert all calculation operations to corresponding rules in the data of a table, the size of the linear transform is restricted to 4 or 8. These chirp transform coefficient values have an appropriate width to perform similar kinds of embedding. Changing the coefficient values enables this watermark to act as a type of a fingerprint to use different coefficients for individual users. An example of the fourth case will be used in the following paragraphs.

Four pixels of the luminance components of the input image data are transformed. The first transformed component is the same class as the so-called direct current (DC). The second transformed component is quantized by a prescribed step-size for watermark embedding. Then, applying an inverse transform to the quantized data,

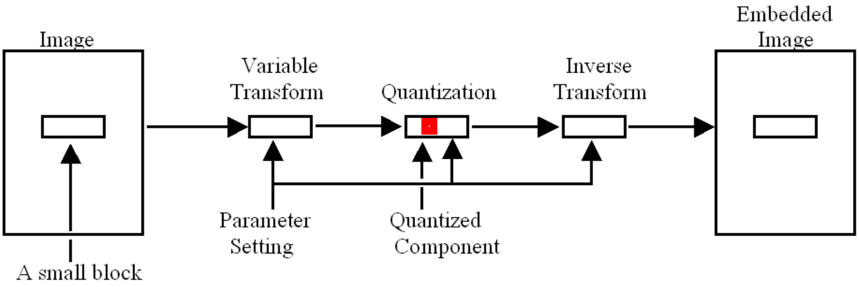


Fig. 2. A basic watermarking embedding structure with a transform of image block data and quantization

we obtain embedded image luminance data. The watermark is detected by inspecting the quantized patterns. If the transformed value is in a range of an interval that is quarter the prescribed step-size, we decide that this is the watermark.

Fig.3 shows the table function structure for 4 pixels of image data, each with 8- bit inputs and 16-bit outputs, with intermediate tables of 24-bit inputs and 16-bit outputs. X,Y,Z and W represent 8-bit input luminance data. To reduce the size of the table, we divide the table into three small tables. This structure corresponds to a single transformed output. Four sets of this structure complete the total four pixel transform.

The intermediate outputs with 16 bits are permutated randomly before being output. They are re-sequenced at the first stage in the next ROM. The inverse transform after quantization can be constructed with the same kind of structure. An evaluation of the table sizes is listed in Table 1. The size of the ROM1 is 128KB, and the ROM2 and the ROM3 are 32MB. The quantization can be included in the final part of the ROM3. Also, the ROM1 can be included in the ROM2. The total size of the ROMs is 64MB for a single output. The inverse transform, whose inputs are rounded to 8 bits after quantization, can be constructed with 32MB for each single output, as in Fig. 3. As mentioned above, the size of the embedding table is (64MB+64MB)x4=512MB, and the size of the detection table is 64MBx4=256MB.

When we carry out watermark embedding by using the ROM method of obfuscation, the size of the image block is practically restricted to about 4 to 8 pixels using present technology. For a larger size of transform, there are many candidate

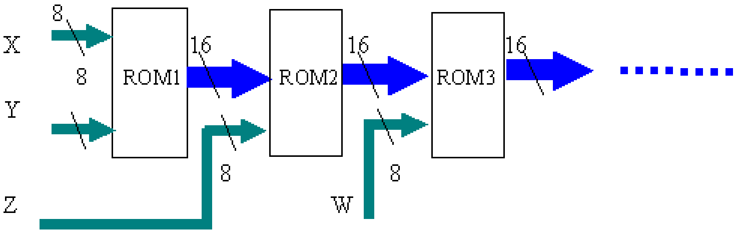


Fig. 3. A table function structure for 4 pixels of image data. X,Y,Z and W represent 8-bit input luminance data. Intermediate tables have 24-bit inputs and 16-bit outputs.

Table 1. Table size estimation

	ROM1	ROM2	ROM3
Input Address	16	24	24
Output	16	16	16
Amount (bits)	$2^{16}\times 16$ $=1048576$	$2^{24}\times 16$ $=268435456$	$2^{24}\times 16$ $=268435456$
Capacity	128KB	32MB	32MB

positions to be embedded, except for the DC component [14]. However, for the case of a transform with a block size of 4, only the second or the third components are candidates for embedding. We will fix the position of the component to the second in this paper. The DCT has symmetrical coefficients in the sense that the sum of positive and negative coefficients, excluding the DC, is zero. The luminance values in a smaller block size of 4 tend to be almost the same value.

The Slant transform [14] could be one of asymmetrical transforms. The Slant transform could have a set of slanted coefficients, as its name suggests. Though the conventional Slant transform has very slanted coefficients, it is still symmetrical, and is similar to DCT.

To observe the embedding problem for a transform with a block size of 4, a preliminary examination was carried out using test images. Data transformed by the DCT with a block size of 4 is shown in Fig. 4. These kinds of distribution are well-known and many examples are also reported for the case of 8 [15]. The distribution varies depending on the image region. For a flat region, image energy concentrates on the first component (DC) and the second component becomes nearly zero. In this situation, embedding is usually skipped [16]. The reason the second component becomes nearly zero is that the coefficients of the DCT are symmetrical. For an even number block size, an asymmetrical transform solves this problem. Or, for an odd number block size, for example if we take a block size of 5, DCT coefficients can be asymmetrical. Here, we use even numbers of 4 and 8 to allow direct comparison with the DCT.

A skew transform, which did not use orthogonal coordinates but skew coordinates, was attempted by Yamane et al [14] for picture coding in order to adapt slanted lines. Yamane et al produced the skew transform by projecting the DCT to the skew coordinates. There were many varieties of the skew transform, depending on the local shapes. Therfore, we hope to integrate the skew features into a single transform. To increase the chances of embedding in the transformed components, it is important that the absolute value of the transformed data is large enough for quantization. For this requirement, the transform coefficients should be unbalanced, not in ways of DCT or Fourier transform. Therefore, to obtain a more effective transform for embedding, one consideration is to have a design that causes conflict with the image data, and another is to include various frequency components in the design. Considering the above views, a chirp signal whose frequency increases as time passes on the horizontal axis is introduced. The chirp signal is described as in the analogue formula,

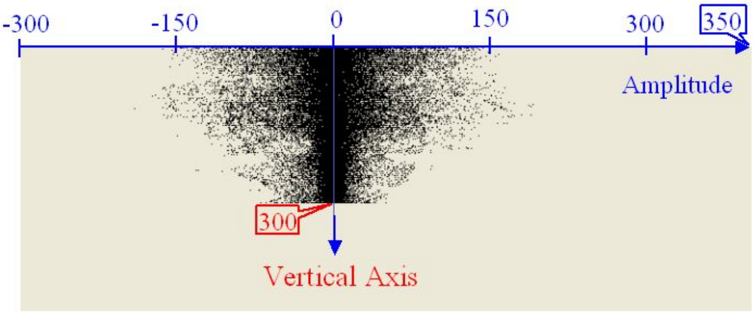


Fig. 4. The second component of data transformed by the DCT with a block size of 4 are displayed as amplitude in the horizontal direction. The original image has a height of 300.

$$C(t) = A \cdot \cos(2\pi f(t) \cdot t + \varphi)\,. \tag{1}$$

The function f(t) generates characteristic. Discrete examples are shown in a caption in Fig.5. The conventional DCT or Fourier transform consists of a single frequency for a single basis and is symmetrical. However, the chirp signal has multiple frequency components and can be asymmetrical when cut out at fractional positions.

Differences to the so-called spread-spectrum methods are listed in table 2. In the chirp method, signal components of low and middle frequency are transformed into a single value, while the spread spectrum transforms a signal to many high components. To embed is to quantize a single component for the chirp, while the spread spectrum uses many distributed components.

The proposed transform aims to preserve the lower frequencies that images usually have, and to include multiple frequency components. Below, we will show the 4th and 8th chirp transforms. We will make use of the second transformed component for watermark embedding. This means that we are not greatly concerned about the moment with other transformed components, i.e. the third and fourth components. Higher frequency components are not necessary for the second coefficient. After manually deciding the second coefficient, other coefficients are supplemented using conventional DCT bases. As the second coefficient is asymmetrical, the first coefficient is not made flat like the DC because of the orthogonal condition. The 4th and 8th chirp transforms are shown in Fig. 5.

Table 2. Comparison between spread-spectrum and chirp methods

items	spread spectrum	chirp
transform	signal to frequency n:n	n:1
frequency range	high	low, middle
embedding	statistically distributed frequency values	quantization of a non-zero value at fixed position
detection	summing up distributed values	decision on a quantized value

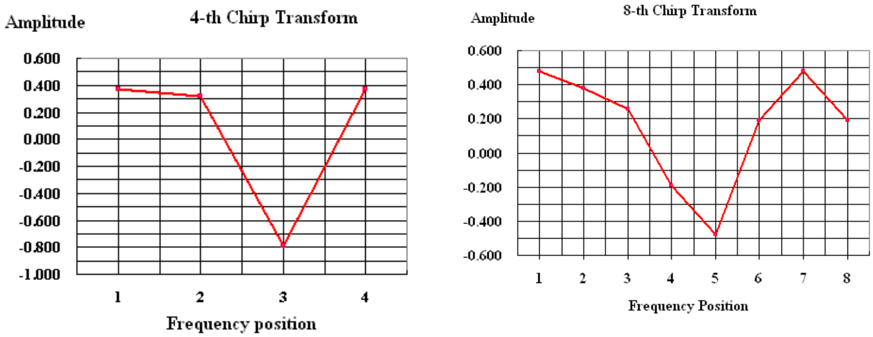


Fig. 5. The second component of the 4th chirp transform (left) and the 8th chirp transform (right). The generating functions are,

$$f_4(t_n) = \frac{1}{2\pi}\left(0.066t_n^2 + 0.351t_n - 0.364\right), \quad \varphi_4 = 1.213, \quad n = 0,..,3$$

$$f_8(t_n) = \frac{1}{2\pi}\left(1.07t_n^6 + 1.18t_n^5 + 1.31t_n^4 + 1.76t_n^3 + 2.07t_n^2 + 4.9t_n + 6.76\right),$$

$$\varphi_8 = 11.19, \quad n = 0,...7.$$

The first basis is made after the second basis, using orthonormal conditions. As an orthonormal transform, it does not need to be flat as in the case of the DCT. In fact, the Karhunen Loeve (KL) transform, which is the optimum transform in the squared error criteria, generated from each set of image data, does not have a flat basis for the first component. Also, as the second basis is asymmetrical, the first basis cannot be flat, by the orthogonal condition. The first basis is defined by manipulating a flat basis to be orthogonal to the second basis. After that, the third and fourth bases are determined. Only the transformed second component is processed for embedding. Other components are not processed for embedding, but they are necessary for the inverse transform. Thus, it is not important to be concerned about the characteristics of the third and fourth bases. Here, we utilize the third and fourth bases of the DCT as seeds to obtain them. For the DCT bases, using Gram-Schmidt orthogonalization, orthonormal bases are produced one after another. The complete chirp transforms are shown in Fig. 6. During the generating process, since absolute values of coefficients near to zero are less contributive to transformation, larger sets of coefficients are selected through iterative trials by adjusting the seed data. Transformed data distributions are shown in Fig.7. These are widely spread and are different from Fig.4. Through this divergence of the transformed data, the number of non-zero components increases and the area in which we can embed watermarks expands.

$$\begin{bmatrix} 0.151 & .200 & .265 & .497 & .646 & .299 & .151 & .299 \\ .476 & .381 & .257 & -.190 & -.476 & .190 & .476 & .190 \\ .299 & -.002 & -.439 & -.242 & .163 & -.345 & -.137 & .704 \\ .416 & -.103 & -.502 & -.277 & .295 & .482 & .089 & -.401 \\ .603 & -.403 & .001 & .565 & -.244 & -.225 & -.150 & .146 \\ -.225 & -.504 & -.170 & .086 & .034 & -.121 & .785 & .105 \\ .017 & -.614 & .388 & -.279 & -.073 & .467 & -.242 & .336 \\ .271 & -.104 & .490 & -.420 & .420 & -,490 & .104 & -.271 \end{bmatrix}$$

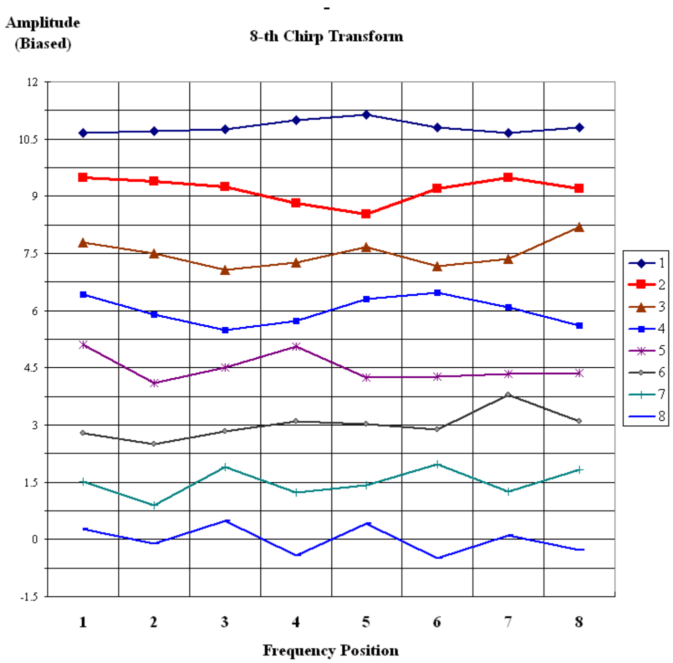


Fig. 6. Eighth chirp transform. The upper shows transform matrix coefficients. The lower shows the waveform of eight vectors.

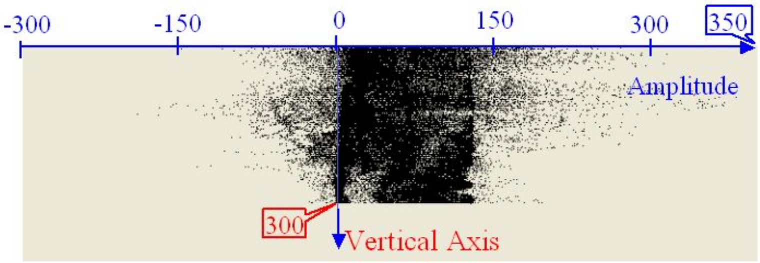


Fig. 7. The second components of transformed data by the chirp transform with a block size of 4 are displayed as amplitude in the horizontal direction. The original image has a height of 300.

4 Experiments and Considerations

Experiments are carried out according to Fig. 8. R,G and B elements of colour image data are first adjusted on the histogram distribution as a pre-process to avoid overflow in the embedding process. R,G B are converted to Y,I,Q, which represent a luminance and two colour difference signals in the formula (2).

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.212 & -0.523 & 0.311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{2}$$

Only the luminance value Y is modified in watermark embedding. The inverse colour conversion formula is shown in (3). Usually, three digits are sufficient to keep LSB invariant after inverse conversion. As embedding quantization is a non-linear operation, it is very important that the figures are precise. Colour differences I and Q are held without any change during the embedding process. Blocks with a size of 4 or 8 in the Y signal are transformed by the chirp transform. Also, DCT is used as a reference. The second components of the transformed data are quantized for watermark embedding (QIM). The quantization step sizes used in these experiments are fixed values of 4, 8, 16, 32, and 64. This quantization acts as watermark embedding. The numbers of embedded blocks are from 2 to 24, depending on the size of the images. After embedding, the inverse orthogonal transforms (chirp or DCT) are applied, which results in obtaining embedded luminance data, Yw. Together with the saved I and Q signals, the embedded colour image data, Rw, Gw, and Bw are recovered using inverse colour conversion (3).

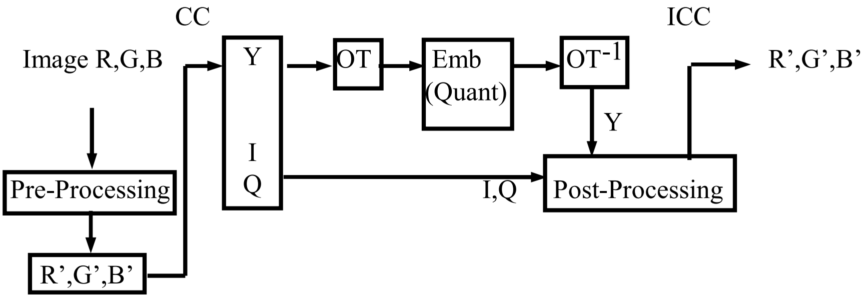
$$\begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = \begin{bmatrix} 1 & 0.954889204321426 & 0.622103935020897 \\ 1 & -0.27135478274584 & -0.647512025865468 \\ 1 & -1.10725100544121 & 1.70246037378756 \end{bmatrix} \begin{bmatrix} Y \\ I \\ Q \end{bmatrix} \tag{3}$$

The precision of the values should be as great as possible. The conversion from YIQ to RGB is a regular linear transform, which has an inverse transform that is reversible. However, the watermark embedding process is not reversible in general. If a point in the YIQ-space is caused to move outside this space by the embedding process, the reconstructed RGB data will exceed the proper range of $0 \le R, G, B \le 255$. There are two possible ways of coping with this problem:

[Y1]: After reconstruction of RGB, all data outside the range of $0 \le R, G, B \le 255$ are moved to the nearest end, 0 or 255.

[Y2]: Depending on the Y value, I or Q values are moved in the direction of the smaller value in an absolute value sense within a two-dimensional space of I and Q, without any change of the Y value.

In these experiments, the method [Y2] is used under the condition that the maximum movement is limited to 20. If the quantization step size during embedding is large, the [Y1] clipping process is applied after [Y2]. This is the post-processing shown in Fig. 8. Finally, the ICC mentioned above is carried out.



CC: Colour Conversion
OT: Orthogonal Transforms (DCT or Chirp_Tr)
Emb:Embedding of Watermark
ICC:Inverse Colour Conversion

Fig. 8. Experimental Embedding

For the embedded image without any change of attack, all watermarks are detected. At first, detection experiments for the original images without any watermarks are tried. The detection ratios are shown in Fig.9. As embedding is quantization, a detection probability of 1/2 is obtained on average for a random signal. For an image with the size of 720x480, there are 24 embedded blocks in our experiment. Then the false detection ratio is,

$$1/2^{24} \tag{4}$$

Robustness is shown in Fig.10. The robustness for the chirp transform is superior to the result of the conventional DCT in the case of a wave attack. The performance is nearly the same in the case of a JPEG attack.

In the next paragraphs, some considerations are described.

[C1]: The Number of Parameters and Attacks

For the fingerprinting applications in watermarking, we can change the coefficient values and quantization step sizes for individual types of embedding usage. In the transforming process, inversion of the DC component provides the maximum effect. For example, for a coefficient whose value is 0.5, the transformed DC value for an average image level of 128 is 256. The error sensitivity that is deduced by the excess of the reconstructed LSB level is about 1/128. Any coefficient change below this value does not usually affect reconstruction. On the contrary, if we change the transform coefficient by 1/128, the resultant LBS will usually change. For a coefficient whose value is 0.5, if we determine an allowable range with an interval from 0.4 to 0.6, then there are 51 varieties of coefficients, 0.400, 0.4004, 0.4008,..,0.600 for the error criteria 1/100. For the 4th transform, which has 16 coefficients, we can obtain 51^{16} varieties of transforms. These changed transforms deviate from the primary orthonormal transform, but it is not necessary that the transform should be strictly orthogonal or normal. It is only necessary that the transform have an inverse. During this deviation processing, the reliability of the deviated transform being a regular matrix should be high because the deviation is very small. For the eighth transform, there are 64 coefficients for variation. As for quantization in the experiments in this paper, the step sizes are fixed at a value from 4 to 64. To maintain robustness, the step sizes should preferably be from 16 to 64. Arbitrary step sizes can be used in this range. One step size value which is a multiple of the another causes the inclusion of the one case with the other case during detection. To make this situation clear, we should use prime numbers as the step size values. There are 11 prime numbers between 16 and 64, so the number of embedding variations increases by more than ten times through the quantization. We must solve simultaneous equations for 64 unknown coefficients even if we know the embedded positions of the image data.

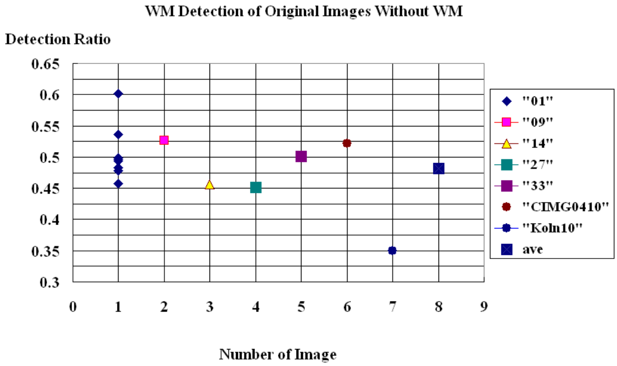


Fig. 9. Detection Ratio for original images without watermarks. The horizontal axis shows the number of images. The total number of blocks is 2214789. For the image “1”, varieties of quantization step-sizes of 2,4,8,16,32,64,128 and 256 are tested and 32 for other images.

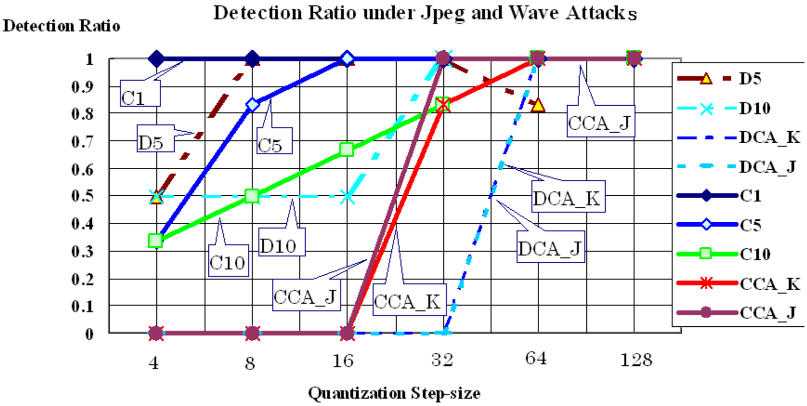


Fig. 10. Comparative view of robustness. Horizontal axis shows step-size of 4,8,16,32,64 and 128. Abbreviations are as follows; D=DCT, C=Chirp. Numbers 1,5,10 after C or D are JPEG compression ratios. CA is a wave attack. The wave attack adds wave components to embedded blocks.

The degree of freedom of parameters for the linear chirp transform with quantization is described in Fig.11. To find out all coefficients is an adversary's task. The structure of Fig.11 has the same structure as the three-layer Perceptron in a neural network. It cannot be solved when the middle layer of quantization exists, according to the theory of neural network studies [17]. The number of iterations to find a matched solution by a full search is estimated in Table 3. The calculation time for the block size of 2000 is 7.97 sec [18]. The time is divided by 7, the size decreasing by half each time, down to 16. For the sizes of 8 and 4, the division ratios are 6 and 5.

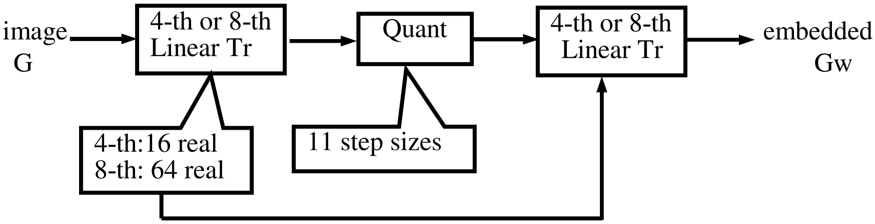


Fig. 11. Degree of freedom of parameters for watermarking embedding

A calculation time estimation for embedding using measured data is listed in Table 4. For the embedding, the calculation time is more than a billion years. The signal to noise ratio for the original image and an embedded image is 69.3dB as shown in Table 5.

The proposed obfuscation method satisfies the basic obfuscation conditions [2]. The execution speed of the obfuscated version by the ROM method is much faster than that of a non-obfuscated version. The output from the obfuscated version by the ROM method is exactly the same as that of a non-obfuscated version. As for the level of difficulty of embedding, it is considered that the proposed method achieves the maximum level because all clues are removed, and the input and output are the only data to be observed. One problem for the proposed ROM method is that it requires a large amount of memory. This places a restriction on realization of the obfuscation conversion, necessitating specific devices for individual cases.

Table 3. Estimation time for matrix multiplication. From a measured value 7.97 sec at the size 2000, empirical division rules by 7 to 5 are applied. values from [35] are averaged.

Size	Time [sec]	
4	3.2259E-07	↑ divided by 5
8	1.61295E-06	↑ divided by 6
16	9.6777E-06	↑ divided by 7
32	6.77439E-05	↑ divided by 7
65	0.000474207	↑ divided by 7
125	0.00331945	↑ divided by 7
250	0.023236152	↑ divided by 7
500	0.162653061	↑ divided by 7
1000	1.138571429	↑ divided by 7
2000	**7.97**	**measured average [18]**

As for detection, the total difficulty of discovering the coefficients was not established because of the small of the number of computations.

Table 4. Estimated calculation time for embedding using measured data, Core2Duo, 2.66GHz (E6700) 21.28GFLOPS and Table 3

size		Matrix Multiplication		Linear Equation		Coefficients		Full Search[sec]
4	(	3.2E-7	x2+	4.9E-7	)x	51^16	=	1.1E20
8	(	1.6E-2	x2+	2.4E-6	)x	70^64	=	5.6E112

Table 5. S/N evaluation data

Image name (size)	Quantization step-size	Block size	Number of blocks	MSE	S/N
01_c20(car) (400,300)	64	8	4	0.077	69.3dB

5 Conclusions and Further Study

A new watermark method is proposed using an obfuscated embedding and detection algorithm. To protect this algorithm is effective when a detection event is called. The obfuscation, called the ROM method, hides all calculation and quantization clues and only exposes input and output data. To realize the ROM method, a small orthonormal transform, the chirp transform, which is preferable to watermarking, is developed. The chirp transform outperforms the conventional DCT with regard to position-free embedding and robustness. The chirp transform has multiple frequency components, and thus provides a larger absolute value of transformed coefficients and distributes the transformed coefficients over a larger range. It is different from the spread spectrum method in the form and the method of embedding. The degradation of this watermark is evaluated by S/N. As the S/N is high, we can increase the number of embedding blocks. Evaluations of robustness take into account JPEG attacks. The robustness of the proposed watermark is nearly the same as conventional methods for JPEG attacks. The proposed obfuscation completely hides the embedding algorithm because, to analyze this watermark, non-linear simultaneous equations for 64 unknown coefficients must be solved even if the embedded positions of the image data are known.

This watermark will contribute to finding the proportion of misused or tampered media from socially distributed images.

We can improve the robustness of the watermark because degradation is small with this construction. As the obfuscation of the detection part is not sufficient at present, a non-linear element should be incorporated in the transform in the future.

References

1. Collberg, C., Thomborson, C., Low, D.: Manufacturing cheap, resilient, and stealthy opaque constructs. In: Proceedings of the 25th ACM SIGPLAN-SIGACT, pp. 184–196 (1998)
2. Drape, S.: Intellectual Property Protecting Using Obfuscation. CS-RR-10-02 (March 2009); Research sponsored by Siemens AG, Munich
3. Barak, B., Goldreich, O., Impagliazzo, R., Rudich, S., Sahai, A., Vadhan, S.P., Yang, K.: On the (Im)possibility of Obfuscating Programs. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 1–18. Springer, Heidelberg (2001)
4. `http://www.cs.princeton.edu/~boaz/Papers/obf_informal.html`
5. Ohzeki, K., Cong, L.: Fingerprinting System Depending On An Anonymous Third Party Authentication Using An Assumption of Computationally Measurable Obfuscation. CSEC-32, pp. 61–66 (March 2006) (in Japanese)
6. Chow, S., et al.: White-Box Cryptography and an AES Implementation. In: Nyberg, K., Heys, H.M. (eds.) SAC 2002. LNCS, vol. 2595, pp. 250–270. Springer, Heidelberg (2003)
7. Upham, D.:Steganographic algorithm JSteg, `http://zooid.org/paul/crypto/jsteg`
8. Kodovský, J., Fridrich, J.: Quantitative Structural Steganalysis of Jsteg. IEEE Transactions on Information Forensics and Security, 681–693 (December 2010)
9. Kodovský, J., Fridrich, J.: Quantitative Steganalysis of LSB Embedding in JPEG Domain. In: Proc. ACM Multimedia and Security Workshop, pp. 187–198 (September 2010)
10. Fahmy, G., et al.: Nonblind and Quasiblind Natural Preserve Transform Watermarking. EURASIP Journal on Advances in Signal Processing, Article ID 452548 (2010)
11. Aliwa, M., et al.: A New Novel Fidelity Digital Watermarking Based on Adaptively Pixel-Most-Significant-Bit-6 in Spatial Domain Gray Scale Images and Robust. American Journal of Applied Sciences (7), 987–1022 (2010)
12. Katzenbeisser, S., et al.: Information Hiding Techniques for Steganography and Digital Watermarking, 1st edn., p. 220. Artech Print, Canton Street Norwood (1999)
13. Yarlagadda, R., Hershey, J.: A naturalness-preserving transform for image coding and reconstruction. IEEE Trans. ASSP 33, 1005–1012 (1985)
14. Yamane, N., Morikawa, Y., Nariai, T., Tsuruhara, A.: An Image Coding Method Using DCT in Skew Coordinates. The Trans. of the IEICE J81-B-1(2), 110–117 (1998) (in Japanese)
15. Adachi, T., Hasegawa, M., Kato, S.: Study on a Watermarking Method for Still Images Using DCT. ITE Technical Report 23(62), 17–22 (1999) (in Japanese)
16. Kodovský, J., Fridrich, J.: Calibration Revisited. In: Proc. ACM Multimedia and Security Workshop, Princeton, NJ, pp. 63–74 (September 2009)
17. Minsky, M., Papert, S.: Perceptrons; an introduction to computational geometry. MIT Press (1969)
18. Kusuhara, H.: Numerical Library (March 2009), `http://www.rcs.arch.t.u-tokyo.ac.jp/kusuhara/fswiki/wiki.cgi`

A Novel Fast Self-restoration Semi-fragile Watermarking Algorithm for Image Content Authentication Resistant to JPEG Compression

Hui Wang[1], Anthony TS Ho[1], and Xi Zhao[2,⋆]

[1] University of Surrey,
Guildford, UK, GU2 7XH
{h.wang,a.ho}@surrey.ac.uk
[2] Shenzhen Institute of Advanced Technology,
Chinese Academy of Science, Shenzhen 518055, China
xi.zhao@siat.ac.cn

Abstract. In the past few years, semi-fragile watermarking has become increasingly important to verify the content of images and localise the tampered areas, while tolerating some non-malicious manipulations. Moreover, some researchers proposed self-restoration schemes to recover the tampered area in semi-fragile watermarking schemes. In this paper, we propose a novel fast self-restoration scheme resisting to JPEG compression for semi-fragile watermarking. In the watermark embedding process, we embed ten watermarks (six for authentication and four for self-restoration) into each 8 × 8 block of the original image. We then utilise four (4×4) sub-blocks' mean pixel values (extracted watermarks) to restore its corresponding (8×8) block's first four DCT coefficients for image content recovering. We compare our results with Li *et al.* and Chamlawi *et al.* DCT related schemes. The PSNR results indicate that the imperceptibility of our watermarked image is high at 37.61 dB and approximately 4 dB greater than the other two schemes. Moreover, the restored image is at 24.71 dB, approximately 2 dB higher than other two methods on average. Our restored image also achieves 24.39 dB, 22.98 dB 21.18 dB and 19.98 dB after JPEG compression QF =95, 85, 75 and 65, respectively, which are approximately 2.5 dB higher than other two self-restoration methods.

Keywords: Semi-fragile Watermarking, Image Content Authentication, Self-restoration, JPEG compression, Linear Regression.

1 Introduction

With the rapid development of multimedia technology, digital image evidence has been used in a variety of applications, such as crime scene investigation, traffic enforcement application, news reporting, medical imaging and electronic

[⋆] Formerly with the University of Surrey, UK but is now with Chinese Academy of Sciences, China.

Y.Q. Shi, H.J. Kim, and F. Perez-Gonzalez (Eds.): IWDW 2011, LNCS 7128, pp. 72–85, 2012.

commerce. However, the popularity and affordability of advanced digital image editing tools, allow users to manipulate images relatively easily and professionally. Fragile and semi-fragile digital watermarking techniques are often utilised for image content authentication applications to verify or authenticate the integrity of the digital media content. Fragile watermarking schemes are designed to detect any possible manipulations that affect the watermarked image pixel values [1,2,3]. In comparison, while fragile watermarking is aptly named because of its sensitivity to any form of attack, semi-fragile watermarking is more robust against attack, and can be used to verify tampered content within images for both malicious and non-malicious manipulations [4,5,6,7]. In addition, semi-fragile schemes make it possible to verify the content of the original image, as well as permitting alterations caused by non-malicious (unintentional) modifications such as system processes. During the image transmission, the mild signal processing errors caused by signal reconstruction and storage, such as transmission noise or JPEG compression, are permissible. However, the image content tampering such as copy and paste attack will be identified as a malicious attack.

Recently, some researchers proposed self-restoration schemes that the content of tampered areas could be recovered after the authentication process [4,8,9,10,11]. Fridrich and Goljan [8] proposed a Least Significant Bit (LSB) based self-correcting scheme and further adopted by Ho *et al.* [4] and Xi *et al.* [9] for semi-fragile watermarking. In these LSB based schemes, the original image is first watermarked. Simultaneously, the original image is also divided into 8×8 sub-blocks, each sub-block is then compressed by discarding the high frequency coefficients. Accordingly, 64 bits for each block are acquired after compression and then encrypted by utilizing a key. Obtained blocks are then shuffled, e.g. the value of block 1 moves to block 50, the value of block 35 moves to block 10. Finally, the LSBs of the watermarked image are replaced with these 64 bits for each block that were compressed from the original image. Therefore, the tampered areas of the image could be restored by decompressing the correlated LSBs of the watermarked image. However, these LSB based recovery schemes could be distorted if the watermarked image has undergone a JPEG compression process.

In order to overcome this drawback, in this paper, we propose a novel fast self-restoration scheme resisting to JPEG compression for semi-fragile watermarking. The rest of this paper is organized as follows: in Section 2, the literature of self-recovery semi-fragile watermarking schemes that could tolerate to JPEG compression is reviewed. Section 3 presents our proposed watermark embedding process, and the detection, authentication and restoration processes are discussed in Section 4. Section 5 illustrates the feasibility of our proposed recovery method in comparing with Discrete Cosine Transform (DCT) coefficients. By comparing with Li *et al.* [12] and Chamlawi *et al.* [13] schemes, the experimental results are compared and analysed in Section 6. This is followed by conclusion and future work in Section 7.

2 Literature Review

As mentioned in Section 1, the LSB based recovery schemes could be distorted after JPEG compression. One of the first recovery methods for semi-fragile watermarking that could resist JPEG compression was proposed by Lin and Chang [14]. In their recovery scheme, the original image was first resized to its half size (e.g. from 512 × 512 to 256 × 256), then divided into 8 × 8 sub-blocks. Each block was applied with DCT and quantilised, and quantisation table was applied to obtain QF=25. The quantilised DCT coefficients were encoded by using Huffman coding. Accordingly, 24 bits information for each block were obtained, and embedded into four 8 × 8 blocks (six bits for each block) of original image for content recovery after the authentication process. However, the quality of recovered image was relatively low. Hasan [10] and Cruz *et al.* [11] proposed schemes based on the concept of region of interest (ROI) and region of embedding (ROE). The ROI (e.g. car registration number) was first selected and encoded to generate watermark sequence, then embedded into ROE (e.g. the rest of the car). Therefore, the ROI could be authenticated and restored by extracting the information from ROE. The results showed that their schemes could restore the tampered regions under JPEG compression. However, their method could only restore the ROI of image and the size of ROI was limited.

Mendoza-Noriega *et al.* [15] proposed a semi-fragile content recovery scheme by utilising a halftoning technique. The original image was converted into a half-tone image as watermarks, then embedded into middle-frequency of DCT coefficient blocks of the image. The Multilayer Perception neural network (MLP) was used to inverse the halftoning process for restoring the tampered areas. Their experimental results showed that the tampered areas could be recovered under mild JPEG compression such as QF=80. However, the quality of restored images was relatively low, and the computational complexity was also high.

Li *et al.* [12] proposed a scheme based on a relationship function of corresponding inter-blocks DCT coefficient of the image. In their scheme, the original image was first divided into 8×8 blocks, DCT was then applied to each block and quantised with a standard JPEG quantization table. These quantised DCT blocks of the image were first assigned into the exclusive precursor block and successor block for each block as pairs. The difference of one DC coefficient and two AC coefficients between these blocks were generated as watermarks, and then embedded into middle-frequency of a successor block. To authenticate the image, each block was analysed by detecting whether its DCT coefficients of neighbouring blocks satisfied the Relationship Function, and utilised the Relationship Vector Recovery (RVR) and the Adjacent Blocks Smooth Estimate Recovery (ABSER) method for content restoration. Furthermore, Chamlawi *et al.* [13] proposed a self-recovering algorithm based on the integer wavelet transform (IWT) and DCT. The original image was first applied with a 1-level IWT and applied DCT to its LL1 sub-band. Then, these DCT coefficients were quantised, zigzag ordered and further scaled into 8192 coefficients. The LH1 and HL1 sub-bands were then decomposed with IWT to obtain the LH2 and HL2 sub-bands, which were used to embed with the 8192 coefficients for the self-restoration process. Our proposed method for self-restoration process is also DCT coefficients based. In Section 6, we will compare our experimental results with Li *et al.* [12] and Chamlawi *et al.* [13] schemes in detail.

3 Proposed Watermark Embedding Method

In this section, we discuss our proposed watermark embedding method, as shown in Figure 1. The original image is divided into non-overlapping 8 × 8 blocks, DCT is then applied to each block. The first set of watermarks is embedded by modifying these DCT coefficients, that are randomly selected from the low frequency band of each 8 × 8 block by using a secret key. The watermark embedding algorithm is adapted and further improved from the QIM method [16], and given as follows:

$$y = \begin{cases} x, & r \in [\frac{T}{2} - \alpha T, \frac{T}{2} + \alpha T) \wedge w = 1 \\ x - r + \frac{T}{2}, & r \in [0, \frac{T}{2} - \alpha T) \bigcup [\frac{T}{2} + \alpha T, \frac{3T}{2}) \wedge w = 1 \\ x - r + \frac{5T}{2}, & r \in [\frac{3T}{2}, 2T) \wedge w = 1 \\ x, & r \in [\frac{3T}{2} - \alpha T, \frac{3T}{2} + \alpha T) \wedge w = 0 \\ x - r + \frac{3T}{2}, & r \in [\frac{T}{2}, \frac{3T}{2} - \alpha T) \bigcup [\frac{T}{2} + \alpha T, 2T) \wedge w = 0 \\ x - r - \frac{T}{2}, & r \in [0, \frac{T}{2}) \wedge w = 0 \end{cases} \tag{1}$$

where w denotes six watermarks for authentication of each block are pseudo-random binary sequence generated by using the key as a seed, and are embedded into the low-mid frequency band of each 8 × 8 block. x is the DCT coefficient of the host, y is the modified DCT coefficient, $T > 0$ determines the perceptual quality of the watermarked image, $\alpha \in (0, \frac{1}{2})$ control the scope, and $r = mod(x, 2T)$.

As shown in Figure 1, the original image is also divided into non-overlapping 8 × 8 blocks. Each block is further divided into four non-overlapping 4 × 4 sub-blocks. Four mean pixel values of each sub-blocks are calculated, and then normalised by multiplying a scaling factor. Figure 2 illustrates the four normalised mean pixel values, calculated from four 4 × 4 sub-blocks of a 8 × 8 block. These normalised mean values belong to the second set of watermarks, which are embedded into their corresponding 8 × 8 blocks by replacing the DCT coefficients, randomly selected from the low-mid frequency band of each block by using the key. To determine the location of corresponding blocks, we adopt the method proposed by Li *et al.* [12], since it provided a circle link that could accurately determine the neighbouring blocks in diagonal location. After the two sets of watermarks have been embedded, each block is applied with an inverse DCT and the watermarked image is then obtained.

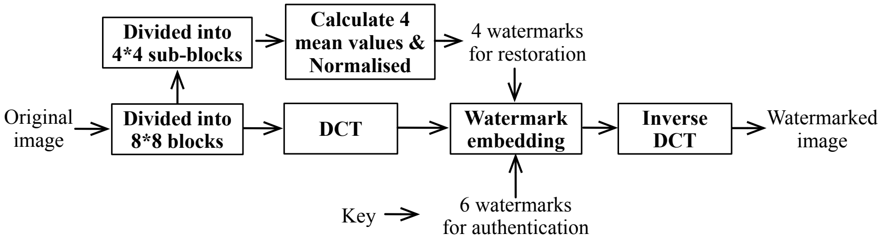
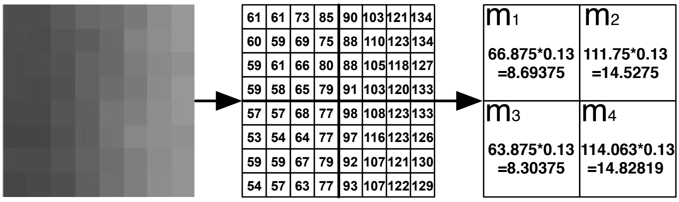

Divided into 4*4 sub-blocks
Calculate 4 mean values & Normalised
4 watermarks for restoration
Original image
Divided into 8*8 blocks
DCT
Watermark embedding
Inverse DCT
Watermarked image
Key
6 watermarks for authentication

Fig. 1. Our proposed semi-fragile watermark embedding process



61	61	73	85	90	103	121	134
60	59	69	75	88	110	123	134
59	61	66	80	88	105	118	127
59	58	65	79	91	103	120	133
57	57	68	77	98	108	123	133
53	54	64	77	97	116	123	126
59	59	67	79	92	107	121	130
54	57	63	77	93	107	122	129

m_1 66.875*0.13 =8.69375	m_2 111.75*0.13 =14.5275
m_3 63.875*0.13 =8.30375	m_4 114.063*0.13 =14.82819

Fig. 2. Example of four normalised mean pixel values that are calculated from a 8×8 block

4 Proposed Watermark Detection, Authentication and Restoration Method

Figure 3 illustrates the proposed watermark detection, authentication and restoration processes. The test image is divided into non-overlapping 8×8 blocks, DCT is then applied to each block. According to the watermark locations that are determined through the key, six watermarks of each block for authentication are first extracted as follows:

$$w' = \begin{cases} 1, & r \geq T \\ 0, & r < T \end{cases} \tag{2}$$

where w' denotes the extracted watermark. The retrieved watermarks of each block are then compared with the original watermarks w (generated from the key) to authenticate the block. If the total number of different watermarks is greater than a pre-determined threshold, then this would indicate that the block has been tampered with. The pre-determined threshold is a tolerance margin that controls the trade-off between false alarm and miss detection rate.

Once the tampered block has been identified, the restoration process is initialised. The second set of watermarks is extracted and denormalised for recovering the tampered blocks by using a secret key. Four extracted watermarks of each block are then restored into one DC and three AC coefficients by using the linear function as follows:

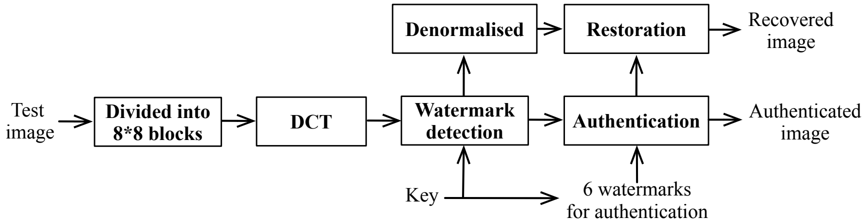
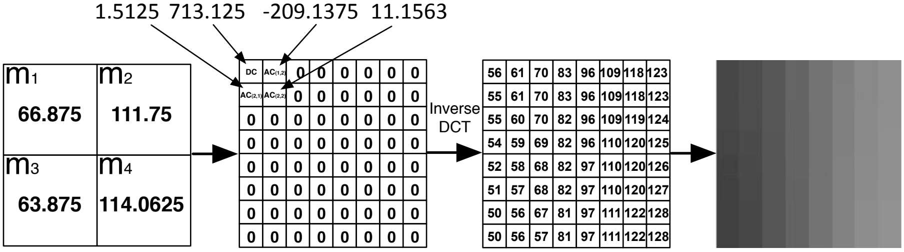


Fig. 3. Our proposed semi-fragile watermark detection, authentication and restoration process

$$
\begin{aligned}
DC_{(1,1)} &= a_1 \times (m_1 + m_2 + m_3 + m_4) + b_1 \\
AC_{(1,2)} &= a_2 \times (m_1 - m_2 + m_3 - m_4) + b_2 \\
AC_{(2,1)} &= a_3 \times (m_1 + m_2 - m_3 - m_4) + b_3 \\
AC_{(2,2)} &= a_4 \times (m_1 - m_2 - m_3 + m_4) + b_4
\end{aligned} \tag{3}
$$

where $DC_{(1,1)}$, $AC_{(1,2)}$, $AC_{(2,1)}$ and $AC_{(2,1)}$ are restored DCT coefficients, $a_1 = 2$, $a_2 = 2.2$, $a_3 = 2.2$ and $a_1 = 2.1$. These coefficients are determined through linear regressive analysis and will be discussed in more detail in the next section. $m_{1,2,3,4}$ are the four extracted watermarks. The values of $b_{1,2,3,4}$ are ignored in our scheme, as their values have negligible influence to the overall results. Finally, the recovered block is applied with an inverse DCT and the restored image is obtained. Figure 4 illustrates an example of four extracted and denormalised watermarks recovered from the first four DCT coefficients by using Equation 3. The remaining 60 coefficients were set to zero in a 8 × 8 block, and then transformed back into spatial domain that approximates the original block, as shown in Figure 2.



DC	AC(1,2)	0	0	0	0	0	0
AC(2,1)	AC(2,2)	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	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

56	61	70	83	96	109	118	123
55	61	70	83	96	109	118	123
55	60	70	82	96	109	119	124
54	59	69	82	96	110	120	125
52	58	68	82	97	110	120	126
51	57	68	82	97	110	120	127
50	56	67	81	97	111	122	128
50	56	57	81	97	111	122	128

Fig. 4. Example of a 8 × 8 block restored from four mean pixel values

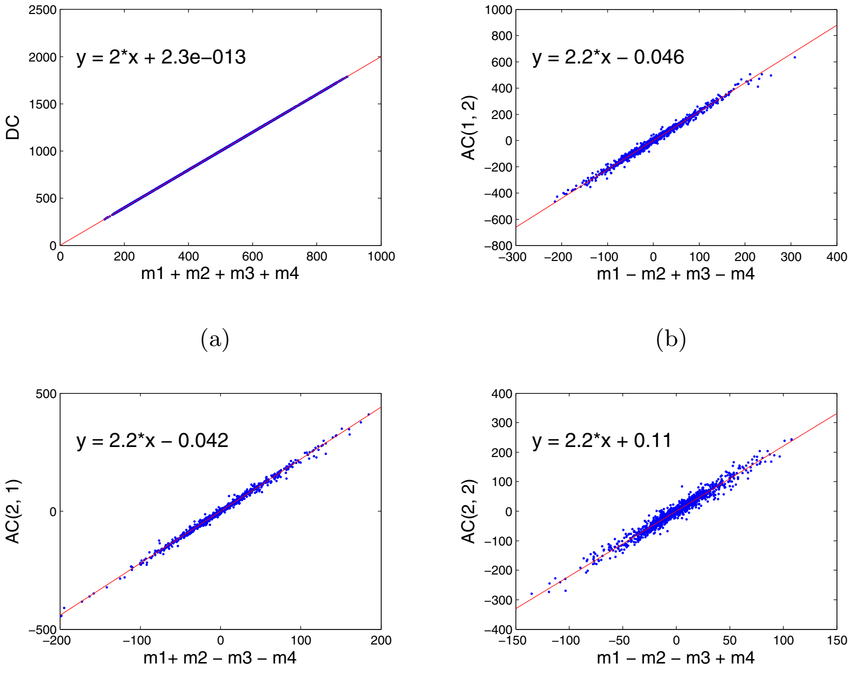
5 Feasibility Study of Proposed Recovery Method

In Sections 3 and 4, we discussed our self-restoration algorithm for semi-fragile watermarking. We utilised four (4 × 4) sub-blocks' mean pixel values to approximately restore its corresponding (8 × 8) block's first four DCT coefficients.

In this section, we will analyse the relationship between the four mean pixel values and their original DCT coefficients by using the linear regression and probability distribution.

5.1 Proposed Recovery Method in Linear Regression Approach

In Figure 5, the four scatter plots demonstrate the relationship between 8×8 DCT coefficients and the values calculated from their 4×4 sub-block's mean pixel values. These include the DC coefficients ($DC_{(1,1)}$) which correspond to the sum of their four mean pixel values as shown in Figure 5(a). The first AC coefficients ($AC_{(1,2)}$) correspond to $m1 - m2 + m3 - m4$ as shown in Figure 5(b). The second AC coefficients ($AC_{(2,1)}$) correspond to $m1 + m2 - m3 - m4$ as shown in Figure 5(c). Finally, the third AC coefficients ($AC_{(2,2)}$) correspond to $m1 - m2 - m3 + m4$ as shown in Figure 5(d). The linear fitting function ($y = a \times x + b$) is used for the linear regression analysis. We found that the cluster points are varying, but mainly closely fit along the diagonal line. Figures 5(a) illustrates the most fitted,



(a)

(b)

(c)

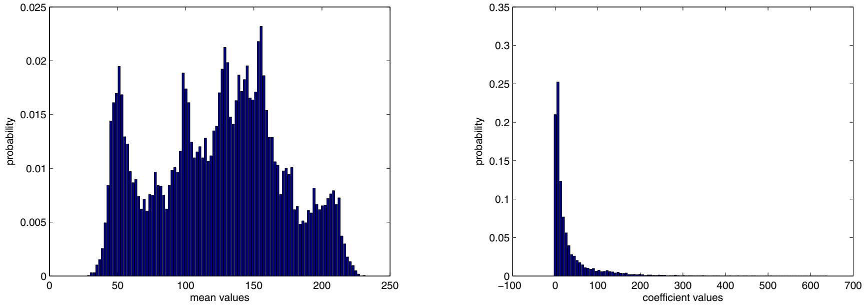
(d)

Fig. 5. Linear linear regression analysis for image *'Lena'*, where $m_{1,2,3,4}$ are the mean pixel values: (a) DC coefficients of each block vs. $m1 + m2 + m3 + m4$ (b) 1^{st} AC coefficients of each block vs. $m1 - m2 + m3 - m4$ (c) 2^{nd} AC coefficients of each block vs. $m1 + m2 - m3 - m4$ (d) 3^{rd} AC coefficients of each block vs. $m1 - m2 - m3 + m4$

whereas Figure 5(d) shows the least fitted into the linear function. Therefore, we conclude that there is a good approximate linear relationship between the first four DCT coefficients and its corresponding sub-block's mean pixel values. This further demonstrates that the four mean pixel values of each 4×4 sub-block are indeed feasible for restoring their corresponding first four DCT coefficients for our proposed self-restoration semi-fragile watermarking algorithm. We analysed 30 standard grayscale test images (512×512) to find the most fitted values during the recovery of the DCT coefficients, as given in Equation 3. The average fitting values used for our self-restoration scheme are $a_1 = 2$ for recovering $DC_{(1,1)}$, $a_2 = 2.2$ for recovering $AC_{(1,2)}$, $a_3 = 2.2$ for recovering $AC_{(2,1)}$ and $a_1 = 2.1$ for recovering $AC_{(2,2)}$, respectively. As a result, our self-restoration method could perform faster due to the simplicity in performing adding and subtracting operations for DCT coefficients during the recovering process.

5.2 Proposed Recovery Method in Probability Distribution Approach

In our experiment, we analysed the probability distributions for mean pixel values and first three AC coefficients, as shown in Figure 6. Figure 6(a) illustrates the histogram of 16384 mean pixel values calculated from all the 4×4 sub-blocks. We found that these mean pixel values were relatively uniformly distributed, with the range between 0 and 255 and the probability just below 0.025. However, the histogram of 12288 coefficients from the first three absolute AC coefficients of each 8×8 block were positively skewed distributed, as shown in Figure 6(b). The absolute values of these AC coefficients ranged from 0 to 700, and most of them concentrated between 0 and 100. By analysing the probability distribution, the AC coefficients were relatively difficult to normalise by using only one scaling factor. However, the mean pixel values could be easily scaled with a scaling factor. Therefore, the AC coefficients are less suitable than mean pixel values

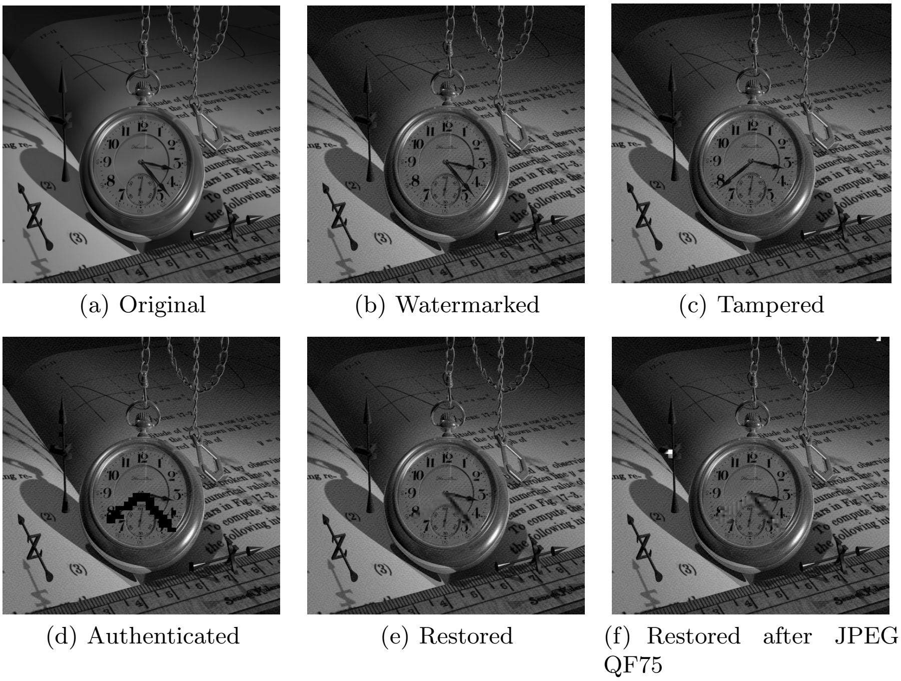


(a) (b)

Fig. 6. (a) Probability distribution of all 4×4 sub-blocks' mean values for image *'Lena'* (b) Probability distribution of all 8×8 blocks' first three absolute AC coefficients of each block for image *'Lena'* for use as watermarks for embedding in our proposed restoration scheme. As a results, our algorithm does not embed the DCT coefficients directly. In addition, we also analysed 30 standard grayscale test images and the results showed similar characteristics.

6 Experimental Results and Analysis

In this section, a number of experiments have been performed to evaluate the performance of the proposed watermarking scheme. Six grayscale images 'Lena', 'Baboon', 'Couple', 'Peppers', 'Boat' and 'Watch' (each of size 512 × 512) are used for our experiments. Figures 7(a)-7(f), show the original, watermarked, tampered, authenticated, restored and restored after JPEG compression QF=75 images for the image 'Watch', respectively. Figure 7(f) illustrates that our proposed semi-fragile watermarking and self-restoration scheme can still authenticate the image, localised and recover the tampered area approximately at JPEG compression QF=75. The quality of watermarked, restored and restored after JPEG compression images of proposed watermarking scheme are also compared with Li *et al.*'s [12] and Chamlawi *et al.*'s [13] watermarking and self-restoration schemes. To measure the quality of the watermarked and restored images objectively, the Peak Signal-to-Noise Ratio (PSNR) is used for comparison.



(a) Original (b) Watermarked (c) Tampered

(d) Authenticated (e) Restored (f) Restored after JPEG QF75

Fig. 7. Demonstration of the image *Watch* in proposed scheme

6.1 Watermarked Images Comparison

In Table 1, the imperceptibility between original image and watermarked images are compared with Li *et al.*'s [12] and Chamlawi *et al.*'s [13] methods. The watermarked image based on our proposed method achieved the highest at 39.12 dB for the higher texture image 'Baboon', while the other two methods' PSNR were just below 30 dB. Our lowest PSNR achieved to 36.55 dB for image 'Watch', which was still 3 dB higher than the other two methods. On average, images watermarked with our proposed method achieved approximately 4 dB higher than other two methods.

Table 1. PSNR (dB) comparison of the watermarked images

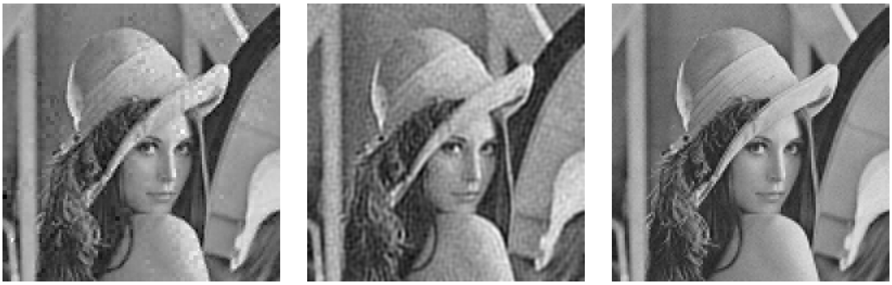
	Li's Method [12]	Chamlawi's Method [13]	Our method
Lena	36.37	36.23	**37.13**
Baboon	28.99	27.87	**39.12**
Couple	33.32	31.42	**37.91**
Peppers	34.96	34.20	**37.07**
Boats	34.45	32.35	**37.88**
Watch	33.30	33.66	**36.55**
Average	33.57	32.62	**37.61**

6.2 Recovered Images Comparison

In Table 2, we compared the restored images with their corresponding original images. The highest PSNR achieved 27.09 dB for image 'Lena', which was higher than Li's and Chamlawi's methods at 25.07 dB and 26.63 dB, respectively. On average, the performance of our method achieved 24.71 dB, which was approximately 2 dB higher than the other two methods. As shown in Figures 8-10, we compared the restored images, such as 'Lena', 'Peppers' and 'Boats' with Li *et al.*'s [12] and Chamlawi *et al.*'s [13] methods. The results also showed the quality of our restored images are higher, smoother and less blockness artifact than Li *et al.*'s and Chamlawi *et al.*'s methods.

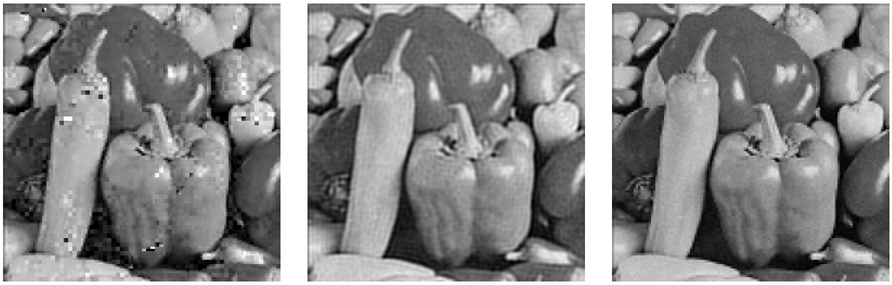
Table 2. PSNR (dB) comparison of the restored images

	Li's Method [12]	Chamlawi's Method [13]	Our method
Lena	25.07	26.63	**27.09**
Baboon	20.40	20.54	**20.78**
Couple	22.59	23.97	**24.46**
Peppers	23.71	25.23	**26.48**
Boats	23.14	24.54	**24.49**
Watch	23.57	12.82	**24.96**
Average	23.08	22.29	**24.71**



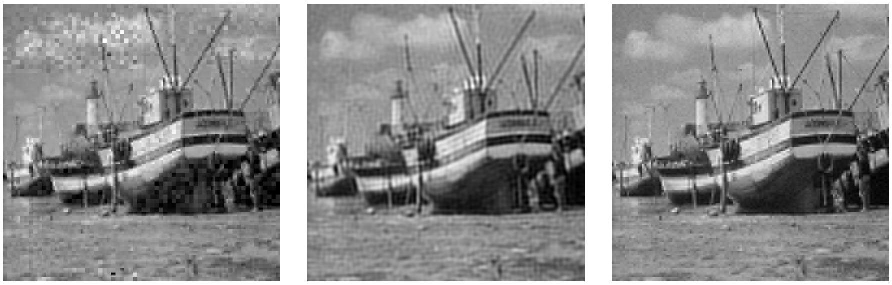
(a) Li's [12] (b) Chamlawi's [13] (c) Our method

Fig. 8. Comparison of restored image *Lena*



(a) Li's [12] (b) Chamlawi's [13] (c) Our method

Fig. 9. Comparison of restored image *Peppers*



(a) Li's [12] (b) Chamlawi's [13] (c) Our method

Fig. 10. Comparison of restored image *Boats*

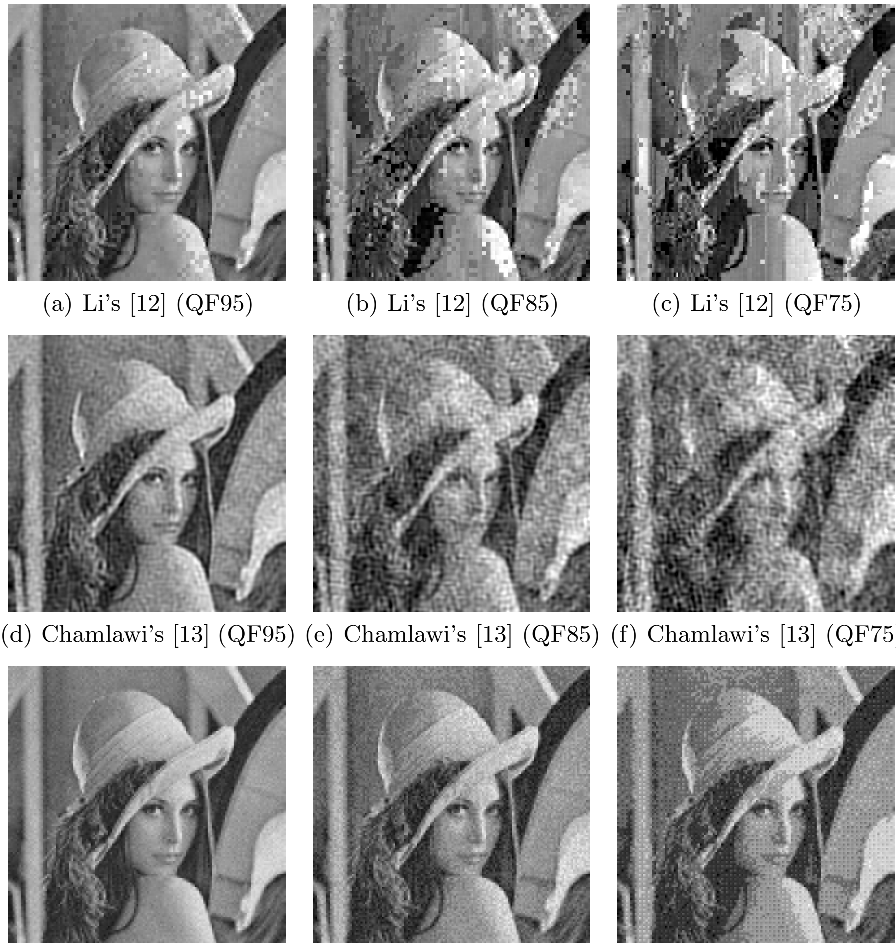
6.3 Recovered Images Robustness Comparison

As discussed in Section 2, semi-fragile watermarking for image authentication can tolerate some mild signal processing, such as JPEG compression. The ability to self-restore the tampered areas would definitely be an advantage. Thus, in this section, we analyse the performance between original images and the restored images after the watermarked images have been JPEG compressed.

Table 3. PSNR (dB) comparison of restored images after JPEG compression

	QF95			QF85			QF75			QF65		
	M1	M2	P	M1	M2	P	M1	M2	P	M1	M2	P
Lena	23.25	25.86	**26.62**	19.71	22.94	**24.67**	14.30	20.93	**22.82**	14.43	19.21	**21.00**
Baboon	19.99	20.32	**20.68**	17.87	19.10	**20.15**	13.46	17.64	**19.35**	13.37	16.58	**18.61**
Couple	21.51	23.50	**24.20**	18.50	21.55	**23.03**	13.86	19.95	**21.38**	13.95	18.64	**19.97**
Peppers	22.19	24.61	**26.09**	18.38	22.16	**24.24**	14.29	20.33	**22.08**	14.27	19.00	**20.92**
Boats	21.80	24.05	**24.24**	19.02	21.99	**23.06**	14.87	20.04	**21.25**	14.42	18.70	**19.92**
Watch	22.50	12.79	**24.50**	18.88	12.54	**22.72**	13.77	12.30	**20.17**	15.30	12.14	**19.45**
Average	21.87	21.86	**24.39**	18.73	20.05	**22.98**	14.09	18.53	**21.18**	14.29	17.38	**19.98**

where M1, M2 and P represent the results of Li *et al.*'s [12], Chamlawi *et al.*'s [13] and our proposed recovery methods, respectively.



(a) Li's [12] (QF95) (b) Li's [12] (QF85) (c) Li's [12] (QF75)

(d) Chamlawi's [13] (QF95) (e) Chamlawi's [13] (QF85) (f) Chamlawi's [13] (QF75)

(g) Our method (QF95) (h) Our method (QF85) (i) Our method (QF75)

Fig. 11. Comparison of restored image *Lena* after JPEG compression

In Table 3, we compare the PSNR results between original images and recovered images after applying JPEG compression at different QFs at 95, 85, 75 and 65 with Li *et al.*'s [12] and Chamlawi *et al.*'s [13] methods. When the JPEG compression QF=95 was applied, the results showed that our method performed on average 2.5 dB better than the two methods. When higher JPEG compressions were applied, the PSNR of our results decreased to 22.98 dB for QF=85, 21.18 dB for QF=75 and 19.98 dB for QF=65, respectively. However, our results were still on average 2.6 dB higher than other two methods. Table 3 shows that the lower texture images 'Lena' and 'Peppers' achieved approximate 2 dB higher than other images, when JPEG compression has been applied.

Figure 11 illustrates the restored image 'Lena' after JPEG compression QF=95, 85 and 75, by comparing our method with Li *et al.*'s [12] and Chamlawi *et al.*'s [13] methods. As shown in Figure 11(i), our proposed method could still recover the image content with some distortion after JPEG compression at QF=75, whereas the other two methods were distorted significantly as shown in Figures 11(c) and 11(f). Overall, the results indicate that our semi-fragile watermarking and self-restoration scheme perform better as compared with the other two schemes.

7 Conclusion and Future Work

In this paper, we presented a novel semi-fragile watermarking scheme for image content authentication, tampered area localisation and self-restoration that could tolerate JPEG compression. We also utilised linear regression and probability distribution approaches to analyse the feasibility of our proposed recovery method. We found that the mean pixel values (as the watermarks) could be restored to DCT coefficients approximately, and were easier to normalise for embedding than DCT coefficients. We compared our results with Li *et al.*'s and Chamlawi *et al.*'s schemes. The results indicated that the imperceptibility of our watermarked image was of high quality at 37.61 dB, and on average achieved a restored image at 24.71 dB. Our restored images also achieved on average at 24.39 dB, 22.98 dB 21.18 dB and 19.98 dB after JPEG compression of QF95, 85, 75 and 65, respectively, which were about 2.5 dB higher than other two self-restoration methods.

For future work, we plan to analyse the quality of restored image when the image has undergone other mild signal processing, such as noise and spatial filtering for semi-fragile watermarking images. We also plan to investigate the merging of watermarks for authentication and self-restoration together to reduce the number of watermarks for each block. This could lead to further improvement in imperceptibility for both watermarked and restored images.

References

1. Li, C.T., Si, H.: Wavelet-based Fragile Watermarking Scheme for Image Authentication. Journal of Electronic Imaging 16, 013009-1–013009-9 (2007)
2. He, H.J., Zhang, J.S., Chen, F.: Adjacent-block based statistical detection method for self-embedding watermarking techniques. Signal Processing 89(8), 1557–1566 (2009)

3. Li, C.T., Yuan, Y.: Digital Watermarking Scheme Exploiting Non-deterministic Dependence for Image Authentication. Optical Engineering 45(12), 127001-1–127001-6 (2006)
4. Ho, A.T.S., Zhu, X., Guan, Y.: Image content authentication using pinned sine transform. EURASIP Journal on Applied Signal Processing 2004, 2174–2184 (2004)
5. Qi, X.J., Xin, X.: A quantization-based semi-fragile watermarking scheme for image content authentication. Journal of Visual Communication and Image Representation 22(2), 187–200 (2011)
6. Lin, C.H., Su, T.S., Hsieh, W.S.: Semi-fragile watermarking Scheme for authentication of JPEG Images. Tamkang Journal of Science and Engineering 10(1), 57–66 (2007)
7. Lin, H.Y.S., Liao, H.Y.M., Lu, C.S., Lin, J.C.: Fragile watermarking for authenticating 3-D polygonal meshes. IEEE Trans. Multimedia 7(6), 997–1006 (2005)
8. Fridrich, J., Goljan, M.: Images with self-correcting capabilities. In: IEEE International Conference on Image Processing, vol. 3, pp. 792–796 (1999)
9. Zhao, X., Ho, A.T.S., Treharne, H., Pankajakshan, V., Culnane, C., Jiang, W.: A Novel Semi-Fragile Image Watermarking, Authentication and Self-restoration Technique Using the Slant Transform. In: 3rd International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP 2007), vol. 1, pp. 26–28 (2007)
10. Hasan, Y.M.Y., Hassan, A.M.: Tamper Detection with Self Correction on Hybrid Spatial-DCT Domains Image Authentication Technique. In: IEEE International Symposium on Signal Processing and Information Technology, pp. 369–374 (2007)
11. Cruz, C., Mendoza, J.A., Miyatake, M.N., Meana, H.P., Kurkoski, B.: Semi-Fragile Watermarking based Image Authentication with Recovery Capability. In: IEEE International Conference on Information Engineering and Computer Science (ICIECS 2009), pp. 1–4 (2009)
12. Li, G., Pei, S., Chen, G., Cao, W., Wu, B.: A Self-embedded Watermarking Scheme Based on Relationship Function of Corresponding Inter-blocks DCT Coefficient. In: 13th International Conference on Computer Supported Cooperative Work in Design (CSCWD 2009), pp. 107–112 (2009)
13. Chamlawi, R., Khan, A., Idris, A.: Wavelet based image authentication and recovery. Journal of Computer Science and Technology 22(6), 795–804 (2007)
14. Lin, C.Y., Chang, S.F.: Semi-Fragile Watermarking for Authenticating JPEG Visual Content. In: SPIE Security and Watermarking of Multimedia Contents II EI 2000, pp. 140–151 (2000)
15. Mendoza-Noriega, J., Kurkoski, B., Nakano-Miyatake, M., Perez-Meana, H.: Halftoning-based Self-embedding Watermarking for Image Authentication and Recovery. In: IEEE International Midwest Symposium on Circuits and Systems (MWSCAS), pp. 612–615 (2010)
16. Chen, B., Wornell, G.: Quantization Index Modulation: A class of provably good methods for digital watermarking and information embedding. IEEE Trans. On Information Theory 48(4), 1423–1444 (2001)

A Robust Audio Watermarking Scheme Based on Lifting Wavelet Transform and Singular Value Decomposition

Baiying Lei, Ing Yann Soon, and Zhen Li

School of Electrical and Electronic Engineering,
Nanyang Technological University, Singapore
leib0001@e.ntu.edu.sg

Abstract. In this paper, a new and robust audio watermarking scheme based on lifting wavelet transform (LWT) and singular value decomposition (SVD) is proposed. Specifically, the watermark data is inserted in the LWT coefficients of the low frequency subband taking advantage of SVD and quantization index modulation (QIM). The use of QIM renders our scheme blind in nature. Furthermore, the synchronization code technique is also integrated with hybrid LWT-SVD audio watermarking. Experimental results demonstrate that the proposed LWT-SVD method is not only robust to both general signal processing and desynchronization attacks but also outperform the selected previous studies.

Keywords: Audio Watermarking, Lifting Wavelet Transform, Robust Watermarking, Singular Value Decomposition, Quantization Index Modulation.

1 Introduction

Recently, audio watermarking is a very hot research topic and attracted a lot of interests as one of the most popular approaches for providing copyright protection. As a result, there are great amounts of state-of-the-art publications in the literature concerning this topic. The required properties and characteristics of the audio watermarking scheme is stated in the International Federation of the Phonographic Industry (IFPI) [1].

In recent years, for robust audio watermarking, the widely used transform domain forms are discrete wavelet transform (DWT) [2], discrete cosine transform (DCT) [3], discrete sine transform (DST) and Fourier transform. Besides, some other transforms such as LWT [4, 5] and SVD [6-8] are also becoming more and more popular and attracted a lot of interest in the audio watermarking field. It is found that the conventional wavelet transform has very good performance because of its multi-resolution property and perfect reconstruction. However, the classic wavelet transform is mainly computed by convolution which results in high computation. Besides, the generated floating numbers increase storage requirements. As a result, a new wavelet is designed and developed to increase the efficiency. First proposed by Sweldens [9], LWT is the second generation wavelet and based on the traditional

Y.Q. Shi, H.J. Kim, and F. Perez-Gonzalez (Eds.): IWDW 2011, LNCS 7128, pp. 86–96, 2012.

wavelet. In the recent year, LWT is widely used in the audio watermarking field [4, 5]. For example, in [4], Tao et al. proposed a robust audio watermarking scheme in the LWT frequency domain based on the statistical characteristics of sub-band coefficients. This invariant watermarking technique in this scheme improves the implementation efficiency with the adoption of the LWT.

Currently, in the perspective of linear algebra, SVD is extensively applied in the robust watermarking to withstand attacks due to its unique and special characteristics [6, 7]. As a factorization of a real matrix and desirable transform, SVD transform [10] has been applied widely in the image watermarking [11] first for ownership protection and extended to audio watermarking quickly [6-8]. Moreover, QIM [12] is also a very popular method for watermark embedding and data hiding. If LWT is combined with QIM method and SVD, it can reduce the operation times, and robust results for copyright protection can be achieved. The organization of this paper is as follows. Section 2 discusses embedding method. Watermark extraction is described in Section 3. Section 4 discusses the experimental results. Finally, Section 5 concludes the paper.

2 Embedding Method

2.1 Watermark Preprocessing

Watermark should be first preprocessed in order to improve the robustness and enhance the confidentiality. Binary image as watermark is scrambled by a chaotic map which is reproduced in a permuted matrix. This paper uses a Skew tent map to enhance the confidentiality of the watermarking method. The skew tent map is defined as follows:

$$x(n+1)=\begin{cases}\dfrac{1}{\alpha}x(n), & 0\le x(n)<\alpha\\ \dfrac{1}{\alpha-1}x(n)+\dfrac{1}{1-\alpha}, & \alpha\le x(n)\le 1\end{cases} \tag{1}$$

where $\alpha\in(0,1)$ can be used as the watermark key.

Then the binary image logo or signature $b(n)$ is scrambled by $x(n)$ with the following rule:

$$w(n)=b(n)\oplus x(n) \tag{2}$$

where $\oplus$ is the exclusive or (*XOR*) operator. After this random chaotic sequence encryption, the watermark is permuted and cannot be guessed by random search.

2.2 Synchronization Code

The watermark will have dislocation of the watermark regions due to the desynchronization attacks such as time scale modification (TSM), shifting and cropping. In the proposed method, we exploit a pseudo random sequence generated by chaotic signal as the synchronization code to increase the security of the synchronization code. By using generators of a strongly chaotic nature we can ensure that the system is cryptographically secure. The synchronization code is generated by thresholding the Bernoulli shift map. The Bernoulli shift map is one of the simple deterministic chaotic maps which contain many chaotic characteristics. A binary shift Bernoulli Map can be defined as:

$$x(k+1)=\begin{cases} 2x(k) & if\, 0 \le x(k) < \dfrac{1}{2} \\ 2x(k)-1 & if\, \dfrac{1}{2} \le x(k) \le 1 \end{cases} \tag{3}$$

where $x(0) \in [0,1]$ (map's initial condition) and must be specified. $x(k)$ is mapped into the synchronization sequence $C = \{c(k), 1 \le k \le \text{Lsyn}\}$ with the following rule:

$$c(k)=\begin{cases} 1 & if\ x(k) > \tau \\ 0 & otherwise \end{cases} \tag{4}$$

τ is a predefined threshold for synchronization code. Time domain embedding has the strength that it is less computational intensive and low cost in finding the synchronization code. The synchronization code insertion part is cut into *Lsyn* audio segments and each audio segment has *P* samples denoted as:

$$SA(k) = A(k \bullet P + u), 1 \le k \le Lsyn, 1 \le u \le P \tag{5}$$

Then each bit of the synchronization code is embedded into each $SA(k)$ as follow:

$$SA'(k)=\begin{cases} round(\dfrac{SA(k)}{\Delta}) \bullet \Delta, & if\ Syn(k)=1 \\ floor(\dfrac{SA(k)}{\Delta}) \bullet \Delta + \dfrac{\Delta}{2}, & if\ Syn(k)=0 \end{cases} \tag{6}$$

where Δ denotes the embedding strength, $round(\bullet)$ means rounding to the nearest integer, $floor(\bullet)$ is rounding to the minus infinity.

After embedding, the embedded and attacked signal $SA''(k)$ is also split into *Lsyn* segments, and then the synchronization code is extracted by the following rule:

$$Syn'(k)=\begin{cases} 0, & if\ \dfrac{\Delta}{4} \le \mathrm{mod}(SA''(k),\Delta) < \dfrac{3\Delta}{4} \\ 1, & otherwise \end{cases} \tag{7}$$

where $\mathrm{mod}(\bullet)$ denotes modulus after division.

2.3 Watermark Embedding

Fig.1 presents the diagram of our watermark embedding algorithm. In our watermarking technique, we choose the popular QIM method in the embedding process because of its good robustness and blind nature. As a result, our method is blind and does not need the original audio for the data extraction. The second part of the host audio, *SB*, signal is used to embed the watermark.

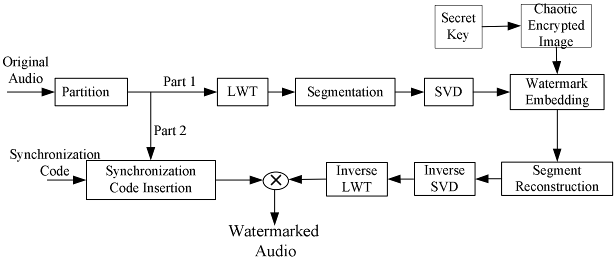


Fig. 1. Diagram of watermark embedding process

Specifically, the embedding process is described by the following steps:

Step 1: Perform LWT on the audio segment, *SB*, of the host audio signal.

$$I = LWT(SB) \tag{8}$$

Step 2: The approximate coefficient after the LWT transform are divided into non-overlapping blocks. The length of audio blocks depends on the amount of data that need to be embedded and the number of LWT decomposition levels. The watermark sequence is embedded successively into the low-frequency subband of blocks.

Step 3: Scrambling the watermarking image with the method mentioned in Section 2.1.

Step 4: For each block, perform SVD transform to obtain the singular values and first singular value, $S(1,1)$.

$$I = USV^T \tag{9}$$

Step 5: Embedding the watermark into singular values with the QIM method. The encrypted watermark $w(i)$ is added to the first singular values, $S(1,1)$, of each block. Our watermark embedding method is based on the popular odd and even parity rule.

Let $Q = round(S(1,1)/\beta)$, $D = \mathrm{mod}(Q,2)$,where β is the quantization step. A small value of β will lead to good imperceptibility of the watermarking scheme but bad robustness to the attacks. Thus we choose an optimal β to tradeoff between inaudibility and robustness of the watermark. The embedding rule is that:

If D is 0 and $w(i)$ is 1, then $Q = Q+1$; if D is 1 and $w(i)$ is 0, then $Q = Q+1$.

Step 6: The first singular values is further modified by the updated Q as follows:

$$S_w(1,1) = \beta \times round(Q) \tag{10}$$

Step 7: $S_w(1,1)$ is used to build the watermarked block I_w by applying inverse SVD:

$$I_w = US_wV^T \tag{11}$$

Step 8: Inverse LWT is conducted to reconstruct the watermarked signal.

$$SB_w = LWT^{-1}(I_w) \tag{12}$$

3 Watermark Extraction

The main step of watermark extraction is as follows:

Step 1: Perform LWT on the watermarked signal.

$$I_e = LWT(SB_w) \tag{13}$$

Step 2: For the obtained wavelet approximation coefficient, block based method is also used, that is, we divide the LWT approximate coefficients into different blocks.

Step 3: SVD is performed in each block.

$$I_e = U_eS_eV_e^T \tag{14}$$

Step 4: Let $Q_e = round(S_e(1,1)/\beta)$, $D_e = \mathrm{mod}(Q_e, 2)$, then the extraction rule is:

$$w'(n) = \begin{cases} 1 & D_e = 1 \\ 0 & D_e = 0 \end{cases} \tag{15}$$

Step 5: Perform the decryption with the same chaotic sequence to get the hidden binary image or signature.

$$b'(n) = w'(n) \oplus x(n) \tag{16}$$

4 Experimental Results

In this section, several experiments are conducted to demonstrate the performance of the proposed LWT-SVD based audio watermarking approach. The performance of our scheme is assessed in terms of robustness and imperceptibility. The test audio signal in our scheme is 44.1 kHz sampled, with 16bits/sample. 32×32 binary image logo is used in our scheme to conduct performance evaluation. LWT decomposition level is 3. In our experiment, SNR and Segmental SNR (SegSNR) are used for the evaluation of the quality of the watermarked audio signals. BER is used for evaluating the reliability of the extracted watermarks. BER, SNR, SegSNR are defined as follows:

$$SNR = 10\log_{10}\left(\sum_{i=1}^{L} S(i) \Big/ \sum_{i=1}^{L} (S'(i) - S(i))^2\right) \tag{17}$$

$$BER = \frac{1}{N_w}\sum_{n=1}^{N_w} w(n) \oplus w'(n) \tag{18}$$

$$SegSNR = \frac{10}{K}\sum_{m=0}^{K-1} \log_{10} \frac{\sum_{i=1}^{r} S^2(i)}{\sum_{i=1}^{r} (S'(i) - S(i))^2} \tag{19}$$

where $S(i)$ and $S'(i)$ correspond to the original and watermarked signal respectively, $w(n)$ and $w'(n)$ are original and extracted watermarks.

4.1 Imperceptibility Test

Fig.2 presents the waveforms of the original, watermarked, and residual signal respectively. Fig. 3 shows the spectrum of the original and watermarked signals respectively. SNR and SegSNR results versus different quantization steps are shown in Fig.4. From the waveforms and spectrums, it can be observed that there is not much distinguishing difference between the original and watermarked audio, which is also verified by the SNR results in Fig. 4 as SNR and SegSNR results are above 20dB even when the quantization step is 1. The SNR, SegSNR results can totally satisfy the IFPI requirements.

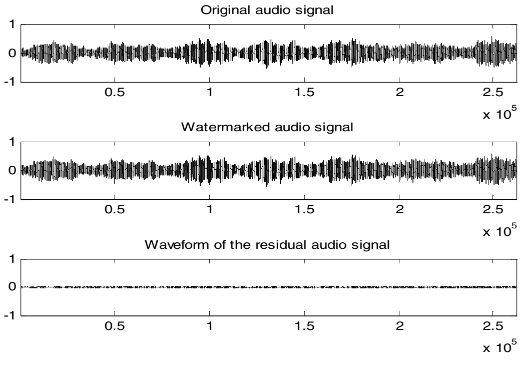


Fig. 2. Waveform of original, watermarked and residual signal

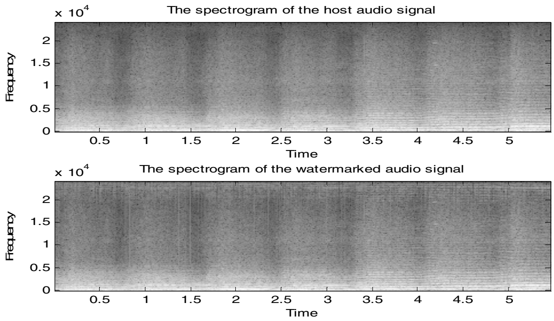


Fig. 3. Spectrum of original and watermarked signal

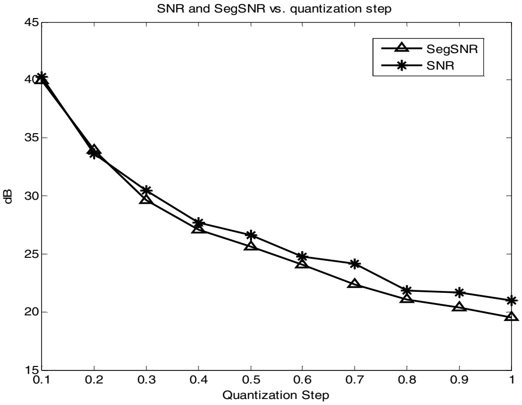


Fig. 4. SNR, SegSNR results versus quantization step

4.2 Robustness to Common Signal Processing

In our experiment, common audio signal processing include re-quantization, re-sampling, noise addition, low-pass filtering, echo addition, equalization, MPEG compression etc, and desynchronization attacks include random cropping, amplitude variation, pitch shifting, jittering etc. The parameters of these common signal processing manipulations are given as follows:

Additive Noise: White Gaussian noise with 1% of the power of the audio signal is added.

Amplitude Variation: The watermarked signal was attenuated up to 120% and down to 80%.

Cropping: 10% samples of each testing signal are cropped out of 5 random positions.

Denoising: The watermarked audio signal is denoised by using the Hiss removal function of GoldWave.

Echo Addition: An echo signal with a delay of 98 ms and a decay of 10% was added to the original audio signal.

Expanding: Expand the watermarked signal with increment of 1 dB and -1dB respectively.

Jittering: Jittering is a form of random cropping and performed uniformly. One sample out of every 100000 samples is removed in our jittering experiment.

Low-Pass Filtering (LPF): Low-pass filtering using a second order Butterworth filter with cut-off frequency of 40 kHz is performed to the watermarked audio signals.

MP3 Compression: The robustness against the low-rate codec was tested by using MPEG 1 Layer III compression (MP3) with compression rates of 56, 64, 96, and 128 kbps.

Pitch Shifting: Tempo-preserved pitch shifting is a difficult attack for audio watermarking algorithms, because it causes frequency fluctuation. In our experiment, the pitch is shifted one degree higher and one degree lower.

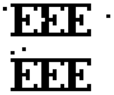
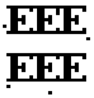
Re-quantization: We tested the process of re-quantization of a 16-bit watermarked audio signal to 8-bit and back to 16-bit.

Resampling: Watermarked audio signals with original sampling rate 44.1 kHz have been subsampled down to 22.05kHz, 11.025kHz, and upsampled back to 44.1kHz.

TSM: TSM processing is done in the watermarked audio signal to change the time scale to an extent of ±1% while preserving the pitch.

The robustness test is evaluated in terms of the above mentioned attacks. BER results and extracted watermarks after attacks are used to show the robustness results as the robustness is of great significance to the robust watermarking. Table 1 demonstrates the robustness results of our proposed scheme. It is obvious that the robustness results of our method are very satisfactory as the BER values after various attacks are very low.

Table 1. Robustness results of various attacks

Attacks	No attack	Additive noise	Amplitude variation
Extract WMs			
BER	0	0	0
Attacks	Cropping	Denoising	Echo addition
Extract WMs			
BER	0	0.005	0.002
Attacks	Expanding	Jittering	LPF
Extract WMs			
BER	0	0	0
Attacks	MP3 (128kbps)	MP3 (96kbps)	MP3 (64kbps)
Extract WMs			
BER	0	0	0
Attacks	MP3 (56kbps)	Pitch shifting	Re-quantization
Extract WMs			
BER	0	0	0
Attacks	Re-sampling 44.1-22.05-44.1	Re-sampling 44.1-11.025-44.1	TSM
Extract WMs			
BER	0	0	0

4.3 Robustness to Stirmark Attacks

The robustness of our scheme is also benchmarked by Stirmark for audio software which is a very popular benchmark tool for audio watermarking. The parameters of these standardized attacks are default which is contained in the software configuration. The detailed benchmark attack results are summarized in Table 2. We also compare our method with the selected state-of-the-art watermarking method in literature [3] and literature [7], from this comparison results, it is noted that our method is slightly better than the watermarking scheme in [7] but very much better than [3] regarding the Stirmark attacks.

Table 2. Robustness of the Stirmark attacks

Attacks	DCT based method in [3]	SVD based method in [7]	Ours	Attacks	DCT based method in [3]	SVD based method in [7]	Ours
Addbrumm	1.25	0	0	Fft_stat1	19.84	19.84	0.5
AddDynNoise	1.56	0	0	Fft_test	19.80	19.80	0.4
AddFFTNoise	51.25	0	0	Flipsample	21.66	21.66	0.75
Addnoise	0.78	0	0	Invert	52.42	52.42	0
Addsinus	0.77	0	0	Lsbzero	0	0	0
Amplify	52.32	0.75	0	Normalize	0	0	0
Bassboost	0	0	0	Nothing	0	0	0
Compressor	0	0	0	Rc_highpass	2.03	2.03	0
Copysample	100	0.5	0.2	Rc_lowpass	0	0	0
Cutsamples	100	0	0	Smooth	0	0	0
Echo	23.43	0	0	Stat1	0	0	0
Exchange	0	0	0	Stat2	0	0	0
Extrastereo	0	0	0	Voiceremove	52.1	52.1	0
Fft_hlpass	0.31	0	0	Zerocross	0	0	0
Fft_invert	52.6	0	0	Zerolength	60.5	60.5	0
Fft_real_reverse	0.78	0	0	Zeroremove	100	100	0
Average all attacks	**22.2937**	**0.0906**	**0.012**				

5 Conclusions

In this paper, a very robust and blind audio watermarking scheme based on SVD-LWT is proposed in this paper as we make good use of features of SVD, LWT, synchronization code technique and QIM. The robustness of our scheme is validated by common signal processing and stirmark attacks. The performance and comparison results demonstrate that our scheme is not only inaudible, but also robust to attacks.

References

1. Katzenbeisser, S., Petitcolas, F.A.P.: Information Hiding Techniques for Steganography and Digital Watermarking. Artech House, Norwood (2000)
2. Wang, X.-Y., Zhao, H.: A Novel Synchronization Invariant Audio Watermarking Scheme Based on DWT and DCT. IEEE Transactions on Signal Processing 54, 4835–4840 (2006)
3. Cox, I., Kilian, J., Leighton, F.T., Shamoon, T.: Secure spread spectrum watermarking for multimedia. IEEE Transactions on Image Processing 6, 1673–1687 (1997)
4. Tao, Z., Zhao, H.-M., Wu, J., Gu, J.-H., Xu, Y.-S., Wu, D.: A lifting wavelet domain audio watermarking algorithm based on the statistical characteristics of sub-band coefficients. Archives of Acoustics 35, 481–491 (2010)
5. Kundur, D., Hatzinakos, D.: Digital watermarking using multiresolution wavelet decomposition. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 2969–2972 (1998)
6. Bhat, K.V., Sengupta, I., Das, A.: An adaptive audio watermarking based on the singular value decomposition in the wavelet domain. Digital Signal Processing: A Review Journal 20, 1547–1558 (2010)
7. Özer., H., Sankur., B., Memon, N.: An SVD-based audio watermarking technique. In: Proceedings of the 7th Workshop on Multimedia and Security, pp. 51–56. ACM, New York (2005)
8. Lei, B.Y., Soon, I.Y., Li, Z.: Blind and robust audio watermarking scheme based on SVD-DCT. Signal Processing 91, 1973–1984 (2011)
9. Sweldens, W.: The lifting scheme: A custom-design construction of biorthogonal wavelets. Applied and Computational Harmonic Analysis 3, 186–200 (1996)
10. Andrews, H.C., Patterson, C.L.: Singular Vale Decomposition and Digital Image Processing. IEEE Transactions on Acoustics, Speech, and Signal Processing ASSP-24, 26–53 (1976)
11. Liu, R., Tan, T.: An SVD-based watermarking scheme for protecting rightful ownership. IEEE Transactions on Multimedia 4, 121–128 (2002)
12. Wornell, G.W., Chen, B.: Quantization index modulation: a class of provably good methods for digital watermarking and information embedding. IEEE Transactions on Information Theory 47, 1423–1443 (2001)

Adaptive Selection of Embedding Locations for Spread Spectrum Watermarking of Compressed Audio

Alper Koz and Claude Delpha

Laboratory Signals and Systems, Univ. Paris Sud-CNRS-SUPELEC,
91192 Gif-sur-Yvette, France
{Alper.KOZ,Claude.DELPHA}@lss.supelec.fr

Abstract. The main stream in audio watermarking, namely spread spectrum (SS) based methods, embeds the watermark into a fixed range in the low-frequency band of the audio in order to correctly match the spectrum coefficients of the tested signal with the watermark pattern during the correlation operation at the detector. However, a watermark that is always inserted to low frequencies could be lost during the coding of the audio files with high frequency content. In this paper, in contrary to the fixed embedding locations, we first propose to adaptively select the watermark embedding locations with respect to the maximum energy region in the frequency spectrum of the cover signal. The proposed method achieves to find the same watermark embedding and extraction locations with over 95 % detection rates down to the coding bitrates of 32 kbps. Then, the proposed adaptive selection is compared to the fixed selection of embedding locations for spread spectrum watermarking of coded audio. The experimental results on watermark bit error rates (BER) have indicated the superiority of the adaptive approach over fixed embedding against the audio coding (32-128 kbps), additive noise (10 dB-45 dB) and low pass filtering (1.4 kHz-12.4kHz). In particular, very high performance of the proposed method for high frequency audio content has demonstrated the better applicability of the scheme for large databases of audio files with various characteristics in exchange networks.

Keywords: Adaptive embedding, Audio Watermarking, Spread Spectrum, Audio Coding.

1 Introduction

Spreading the spectrum of a signal into a band wider than the required minimum for transmission has been verified as a robust and secure way of sending information in communication research [1]. This strategy has formed the basic inspiration for spread spectrum watermarking [2] where a narrow band *message* is spread over the spectrum of the *cover signal* such that the energy in any frequency bin is very small and undetectable [2] . In the last decade, the proposed spread spectrum (SS) approach has found a significant attention in the watermarking of audio signals [3]-[9] as in previous studies for image [10] and video [11]-[12].

Y.Q. Shi, H.J. Kim, and F. Perez-Gonzalez (Eds.): IWDW 2011, LNCS 7128, pp. 97–110, 2012.

Table 1. Watermark embedding locations and payloads in some of the SS-based watermarking methods (Sampling Rate: 44.1 kHz; int-MDCT: Integer Modified Discrete Cosine Transform; MCLT: Modulated Complex Lap Transform)

Method	Utilized Transform	Watermarked coefficients	Frequency Range (~ Hz)	Payload (bits/sec)
Li *et al.* [5]	1024 point int-MDCT	1- 40	0- 1700	2-13 bits
Kirovski *et al.* [6]	2048 point MCLT	8 -83	200-2000	0.5-1 bit

The success of SS methods lies in a number of advantages over the preceding methods which mostly embed the watermark directly on the time samples of the audio signals by changing the energy relations of the subframes [13], by introducing inaudible delays on the audio signal [14]-[15] or by adding a noise-like signal weighted according to Human Auditory System (HAS) [16]. Compared to these methods, SS approach first provides a high robustness as the removal of a watermark that is spread to the spectrum would require a noise of high amplitude to be added to all frequencies [2]. Such an attack against the watermark would not be preferable as it would also produce a severe degradation on the original data. Second, the perceptual characteristics of HAS are better represented in the frequency spectrum to properly determine the significant and insignificant part of an audio signal to embed the watermark. Finally, the utilization of frequency domain offers a suitable framework consistent with the coding standards to embed and extract the watermark without needing a full decoding of the audio files.

However, a major limitation in SS approach is to correctly determine the embedding locations of the watermark in order to properly match the spectrum coefficients with the watermark pattern during the correlation at the detector [5]-[6]. The existing methods solve this problem by fixing the embedding locations in the low frequency band of the spectrum assuming that the significant part of the audio files are mostly located at the low-band. Table 1 shows the watermarked coefficients in each frame of the audio file, the corresponding low-frequency range to these coefficients, and the payload (in bits/sec) of the watermark in some of the state-of-the art methods [5]-[6]. The selected frequency range for watermarking is fixed to about 0-2 kHz range in these methods and the payload is a few bits (1-13 bits) for every second of the utilized audio files.

Although low frequency band represent the significant part of the most audio files, this assumption is not valid for the audio files, or for some frames of the audio files, with high frequency content. A watermark that is directly embedded to low frequencies for these frames and files would be audible or would be lost during the lossy compression. Therefore, a better option for watermarking would be to adaptively select the most significant coefficients that will remain after the typical processing of audio files. The positions of these coefficients should also be correctly determined both during the embedding and extraction stages in order to achieve the synchronization during the correlation operation in the detector.

In this work, we handle this problem, namely the selection of the most significant coefficients for watermarking with the mentioned constraint for the matching of embedding and extraction locations. We first point out that a direct solution, just taking the greatest fixed number of spectrum coefficients, would also produce a synchronization problem during the extraction as the order of the coefficients is very sensitive to the lossy compression. Then, a grouping strategy is proposed to increase the robustness of the selected coefficients against the audio coding. The proposed strategy mainly solves the basic trade-off in selection between the matching rates of embedding and extraction locations, and the total amount of signal energy contained in the selected coefficients. Finally, the superiority of the proposed adaptive scheme over the traditional approach using fixed watermark locations [5]-[8] is verified with detailed imperceptibility and robustness tests.

Our focus in this paper would be a compressed domain watermarking approach for the audio downloading applications in exchange networks. As the audio files are stored in the coded form at the server side, a watermarking approach directly working on the spectrum coefficients without needing the full decoding would increase the efficiency of the system during streaming. The common distortions that might occur in the transmission chain of a watermarked audio file such as coding at different bit rates, noise addition and low pass filtering are taken into account in the comparisons to reveal the benefit of the adaptive selection.

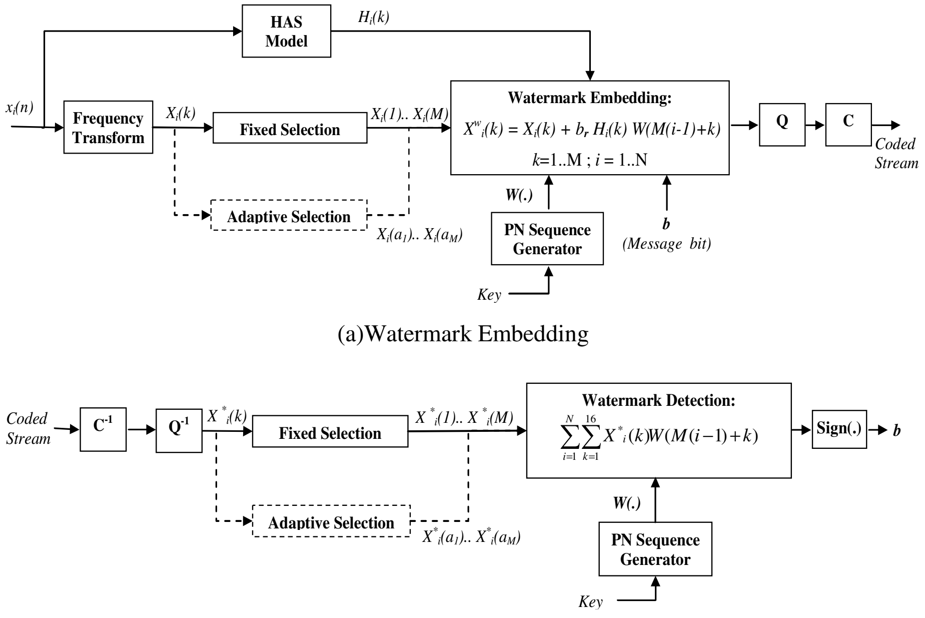
The desynchronozation attacks are out of the core point of the paper and can be solved by implementing the existing solutions to the top of the proposed SS methods regardless of the selection scheme being adaptive or fixed. These solutions can involve seeking for the best correlation with an exhaustive search on the possible scales of time and frequency fluctuations [17], repetitive pattern embedding [18]-[19] and using salient points that are invariant under the common audio processings [20]-[21]. Without loss of generality, the proposed selection of embedding locations can also be applied to the other basic schemes in audio watermarking such as quantization index modulation (QIM) [4], [19], [22].

The next section gives the details of the proposed selection method and the experimental results on its performance. Section 3 gives and compares the experimental results for the robustness and imperceptibility of the watermark for the adaptive and fixed selection approaches. Then, the paper is concluded in section 4.

2 Proposed Adaptive Selection

In SS based watermarking of compressed audio (Fig. 1), a pseudo sequence of length (L) is generated to embed one bit of watermark [5]-[8]. After being modulated with the HAS thresholds and the sign of the watermark bit, this sequence is added to the spectrum coefficients of N frames of the coded audio file in accordance with the targeted payload (in bits) and total number of frames in one second. Then, the resulting coefficients are quantized and entropy coded. Such a procedure results in M (i.e. L/N) spectrum coefficients in each audio frame to be used for watermarking. These M coefficients are selected as the coefficients from 1 to M as illustrated with the *fixed selection* block in Figure 1 [5]-[6].

During the detection stage, the coded stream of the audio frame is first entropy decoded and dequantized. The first M spectrum coefficients are then used in the correlation operation to detect the watermark. As the indices of the selected coefficients



(a)Watermark Embedding

(b) Watermark Detection

Fig. 1. A generic structure for SS based watermarking of coded audio and the location of the proposed adaptive selection in the structure as an alternative to fixed selection. The generated sequence for one bit of message is embedded to N audio frames. (Q: Quantization; C: Entropy Coding)

do not change during the embedding and detection, such a procedure do not yield any synchronization problem with the watermark pattern. However, performance of such a scheme can be severely degraded, if the content mainly consists of high frequency components outside the selected region.

Our proposal in this paper is to replace the fixed embedding block during watermark embedding and detection with the *adaptive selection block* illustrated in Figure 1.

2.1 Adaptive Selection of Spectrum Coefficients

The problem of selection is to determine the same M spectrum coefficients both during the embedding and extraction after the audio coding, while representing at the same time the highest possible energy.

A direct approach for such a selection is to use the highest M coefficients for watermarking. Table 2 gives the average ratio of the coefficients among $M=16$ highest coefficients in 512-point audio frames (MPEG-1) whose locations found correctly after the coding at different bitrates for *Madonna-Music* file. The ratio of the coefficients whose locations are found correctly is about 35 % for a typical bitrate of 64 kbps, which indicates the very poor performance of such an individual selection of

Table 2. Ratio of the correctly found coefficients among *M* highest coefficients after bit rate coding (File: *Madonna-Music*, M=16)

Bitrate (kbps)	128	96	64	32
Ratio (%)	44.8	39.2	34.8	29.2

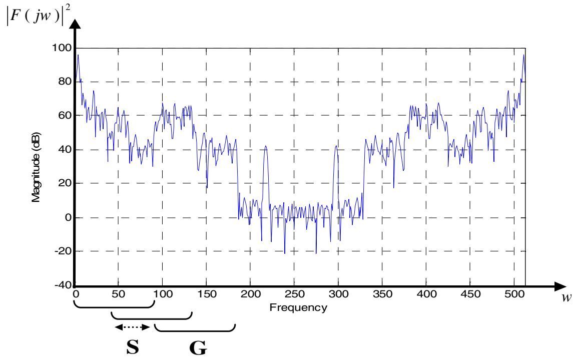


Fig. 2. Selection of the watermarking coefficients with a grouping of G elements

the highest coefficients for watermarking. As the order of the coefficients in such an approach is sensitive to bitrate coding, an error in the location of one coefficient in the order changes all the locations after that coefficient.

Our proposed solution is to select the M coefficients not individually but in groups to provide robustness against coding. The proposed algorithm is as follows:

- For each frame of the audio file, group each *G* number of consecutive coefficients with a shift of *S* between the groups (Fig. 2).
- Calculate the signal energy for each group and order the groups with respect to their energy.
- Select (M/G) number of groups with highest energy for watermarking.

The performance of the proposed algorithm is evaluated by testing the invariance of the selected embedding regions against the coding at different bit rates for different values of *G* and *S*. The performance measure is selected as the ratio of the number of frames with correctly found embedding regions to the total number of frames in the audio files.

2.2 Experimental Results for the Invariance of Selected Region

The performance of the algorithm is tested over a database of about 600 audio songs with various characteristics. Table 3 presents the durations, bitrates and the number of frames for the ten songs from our database. The total frequency range for watermarking is taken about 1.4 kHz, which corresponds to *M=16* coefficients (2 subbands) of 512 (32 subbands) point FFT of audio frames. This range corresponds to a comparable band range that is utilized in the two SS domain methods [5]-[6].

Table 3. Utilized Songs in the experiments

Name	Duration (min:sec)	Smp. Rate (KHz)	Bitrate (kbps)	# of Frames
1. Madonna *Music*	*3:47*	*44.1*	*128*	*26070*
2. L. Fabian *I am who I am*	*3:47*	*44.1*	*96*	*26070*
3.Ace of Base *Unspeakable*	*3:14*	*22.05*	*192*	*11140*
4. M. Jackson *Speechless*	*3:18*	*44.1*	*128*	*22739*
5. Sugarland *Stay*	*4:43*	*44.1*	*224*	*32501*
6. C. Aguleria *Hurt*	*4:03*	*44.1*	*181*	*27907*
7. Dj mhd *Ya Zina Club*	*4:30*	*44.1*	*256*	*31008*
8. Greg Cerrone *Pilling me*	*5:51*	*44.1*	*320*	*40310*
9. Gwen Stefani *The sweet escape*	*4:06*	*44.1*	*128*	*28252*
10. The police *Roxanne*	*3:12*	*44.1*	*192*	*22050*

For this fixed value of $M=16$, the ratio of correctly detected embedding regions are determined for different G values after the test songs are passed from a bitrate coding at 64 kbps, as a lower bitrate than the originals. The shift (S) during the grouping is kept equal to G (i.e. non-overlapping groups). Figure 3 indicates very rapid increases in detection rates for increasing values of G. As the grouping has mainly low pass filtering characteristics, the group of coefficients is becoming more robust against the compression noise. After $G=8$, the detection rates is exceeding 90 % while the increase is getting more saturated.

On the other hand, the trade-off for using larger values of G is the decreased amount of energy in the selected embedded region as the resolution of the algorithm for finding the maximum energy region decreases with higher G values. Figure 3 shows also this decrease in the amount of energy. However, the decrease has a slow characteristics and the selected embedded region for watermarking contains the significant part of the content with more than 76 % signal energies for the worst case (G=16). Considering the highest detection rate and the comparable signal energy with respect to other values, we select the G value as 16.

In our second experiment, we examine the effect of parameter S to the detection rates after a coding of 64 kbps by fixing G to 16. Figure 4 indicates an increasing detection rate for increasing values of S. As the distance between the consecutive groups are becoming larger with larger values of S, the probability of detecting the neighbor group instead of the original is getting lower. Similar to the previous case, the trade-off is the decreased amount of energy in the selected regions due to the lower number of groups generated for selection with higher S values. However, the difference of the signal energy between the maximum and minimum cases is only about 2 %. Accordingly, we choose the S value as 16.

Figure 5 shows the decrease in the detection w.r.t decreasing bit rates for fixed values of *G*=16 and *S*=16 in our last experiment. Although the bit rate is decreased down to 32 kbps, the algorithm still finds the correct embedding regions with more than 95 % detection rates. This indicates that the energy of the group of coefficients is highly robust to the changes in coding bit rate.

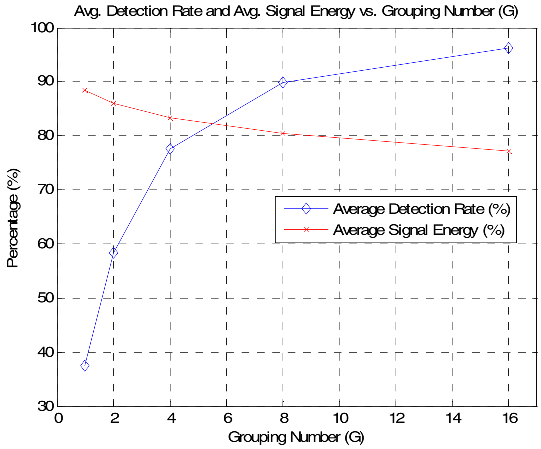


Fig. 3. Average Detection Rate (%) of correct embedding regions and Average Signal Energy in the embedding regions *vs.* Grouping Number (G) after a coding at 64 kbps

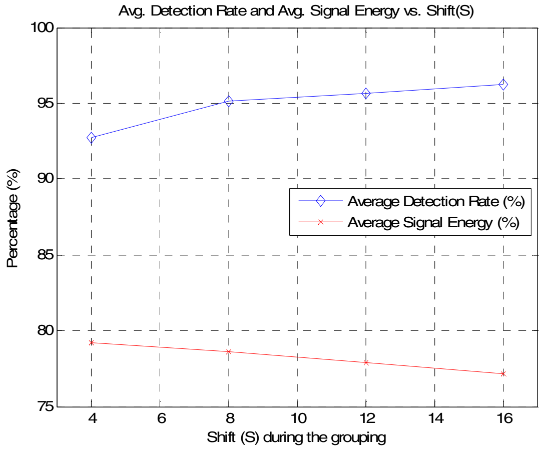


Fig. 4. Average Detection Rate (%) of correct embedding regions and Average Signal Energy in the embedding regions *vs.* Shift (S) after a coding at 64 kbps

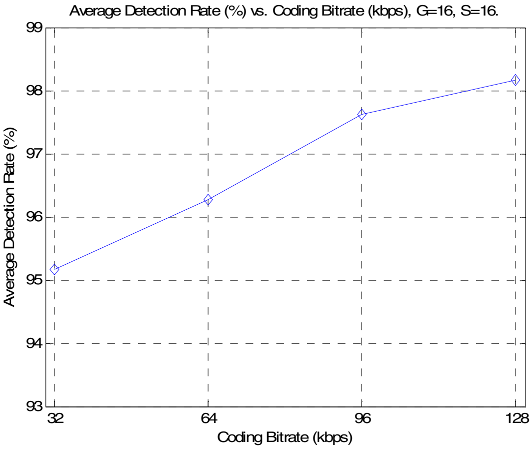


Fig. 5. Average Detection Rate (%) of Correct Embedding Regions vs. Coding Bitrate (kbps), S=16, G=16

3 Comparison of the Adaptive Approach to Fixed Embedding

In order to compare the two embedding approaches, we implement the SS method proposed in [5] for the coded audio files. In the fixed approach, an m-sequence of length 511 is generated to embed one bit of watermark. This sequence is added to the fixed range of 512 point frequency spectrum of 32 audio frames as in [5] as follows:

$$X_i(k) = X_i(k) + b\,H_i(k)\,W(16(i-1)+k), \tag{1}$$

for i = 1..31, k = 1..16 ; for i = 32, k = 1..15.

$X_i(k)$ and $H_i(k)$ denote the k'th spectrum coefficient of i'th frame and the corresponding HAS threshold for that coefficient; W is the generated m-sequence and b corresponds to watermark bit of +1 or -1. Psychoacoustic model I in ISO standard [23] is utilized for the calculation of HAS thresholds.

The same operation is repeated for every 32 frames to embed one bit of information. Such a scheme changes about $M=16$ coefficients for each audio frame and achieves a payload of about 3.6 bits/sec similar to the given payloads in Table 1. The detection is performed by means of correlating the generated m-sequence with the spectrum coefficients of the 32 audio frames of tested audio in the same fixed embedding range. Then, the sign of the correlation is accepted as the embedded watermark bit [5].

In the adaptive approach, the same procedure is followed by selecting the embedding regions in each audio frame as described in section 2 during the embedding and detection (Fig. 2). G and S values are selected as 16.

Table 4 shows the resulting SNR due to watermarking in the tested audio files and the number of the embedded bits for each audio file in accordance with their durations. The table also gives the number of *high frequency watermark blocks* in each audio file. If the most repetitive selected embedding region among 32 audio frames is not the first window, then that block of 32 frames is assumed as a *high frequency* (HF) watermark block. The resulted SNR values in the table are smaller for the adaptive approach as the HAS threshold values are greater when the selected embedding region is in high frequency regions. The ratio of the HF watermark blocks to the total number of watermark blocks is in the range of 5-21 % in the tested songs.

Table 4. The resulting SNR values, # of embedded bits and # of HF watermark blocks

Song	**SNR (dB)** ***Fixed***	**SNR (dB)** ***Adaptive***	**Embedded Bits**	**HF Blocks**
1. Madonna *Music*	*18.94*	*18.67*	*800*	*85*
2. L. Fabian *I am who I am*	*21.56*	*20.58*	*341*	*35*
3.Ace of Base *Unspeakable*	*19.40*	*19.30*	*695*	*27*
4. M. Jackson *Speechless*	*19.80*	*19.67*	*798*	*27*
5. Sugarland *Stay*	*24.17*	*22.47*	*1006*	*108*
6. C. Aguleria *Hurt*	*21.82*	*21.37*	*864*	*29*
7. Dj mhd *Ya Zina Club*	*18.86*	*18.49*	*968*	*212*
8. Greg Cerrone *Pilling me*	*18.81*	*18.55*	*1253*	*206*
9. Gwen Stefani *The sweet escape*	*19.52*	*19.23*	*880*	*66*
10. The police *Roxanne*	*20.79*	*20.56*	*664*	*48*

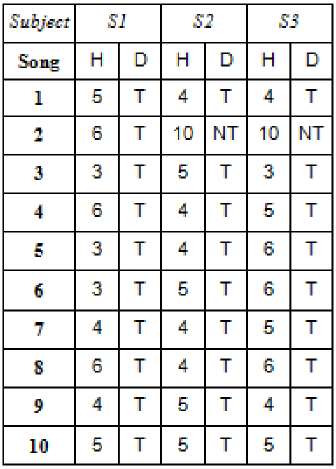
3.1 Imperceptibility Tests

The two alternative two choice test is utilized to test the transparency of the watermarked audio files [8]. For every item, we generate a set of 10 pairs each of which is selected randomly from the pairs {(O,O), (O,W), (W,O), (W,W)}, where "O" denotes the original and "W" the watermarked file. The subject is asked whether both items are equal or not. Each correct decision about items being equal or different is assumed a "hit". As the subjects are mostly graduate students and researchers familiar with audio processing, the probability to detect any distortions is assumed as greater than 0.7 as in [8]. The level of significance is taken as 0.05 as common in hypothesis testing. With these settings, if the number of the hits is {8,9,10} out of 10 pairs, we assume this item to be non-transparent with 95 % probability [8]. If the number of hits is less than or equal to 6 hits, we assume this item to be transparent with a 95 % probability.

Table 5 (a) and (b) shows the number of hits and the concluded decision as being transparent or non-transparent for each item and subject for the fixed approach and adaptive approach, respectively. The results have indicated that watermarked audio

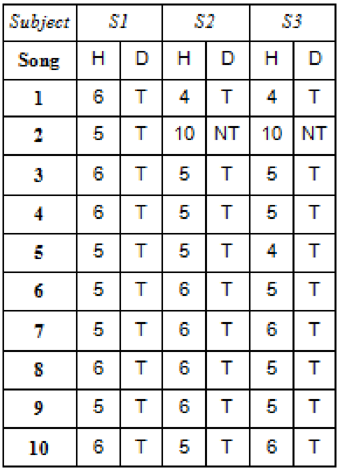
Table 5. Results of the listening tests, for the fixed embedding (a) and adaptive embedding (b). H: "Number of Hits", D: "Concluded Decision", T : "Transparent" (T), NT:"Non-transparent", X: "Not transparent nor non-transparent".

(a)



Subject	*S1*		*S2*		*S3*	
Song	H	D	H	D	H	D
1	5	T	4	T	4	T
2	6	T	10	NT	10	NT
3	3	T	5	T	3	T
4	6	T	4	T	5	T
5	3	T	4	T	6	T
6	3	T	5	T	6	T
7	4	T	4	T	5	T
8	6	T	4	T	6	T
9	4	T	5	T	4	T
10	5	T	5	T	5	T

(b)



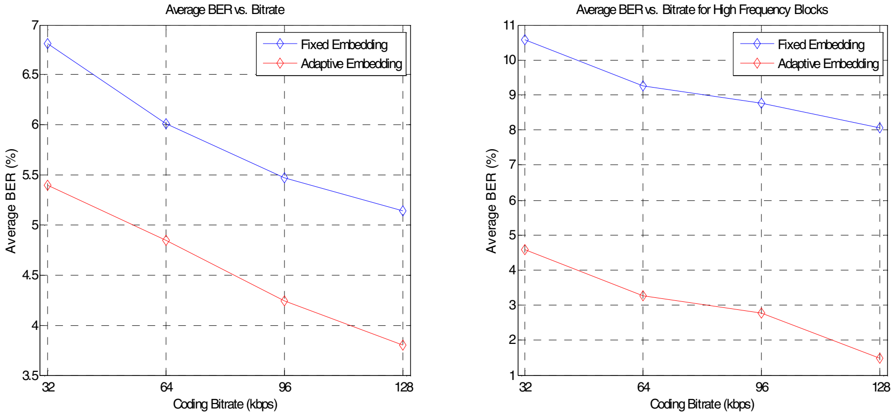
Subject	*S1*		*S2*		*S3*	
Song	H	D	H	D	H	D
1	6	T	4	T	4	T
2	5	T	10	NT	10	NT
3	6	T	5	T	5	T
4	6	T	5	T	5	T
5	5	T	5	T	4	T
6	5	T	6	T	5	T
7	5	T	6	T	6	T
8	6	T	6	T	5	T
9	5	T	6	T	5	T
10	6	T	5	T	6	T

files are transparent for the subjects almost in all cases although the scaling of the thresholds is equal to unity in the tests. Note that the utilized model [23] only takes the frequency masking inside the audio frames into account to compute the HAS thresholds while ignoring the temporal masking between the audio frames. The only exception in the results is in song 2 for subject 2 and 3. The perceptual quality of this song is however regarded as *perceptible, but not annoying* by the subjects both for the fixed and adaptive approaches. Ultimately, as both the approaches adjust the strength of the watermark w.r.t. HAS thresholds, there is no significant difference in the transparency of the watermark.

3.2 Robustness Tests

The watermarked audio files are passed from coding at different bitrates as the first and most relevant attack for the comparison of two embedding approaches in the compressed domain. Figure 6 (a) shows a decrease of about 1.5 % in the average watermark BER rates of 10 utilized audio files for the *adaptive embedding* compared to the *fixed embedding* for the same coding bitrates down to 32 kbps. For the same watermark BERs in the graph, the gap between the adaptive and fixed embedding is at least 64 kbps. In other words, the adaptive approach requires at least two times more compression than the one in fixed approach in order to observe the same watermark BERs. The results have indicated that a watermark that is adaptively inserted to a high energy region of the content persists better against the audio coding than the one which is always embedded to the low frequency region.

In order to put more contrast to this conclusion, the watermark BER for only high frequency watermark blocks are shown in Figure 6 (b) for different coding bitrates.



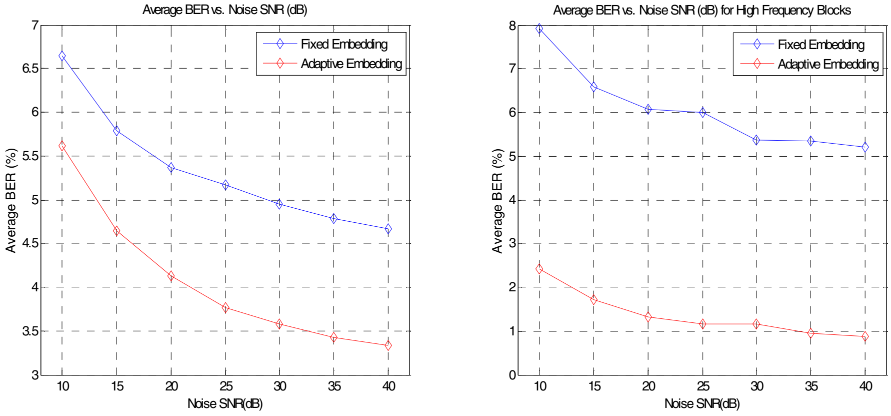
(a) (b)

Fig. 6. Average Bit Error Rate for the watermark *(%) vs.* Coding Bitrate *(kbps)* for the *fixed embedding* [5] and *adaptive embedding*, (a) for all the utilized files, (b) for only high frequency watermark blocks.

BER values are mostly greater than 8 % for the fixed approach. The adaptive approach provides a gain more than 6 % compared to the fixed case. Considering the automation of the watermarking process in a large database of compressed audio files with different characteristics, selecting the watermarking locations with respect to their energies offers a more coherent solution with MPEG coding.

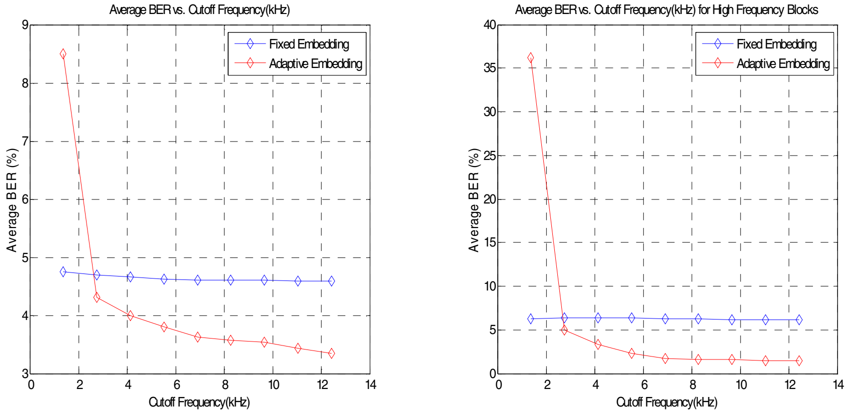
Figure 7 (a) shows the average watermark BER against additive white Gaussian noise in different strengths in our second experiment. The adaptive approach gives 1-1.5 % lower watermark BERs than the fixed approach. For the same watermark BERs in the graph, the gain in the adaptive approach against the noise attack is almost more than 10 dB for all BER values in the given range. The gain of the adaptive approach for high frequency watermark blocks is much apparent with the decrease of 4-5 % in BER values, as illustrated in Figure 7 (b). As the white noise affects all the frequencies equivalently, the adaptive approach has a better robustness due to the more watermark energy embedded to high frequencies with larger HAS thresholds.

Figure 8 (a) and (b) indicates the average watermark *BER* against low pass filtering for all the utilized files and for only high frequency watermark blocks, respectively, in our last experiment. Watermark BERs are computed for each case after a low pass filter with a different cut-off frequency is applied to the signal. As the watermark is always embedded to the first two subbands of the coded audio in the *fixed approach* [5]-[6], the watermarked components are not much affected by low pass filtering and a steady behavior is observed in both graphs. Down to the cutoff frequency of 2.8 kHz, the *adaptive* approach is giving better performance for both figures. As expected, the gain in BER rates for the high frequency blocks is higher again. However, after the point of 2.8 kHz there is a sudden increase in watermark BER of adaptive approach.



(a) (b)

Fig. 7. Average Bit Error Rate for the watermark *(%) vs.* Additive Noise *(dB)* for the *fixed embedding* [5] and *adaptive embedding*, (a) for all the utilized files, (b) for only high frequency watermark blocks



(a) (b)

Fig. 8. Average Bit Error Rate for the watermark *(%) vs.* Cutoff Frequency of the Low-pass filter *(kHz)* for the *fixed* [5] and *adaptive embedding*, (a) for all the utilized files, (b) for only high frequency watermark blocks

In order to explain this situation, we give the distribution of the highest energy subband in each frame for all the utilized audio files in Figure 9. After a low pass filtering with a cutoff frequency of 1.4 kHz. (corresponding to the first 2 subbands of 32 channels), about 10 % of the highest energy subbands including the watermark is being blocked, which cause a sudden increase in BER. However, after such a filtering SNR values have fall down to the 7-10 dBs and very annoying distortions in the high frequency regions have been experienced in the utilized songs.

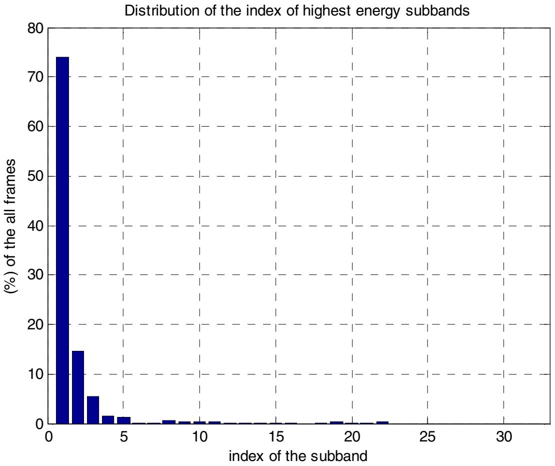


Fig. 9. Distribution (%) of the index of highest energy subbands for the utilized audio files

4 Conclusions

We investigate the selection of the watermark embedding locations in SS based audio watermarking methods in this paper. The results have indicated the weaknesses of fixing the embedding locations always in the low frequency band in the previous methods. The proposed adaptive method has solved the problem for the audio files with high frequency content by adaptively selecting the watermark locations according to the highest energy region in the frequency spectrum. Such a method enables the automation of watermarking process for large databases, in particular for the applications in exchange networks. The proposed method can also form a base for the selection of embedding locations for the other schemes in the literature. Future work will focus on the modeling of the proposed selection algorithm with respect to the bitrate coding to optimize the parameters of the proposed method.

References

[1] Pickholtz, R.L., Schilling, D.L., Milstein, L.B.: Theory of spread spectrum communications—a tutorial. IEEE Transactions on Communications 30, 855–884 (1982)

[2] Cox, I.J., Killian, J., Leighton, T., Shamoon, T.: Secure spread spectrum watermarking for multimedia. IEEE Trans. on Image Processing 12(6), 1673–1687 (1997)

[3] Drajic, D., Cvejic, N.: Audio Watermarking: State-of-the-Art. In: Al-Haj, A.M. (ed.) Advanced Techniques in Multimedia Watermarking: Image, Video and Audio Applications (May 2010)

[4] Chen, X.-M., Doerr, G., Arnold, M., Baum, P.G.: Efficient Coherent Phase Quantization for Audio Watermarking. In: IEEE International Conference on Signal Processing, pp. 1844–1847 (May 2011)

[5] Li, Z., Sun, Q., Lian, Y.: Design and Analysis of a Scalable Watermarking Scheme for the Scalable Audio Coder. IEEE Trans. Signal Process. 54(8), 3064–3077 (2006)

[6] Kirovski, D., Malvar, H.S.: Spread-spectrum watermarking of audio signals. IEEE Trans. Signal Process. 51(4), 1020–1033 (2003)

[7] Neubauer, C., Hurre, J.: Audio Watermarking for MPEG-2 AAC Bit streams. In: AES 108th Convention, Paris (February 2000)

[8] Neubauer, C., Hurre, J.: Digital watermarking and its influence on audio quality. In: 105th AES Convention, San Francisco (September 1998)

[9] van der Veen, M., Bruekers, F., Haitsma, J., Kalker, T., Lemma, A.N., Oomen, W.: Robust, multi-functional and high-quality audio watermarking technology. In: AES 110th Convention, Amsterdam, The Netherlands, May 12–15 (2001)

[10] Podilchuk, C.I., Zeng, W.: Image-Adaptive Watermarking Using Visual Models. IEEE J-SAC 16, 525–539 (1998)

[11] Hartung, F., Girod, B.: Watermarking of Uncompressed and Compressed Video. Signal Processing 66(3) (Special issue on Watermarking), 283–301 (1998)

[12] Langelaar, G.C., Setyawan, I., Lagendijk, R.L.: Watermarking Digital Image and Video Data. IEEE Signal Processing Magazine 17, 20–46 (2000)

[13] Lie, W.N., Chang, L.C.: Robust and high-quality time-domain audio watermarking subject to psychoacoustic masking. In: IEEE International Symposium on Circuits and Systems, vol. 2, p. 4548 (2001)

[14] Xu, C., Zhu, Y., Feng, D.D.: A Robust and Fast Watermarking Scheme for Compressed Audio. In: IEEE International Conference on Multimedia and Expo. (ICME 2001), p. 48 (2001)

[15] Foote, J., Adcock, J., Girgensohn, A.: Time base modulation: a new approach to watermarking audio. In: International Conference on Multimedia and Expo. (ICME 2003), vol. 1, pp. 221–224 (2003)

[16] Swanson, M.D., Zhu, B., Tewfik, A.H., Boney, L.: Robust audio watermarking using perceptual coding. Signal Process., Special Issue 66(3), 337–355 (1998)

[17] Kirovski, D., Attias, H.: Audio Watermark Robustness to Desynchronization via Beat Detection. In: Petitcolas, F.A.P. (ed.) IH 2002. LNCS, vol. 2578, pp. 160–176. Springer, Heidelberg (2003)

[18] Tachibana, R., Shimizu, S., Nakamura, T., Kobayashi, S.: An audio watermarking method robust again time-and frequency fluctuation. In: Proc. Security and Watermarking of Multimedia Contents III, vol. 4314, pp. 104–115 (January 2001)

[19] Wang, X.-Y., Zhao, H.: A Novel Synchronization Invariant Audio Watermarking Scheme Based on DWT and DCT. IEEE TSP 54, 4835–4840 (2006)

[20] Fulchiron, P.-Y., et al.: A Synchronization Method for Informed Spread-Spectrum Audio Watermarking. Journal of Systemics, Cybernetics and Informatics 1(6), 11–17 (2003)

[21] Mansour, M.F., Tewfik, A.H.: Audio Watermarking by time-scale Modification. In: IEEE ICASSP, vol. 2, pp. 1353–1356 (May 2001)

[22] Zeng, G., Qiu, Z.: Audio Watermarking in DCT: Embedding Strategy and Algorithm. In: IEEE ICSP, pp. 2193–2196 (December 2008)

[23] Information technology, Coding of moving pictures and associated audio for digital storage media at up to 1,5 Mbit/S, Part3: audio. British standard. BSI, London (October 1993)

IR Hiding: Method to Prevent Re-recording Screen Image Built in Short Wavelength Pass Filter Detection Method Using Specular Reflection

Takayuki Yamada[1], Seiichi Gohshi[2], and Isao Echizen[1,3]

[1] Graduate University for Advanced Studies, Japan
[2] Kogakuin University, Japan
[3] National Institute of Informatics, Japan
{nii20081705,iechizen}@nii.ac.jp, gohshi@cc.kogakuin.ac.jp

Abstract. We previously proposed using infrared (IR) LEDs to corrupt recorded content to prevent the re-recording of images displayed on a screen. This method is based on the difference in sensory perception between humans and devices and prevents re-recording by adding IR noise to the images displayed on the screen without it being detected by the human eye. However, it cannot prevent re-recording using digital camcorders equipped with a short wavelength pass filter to eliminate the noise. We have now improved our method by adding a simple countermeasure against such attacks. It detects IR light reflected off the filter by using the IR specular reflection properties of the filter, and thereby detects re-recording using digital camcorders equipped with a short wavelength pass filter. We implemented this countermeasure by adding one of two types of IR LEDs (bullet type and chip type with lens) to our prototype re-recording prevention system, which is installed on a B3-size screen. Testing showed that this enhanced system can detect camcorders with an attached short wavelength pass filter.

Keywords: short wavelength pass filter, infrared cut filter, infrared absorption filter, specular reflection, infrared camcorder.

1 Introduction

Today's small and highly functional digital camcorders can easily be carried into movie theaters and used to secretly record the displayed images with vivid color and high resolution. This has led in recent years to a growing problem of illegal recording of movies when they are shown in movie theaters. Moreover, the recording media is now digital, often in the form of a memory card. Data recorded on such media can easily be copied, written to other media, such as DVDs, and uploaded to the Internet, without degradation in the image or sound. This makes it possible to distribute the data worldwide immediately. The Motion Picture Association of America (MPAA) [1] estimates the damage caused by bootleg film recording to be three billion dollars per year [2]. This has contributed to the reduction in the number of people paying to

Y.Q. Shi, H.J. Kim, and F. Perez-Gonzalez (Eds.): IWDW 2011, LNCS 7128, pp. 111–125, 2012.

view movies in a movie theater, buying movies through legal channels, and renting movies for home viewing, resulting in serious financial losses by the movie industry.

Preventing the re-recording of movies being shown in a movie theater so as to protect the owner's copyright has thus become an even more urgent problem. One approach to solving it is to embed digital watermarks so that copyrighted materials can be detected. Unique information, such as ID number, are embedded in each image or each voice segment using digital watermarking technology. Information such as the movie theater in which the re-recorded movie was shown and the time it was re-recorded can be obtained by detecting the embedded watermarks [3–6]. Although such methods are effective in terms of obtaining such information, they are ineffective in identifying the person responsible for the re-recording unless the theater was equipped with an audience monitoring system. Moreover, they do nothing to prevent the re-recording act itself.

We previously proposed a different approach: using a re-recording prevention system based on the difference in sensory perception between humans and devices that prevents re-recording by adding infrared (IR) noise to the images displayed on the screen without it being detected by the human eye [7]. However, a prototype implementing such a method was unable to prevent re-recording using digital camcorders equipped with a short wavelength pass filter (SWPF) to eliminate the noise. We have now improved our method by adding a simple countermeasure against such attacks: the IR light reflected off the filter is detected using the IR specular reflection properties of the filter, and thereby detects re-recording using digital camcorders equipped with a short wavelength pass filter.

Section 2 briefly describes how an SWPF mounted on a camcorder eliminates infrared noise. Section 3 describes our countermeasure based on IR specular reflection. Section 4 describes its addition to our prototype system. Section 5 explains how we evaluated the enhanced prototype system, presents the results, and discusses the effectiveness of our countermeasure. Section 6 briefly summarizes the key points and mentions future work.

2 Elimination of IR Noise Using SWPF

According to the International Commission on Illumination (CIE), the wavelengths of visible light range from 380 to 780 nm [8]. However, the image sensor devices, such as CCDs and CMOSs, used in digital cameras and camcorders can generally detect light between 200 and 1100 nm, which gives them the high level of luminous sensitivity needed for shooting in the dark [9]. Our previously proposed re-recording prevention method [7] uses IR light of 870 nm to add noise to images displayed on a screen without it being detected by the human eye. It is, however, ineffective when a SWPF is used to filter out the IR light. Such a filter allows short wavelength light to pass and blocks long wavelength light, i.e., IR light.

SWPFs can be classified as either IR cut or IR absorption. An IR cut filter is a planar object with a dielectric multilayer. It reflects the IR light received from a single direction back in a single outgoing direction (specular reflection). An IR absorption filter is also a planar object that reflects the incoming IR light back in only one direction. However, since the most of IR light penetrates through the surface then is absorbed into the absorber mixed into the glass, the IR reflection is lower than that of an IR cut filter, and the reflectance can be almost the same as that of a glass surface.

In contrast, non-specular reflectors, such as scatter plates, have various shapes and surface treatments, so they reflect the incident IR light back in various directions (diffuse reflection). The filter detection algorithm can thus detect the use of an SWPF by analyzing the reflection images picked up by the IR camcorder. Establishing a countermeasure against the use of an SWPF is essential for making this method practical. The simple countermeasure we developed uses the IR specular reflection properties of the SWPF.

3 Countermeasure

3.1 Principle

Re-recording is basically done by pointing a camcorder towards the screen and pressing the record button. Therefore, as illustrated in Fig. 1, an SWPF attached to the lens of a camcorder would be parallel to the screen.

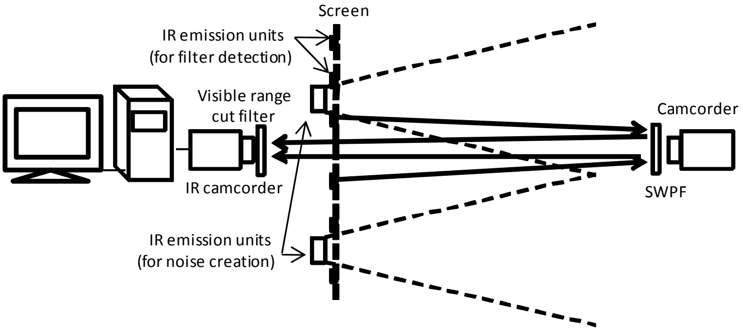


Fig. 1. Principle of SWPF detection

In our proposed countermeasure, as illustrated in Fig. 1, IR emission units for filter detection and for noise creation are attached at regular intervals on the backside of the screen. An IR camcorder with a visible range cut filter and placed behind the screen captures the IR light reflected by various objects, and an algorithm running on a PC analyzes the reflected light. The SWPF on a camcorder in front of the screen is a planar filter with a dielectric multilayer. As described above, it reflects IR light from a single incoming direction in a single outgoing direction (specular reflection). An IR absorption filter also reflects incoming IR light in only one direction. However, since it is a planar object, the wavelengths that it transmits depend on the quality of the absorber used in its glass plates. As mentioned above, its IR reflection is low compared with that of an IR cut filter.

The non-specular reflective objects, which have various shapes and surface treatments, reflect the incident IR light in various directions (diffuse reflection).

The filter detection algorithm thus detects the use of an SWPF by analyzing the images picked up by the IR camcorder and identifying the specular reflections. In the following section, we describe the requirements for the countermeasure in more detail.

3.2 Reflections Off Object Surfaces

The key to our countermeasure is distinguishing the reflections from an SWPF from those from other objects. The algorithm we use to do this is based on the Phong shading model [10]. In this model, there is a light source, an object, and a camcorder, and the spectral radiance $L_Q(\lambda)$ for one pixel is expressed as follows.

$$L_Q(\lambda) = I_e(\lambda)K_d(\lambda)cos\theta/ r^2 + I_e(\lambda)K_s(\lambda)(cos\varphi)^n + I_a(\lambda)K_a(\lambda). \quad (1)$$

r: distance from light source
θ: angle between light source and normal vector of object surface
φ: angle between camcorder and regular reflection
$I_e(\lambda)$: radian intensity of light source
$I_a(\lambda)$: radian intensity of ambient light source
$K_d(\lambda)$: diffuse reflectance of light source
$K_s(\lambda)$: specular reflectance of light source
$K_a(\lambda)$: reflectance of ambient light
$(0 \leq K_d(\lambda), K_s(\lambda), K_a(\lambda) \leq 1)$

The first term in Eq. (1) is called the diffuse reflection element, and it shows that the light reflects randomly and diffuses equally. The second term is called the specular reflection element, and it shows that the light reflects more strongly on object surfaces. The n is the decrease in reflection intensity; when the value is large, the object has properties of specular reflection, and when it is small, the object has those of diffuse reflection. The third term is called the ambient light element, and it shows the brightness of the light on the object surface that is not directly from a light source. Here, the light source is the IR emission units for filter detection, the object is the SWPF, and the camcorder is the IR camcorder. In the enhanced method, a short wavelength cut filter is attached to the IR camcorder to remove the effects of visible range light, thereby excluding the effects of visible light. From the information presented above, K_d, K_a, *and* φ are

$$K_d(\lambda) \cong 0, K_a(\lambda) \cong 0, \varphi \cong 0, \quad (2)$$

and Equation (1) becomes

$$L_Q(\lambda) = K_s(\lambda) \times I_e(\lambda) \ . \quad (3)$$

Here, the reflection of the IR cut filter coefficient to $K_s(\lambda)$, and the reflection of the IR absorption filter to $K_s'(\lambda)$ have the relationship

$$0 \le K_s(\lambda) < K_s'(\lambda) \le 1 . \tag{4}$$

If the object surface is curved one or is a diffused one, the specular reflection is lower, and the diffuse reflection is higher. Since the increment in the diffuse reflection element is inversely proportional to the square of the object's distance from the light source, the diffuse reflection is smaller than the spectral radiance when the object is an SWPF. We can express these characteristics using the following relationships.

(a) spectral radiance *$L_Q(\lambda)$* of IR cut filter:

$$L_Q(\lambda) \cong I_e(\lambda) \tag{5}$$

(b) spectral radiance *$L'_Q(\lambda)$* of IR absorption filter:

$$L_Q'(\lambda) \le I_e(\lambda) \tag{6}$$

(c) spectral radiance *$L''_Q(\lambda)$* of curved shape that is not a mirror:

$$L_Q''(\lambda) \le L_Q'(\lambda) \tag{7}$$

From these relationships, we can order the spectral radiances:

$$L_Q''(\lambda) \le L_Q'(\lambda) \le L_Q(\lambda) . \tag{8}$$

This makes it possible to identify an SWPF and other reflecting objects.

Equality holds in two cases.

(a) specular reflective objects such as a mirror with almost the same reflectance as an IR cut filter

(b) specular reflective objects such as a glass with almost the same reflectance as an IR absorption filter

Both types of reflective objects are unlikely to be in fixed position "facing" the screen during a certain period of time. Even if they did happen to be facing the screen, they would be automatically excluded as candidates by the motion estimation algorithm as soon as they were moved. In the unlikely event that a reflective object remained in a fixed state during a certain period of time, a size-and-shape algorithm would determine whether it was an SWPF.

3.3 Filter Detection Method

As shown in Fig. 1, the relationship between the positions of the IR emission units for filter detection and of the IR camcorder behind the screen is essential because the camcorder must be able to capture the emitted and reflected IR light source. The following section describes their arrangement.

Arrangement of IR Emission Units for Detection

Since the IR camcorder can stably detect the specular reflection from a SWPF with which the digital camcorders equipped, the IR emission units for detection should be arranged as shown in Fig. 1. The interval between them is derived as follows.

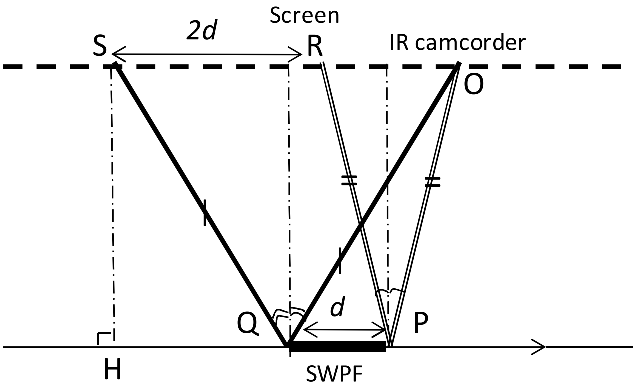
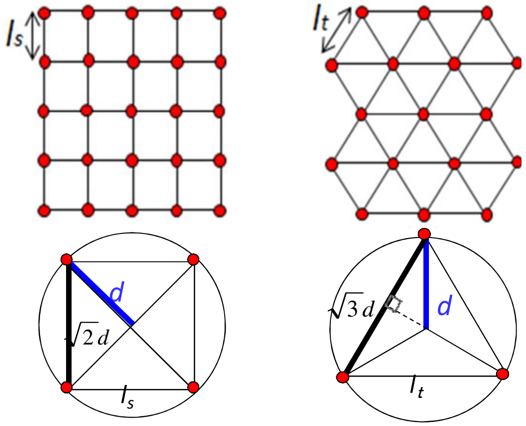


Fig. 2. Relationship between screen and SWPF



(a) Square lattice (b) Triangular lattice

Fig. 3. Arrangement of IR emission units for filter detection and interval calculation

Figure 2 illustrates the physical relationship between the screen and the SWPF. The IR reflection from the SWPF is detected along segment *QP* (length *d*) using the IR camcorder (point *O*). Since the IR light incident angle and reflection angle are equal to segment *QP* (from the property of specular reflection), we must place one or more IR emission units along segment *SR* (length of 2*d*). Since a screen is generally flat, the units should be placed on the detection plane.

The detection plane generally has a square lattice or a triangular lattice arrangement, as shown in Fig. 3. Since the position interval of the IR emission units depends on the size of the SWPF and on the filter form (generally square or circular), if the length of one side of a square filter and the diameter of a circle filter are set to *d*, we derive position interval l_s for a square lattice and l_t for a triangular lattice. Given the two types of lattice, it is necessary to determine the position interval of a square lattice and a triangular lattice in a square area with a side length of *2d* and in a circular area with a diameter of *2d* so that at least one IR emission unit is positioned to cover that area. A circle with a diameter of *2d* is inscribed in a square with a side length of *2d*. We thus need to determine the position interval on the basis of a circular area with a diameter of *2d*. Since we have to make the intervals of a square lattice and triangular lattice (l_s and l_t) shorter than the length of one side of the square and triangle that are inscribed in a circle *2d* in diameter, as shown in Fig. 3, we can derive

$$l_s \le \sqrt{2}d \tag{9}$$

$$l_t \le \sqrt{3}d \ . \tag{10}$$

Position of IR Camcorder

Someone attempting to illegally record the images displayed on a screen usually tries to capture the entire displayed image in the camcorder's viewfinder so as to reduce distortion of the recorded images. As a result, the normal vector of the surface of the SWPF attached to the camcorder is generally fixed and facing the center of the screen for a certain period of time. Thus, we can efficiently detect the reflection of IR light by placing the IR camcorder at the center of the screen. There are many tiny holes in the screen for sound transmission, and an IR camcorder behind the screen can capture the video image through them[1]. To determine the locations of the IR light sources, which depend on the screen size (number of IR emission units) and/or the theater size, we performed a preliminary assessment using our prototype SWPF detection system.

3.4 Filter Detection Algorithm

The video images recorded as described above are analyzed using an algorithm that detects specular reflection. It uses two sets of video images.

Video (a): shot in a room without an audience
Video (b): shot in the same room with an audience

By comparing the images between the two, the algorithm can eliminate the reflections from objects already in the room. The detection steps are listed below, and the flow is illustrated in Fig. 4.

Filter Detection Procedure

Step 1. Input image frames of video (a) and eliminate effect of flashing noise (from IR emission units).

Step 2. Average processed image frames and generate one averaged image frame.

Step 3. Do steps 4 through 8 for each series of image frames of video (b).

[1] The camcorder is positioned as close to the screen as possible, and the focus is set to infinity. As a result, the screen is blurred and the image recorded through the screen holes is sufficiently clear.

Step 4. Input image frames of video (b) and eliminate effect of flashing noise.

Step 5. Subtract pixel values of averaged image frame of video (a) generated in step 2 from those of each image frame of video (b) processed in step 4.

Step 6. Estimate motion areas for video (b) from image frames processed in step 4.

Step 7. Eliminate motion areas for video (b) using results of motion estimation (step 6) and eliminate diffuse reflective objects for video (b).

Step 8. Calculate areas for each reflection area, *S*, for video (b) and compare them with threshold *T*. Do the next step if the area is larger than the threshold.

Step 9. The object for detection is recognized by a labeling. Do the next step if detected camcorder faces screen for a certain period of time.

Step 10. If reflective object shape is circle or square, detect attack and display position of SWPF-equipped camcorder to PC display for analyzing.

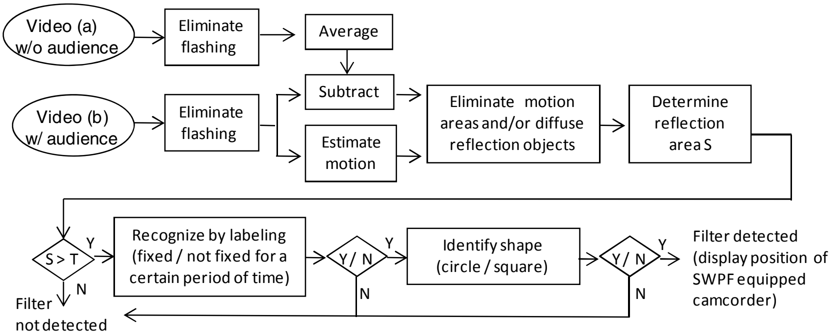

Video (a) w/o audience
Eliminate flashing
Average
Video (b) w/ audience
Eliminate flashing
Subtract
Estimate motion
Eliminate motion areas and/or diffuse reflection objects
Determine reflection area S
S > T
Y
N
Filter not detected
Recognize by labeling (fixed / not fixed for a certain period of time)
Y/ N
Y
N
Identify shape (circle / square)
Y/ N
Y
N
Filter detected (display position of SWPF equipped camcorder)

Fig. 4. Filter detection algorithm

4 Implementation

4.1 Description

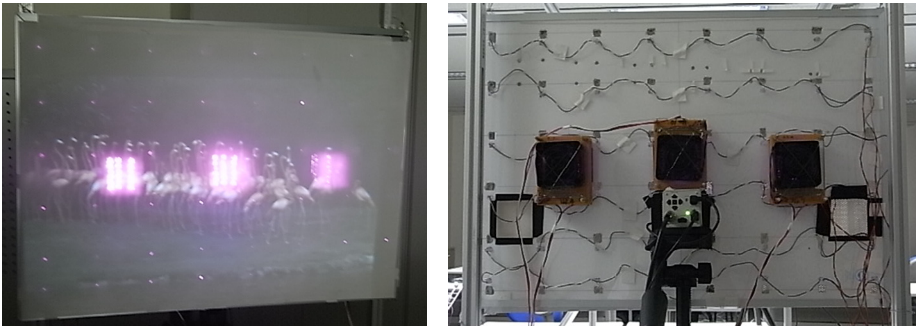
We added the proposed countermeasure to our re-recording prevention prototype system. This system consists of 3 IR emission units for noise creation, 24 or 48 IR emission units for filter detection, and an IR camcorder for recording images of the reflected IR light. The IR emission units for noise creation comprise 18 reflection-type IR LEDs, a short wavelength cut filter (cut-on wavelength of 870 nm) attached to the front, a cooling fan attached to the rear, and three IR emission units on the back of the screen are arranged horizontally. We used one of two types of IR-emitting LEDs (bullet type and chip type with lens) as the IR emission units for filter detection, thereby creating two prototypes for evaluation. Their specifications are summarized in Table 1, an overview of the bullet-type system is shown in Fig. 5, and one of the chip-type-with-lens system is shown in Fig. 6. The strength of the IR LED radiation can be compared by using a value representing the radiation intensity I_Q, which is the strength of the intensity of a point radiation source. The radiant intensity, which represents the amount of radiant energy Q per unit time t as the radiant flux φ radiates per unit solid angle Ω, is defined as

$$I_Q = d\varphi / d\Omega \,. \tag{11}$$

The perceived intensity of the light source depends on the wavelength of the light source, the viewer's visual sensitivity, and the spectral sensitivity of the camcorder, and it is highly dependent on the system configuration. Moreover, the radiation angle of each IR LED is shown at the half power angle, which is the range in which the radiant intensity of an IR LED becomes half.

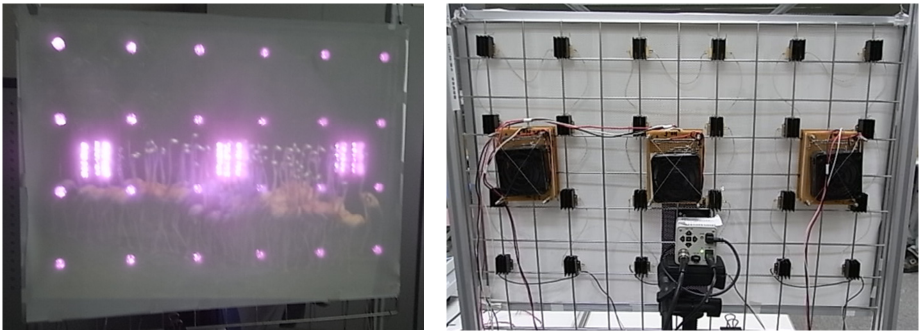
The bullet-type system uses bullet-type IR LEDs as the IR light sources for filter detection. As shown in Table 1, they have a radiation angle of ±7° and a wavelength of 940 nm. They are arranged behind the screen in a rectangular lattice. The chip-type-with-lens system uses chip-type IR LEDs as the IR light sources for detection. They have a narrower radiation angle (±4°) and a wavelength of 940 nm as well. They are also arranged behind the screen in a rectangular lattice.

The radiation angle is narrower in the chip-type-with lens system because a lens is placed in front of each LED. As a result, this system also has a detection range at far distance than the bullet-type system. The IR camcorder used for detection has high sensitivity in the IR wavelength range and is placed behind the screen at the center.



(a) Front (b) Back

Fig. 5. Bullet-type system overview



(a) Front (b) Back

Fig. 6. Chip-type-with-lens system overview

Table 1. Specifications of IR-emitting LEDs

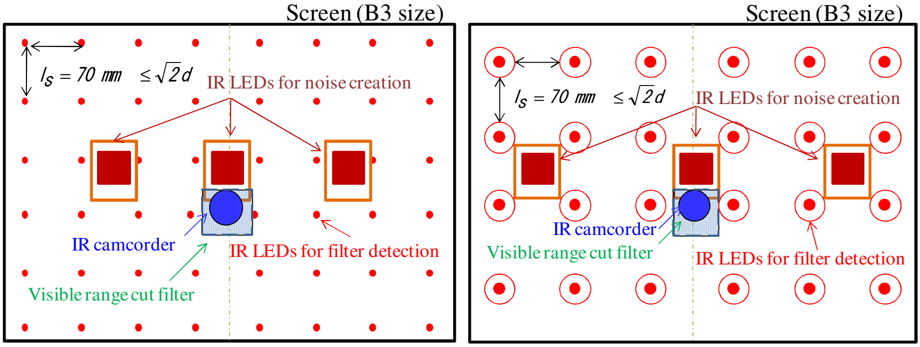
	Wavelength	Radiation angle	Radiant intensity
Bullet type	940 nm	±7°	0.16 W/sr
Chip type with lens	940 nm	±4°	5.20 W/sr

4.2 Arrangement of IR Emitting Units

In our evaluation, the detection targets were square SWPFs with a side length of 50 mm and circular SWPFs with a diameter of 50 mm. In accordance with Eq. (9), the distance between the IR LEDs for filter detection was set to 70 mm. The preliminary assessment mentioned above for determining the locations of the IR light sources was done using the following procedure.

1. An IR absorption filter with lower IR reflectance than an IR cut filter was attached to a camcorder.
2. The camcorder was sequentially placed in five locations in a dark room, and the arrangement of the IR LEDs was adjusted each time so that the IR camcorder could detect the specular reflection from the IR absorption filter. The camcorder was positioned in each case so that it could capture the entire image on the screen.

On the basis of the results of this preliminary assessment, we arranged the bullet-type IR LEDs in a rectangular lattice of eight columns and six rows (Fig. 7 (a)) and the chip-type IR LEDs in a rectangular lattice with six columns and four rows (Fig. 7 (b)).


Screen (B3 size)
$l_S = 70\ mm \leq \sqrt{2}d$
IR LEDs for noise creation
IR camcorder
IR LEDs for filter detection
Visible range cut filter
Screen (B3 size)
$l_S = 70\ mm \leq \sqrt{2}d$
IR LEDs for noise creation
IR camcorder
Visible range cut filter
IR LEDs for filter detection

(a) Bullet-type system (b) Chip-type-with-lens system

Fig. 7. Arrangement of IR-emitting LEDs for two prototype systems

5 Evaluation

5.1 Method

We used a dark room in our laboratory instead of an actual movie theater and the reflective objects shown in Fig. 8.

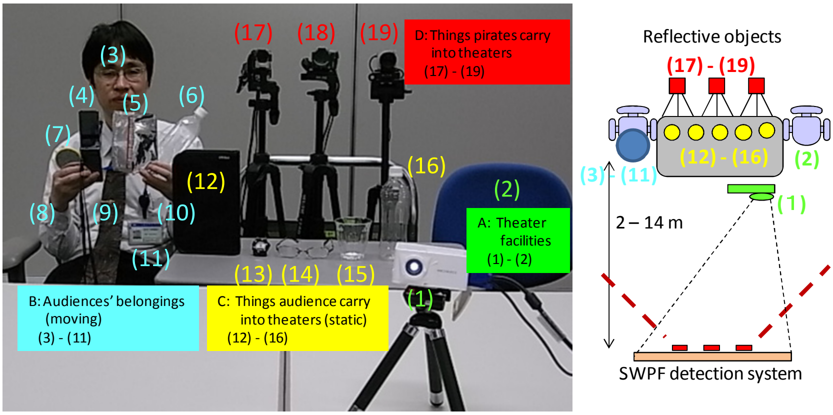

(3)
(4)
(5)
(6)
(7)
(8)
(9)
(10)
(11)
(12)
(17)
(18)
(19)
D: Things pirates carry into theaters (17) - (19)
(16)
(2)
A: Theater facilities (1) - (2)
(13) (14) (15)
(1)
B: Audiences' belongings (moving) (3) - (11)
C: Things audience carry into theaters (static) (12) - (16)
Reflective objects
(17) - (19)
(12) - (16)
(2)
(3) - (11)
(1)
2 – 14 m
SWPF detection system

Fig. 8. Experimental setup

As shown in Table 2, the objects can be divided into four groups. They were placed anywhere from 2 to 14 m from the screen, except for the three camcorders. They were positioned so that each one could capture the complete image on the screen. We set the comparison threshold at six pixels so that the IR absorption filter, which was placed at a distance of 14 m, could be detected. Using the results of the detection algorithm described in section 3.4, we evaluated the detection ability of each prototype.

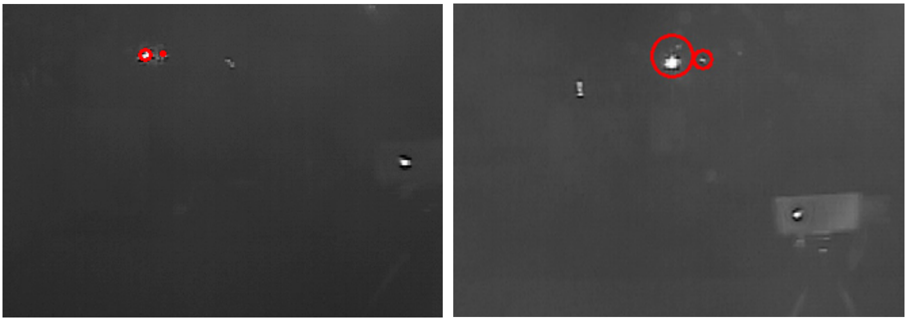
Table 2. Reflective objects used

Group	Object Type	Objects		
A	Theater facilities	(1) Beam projector	(2) Chair	
B	Audiences' belongings (moving)	(3) Eyeglasses (6) Plastic bottle (9) Tie clip	(4) Mobile phone (7) Hand mirror (10) Pen	(5) Snack package (8) Watch (11) ID card
C	Things audience carry into theaters (static)	(12) Nylon bag (15) Drinking glass	(13) Watch (16) Plastic bottle	(14) Eyeglasses
D	Things pirates carry into theaters	(17) Camcorder with attached IR cut filter (18) Camcorder with attached IR absorption filter (19) Camcorder (without filter)		

5.2 Results

An example evaluation image when detecting at a distance of 2 m is shown in Fig. 9 for each prototype. The red circles indicate areas with a threshold of six pixels or more. They were detected as an IR cut filter (object 17) and an IR absorption filter (object 18). These areas correctly correspond to the two camcorders with a filter, as shown Fig. 8.

The light source of the beam projector (object 1) was eliminated by the background difference step in the detection algorithm, and the moving objects (3–11) were eliminated by the movement detection algorithm; so only the two filters were detected. Filter detection takes about one second, so it is virtually done in real time. The proposed countermeasure is thus effective against attacks using an SWPF.



(a) Bullet-type system. (b) Chip-type-with-lens system.

Fig. 9. Evaluation images for two prototype systems

5.3 Comparison between Prototype Systems

Considering that the target use is in a movie theater, we evaluated the two prototype systems under various conditions. These conditions include lighting, background, size and direction of filter, type and condition of reflective objects, and detection distance. However, the use of a visible cut filter on the IR camcorder makes lighting a moot point. Moreover, the use of the background difference step in the detection algorithm eliminates the effect of the background. And because the SWPF is attached to the camcorder, it is probably about the same size as the camcorder lens. Furthermore, given the aim of the pirate, the SWPF attached to the camcorder is most likely parallel to the screen.

We thus placed the 19 reflective objects listed in Table 2 at various distances from 2 to 14 m from the screen and measured the detection rate. The detection algorithm can independently detect an SWPF attack from three consecutive image frames, which means that an SWPF can be detected ten times per second, assuming a video frame rate of 30 fps. For each measurement of the detection rate, we used a 20-second video clip, so the total number of detections, n, was 200 (= 20 seconds × 10). The detection rate, r, is thus given by $r = n_c/n \times 100$, where n_c is the number of detections in which the SWPF was correctly detected.

5.4 Results of Comparison

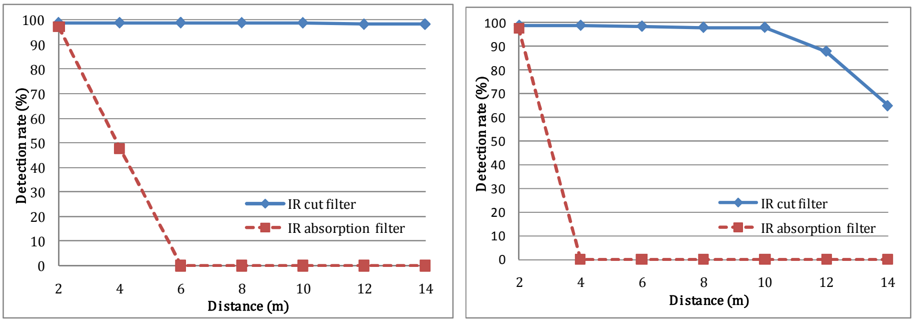
The detection rates are summarized in Table 2 for various distances from the front or the diagonal (5°). They are plotted in Figs. 10 and 11. The detection rate dropped at distances greater than 4 m with the bullet-type system (Fig. 10) and dropped at distances greater than 12 m with the chip-type-with-lens system. This is because IR light sufficiently strong for detection did not reach the filters on the camcorders. In general, LED radiation was centered at zero degrees (peak) and was distributed in a bell shape to the right and left. Therefore, a larger angle between the camcorder and an IR reflective object makes it more difficult to detect the reflected IR light. With both systems, the detection rates were higher when a reflective object was placed directly in front of the system.

In marketing, it is generally said that the accuracy of a number count by a person is about 90% [11]. Therefore, we evaluated the detection rate for near distance, middle distance, and far distance for four grades (Excellent, Good, Fair, Poor), as summarized in Table 3. The results show that the chip-type-with-lens system was marginally better at far distances than the bullet-type system, meaning that it is better for large places, such as a movie theater. In the diagonal case for the chip-type-with-lens system, the grades were "Poor." This can be resolved by widening the radiation angle of the chip-type IR LEDs while maintaining the radiant intensity and by attaching the IR LEDs at different angles.

Table 3. Detection grades* for two prototype systems

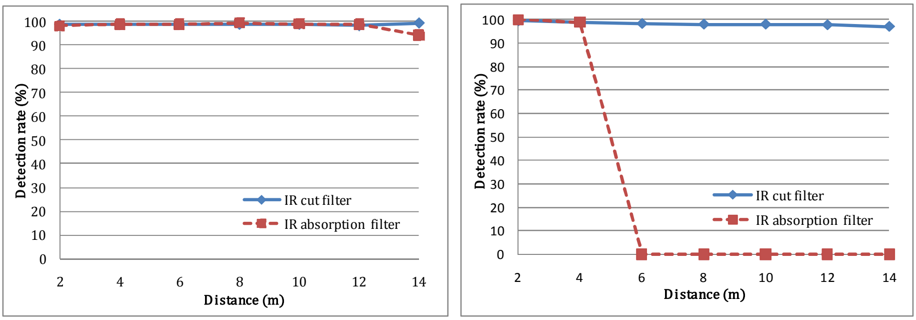
		Bullet type		Chip type with lens	
		IR cut filter	IR absorption filter	IR cut filter	IR absorption filter
Front	Near distance (2, 4 m)	Excellent	Fair	Excellent	Excellent
	Middle distance (6, 8 m)	Excellent	Poor	Excellent	Excellent
	Far distance (10,12,14 m)	Excellent	Poor	Excellent	Excellent
Diagonal (5°)	Near distance (2, 4 m)	Excellent	Poor	Excellent	Excellent
	Middle distance (6, 8 m)	Excellent	Poor	Excellent	Poor
	Far distance (10,12,14 m)	Fair	Poor	Excellent	Poor

*The four grades are defined every 2 m in accordance with the average of the detection rate for the measurement point: detection rate over 95%= "Excellent," more than 90 to less than 95% = "Good," more than 50 to less than 90% = "Fair," and less than 50% = "Poor."



(a) Front (b) Diagonal (5°)

Fig. 10. Detection rates for bullet-type system



(a) Front (b) Diagonal (5°)

Fig. 11. Detection rates for chip-type-with-lens system

6 Conclusion

The re-recording of images shown in a movie theater has become a social problem. Even though existing technical countermeasures using digital watermarking might create a mental deterrence, they are unable to prevent it. Therefore, we developed a method to prevent re-recording that actually prevents re-recording. However, it could be thwarted by attaching a short wavelength pass filter to the camcorder to cut or absorb the IR light used to create noise in the image. We have now developed a countermeasure against such attacks that uses the specular reflection properties of the filter. The results of an evaluation showed that its implementation using chip-type LEDs with a lens system works better than one using bullet-type LEDs in large places, such as a movie theater.

Digitization of images is progressing and sound facilities are growing rapidly, resulting in environments where contents other than movies can be shown at low cost. Realistic and powerful images, such as 3D images of a sporting event, are growing in popularity. The number of people who enjoy viewing sporting events and music concerts in a cinema complex is increasing. Consequently, the re-recording of images displayed on various types of devices in various environments will continue to proliferate. We thus plan to apply our re-recording detection method to various types of display equipment, such as CRTs and LCDs.

References

1. The Motion Picture Association of America (MPAA), `http://www.mpaa.org/`
2. Ezra, E., Rowden, T. (eds.): Transnational Cinema: The Film Reader. Routledge (2006)
3. Haitsma, J., Kaler, T.: A Watermarking Scheme for Digital Cinema. In: Proc. 2001 International Conference on Image Processing (ICIP 2001), vol. 2, pp. 487–489 (2001)
4. Gohshi, S., Nakamura, H., Ito, H., Fujii, R., Suzuki, M., Takai, S., Tani, Y.: A New Watermark Surviving After Re-shooting the Images Displayed on a Screen. In: Khosla, R., Howlett, R.J., Jain, L.C. (eds.) KES 2005. LNCS (LNAI), vol. 3682, pp. 1099–1107. Springer, Heidelberg (2005)

5. Nakamura, H., Gohshi, S., Fujii, R., Ito, H., Suzuki, M., Takai, S., Tani, Y.: A Digital Watermark that Survives after Re-shooting the Images Displayed on a CRT Screen. Journal of the Institute of Image Information and Television Engineers 60(11), 1778–1788 (2006)
6. Nakashima, Y., Tachibana, R., Babaguchi, N.: Watermarked Movie Soundtrack Finds the Position of the Camcorder in Theater. IEEE Transactions on Multimedia 11(3), 443–454 (2009)
7. Yamada, T., Gohshi, S., Echizen, I.: IR Hiding: A Method to Prevent Video Re-shooting by Exploiting Differences between Human Perceptions and Recording Device Characteristics. In: Kim, H.-J., Shi, Y.Q., Barni, M. (eds.) IWDW 2010. LNCS, vol. 6526, pp. 280–292. Springer, Heidelberg (2011)
8. Schanda, J. (ed.): Colorimetry: Understanding the CIE system. Wiley-Interscience (2007)
9. Holst, G., Lomheim, T.: CMOS/CCD Sensors and Camera Systems. SPIE-International Society for Optical Engineering (2007)
10. Zhang, H., Liang, Y.: Computer graphics using Java 2D and 3D. Prentice Hall (2006)
11. Mitsui, T., Yamauchi, Y., Fujiyoshi, H.: Human Detection by Two Stages AdaBoost with Joint HOG. SSII08, IN1-06 (2008)

IRIW: Image Retrieval Based Image Watermarking for Large-Scale Image Databases

Jong Yun Jun[1], Kunho Kim[1], Jae-Pil Heo[1], and Sung-eui Yoon[1,2]

[1] Dept. of Computer Science, KAIST, South Korea
[2] Div. of Web Sci. and Tech., KAIST, South Korea

Abstract. We present a novel, Image Retrieval based Image Watermark (IRIW) framework to identify copyright-violated images in both efficient and accurate manner for large-scale image databases. We first perform SIFT-based image retrieval to identify similar images given a query image and store them as an output list. Then we extract watermark patterns and check watermark similarity only for images stored in the list. As a final step, we re-rank images by considering various information available between each image in the list and the query image and by utilizing information even among images in the list. Also, in order to reduce any negative impacts on image retrieval by embedding watermark patterns on images, we propose to use a SIFT-aware image watermark detection method. Compared with the exhaustive method that checks all the images stored in an image database that consists of 10 K images, our method achieves more than two orders of magnitude performance improvement. More importantly, by identifying similar images given a query image and focusing on checking watermark similarities among those similar images, we are able to reduce false positive and false negative cases by a factor of up to two over the exhaustive method.

1 Introduction

Thanks to rapid advances of digital camera and various image processing tools, we can easily create new pictures and images for various purposes. This in turn results in a huge amount of images in the internet and even in personal computers. For example, flickr, an image hosting website, contains more than five billion images and flickr members update more than three thousand image every minute[1].

These huge image databases pose numerous technical challenges in terms of image processing, searching, storing, etc. One of the many challenging problems caused by easy image processing and modification technologies is the security problem. By the nature of digital data, it is very easy to copy, modify, redistribute the original image data. In order to address the security problem related to images, image watermark techniques have been studied actively in the last decade [21].

[1] `http://blog.flickr.net/en/2010/09/19/5000000000/`

Y.Q. Shi, H.J. Kim, and F. Perez-Gonzalez (Eds.): IWDW 2011, LNCS 7128, pp. 126–141, 2012.

The main concept of image watermarking is to embed visually imperceptible patterns on images so that a copyright holder of images can claim his or her ownership by extracting those patterns. Therefore, most image watermark techniques focus on extracting the embedded watermark patterns in a highly accurate manner against many different image attack scenarios (e.g., geometric transformation, cropping, and noise addition).

Even with drastic advances on image watermarking, the state-of-the-art image watermark techniques have certain false negative and false positive probabilities. As a result, a high number of false negative and false positive cases can occur, if we attempt to identify copyright-violated images solely based on image watermark techniques for web-scale image databases such as flickr. Furthermore, extracting watermark patterns and matching those patterns against the watermark pattern of the input query image can take prohibitive time for a large-scale image database consisting of millions of images or more.

Main Contributions: In order to efficiently and accurately identify images that are modified or are the exactly same images from a query image in large-scale image databases, we present a novel, Image Retrieval based Image Watermark (IRIW) framework. Instead of exhaustively scanning and extracting watermark patterns from all the images in the image database, we first identify similar images given a query image by using a SIFT-based image retrieval method (Sec. 4.1). Then we extract watermark patterns only from those similar images and measure watermark pattern similarities against the query image. Finally, we re-rank images by considering both image and watermark pattern similarities against the query image, in order to place images that are more likely to be copyright-violated in higher ranks in a final image list (Sec. 4.2). We propose to use a SIFT-aware image watermark method (Sec. 4.3) that does not embed watermark patterns on image regions where we get SIFT features, in order to minimize negative effects on our SIFT-based image retrieval method.

In order to verify the benefits of our method, we test our method in an image database that consists of 10 K images (Sec. 5). We found that our method improves the performance of searching copyright-violated images given a query image by more than two orders of magnitude over the exhaustive method that searches those copyright-violated images by accessing all the images in the database. More importantly, our method improves the accuracy of search results by reducing ratios of false negative and false positive cases up to two times over the exhaustive method. The performance and accuracy improvements of our method is mainly caused by identifying similar images based on image retrieval and by checking watermark similarities only for those images.

2 Related Work

In this section we review prior work on content-based image retrieval (CBIR), image watermarking, and their combinations.

2.1 Image Retrieval

CBIR has been drawing significant attention in recent years, and an excellent survey [7] is available. One of the most successful classes of CBIR techniques is based on using Scale Invariant Feature Transform (SIFT) [11] and the concept of visual words [17]. A visual word is a clustered set of similar SIFT features. An input query image is decomposed into a number (e.g., a few thousands) of SIFT features. Then, each SIFT feature of the query image is assigned to one or multiple visual words, which are precomputed with images stored in a database. Once we represent the query image with a set of visual words, then we find similar images from the database; the image similarity is defined in terms of the associated visual words for each image.

Since it can take a huge amount of time to identify similar images among a large number of images, hierarchical computation for visual words [13] or approximate computation [15] for similar images have been proposed. Our technique is based on one [13] of recent techniques that shows high runtime query performance and accuracy. However, our approach can be integrated with other SIFT-based image retrieval techniques.

A few CBIR techniques have been used to identify copyright-violated images by relying only on image features [4,8]. These techniques can be classified as CBIR methods designed for near-duplicate (or near-identical) image detection [5,20]. Since they do not use any watermarking techniques, it is unclear how robustly they can handle differently attacked images. Moreover, even though we identify copyright-violated images based on these near-duplicate image detection methods, these results provide limited legal claims over identifying copyright-violated images based on watermark techniques.

2.2 Image Watermarking

Image watermark algorithms have been extensively studied, and major image watermark techniques are well explained in a recent survey [21]. Most image watermark techniques are classified as spatial and transform/spectral domain techniques. Spatial domain techniques are easier to implement, but transform/spectral domain techniques [6] have been proven to be more robust for various image editing attacks.

In recent years, research on image watermark algorithms targets on achieving a higher robustness against to various geometrical distortions including RST (Rotation, Scaling, and Translation) attacks. Different approaches [21, p.26] have been proposed for these RST attacks. One class of techniques that are robust for RST attacks relies on using salient image features such as corners and edges of images. Utilizing such image features is useful, since the problem of geometric synchronization necessary for watermark extraction can be addressed by aligning those image features that are invariant to such geometric transformations.

Among image watermark techniques utilizing image features, Bas et al. [2] proposed a content-based synchronization algorithm by using image corners. They perform Delaunay triangulation [3] with the computed image corner points. Watermark patterns are embedded into each triangle of the constructed Delaunay triangulation. Also, Tang and Hang [18] use feature points computed by the Mexican Hat wavelet scale interaction that considers the intensity changes in images. Lee et al. [10] utilize SIFT features, the well-known image feature for image retrieval, for image watermarking. We propose to use this kind of techniques within our IRIW framework, in order to minimize any negative effects on the accuracy of our image retrieval component.

2.3 Image Retrieval with Watermarking

CBIR and image watermark techniques have been developed in separate fields. Recently there have been a few approaches that combine these two techniques.

Lu et al. [12] introduced a multipurpose watermarking scheme that embeds robust and fragile watermarks simultaneously in images. They also use image features that can be used for image retrieval as watermark patterns for images. Xu et al. [19] proposed an image retrieval technique that utilizes watermark patterns as features for image retrieval, and showed its retrieval performance in a small number of image data consisting of only eight different images. This method can allow users to identify images that have the exactly same watermark patterns. However, if watermark patterns of images are broken, this technique cannot identify similar images, since the method relies solely on watermark patterns for image retrieval. Furthermore, these two prior methods do not use image retrieval to improve the performance and accuracy of image watermark methods. In other words, results computed only based on watermark patterns may not include severely attacked images if their watermark patterns are broken. Also, this approach may report completely different images especially in large-scale image databases, because of certain false positive ratios of any watermarking techniques.

Unlike prior approaches that use image features or watermark patterns either for image retrieval or for image watermarking, we propose a novel, holistic framework that combines image watermark and retrieval techniques together such that it can improve both the performance and accuracy of image watermarking for large-scale image databases.

3 Overview

In this section we summarize issues with large-scale image databases and present the overview of our approach.

3.1 Issues with Large-Scale Image Databases

Suppose that a copyright holder wants to identify illegal usages (e.g., using the exact or modified images) of his/her images among images available on the internet. Even though addressing this kind of scenario is necessary because of the rapid advances of the internet, effective ways of handling large-scale image databases have not been actively studied in the context of image watermarking [21].

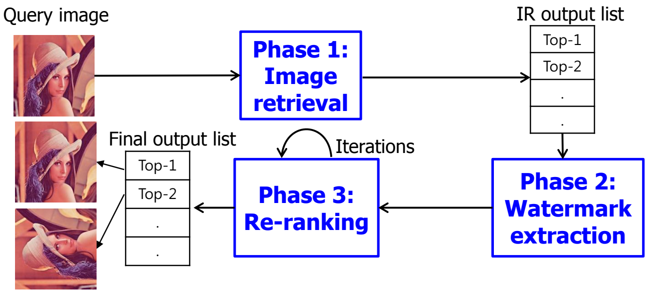


Fig. 1. This figure shows an overview of our IRIW framework

The most simple, but general approach for dealing with large-scale image databases is to exhaustively scan and extract watermark patterns from all the images in the database. More specifically, for each image on the internet, we can attempt to extract a watermark pattern and perform a *watermark similarity test* that measures a watermark similarity value by comparing the extracted pattern against the watermark pattern of the copyright holder. Then the exhaustive method reports a list of k images that have top k watermark similarity values in the image database.

In the list, however, we may fail to include copyright-violated images (e.g., the exact or modified images) given the query image or may incorrectly include irrelevant images, given an image watermark method, since any image watermark method has certain probabilities for false negative and false positive. Moreover, it is prohibitively expensive to search copyright-violated images by exhaustively scanning images in the image database and performing the watermark similarity tests.

One may want to accelerate the performance of identifying images that have top k watermark similarity values in the database, by transforming the problem of identifying such images into the problem of finding k nearest neighbors [1]. Then we can borrow well-established acceleration techniques for the nearest neighbor problem. One of the main acceleration techniques is to use a hierarchy (e.g., kd-trees) computed from image watermark patterns that are pre-extracted from images of the database, and to perform the nearest neighbor search given the watermark pattern of the query image.

This hierarchical approach, however, has a major limitation that makes the approach impractical. Since most image watermark techniques require a private key of the copyright holder to extract watermark patterns from images [21], it is impossible to even pre-extract watermark patterns until query images are available.

In this paper we aim to improve both the performance and accuracy of the exhaustive by adopting an image-retrieval technique as a culling step that does not need to pre-extract watermark patterns and still handles large-scale image databases.

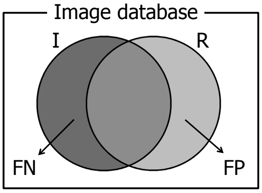


Fig. 2. The ground-truth images, I, that are modified from a query image, and a result set, R, computed by an image watermark method

3.2 Overview of Our Approach and Expected Benefits

As a pre-computation step of our IRIW approach, we construct a vocabulary tree with image features (e.g., SIFTs) of images in the database. Then, we perform our runtime algorithm that consists of three phases (Fig. 1): 1) image retrieval, 2) on-demand watermark extraction, and 3) re-ranking phases. Given a query image, we first identify similar images by performing our SIFT-based image retrieval method and store them in an output list, called *IR output list*. Then we extract a watermark pattern on demand for each image in the IR output list, followed by performing the watermark similarity tests between images in the output list and the query image. As a final step, we re-rank images in the output list by considering the computed similarity values and other additional information (e.g., similarity values among images in the output list), and provide our final output list to users. Also, we use a SIFT-aware image watermark technique that does not interfere with our SIFT-based image retrieval with watermarked images.

Our proposed method has the following benefits:

- **Higher Performance:** By identifying similar images given a query image and then performing the watermark similarity tests only against those similar images, we can drastically reduce the number of images that we need to consider for image watermarking, leading to fast runtime performance for large-scale image databases. Note that the image retrieval component serves as a culling step for an image watermark method employed in our IRIW framework.
- **Higher Accuracy:** By excluding dissimilar images based on our image retrieval component from the IR output list and by measuring watermark similarities against images stored only in the list, we can reduce the number of false positive cases in the final output list. This is because that it is likely that strongly dissimilar images are not modified from the query image and thus they are not copyright-violated with respect to the query image. Moreover, we also reduce the number of false negative cases by identifying similar images and placing them in the final output list, even though they may have low watermark similarity values caused by severe image editing attacks.

4 Our Approach

In this section we describe different steps of our approach in a detailed manner.

4.1 Image Retrieval Phase

As the first step of our method, we perform our image retrieval method to identify images that are similar to the given query image.

Suppose that given a query image, I_q, we have a set, I, of images modified from the query image I_q in an image database (Fig. 2); I serves as ground-truth results that are modified from the query image. Any image watermark methods aim to produce a set, R, of result images that contains all of those modified images. However, because of inaccuracy of image watermark methods, we may get a set, FP, of false positive images, which are irrelevant images (i.e. $FP \cap I = \phi$) given the query image, but are included in R. Moreover, we may fail to identify a subset, FN, of those modified images as false negative images; therefore, $FN \subseteq I$, but $FN \cap R = \phi$.

The goal of our image retrieval phase is to compute an image output list such that the list reduces the cardinalities of two sets FP and FN. To achieve our goal, we propose to use a SIFT-based image retrieval method, since it has been studied extensively recently and reported to perform well in terms of identifying images that have similar image features [7]. By performing our SIFT-based image retrieval method, we compute an output list, called *IR output list*, of images sorted in terms of *image similarity*, which will be explained later.

Note that we identify images that are similar to the query image and report them in the IR output list. Dissimilar images cannot be in the IR output list and thus will be excluded in the final output list (Fig. 1), even when some of dissimilar images happen to have relatively high watermark similarity values against the query image. As a result, we can reduce false positive cases. Also, severely modified or attacked images may have low watermark similarity values against the query image. It is possible that they may not be included in the final output list, if the list is computed from the exhaustive method that reports images sorted only in terms of watermark similarity values. Nonetheless, those severely attacked images may still have similar image features and thus can be included in the IR output list computed from our SIFT-based image retrieval method. Since our final output list contains all the images of the IR output list with different ranks in the list, those severely attacked image can be included in the final output list.

Pre-Computation: We perform our retrieval method based on the concept of visual words [17]. For all the images in the image database, we extract SIFT features and cluster them into visual words. In order to accelerate the clustering process, we adopt a hierarchical clustering method [13]. Starting from the root cluster that contains all the SIFT features, we recursively partition it into t different child clusters. We stop the recursive process if the depth of a cluster reaches a pre-defined threshold. Then we make those clusters leaf clusters that serve as visual words. For each leaf cluster, we compute a representative SIFT feature by averaging SIFT features assigned to the cluster and record images that are related to those contained SIFT features. This hierarchical construction method creates a t-ary tree that serves as a vocabulary tree.

Runtime Process: Once a user provides a query image at runtime, we extract SIFT features from the query image. Then for each SIFT feature, we traverse the vocabulary tree and find a leaf cluster whose representative SIFT feature is closest to the SIFT feature. We also add the images associated with the leaf cluster into a *similar image list*. Once we represent the query image with a set of visual words, then we compute the *image similarity value* based on the visual words of the query image and those of images stored in the similar image list [13]. As the final step, we sort images in the similar image list based on the computed image similarity values and store top r images in the IR output list, which is fed to the next phase.

4.2 Watermark Detection and Re-ranking Phases

After computing the IR output list, we measure watermark similarity values between the query image and images in the list; we will explain our image watermark method in the later section. Then, we re-rank images in the list by considering both image and watermark similarity values and store them in the final output list.

One can return the final output list, whose images are sorted only by the watermark similarity values. Note that it is highly likely that we get a very low watermark similarity values for severely modified or attacked images, even though our image retrieval method identifies them in the IR output list. As a result, these images are likely to be located near the bottom of the list and thus it hinders users to identify those modified images in an efficient manner. It is desirable to locate them higher in the list, even though they have low watermark similarity values.

In order to address this problem, we propose to re-rank images by utilizing information among the images stored in the list. Moreover, we re-rank images based on a weighted sum of image and watermark similarity values, instead of reporting images only according to the watermark similarity values for the final output list.

As an initial step, we associate a score value with each image in the IR output list, where the score value is initialized with the sum of image and watermark similarity values computed against the query image. According to the current score values, we sort images and store them in the list.

Then we perform our re-ranking by utilizing information available among images in the list. In each iteration of our re-ranking phase, we compute image similarity values between the first-ranked image and other images in the list. We accumulate the similarity value computed with each image in the list to the score associated with the image. As the final step of the iteration, we sort images in the list according to the current scores of those images. We iterate this

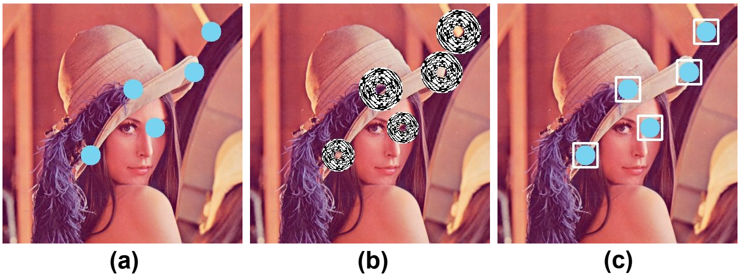


Fig. 3. This figure shows (a) the original Lena image with its SIFT features shown as circles, (b) watermark patterns that will be embedded around the SIFT features, and (c) watermarked image and its extracted SIFT features shown as rectangles with SIFT features of the original image shown in circles. We show only five SIFT features that have the top-five highest strength values.

process again with the next ranked image in the list. We found that running two iterations works well in our experiments.

4.3 SIFT-Aware Image Watermarking

Our image retrieval phase works by considering SIFT image features. If the image regions that contain SIFT image features are affected by embedded watermark patterns, results of image retrieval with watermarked images would be different from those before embedding watermark patterns on images. At the worst case, certain image features may not be extracted from the watermarked images. As a result, image retrieval may fail to identify similar images. This can deteriorate the accuracy of our framework, since our method performs image watermarking only with the IR output list computed from the image retrieval phase.

In order to prevent this problem, we propose to use an image watermark technique that takes advantage of SIFT features of images, inspired by image watermark techniques that utilize invariant image features [10]. We generate a donut-shaped watermark pattern (Fig. 3) and identify SIFT image features for each image. Then we embed the donut-shaped watermark pattern whose position is at the center of each extracted SIFT image feature.

Since a SIFT image feature is extracted from a 16 by 16 image region, the inner circle of each donut-shaped watermark pattern is computed to have a radius such that the inner circle can contain its associated 16 by 16 image region. For each image, about one thousand SIFT features are extracted. A *strength* variable for each SIFT feature is defined as the difference of Gaussians in two varying resolutions that contain the scale of the feature. Note that as a SIFT feature has a higher strength value, it is more likely that the SIFT feature survives with various attacks. As a result, we choose SIFT features that have high strength values and embed the donut-shaped watermark pattern at those SIFT features. More specifically, we choose SIFT features in the order of decreasing strength values, while avoiding any overlaps among the patterns associated with the SIFT features that are considered currently and were chosen previously.

Also, we found that in this configuration, the chosen SIFT features are well distributed across the image and thus our technique can be robust for attacks such as image cropping.

Since we embed the donut-shaped watermark pattern on the SIFT image features, local gradient values around the center point of each SIFT image feature is not changed. As a result, even after embedding watermark patterns, we can extract most of the same SIFT image features and thus achieve a similar result with image retrieval even after embedding watermark patterns.

5 Results and Discussions

We have implemented our IRIW method and performed various tests with a 32 bit machine that consists of 2 GB memory and 3 GHz CPU.

Image Benchmark: In order to test our method, we prepare an image benchmark that consists of 10 K images. The benchmark includes the well known images (e.g., Lena, Mandrill, and Goldhill) and images from the CalTech 101 and UKBench image datasets. In our image benchmark, 100 different categories (e.g., airplanes, cups, cars, etc.) are defined. Also, each category has ten different, but similar images. In each category, we select two images among ten similar images and embed two different watermarks into them. We leave the original un-watermarked images in our image benchmark. Since these original un-watermarked images do not have any watermark patterns, they can serve as images that could have been generated with ideal attacks, when we use watermarked query images. Also, to represent various attack scenarios, we attack each of watermarked images in eight different ways; we use the standard image attack generation tool, called Stirmark [14]. More precisely, these different attack scenarios include addictive noise (2% of the average pixel value), median filtering (3×3 box filter), center-cropping (75%), JPEG compression (lossy 70%), scaling (75%), rotation (45° and 90°), and shearing (1% extension along X and Y directions). Note that both JPEG compression and median filtering cause blurring that can affect SIFT features of images.

Vocabulary Tree Construction: Our image retrieval method is based on SIFT image features and uses the concept of vocabulary trees [13]. We perform the hierarchical k-means construction with SIFT features in order to construct a vocabulary tree. Our vocabulary tree has a depth of four with ten branches for each intermediate node; therefore, the tree has 10 K leaf nodes. From our image benchmark, we extract 4.5 million SIFT features, and it takes about 56 min to construct the vocabulary tree for the benchmark.

Comparison Setting: In order to show the benefits of our method, we compare the runtime performance and accuracy of our method against those of the exhaustive method that checks all the images in the image database. In both methods, we set them to report 30 different images as their results given a query image. The image retrieval component of our method also computes the IR output list that contains 30 different images. In all the tests, we perform 100 different search queries to identify copyright-violated images, and compare the average performance and accuracy between these two different methods.

5.1 Runtime Performance

Achieving a higher runtime performance for identifying copyright-violated images is very important to support search queries in large-scale image databases for a more number of users. Therefore, we compare the runtime performance of our method against the exhaustive method.

The exhaustive method computes the watermark similarity value for each image in our database, and spends about 19 min. to compute top-30 images sorted according to only watermark similarity values. The exhaustive method spends most of its running time of extracting and comparing watermark patterns. On the other hand, our method spends 5.9 sec. to compute the top-30 images according to the sum of image and watermark similarity values. Since our image retrieval component identifies a small subset (e.g., 30 images) of images that serve as candidates for potentially modified images from the query image and we perform our image watermark extraction for only those images, our method achieves a much higher runtime performance, more than two orders of magnitude performance improvement, over the exhaustive method.

Within the average running time, 5.9 sec. of our method, our method spends 0.34 sec. and 0.71 sec. to extract SIFT features from the query image and identify top-30 similar images. The rest of the running time, 4.9 sec., is spent on performing watermark extraction and watermark similarity tests.

One may think that we can pre-compute watermark patterns for images stored in the database and construct a hierarchical acceleration structure to improve the performance of the exhaustive method. However, as highlighted in Sec. 3.1, it is impossible in practice to pre-extract watermark patterns from images because many watermark methods can be used and some of them can use private keys associated with query images that disallow the pre-extraction. Therefore, we decide to compare our method against the exhaustive method that does not have such problems and works in a wide variety of usage scenarios for detecting copyright-violated images.

5.2 Accuracy

We measure the accuracy of two methods in terms of ratios of false negative and false positive results given the ground-truth results of query images. Inspired by notions of *precision* and *recall* used for image retrieval, we also connect ratios of false positive and false negative results with precision and recall respectively for image watermark methods.

We define the ratio, FP_r, of false positive results to be a ratio of the number of irrelevant images that are not in the ground-truth result of the query image, but are in the final output list, to the size of the final output list; therefore, $1 - FP_r$ can be interpreted as precision. We also define the ratio, FN_r, of false negative results to be a ratio of the number of ground-truth images given a query image that are not in the final output list, to the size of the final output list.

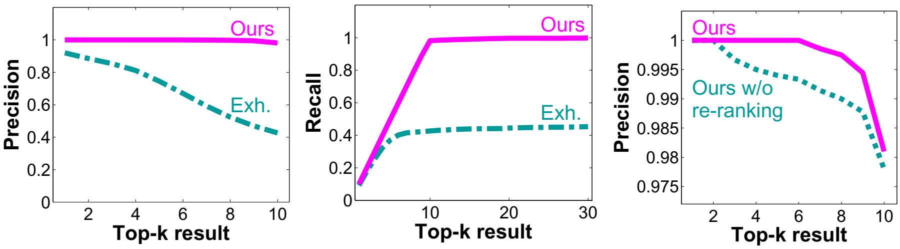


Fig. 4. The left and middle graphs show precision and recall curves of our method and the exhaustive method. The right graph shows precision curves w/ and w/o re-ranking images.

As a result, $1 - FN_r$ can be thought of as recall. Since the concepts of recall and precision are more intuitive, we represent the accuracy of different techniques in terms of those two concepts.

Fig. 4 shows the precision and recall curves of our and the exhaustive methods. Note that in our image benchmark there are ten ground-truth images (i.e. one original image, its watermarked image, and eight differently attacked images from the watermarked image) given a (watermarked) query image; ground-truth images for query images used in Fig. 5 are shown in Fig. 1 in the supplementary report, which is available at `http://sglab.kaist.ac.kr/IRIW`. As can be seen in the recall curve, our method achieves a near-linear recall curve up to the top-8 image in the final output list and reaches a recall value close to 1 around the top-12 and the top-13 images in the list. On the other hand, the exhaustive method does not achieve a recall value of more than 0.5, even though we allow up to top-30 images in the list. This is because many irrelevant images have more higher watermark similarity values than those of ground-truth images in the exhaustive method. Similarly, our method achieves up to two times higher precision results over the exhaustive method as we vary the size of the final output list. Improvement achieved by our re-ranking method is shown in the right graph of Fig. 4. Results before and after re-ranking are available in Fig. 2 of the supplementary report.

Examples of our results given two different query images are shown in Fig. 5. The exhaustive method achieves comparable results over our method for the Mandrill image up to top seven images. However, its result deteriorates after the top-7 images, while our method achieves accurate results up to top ten images; see Fig. 3 of the supplementary report for top-6 to top-10 images. In the Mona Lisa image, the exhaustive method reports an irrelevant image (Fig. 5-(p)) at the top-5 place, while our method reports one of ground-truth images, the original image, at the top-5 place. Since the original image does not have any watermark in it, it serves as one of images attacked by ideal image editing scenarios and– thus is very hard to be identified by prior image watermark methods. This result supports that our approach can detect copyright violated images even if their watermark patterns has been removed.

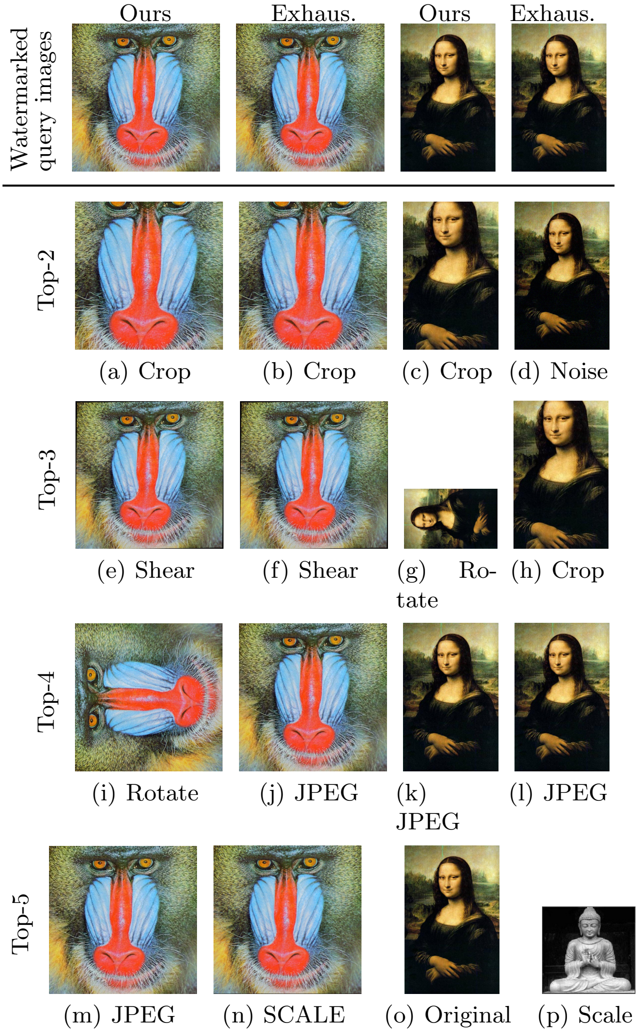


Fig. 5. This figure shows returned results in the top-5 images of our and exhaustive methods given the watermarked query images shown in the top row. We do not show the top-1 images since the query images are returned at the top-1 images in all the cases. Sub-captions from (a) to (p) represent image attack used to create the corresponding images. Top-6 to top-10 images are available at the supplementary report.

5.3 Discussions

Our Approach with Other Image Watermark Methods: To show benefits of our IRIW approach even with other image watermark methods, we combine our approach with a DCT-based image watermark method [16]. This DCT-based image watermark method works in the frequency domain, while our SIFT-aware image watermark method works in the spatial domain. Compared with the exhaustive method that uses the DCT-based image watermark method, our IRIW approach with the DCT-based method still achieves 233:1 performance improvement. Moreover, our IRIW approach with the DCT-based one achieves up to 2:1 accuracy (i.e. precision and recall) improvements in a similar manner shown in Fig. 4.

Effects on the Accuracy of Image Retrieval: To further verify the amount of effects of our SIFT-aware image watermark method on the accuracy of our SIFT-based image retrieval, we measure the mean Average Precision (mAP) of our SIFT-based image retrieval method. Our image retrieval method shows 0.99 mAP with images that do not have any watermarks. We also measure the mAP after embedding watermarks on all the images and mAP is changed only a bit (e.g., less than 1% changes). This result verifies that our SIFT-based image watermark method does not have major effects on the image retrieval accuracy even after embedding watermark patterns on images.

Limitations: Our IRIW approach employs an image retrieval component to cull most of irrelevant images given a query image. If our image retrieval component fails to identify similar images that are copyright-violated, our method cannot report such images in the final output list. However, we found that our SIFT-based image retrieval method works quite well in our tested image benchmark. Also, there may be attack scenarios, where our IRIW method may not work well. For example, one can apply severe blurring on images to affect most SIFT features and then deblur the blurred images based on recent advanced deblurring techniques. We expect that our method may not work well in such extreme cases, while the exhaustive method is also expected not to work well. Also, we can improve the accuracy of the exhaustive method by adopting simple geometric verifications [15] and culling irrelevant images based on simple image information (e.g., color histrom) given a query image. However, we can also adopt the same approach to our IRIW approach to further improve its accuracy.

6 Conclusion

We have presented a novel, Image Retrieval based Image Watermark (IRIW) approach that uses a SIFT-based image retrieval component to efficiently and accurately identify similar images from a query image. Our method extracts watermark patterns and measures watermark similarity values against images only in similar images identified from our image retrieval component. We have also proposed a re-ranking method to place severely attacked images even in higher positions in the final output list. Finally, we have proposed to use a SIFT-aware image watermark method that does not have negative effects on the image retrieval component. As a result, we were able to show more than two orders of magnitude performance improvement and up to two times accuracy improvement over the exhaustive method that scans all the images in an image database.

There are many interesting future research directions. In addition to addressing current limitations of our system, we would like to design an interactive

IRIW system for web-scale image databases based on a recent large-scale image retrieval method [9]. It would require massive parallelization on all the components of our current system. Also, we would like to investigate efficient watermark extraction methods by utilizing GPUs. Also, we found that sometimes users can provide additional information about similarities among images. Therefore, we would like to design effective visualization and browsing tools for large-scale image databases. Finally, we would like to design a frequency-domain image watermark method that maintains SIFT image features even after embedding watermark patterns.

Acknowledgements. We would like to thank anonymous reviewers and members of SGLab for their constructive comments and feedbacks. The first author was supported in part by the Korea Student Aid Foundation (KOSAF) grant funded by the Korea government (MEST) (S2-2009-000-00971-1). Also, this research is supported in part by MKE/KEIT [KI001810035261], MSRA, MKE/MCST/IITA [2008-F-033-02], KRF-2008-313-D00922, BK, DAPA/ADD (UD080042AD), MEST/NRF/WCU (R31-2010-000-30007-0), KMCC, and MCST/KOCCA/CT/R&D 2011.

References

1. Arya, S., Mount, D.M., Netanyahu, N.S., Silverman, R., Wu, A.: An optimal algorithm for approximate nearest neighbor searching. In: Symp. on Discrete Alg., pp. 573–582 (1994)
2. Bas, P., Chassery, J.-M., Macq, B.: Geometrically invariant watermarking using feature points. IEEE Trans. on Image Processing 11, 1014–1028 (2002)
3. de Berg, M., Cheong, O., van Kreveld, M., Overmars, M.: Computational Geometry: Algorithms and Applications. Springer-Verlag TELOS, Santa Clara (2008)
4. Berrani, S.A., Amsaleg, L., Gros, P.: Robust content-based image searches for copyright protection. In: Proceedings of the 1st ACM International Workshop on Multimedia Databases, pp. 70–77 (2003)
5. Chum, O., Philbin, J., Isard, M., Zisserman, A.: Scalable near identical image and shot detection. In: ACM International Conference on Image and Video Retrieval, pp. 549–556 (2007)
6. Cox, I.J., Kilian, J., Leighton, F.T., Shamoon, T.: Secure spread spectrum watermarking for multimedia. IEEE Transactions on Image Processing 6(12), 1673–1687 (1997)
7. Datta, R., Joshi, D., Li, J., Wang, J.Z.: Image retrieval: Ideas, influences, and trends of the new age. ACM Computing Survey 40(2), 1–60 (2008)
8. Huston, L., Sukthankar, R., Ke, Y.: Evaluating keypoint methods for content-based copyright protection of digital images. In: IEEE International Conference on Multimedia and Expo (ICME), p. 4 (July 2005)
9. Jégou, H., Douze, M., Schmid, C., Pérez, P.: Aggregating local descriptors into a compact image representation. In: CVPR, pp. 3304–3311 (2010)
10. Lee, H.Y., Kim, H.S., Lee, H.K.: Robust image watermarking using invariant features. Optical Engineering 45(3), 1–11 (2006)
11. Lowe, D.: Distinctive image features from scale-invariant keypoints. IJCV 60(2), 91–110 (2004)

12. Lu, Z.-M., Skibbe, H., Burkhardt, H.: Image Retrieval Based on a Multipurpose Watermarking Scheme. In: Khosla, R., Howlett, R.J., Jain, L.C. (eds.) KES 2005. LNCS (LNAI), vol. 3682, pp. 573–579. Springer, Heidelberg (2005)
13. Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. In: CVPR, pp. 2161–2168 (2006)
14. Petitcolas, F.: Watermarking schemes evaluation. IEEE Signal Processing Magazine 17(5), 58–64 (2000)
15. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: CVPR, pp. 1–8 (2007)
16. Piva, A., Barni, M., Bartolini, F., Cappellini, V.: Dct-based watermark recovering without resorting to the uncorrupted original image. In: ICIP, p. 520 (1997)
17. Sivic, J., Zisserman, A.: Video google: A text retrieval approach to object matching in videos. In: ICCV, vol. 2, pp. 1470–1477 (2003)
18. Tang, C.W., Hang, H.M.: A feature-based robust digital image watermarking scheme. IEEE Trans. on Signal Processing 51(4), 950–959 (2003)
19. Xu, J., Hua Qin, W., Ying Ni, M.: A new scheme of image retrieval based upon digital watermarking. In: Int. Symp. on Computer Science and Computational Tech., pp. 617–620 (2008)
20. Zhao, W.L., Ngo, C.W.: Scale-rotation invariant pattern entropy for keypoint-based near-duplicate detection. IEEE Transactions on Image Processing 18(2), 412–423 (2009)
21. Zheng, D., Liu, Y., Zhao, J., Saddik, A.E.: A survey of rst invariant image watermarking algorithms. ACM Computing Survey 39(2), 5 (2007)

Self-recovery Fragile Watermarking Scheme with Variable Watermark Payload

Fan Chen[1], Hongjie He[2], Yaoran Huo[2], and Hongxia Wang[1]

[1] Information Security and National Computing Grid Lab,
Southwest Jiaotong University, Chengdu 610031, China
[2] Sichuan Key Lab of Signal and Information Processing,
Southwest Jiaotong University, Chengdu 610031, China

Abstract. To take into account security, invisibility and recovery quality, this work proposes a variable-payload self-recovery fragile watermarking scheme. For each block, the watermarks include the total-watermark with variable number of bits and the basic-watermark of length 24 bits. The two watermark versions of each block are embedded into the less significant bit planes of the different blocks based on the secret key, respectively. They not only can partially resolve the coincidence tampering problem, but also improve the performance of tamper detection. The variable watermark payload preserves the adequate information of image block to as few bits as possible. Simulation results demonstrate that the proposed scheme not only provides a better invisibility and security against the known counterfeiting attacks, but also allows image recovery with an acceptable visual quality up to 60% tampering.

Keywords: fragile watermarking, self-recovery, variable watermark payload.

1 Introduction

The purpose of fragile watermarking is to achieve multimedia content authentication by imperceptibly embedding additional information into the host media [1, 2]. Many fragile watermarking schemes have been developed for digital images to detect accurately the tampered areas [3]. To further provide the tampering proofing, some fragile watermarking can recover approximately the original content in the tampered areas, which is also called as self-recovery watermarking [4].

Self-recovery watermarking techniques for image authentication usually partition the image into blocks with the same size. The watermark, or a part of it, is a compressed version of an image block (block-coding) so that the original content in the tampered regions can be reconstructed. For example, the important quantized DCT coefficients of image block of size 8×8 pixels [4-5], the average intensity of image block with size of 2×2 pixels [6-7] and the VQ indexing of block with size of 4×4 pixels [8]. In their methods, the length of block-coding are the same for different blocks, no matter the block is smooth or rough. As pointed in [9], the block-coding with fixed length is overmuch for smooth blocks, but is inadequate for a rough block. Therefore, Qian *et al* [9] proposed an algorithm of multilevel encoding, in which the blocks of size 8×8

Y.Q. Shi, H.J. Kim, and F. Perez-Gonzalez (Eds.): IWDW 2011, LNCS 7128, pp. 142–155, 2012.

pixels are dynamically classified into six types. For different types, the corresponding blocks are encoded to different number of bits so that the rougher blocks have more bits and the smoother blocks have fewer bits. This may improve the restoration quality of the tampered rough blocks and the security against the constant-feature attack [10]. Nevertheless, the average code-length of all blocks in host image must be fixed due to the limitation of watermark embedding method in Qian's scheme [9]. As a result, the block-coding may be overmuch for some smooth blocks in the smooth image and be inadequate for some rough blocks in the rough image. The overmuch code of image blocks decreases the invisibility due to the increasing watermark payload, and the inadequate code of image blocks would be impair the quality of reconstructed image.

In most self-embedding schemes, the watermark payload is more than the average length of the block codes in a host image. The reason of which lies in the following two aspects. First, the redundant information were introduced to resolve the tampering coincidence problem [2]. The content of a tampered block will fail to reconstruct if both the block and its hidden watermark in the other block are tampered. To address this problem, Lee and Lin [7] proposed a dual watermarking method. This scheme maintains two watermark copies of the whole image and provides a second chance for block recovery in case one copy is destroyed. Yang and Shen [8] maintained four watermark copies of the whole image to provide the many chances for block recovery. Qian *et al* [9] enlarged the compressed features from 64 bits to 160 bits, which bring redundancy to the initial block-coding for error correction. Second, the authentication data of each block were added to resolve the tamper detection problem. In most of the mentioned self-embedding schemes [2, 6-9], the watermark payload consists of authentication data and recovery data. The authentication data of a block were embedded in the block itself and used to determine the validity of itself. Additional authentication data increase watermark payload of the self-recovery system. Moreover, the block-wise independence of authentication data makes these self-embedding schemes vulnerable to the collage attack [11].

To address the aforementioned problems, this study proposes a self-recovery fragile watermarking scheme with variable watermark payload for image authentication. The block of 8×8 pixels is classified into six types according to the roughness of the blocks. The codes of a block includes the basic-code of length 24 bits and the total-code with different number of bits for different types of blocks. For each block, the watermarks are generated by encrypting the total-code and the basic-code and inserted into the less significant bit (LSB) planes of the different blocks based on the secret key, respectively. The embedded watermarks contribute to the tamper detection and content recovery. The watermark payload is variable due to the variable number of bits of total-code. The flexible watermark payload preserves the adequate information of image block with as few bits as possible, while taking into consideration the invisibility, security and recovery quality. Experimental results show that the proposed scheme provides a better invisibility and security against the known forgery attack such as the collage attack and the constant-feature attack. Moreover, the proposed scheme allows image recovery with an acceptable visual quality up to 60% tampering.

The remainder of this paper is organized as follows. In Section 2 the watermarked image generation procedure is described. Section 3 presents the tamper detection and recovery. Experimental results are given in Section 4 and conclusions are given in Section 5.

2 Watermarked Image Generation

The procedure of a watermarked image generation is described in two phases: variable-length block coding and variable-payload watermark insertion.

2.1 Variable-Length Block Coding

In this work, a host image X is divided into blocks $X=\{X_i|i=1,2,...,N\}$, where N is a number of blocks in the host image. For a block of 8×8 pixels $X_i = (x_{i1}, ... x_{i64})$, the corresponding quantized vector $Q_i=(q_{i1},...,q_{i64})$ (Zigzag scanning the 8×8 coefficient matrix) is computed by,

$$Q_i = \left[\frac{D_{ct}\left(4\times\left\lfloor\frac{X_i}{4}\right\rfloor\right)}{Q_T}\right] \tag{1}$$

where $\lfloor a \rfloor$ denotes the largest integer less than or equal to a, $[a]$ denotes the nearest integer of a, $D_{CT}(.)$ represents the DCT transformation, Q_T is the typical JPEG quantization table corresponding to the quality factor 50 [9]. Let α_i be the index of the last non-zero coefficient in the Q_i. If all coefficients in Q_i are zero, the value of α_{i} would be assigned to zero. Thus the value of α_i is an integer ranging from 0 to 64. For the sake of comparison, this work directly adopt the method of block classification in Qian [9]. The indexes [0, 64] are divided into 6 parts, {[0, 3], [4, 6], [7, 9], [10, 12], [13, 16], [17, 64]}, denoted as $[L_k , U_k]$ (k=1,2,...,6). The type of the block is set to the k if the value of α_i belongs to interval $[L_k , U_k]$.

The codes of a block X_i have two parts: the total-code and the basic-code, denoted as C_i^T and C_i^B, respectively. The length of total-code is variable and the length of basic-code is fixed to 24 bits for the different types of blocks. For an image block X_i, the proposed block coding procedure consists of two steps.

(1) Basic-code generation: the basic-code of a block X_i is formed by encoding the first three coefficients in Q_i. That is,

$$C_i^B = \llbracket q_{i1} \rrbracket^U || \llbracket q_{i2} \rrbracket^S || \llbracket q_{i3} \rrbracket^S = (c_{i1}^B, c_{i2}^B, \ldots, c_{i24}^B) \tag{2}$$

where || denotes the concatenation operator, $\llbracket a \rrbracket^{\mathrm{U}}$ and $\llbracket a \rrbracket^{\mathrm{S}}$ are the unsigned and signed binary code of an integer a, respectively. The code length of q_{i1}, q_{i2} and q_{i3} is 8 bits, thus the length of the basic-code C_i^B is 24 bits.

(2) Total-code generation: the total-code of a block X_i is formed by encoding the block type and the chosen coefficients in the corresponding quantized vector Q_i. That is,

$$C_i^T = \llbracket k \rrbracket^U || \llbracket q_{i1} \rrbracket^U || \llbracket q_{i2} \rrbracket^S || \cdots || \llbracket q_{l_i} \rrbracket^S = (c_{i1}^T, c_{i2}^T, \ldots, c_{iv_i}^T) \tag{3}$$

where l_i is the number of the encoded coefficients and v_i is the length of total-code of block X_i. For the sake of comparison, the number of the encoded coefficients and its corresponding code length for each type of blocks are the same as that of Qian's method [9]. Table 1. shows the code-length of the chosen coefficients for blocks corresponding to different types. From Table 1, the number of the encoded coefficients and length of the total-code are different for different types of blocks. Table 2 summarizes the number of the encoded coefficients and the length of total-code for six types of blocks.

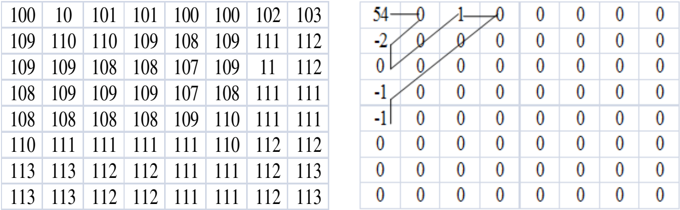
Table 1. Code-length of the chosen coefficients for blocks corresponding to different types

Types	Code-length of the chosen coefficients															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17~21
1	8	8	8														
2	8	7	7	6	6	6											
3	8	7	7	6	6	6	5	5	5								
4	8	7	7	6	6	6	5	5	5	5	4	3					
5	8	7	7	6	6	6	5	5	5	5	4	4	4	4	4	2	
6	8	7	7	6	6	6	5	5	5	5	4	4	4	4	4	3	3

Table 2. Total-code information for different types of blocks

Types	Type-code	Number of the chosen coefficients l_i	Code length v_i (bits)
1	001	3	27
2	010	6	43
3	011	9	58
4	100	12	70
5	101	16	85
6	110	21	101

Fig.1 is a instance of a block X_i and its quantized vector Q_i. Since the first three coefficients in Q_i are 54, 0 and -2 respectively, the basic-code of the block X_i is {00110110 0000000 10000010} according to (2). The type of the block X_i is 4 because the index of the last non-zero coefficient in the Q_i is 11. From Table 1 and Table 2, the twelve coefficients chosen for encoding are 54, 0, -2, 0, 0, 1, 0, 0, 0, -1, -1 and 0 in turn. According to (3), the total-code of the block X_i is {100 00110110 0000000 1000010 000000 000000 000001 00000 00000 00000 10001 1001 000}, whose length is 70 bits.



100	10	101	101	100	100	102	103
109	110	110	109	108	109	111	112
109	109	108	108	107	109	11	112
108	109	109	109	107	108	111	111
108	108	108	108	109	110	111	111
110	111	111	111	111	110	112	112
113	113	112	112	111	111	112	113
113	113	112	112	111	111	112	113

(a) Block X_i

54	0	1	0	0	0	0	0
-2	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
-1	0	0	0	0	0	0	0
-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

(b) Quantized vector Q_i

Fig. 1. A block and its corresponding quantized vector

2.2 Variable-Payload Watermark Insertion

To reduce the distortion caused by watermark insertion, the length of watermark data of each block should equal to that of block-coding of it. This requires that the watermark embedding method may differ for the watermark data with different length. Moreover, the basic-code and total-code of a block are encrypted and hidden in the different blocks based on secret key to improve the performance of detection and recovery. The embedding process consists of four operations.

Step 1: Partitioning and Block-Mapping. The host image $\boldsymbol{X}$ is partitioned into N blocks X_i (i=1,2,...,N) of 8×8 pixels. From two key-based pseudo-random permutation, two block-mapping sequences $\Psi = (\varphi_1, \dots, \varphi_N)$ and $\Lambda = (\sigma_1, \dots, \sigma_N)$ of the integer interval [1, N] are obtained. The detailed procedure of generating block-mapping sequence refers to Ref. [5].

Step 2: Watermarks Generation. For each block X_i , the basic-code $C_i^B = (c_{i1}^B, \dots, c_{i24}^B)$ and the total-code $C_i^T = (c_{i1}^T, \dots, c_{iv_i}^T)$ are generated from (2) and (3). They are encrypted with a secret key to construct the basic-watermark $W_i^B = (w_{i1}^B, \dots, w_{i24}^B)$ and the total-watermark $W_i^T = \left(w_{i1}^T, \dots, w_{iv_i}^T\right)$, respectively.

$$\begin{cases} w_{ij}^B = c_{ij}^B \oplus r_{ij}, & j = 1, \dots, 24 \\ w_{ij}^T = c_{ij}^T \oplus r_{i(j+24)} & j = 1, \dots, v_i \end{cases} \tag{4}$$

where $\{r_{i1}, r_{i2}, \dots r_{i128}\}$ is a key-generated random bit pattern, different for each block X_i.

Step 3: Basic-watermark embedding. For each block X_i, the basic-watermark $W_i^B = (w_{i1}^B, \dots, w_{i24}^B)$ is hidden in the second LSB of the last 24 pixels in the mapping block X_p, where $p = \varphi_i$. That is, the intensity of last 24 pixels in X_p are updated,

$$x_{p(j+40)} = 4\lfloor x_{p(j+40)}/4 \rfloor + 2w_{ij}^B + mod\left(x_{p(j+40)}, 2\right), j = 1,2, \dots, 24 \tag{5}$$

where $\lfloor a \rfloor$ denotes the largest integer less than or equal to a, and *mod(,)* is the modulo operation.

Step 4: Total-Watermark Embedding. Setting $q = \sigma_i$, the total-watermark $W_i^T = \left(w_{i1}^T, \dots, w_{iv_i}^T\right)$ of block X_i is hidden in the mapping block X_q. The watermarked block Y_q =(y_{q1},..,y_{q64}) is generated by one of the two cases. If v_i is not more than 64,

$$y_{qj} = \begin{cases} 2\lfloor x_{qj}/2 \rfloor + w_{ij}^T, & j = 1, \dots, v_i \\ x_{qj}, & j = (v_i + 1), \dots, 64 \end{cases} \tag{6}$$

otherwise,

$$y_{qj} = \begin{cases} 4\lfloor x_{q(j-64)}/4 \rfloor + 2w_{ij}^T + w_{i(j-64)}^T, & j = 65, \dots, v_i \\ 2\lfloor x_{q(j-64)}/2 \rfloor + w_{i(j-64)}^T, & j = (v_i + 1), \dots, 128 \end{cases} \tag{7}$$

3 Tamper Detection and Recovery

Suppose $\boldsymbol{Z}$ represents the tested image, which can be a distorted watermarked image or unaltered one. A binary sequence T=(t_i|i=1,2,...,N) called the tamper detection mark (TDM) is used to represent the location of tampering [5].

3.1 Tamper Detection

The tamper detection procedure includes the following steps.

Step 1: Partitioning and Block-Mapping. As in the watermark insertion process, the tested image **Z** is divided into non-overlapping 8×8 blocks Z_i and two block mapping sequences $\Psi = (\varphi_1, \dots, \varphi_N)$ and $\Lambda = (\sigma_1, \dots, \sigma_N)$ are obtained by the secret key.

Step 2: Watermarks generation and extraction. According to the content of the block Z_i and the secret key, the basic-watermark $W_i^B = (w_{i1}^B, \dots, w_{i24}^B)$ and total-watermark $W_i^T = \left(w_{i1}^T, \dots, w_{iv_i}^T\right)$ are computed from (4), respectively. Meanwhile, the extracted basic-watermark and total-watermark from the block Z_i, denoted as $E_i^B = (e_{i1}^B, \dots, e_{i24}^B)$ and $E_i^T = (e_{i1}^T, \dots, e_{i101}^T)$, are obtained by,

$$e_{ij}^B = mod\left(\left\lfloor \frac{z_{i(j+40)}}{2} \right\rfloor, 2\right), j = 1,2, \dots ,24 \tag{8}$$

$$e_{ij}^T = \begin{cases} mod\left(z_{ij}, 2\right), & j = 1,2, \dots ,64 \\ mod\left(\left\lfloor \frac{z_{i(j-64)}}{2} \right\rfloor, 2\right), & j = 65, \dots ,101 \end{cases} \tag{9}$$

Step 3: Watermarks Matching. By comparing the computed watermarks of block Z_i with the extracted watermarks from their corresponding mapping blocks Z_p and Z_q (where $p = \varphi_i$ and $q = \sigma_i$), the basic-watermark match-matrix $D^B = (d_1^B, \dots, d_N^B)$ and the total-watermark match-matrix $D^T = (d_1^T, \dots, d_N^T)$ are calculated by,

$$d_i^B = \begin{cases} 0 & , \ if\ W_i^B = E_p^B \\ 1 & , \ \text{otherwise} \end{cases} \tag{10}$$

$$d_i^T = \begin{cases} 0 & , \ if\ w_{ij}^T = e_{qj}^T\ \forall j \le v_i \\ 1 & , \ \text{otherwise} \end{cases} \tag{11}$$

where v_i is the length of total-code of block Z_i. From (10) and (11), all 24 bits in the basic-code but the first v_i bits in the total-code are used to detect the consistency of blocks.

Step 4 Adjacent-Based SDM: According to the basic-watermark match-matrix D^B and the corresponding block mapping Ψ, the basic-watermark TDM $T^B = (t_i^B | i=1,2,\dots,N)$ is obtained by the adjacent-block based statistical detection method (SDM) proposed in [5]. That is,

$$t_i^B = \begin{cases} 1 & , \ if\ (d_i^B = 1) \& \left(\Gamma_i \ge \Gamma_p\right) \\ 0 & , \ otherwise \end{cases} \tag{12}$$

where $p = \varphi_i$ and Γ_i denotes the number of nonzero pixels that are adjacent to the i^{th} pixel in the D^{B}. Similarly, the total-watermark TDM $T^T = (t_i^T | i=1,2,\dots,N)$ can be obtained by the D^T and the corresponding block mapping Λ.

Step 5 Tamper Detection: Setting $\Omega = (\omega_1, \dots, \omega_N)$, where $\omega_i = t_i^B + t_i^T$. The value of ω_i is an integer ranging from 0 to 2 since the value of both $t_i^B\ and\ t_i^T are$ 0 or 1. Let ξ_i denotes the sum of eight pixels adjacent to the pixel ω_i in the Ω. The TDM $T = (t_i | i=1,2,\dots,N)$ is obtained by,

$$t_i = \begin{cases} 1 & , \ if\ (\omega_i + \xi_i) > 6 \\ 0 & , \ otherwise \end{cases} \tag{13}$$

3.2 Tamper Recovery

After tamper detection, all blocks in test image are marked as either valid or invalid. The proposed recovery procedure is only for the invalid blocks. The tampered blocks can be classified into two categories: watermark-destroyed and watermark-reserved tampered blocks. If two mapping blocks of a tampered block are invalid, the tampered block is the former, otherwise it is the latter. The proposed tamper recovery procedure includes the following steps.

Step 1: Recovery for Watermark-Reserved Tampered Blocks. According to the TDM *T* and two block mapping sequence Ψ and Λ, the recovered image $R = \{R_i | i = 1,2,\ldots,N\}$ of the test image *Z* is initialized by,

$$R_i = \begin{cases} D_{ec}(E_q^T), & if\ (t_i = 1)\&(t_q = 0) \\ D_{ec}(E_p^B), & if\ (t_i = 1)\&(t_q = 1)\&(t_p = 0) \\ Z_i, & otherwise \end{cases} \tag{14}$$

where $q=\sigma_i$,, $p=\varphi_i$, and $D_{ec}(.)$ represents the inverse of the block coding described in Section 2.1.

Step 2: Recovery for Watermark-Destroyed Tampered Blocks. To mark the watermark-destroyed tampered blocks, the mark matrix $M = \{m_i | i = 1,2,\ldots,N\}$ is obtained by the following expression,

$$m_i = \begin{cases} 1, & if\ (t_i = 1)\&(t_q = 1)\&(t_p = 1) \\ 0, & otherwise \end{cases} \tag{15}$$

For each block R_i with m_i=1, the valid blocks adjacent to the block R_i are used to reconstruct it. Let $\{R_{i1},..,R_{i8}\}$ denote the eight blocks adjacent to block R_i and $\{\mathrm{m}_{i1},..,\mathrm{m}_{i8}\}$ be the corresponding mark of them. If there are one or more zero pixels in $\{\mathrm{m}_{i1},..,\mathrm{m}_{i8}\}$, the recovered block R_i is recovered by,

$$R_i = D_{ct}^{-1}\left(\frac{\sum_{k=1}^{8}(1-m_{ik})\times D_{ct}(R_{ik})}{\sum_{k=1}^{8}(1-m_{ik})}\right) \tag{16}$$

where $D_{ct}^{-1}(.)$ denotes the DCT inverse transformation. At the same time, the value of m_i is updated to 0.

Step 3: If $\exists\ m_i = 1$ in the mark matrix *M*, *steps 2* is repeated until the value of each pixel in *M* is zero.

4 Experimental Results

We conduct numerous experiments to demonstrate the effectiveness of the proposed self-recovery fragile watermarking scheme and compare with the typical self-recovery watermarking schemes in the performance of tamper restoration. For quantitative evaluation, several measurements are introduced. (a) Invisibility: PSNR (peak signal-to-noise ratio) between the original image and watermarked one, (b) Watermark payload: the number of bits per pixel (bpp), and (c) Restoration performance : PSNR between the recovered image and watermarked one.

4.1 Watermark Payload and Invisibility

Generally, the watermark payload ranges from 1 to 3 bpp (bit per pixel) in the mentioned self-embedding schemes [2, 7-9]. To ensure the invisibility of watermark, the watermarked image is commonly generated by substituting for the b (=1,2,3) LSB planes while keeping the MSB planes of the original image intact. Suppose that the original distribution of the data in the LSB planes is uniform. The average energy of distortion caused by embedding b bits watermark is,

$$E_D = \frac{1}{2^{2\mathrm{b}}} \sum_{i=0}^{(2^b-1)} \sum_{j=0}^{(2^b-1)} (i-j)^2 \tag{17}$$

Then the approximate PSNR of the watermarked image with respect to the original one is,

$$\mathrm{PSNR} \approx 10 \cdot \log_{10}(255^2/\mathrm{E_D}) = \begin{cases} 51.14\, dB & , \quad b=1 \\ 44.15\, dB & , \quad b=2 \\ 37.92\, dB & , \quad b=3 \end{cases} \tag{18}$$

The PSNR value of the watermarked image decreases with the increase of watermark payload. This indicates that the smaller the watermark payload is, the better the quality of watermarked image is.

To provide more information with a certain capacity, this work generates the block coding of an image block with an unfixed length. The watermark payload is generated by encrypting a block coding without adding redundant information and all embedded watermark data contribute to both tamper detection and content recovery. In the proposed watermarking scheme, the embedded watermarks in a block of 8×8 pixels include two parts: the basic-watermark of length 24 bits and the total-watermark of length ranging from 27 to 101 bits. As a result, the watermark payload of the proposed scheme no more than (101+24)/64=1.96 bpp. From (18), PSNR of watermarked images should be generally no less than 44 dB. On the contrary, in most of the mentioned self-embedding schemes, a part of watermark data (authentication data) were used to tamper detection and the other ones (recovery data) were used to tamper recovery. The watermark payload of these schemes [2, 7-9] was 3 bpp and the PSNR of watermarked images was about 37.92dB. Table 1 shows the watermark payload and invisibility for different watermarked images. From Table 1, the watermark payload of the proposed scheme ranges from 1.35 to 1.95 bpp, but that of Qina's scheme [9] is fixed to 3 bpp for different images. As a result, the quality of the watermarked images is better than that by Qian's scheme, indicated by PSNR of the proposed scheme ranges from 44.36 dB to 46.49 dB, which is about 7 dB higher than that of Qian's scheme.

4.2 Restoration Performance

Self-recovery watermarking schemes enable the detection of tampering or replacement of a watermarked image. The distinction mainly lies in the tamper localization accuracy and the quality of recovered images. The quality of a recovered image depends highly on the size of tampered regions and the complexity of the image content. Two watermarked images of size 512×512, a rough Fingerprint and a smooth Elaine, are used to demonstrate the performance of the proposed scheme in general tampering. Two tested images were randomly modified with different tamper ratios and the tampered blocks were detected and recovered. Fig. 2 shows the experimental results of the quality of the recovered images (PSNR) by the proposed scheme and Qian’s [9], respectively.

Table 3. The watermark payload and invisibility for different watermarked images

Images	Watermark Payload (bpp)		Invisibility (dB)	
	Proposed	Qian[9]	Proposed	Qian [9]
Boat	1.35	3	46.49	37.92
Lena	1.46	3	46.16	37.94
Airplane	1.47	3	46.02	37.93
Barbara	1.63	3	45.41	37.87
Mona Lisa	1.63	3	45.70	37.97
Elaine	1.64	3	45.52	37.94
Goldhill	1.73	3	45.12	37.93
Napoleon	1.74	3	45.06	37.96
Man	1.77	3	45.02	37.93
Fingerprint	1.95	3	44.36	37.79

Fig. 2 implies that the complexity of the image content has much impact on the performance of tamper recovery. The recovery quality of Elaine image is better than that of Fingerprint image in the same tampering ratio for both the proposed scheme and Qian's one. In the proposed and Qian's schemes, the watermarks of an image block were generated by encoding the important quantized lower order (1~21) DCT coefficients. The number of non-zero higher order DCT coefficients in a rough block is much more than that in a smooth block. As seen from Fig.2, the PSNRs of the proposed scheme are higher than those of Qian's scheme as long as the tampered ratio is no more than 20%, especially for a rough image. This may be due to the fact that some rough blocks were not adequately coding in Qian's scheme. Fig. 2 also shows that the restoration performance achieved by Qian's scheme suddenly drops when the tampering ratio is up to 35%. This is because the original reference-bit would not be recovered according to the extracted useful reference-bit if the tampering ratio is larger than 35% [9]. On the contrary, the restoration performance achieved by the proposed scheme decreases gradually as the tampering ratio increases.

Two particular examples of the experiment mentioned in Fig. 2 are shown in Fig. 3. Fig.3(a) is a tampered Fingerprint image, in which the tampered region is about 8.8% of the host image. The PSNR values of the recovered image by the proposed and Qian's schemes are about 35.49 dB and 31.77 dB, respectively. For the sake of comparison, the actual-sized contents corresponding to the tampered area in the watermarked image, the reconstructed image by the Qian's, and that by the proposed scheme are shown in Figs. 3(b), 3(c) and 3(d), respectively. There are some serious diamond effects in the Figs. 3(c) and 3(d). The reasons of creating diamond effects in Fig. 3(c) and Fig. 3(d) are different. The diamond effects in Fig. 3(c) are caused by the inadequate code of image blocks, while those in Fig. 3(d) are caused by the corresponding total-watermark being destroyed. The number of the blocks whose total-watermark is destroyed is larger with the increase of the tampering ratio. This is also the reason for the lower PSNR of the reconstructed image by the proposed scheme with the tamper ratios ranging from 20% to 35%. As the tampered ratio is more than 35%, Qian's scheme could not retrieve the reference-bits with the extracted bits, thus the PSNRs of the proposed scheme are higher than that of Qian's scheme. Fig. 3(e) is the tampered Elaine with 63.5% tampering ratio. The recovered Elaine of Fig. 3(e) by

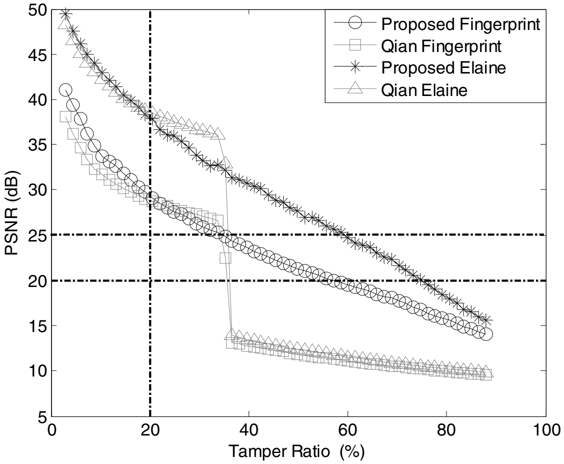


Fig. 2. Performance comparison of the restoration quality under general tampering with different tampering ratio

the proposed scheme, shown in Fig. 3(f), has the PSNR of 23.47 dB. These results indicate that the tampered image can be recovered by the proposed scheme with an acceptable visual quality even the tamper ratio is up to 60% of the host image.

4.3 Security

Self-recovery watermarking schemes enable the detection of general tampering of a watermarked image. However, not all self-recovery watermarking schemes have an ability against the collage attack proposed in [11]. This experiment considers the effect of the collage attack. In this test, two images, ‘Mona Lisa’ and ‘Napoleon’, both of size 372×288, were watermarked using the same key. The watermarked images of Napoleon and Mona Lisa are shown in Fig. 4(a) and Fig. 4(b), respectively. The collaged image, Fig. 4(c), was constructed by copying the face of Mona Lisa and pasting it onto the Napoleon image while preserving their relative spatial location within image. The tampering ratio of the collaged image is 19.86%. Figs. 4(d), 4(e) and 4(f) are the recovered images by the proposed, Qian [9] and He [5] schemes. In the collage attack, Qian's scheme could not recover the collaged image, indicated by PSNR of the recovered image is 17.22 dB. This is because Qian's scheme was not capable of withstanding the collage attack since the authentication watermark of each block was the block-wise independent. The invalid reference-bits extracted from the collaged region would be wrongly accepted as true. As a result, it is almost impossible to obtain the correct compressed codes of the tampered blocks by the wrongly reference-bits. In contrast, the proposed and He's [5] methods would effectively resist the collage attack. Since the blocks whose watermarks hidden in the other block are tampered could not be recovered in He's scheme [5], the proposed method has the better recovery quality. PSNR of the recovered image by the proposed scheme is 35.27 dB, which is 12 dB higher than that of He’s scheme. It indicates that the quality of the proposed scheme is the best in the collage attack, as evidenced by Figs. 4(d), 4(e) and 4(f).

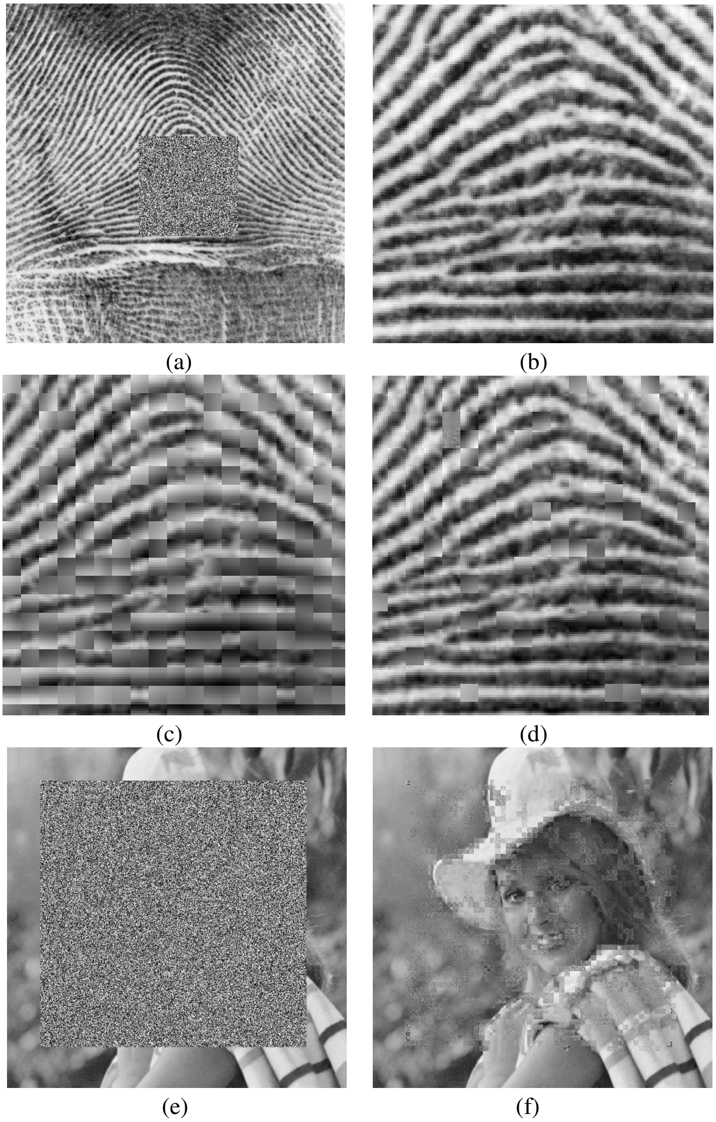
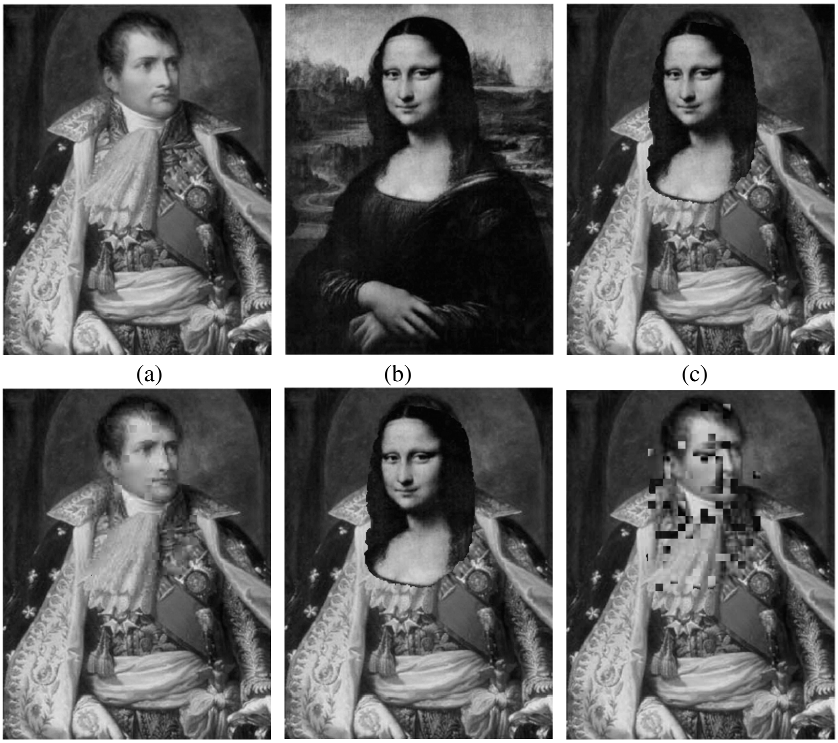


Fig. 3. Restoration quality comparison with different tampering ratios (a) tampered Fingerprint, the true size content corresponding to the tampered region in (b) the watermarked Fingerprint, (c) the recovered one by Qian's scheme [9], (d) the recovered one by the proposed scheme, (e) tampered Elaine, (f) the recovered Elaine by the proposed scheme



(a) (b) (c)

(d) (e) (f)

Fig. 4. Restoration quality comparison by the collage attack (a) Watermarked Napoleon, (b) Watermarked Mona Lisa, (c) Collaged image, Recovered images by (d) the proposed method, (e) Qian [9], and (f) He [5]

In the previous experiments, we only consider single tampered area under single attack. Here we examine the performance of the proposed scheme under multi-region and multi-attack tampering. The Barbara and Lena images with size of 512×512 pixels are chosen. The watermarked Barbara and Lena were generated by the proposed scheme with the same secret key, as shown in Fig. 5(a) and Fig. 5(b). Fig. 5(c) shows the multi-region and multi-attack tampered Lena image, in which four attacks occurred: (1) General tampering with two big flowers and several small ones; (2) Only content tampering: the content (5 MSBs) of a 106×121 rectangular region in the top-left side of the watermarked Lena was replaced by that of the Barbara image; (3) Collage attack: the face of the watermarked Barbara was collaged onto the watermarked Lena; and (4) Constant-feature attack: the region of Lena's face was modified by changing some uncoded DCT coefficients, and the size of modification was of 128×112 pixels.

Figs. 5(d), 5(e) and 5(f) show the recovered images by the proposed, Qian [9] and He [5] schemes, respectively. For the multi-region and multi-attack tampering, the proposed scheme exhibits much better tamper recovery performance than other schemes, indicated by the PSNR of 34.38 dB compared to the 16.58 dB by Qian and 23.89 dB by He. This is due to the fact that the methods by Qian [9] and He [5] cannot resist all the counterfeiting attacks such as the collage attack and the constant-feature attack, while the proposed method could effectively resist the counterfeiting attacks. This demonstrates that the proposed method outperforms other self-recovery watermarking algorithms in tamper recovery under multi-region and multi-attack tampering.

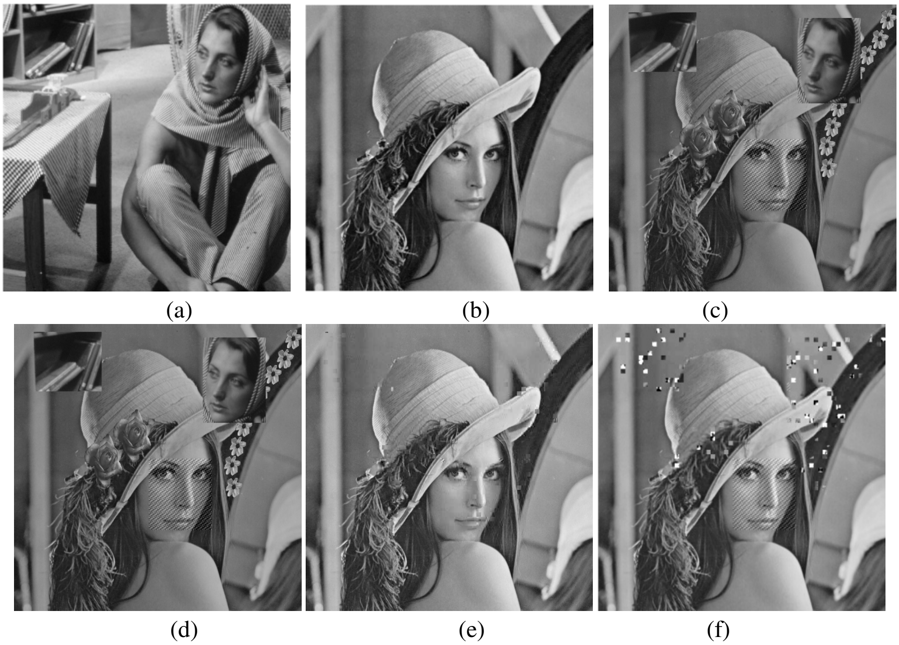


Fig. 5. Restoration quality comparison by the multi-region and multi-attack tampering. (a) watermarked Barbara, (b) watermarked Lena, (c) tampered Lena image, Recovered images by (d) the proposed method, (e) Qian [9], and (f) He [5]

5 Conclusion

We have proposed a self-recovery fragile watermarking method with variable watermark payload. The watermarks of an image block, including the total-watermark and the basic-watermark, are variable of bits. The variable watermark payload not only preserves the adequate information of image block to as few bits as possible, but also makes the distortion caused by watermark insertion as small as possible. The embedded watermark bits are used to both tamper detection and tamper recovery to further decrease watermark payload and improve security. Experiment results have demonstrated the superiority of the proposed scheme in comparison to other self-recovery fragile watermarking algorithms. Future research includes extending this approach to resist mild distortion such as JPEG compression, and analytic investigation on the tamper detection performance.

Acknowledgments. This work is supported in part by the National Natural Science Foundation of China (60970122, 60702025), by the Research Fund for the Doctoral Program of Higher Education (20090184120021), by the Fundamental Research Funds for the Central Universities (SWJTU09CX039, SWJTU10CX09), and by Beijing Key Laboratory of Advanced Information Science & Network Technology and Railway Key Laboratory of Information Science & Engineering. (No. XDXX1007).

References

1. Vleeschouwer, C., Delaigle, J.-F., Macq, B.: Invisibility and application functionalities in perceptual watermarking–An overview. Proc. IEEE 90(1), 64–77 (2002)
2. Zhang, X., Wang, S., Qian, Z., Feng, G.: Reference Sharing Mechanism for Watermark Self-Embedding. IEEE Trans. On Image Processing 20(2), 485–495 (2011)
3. Wong, P.W., Memon, N.: Secret and public key image watermarking schemes for image authentication and ownership verification. IEEE Trans on Image Processing (10), 1593–1601 (2001)
4. Fridrich, J., Goljan, M.: Images with Self-Correcting Capabilities. In: ICIP 1999, Kobe, Japan, October 25-28 (1999)
5. He, H., Zhang, J., Chen, F.: Adjacent-Block Based Statistical Detection Method for Self-Embedding Watermarking Techniques. Signal Processing 89, 1557–1566 (2009)
6. Lin, P.L., Hsieh, C.K., Huang, P.W.: A hierarchical digital watermarking method for image tamper detection and recovery. Pattern Recognition 38(12), 2519–2529 (2005)
7. Lee, T.Y., Lin, S.D.: Dual watermark for image tamper detection and recovery. Pattern Recognition 41(11), 3497–3506 (2008)
8. Yang, C.W., Shen, J.J.: Recover the tampered image based on VQ indexing. Signal Processing 90, 331–343 (2010)
9. Qian, Z., Feng, G., Zhang, X., Wang, S.: Image self-embedding with high-quaility restoration capability. Digital Signal Processing 21, 278–286 (2011)
10. Chang, C., Fan, Y.-H., Tai, W.-L.: Four-scanning attack on hierarchical digital watermarking method for image tamper detection and recovery. Pattern Recognition 41, 654–661 (2008)
11. Fridrich, J., Goljan, M., Memon, N.: Cryptanalysis of the Yeung-Mintzer Fragile Watermarking Technique. Electron. Imaging 11(4), 262–274 (2002)

Spread Spectrum-Based Multi-bit Watermarking for Free-View Video

Huawei Tian[1,2], Zheng Wang[1,2], Yao Zhao[1,2], Rongrong Ni[1,2], and Lunming Qin[1,2]

[1] Institute of Information Science, Beijing Jiaotong University, Beijing 100044, China
[2] Beijing Key Laboratory of Advanced Information Science and Network Technology,
Beijing 100044, China
hwtian@live.cn

Abstract. In Free-View Television (FTV) system, the user can freely generate a realistic arbitrary view of a scene from a number of original views. The copyright problem for free-view video content has been produced in the emerging FTV system. In this paper, we propose a spread spectrum-based multibit watermarking scheme for free-view video. The same watermark sequence is embedded into every frame of multiple views. The watermarking extraction is carried out in the DCT domain of virtual frame generated for an arbitrary view. Experimental results show that the watermark in FTV not only can be resistant to common signal processing but also can be detected from the virtual view generated for an arbitrary view.

Keywords: Free-view television, light field rendering, multi-view video, multi-bit watermarking, 3-D watermarking.

1 Introduction

Digital media widely spread along with the prosperity of information science and Internet technology. However, convenient manipulation and unrestricted copying of digital media bring on a considerable financial loss to the content providers and the media creators. Digital watermarking is introduced to prevent the above infringement.

Mono-view video watermarking has been widely studied [1-3] as a popular and powerful technique of copyright protection in the video transmission and processing. It embeds copyright information in the mono-view video. The ownership of the video can be verified by detecting the embedded copyright information.

Recently, generating a realistic arbitrary view of a scene from a number of original views has become faster and cheaper with the advances in image based rendering (IBR) [4]. One of the main applications is FTV, where viewers can select freely the viewing position and angle via IBR on the transmitted multi-view video. As in previous copyright problems for mono-view video, the copyright problem for multi-view video can also be treated by the using of watermarking. However, there are more challenging requirements, compared to well-studied mono-view video watermarking [5]. The owner of the multi-view video should prove his/her ownership, not only on

Y.Q. Shi, H.J. Kim, and F. Perez-Gonzalez (Eds.): IWDW 2011, LNCS 7128, pp. 156–166, 2012.

the original views of the multi-view video, but also on any virtual view, which is generated by the user using IBR from the original views.

Apler Koz *et al.* propose a watermarking approach inserts the watermark into each view frame of multi-view video in [5] and [6]. The watermark is modulated with the resulting output image which is obtained after filtering each view frame by a high-pass filter, and spatially added onto the view frame. The watermark is a sequence generated from a Gaussian distribution with zero mean and unit variance. The well-known correlation-based detection scheme is utilized during watermark extraction. If the correlation coefficient is big enough, the watermarking scheme claims to be success. In fact, this approach is intended to embed only one bit of information, i.e., presence or absence of the watermark.

In this work, we propose a multi-bit watermarking scheme for free-view video. Spread Spectrum-based direct-sequence code division multiple access (DS-CDMA) [7] watermarking method is used to embed multi-bit wateramrk sequence into discrete cosine transform (DCT) domain of each view frame. The watermark sequence is extracted bit-by-bit with a correlation detector from a watermarked view frame or a virtual frame generated for an arbitrary view. The detection algorithm should include a procedure to determine the position and direction of the virtual camera, because the watermark detector does not know the information. However, the research of determining the position and direction has been investigated by Koz *et al.* in [5], so we assume that the position and the direction of the virtual camera is priori in our proposed watermarking extraction method.

The rest of the paper is organized as follows. In Section 2, the light field rendering (LFR) [8] approach is introduced firstly, which is one of the competing IBR technology for FTV systems [9]. Section 3 describes the details of watermark embedding and detection procedure. The experimental results are illustrated in Section 4, and finally some conclusions are drawn in Section 5.

2 Light Field Rendering

In the literatures, light field approach is the most well known and preferred IBR technique. The first reason is that it does not require any geometry information but only relies on scene images which are easy to capture by common digital products. Second, it avoids building complex models, such as depth values or image correspondences, to extract the image values. Third, the new views can be constructed in real time and is independent of the scene complexity (only related with the size of the rendered image).

The basic assumption behind this technique is that the radiance along a ray remains constant if there are no blockers. Then a light field is built to capture all the necessary rays within a certain sub-space so that every possible view within a region can be synthesized [8].

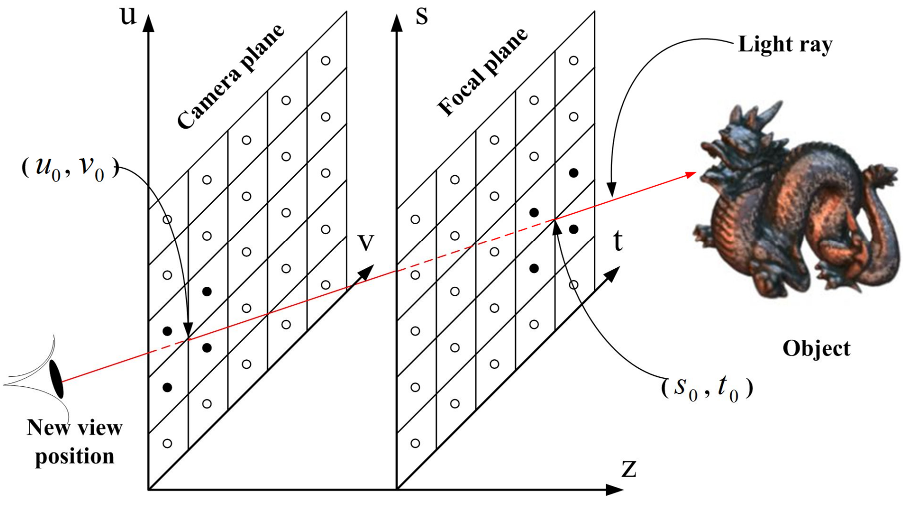


Fig. 1. A representation of the light field

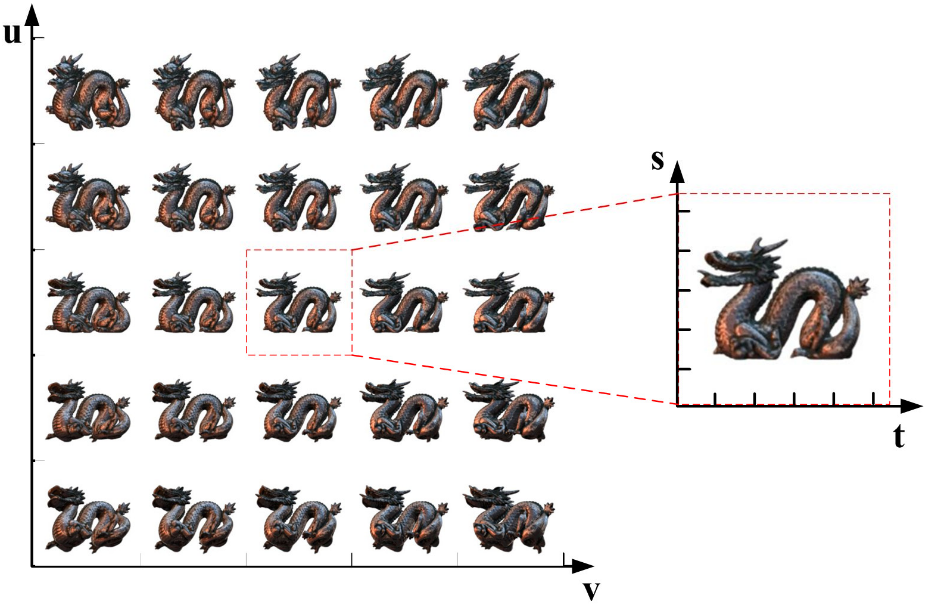


Fig. 2. A sample light field image array: *Dragon* [11]

In practice, a light ray is usually parameterized as lines by its intersections with two parallel planes, namely the camera plane and the focal plane (see Fig. 1). In Fig.1, a light ray is shown and indexed as an integer 4-tuple (u_0, v_0, s_0, t_0), where (u_0, v_0) and (s_0, t_0) are the intersections of the light ray with camera and focal planes, respectively. The two planes are usually discrete so that a finite number of light rays can be recorded.

If the light rays from all the points on the focal plane arrive at one point on the camera plane, then an image is generated (2D array of light rays). Therefore, the two planes can also be interpreted as a 2D array of images, as shown in Fig. 2. To generate a virtual view of the object for a random selected viewpoint, the light ray for each pixel of the rendered image is calculated by quadlinear interpolation existing nearby light rays in the image array. *Nearest neighborhood* interpolation and *bilinear* interpolation are two interpolation methods in LFR. Bilinear interpolation gives more natural and subjectively pleasant outputs than nearest neighborhood interpolation, so we choose the bilinear interpolation in our simulations (in Section 4 of the paper).

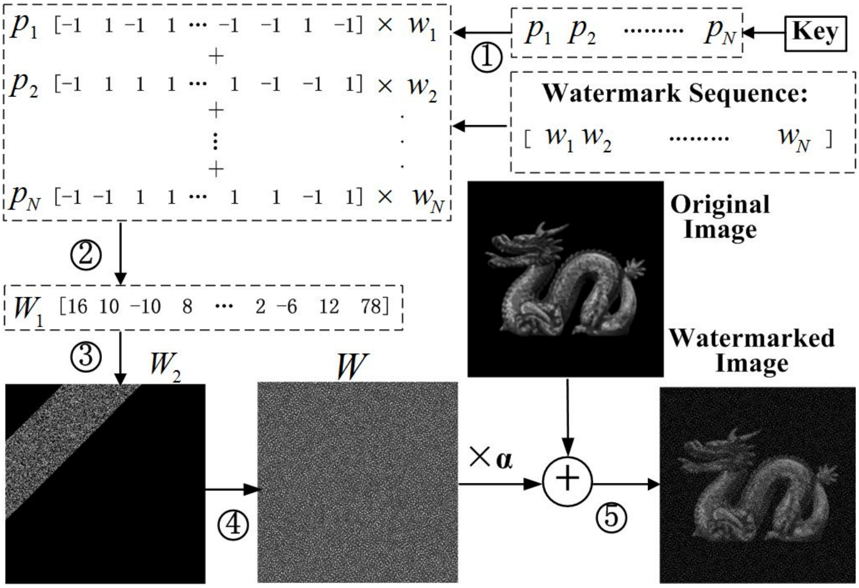

p_1 [-1 1 -1 1 ··· -1 -1 1 -1] × w_1
+
p_2 [-1 1 1 1 ··· 1 -1 -1 1] × w_2
+
⋮
+
p_N [-1 -1 1 1 ··· 1 1 -1 1] × w_N
①
p_1 p_2 ········· p_N
Key
Watermark Sequence:
[w_1 w_2 ········· w_N]
Original Image
②
W_1 [16 10 −10 8 ··· 2 −6 12 78]
③
W_2
W
Watermarked Image
×α
④
⑤

Fig. 3. Illustration of watermark embedding procedure. ①~⑤ indicates Step 1) ~ Step 5).

3 Proposed Watermarking Method

3.1 Watermarking Embedding

In the proposed watermarking scheme, the spread spectrum-based DS-CDMA watermarking scheme [10] which is well known for its robustness to common signal processing attacks is used to embed watermark into every images of the light field image array. The watermarking embedding procedure is demonstrated in Fig.3 and summarized as follows:

1) Generate M 1-D binary pseudo random sequence p_i, $i = 1,...,M$, as signature patterns using the private key as seed. Each of these sequences has zero mean and takes values from binary alphabet {-1, 1}. M is the number of bits in the watermark message. The length of p_i is N, $N > M$;

2) Create a 1-D DS-CDMA watermark signature W_1 by modulating the watermark message with the patterns generated in Setp 1), i.e., $W_1 = \sum_{i=1}^{M} w_i p_i$, where w_i is the ith bit (i.e., -1 or 1) in the watermark message $w = [w_1\ w_2 \cdots w_i \cdots w_M]$;
3) Convert the 1-D signature W_1 into a 2-D signature W_2 in a pre-selected zigzag scan (e.g., mid-range DCT coefficients); other coefficients are set to zero;
4) Apply the inverse discrete cosine transform (IDCT) to the 2-D signature W_2 to produce W ;
5) The final watermark signature W is embedded into each original light field image I using the formula:

$$I_w = I + \alpha W$$

where α is the watermarking strength. It produces the watermarked light field image I_w .

The whole procedure is equivalent to embedding the watermark signature W into the DCT domain of the light field image. The advantage is that it avoids any distortion which might have incurred to the original image [10].

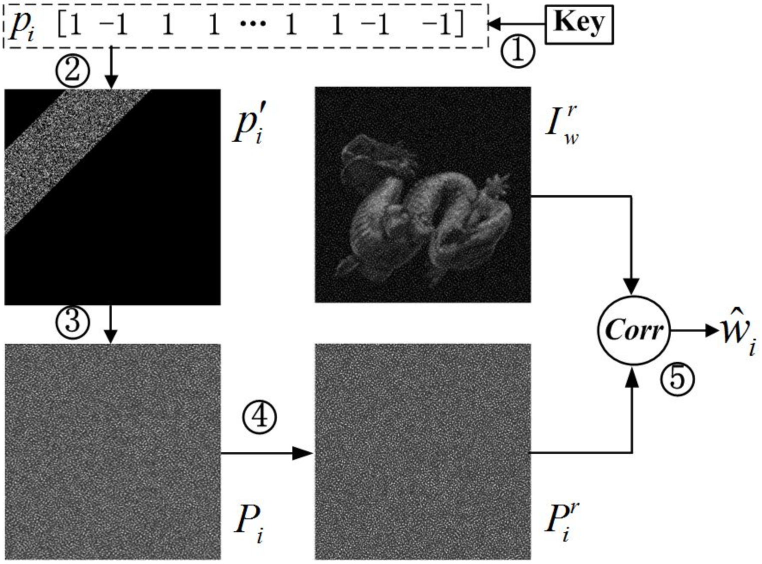


Fig. 4. Illustration of watermark extraction procedure. ①~⑤ indicates Step 1) ~ Step 5).

3.2 Watermarking Extraction

Rather than dealing with the general attacks for image and video watermarking, the major challenge of FTV is extracting the watermark message from an arbitrary view generated by LFR. The strategy of estimating the position and rotation for the imagery view has been investigated by Koz *et al.* in [5], so we can only focus on the state that the position and rotation of the virtual camera is known. The following steps are taken to decode the embedded watermark message in a rendered image I_w^r:

1) Regenerate 1-D binary pseudo random sequence p_i , $i = 1,...,M$, using the same key as in Step 1) of watermarking embedding. M is the number of bits in the watermark message. Each of these sequences has zero mean and take values from binary alphabet {-1,1};
2) Convert the 1-D pseudo random sequence p_i into a 2-D p_i' in a pre-selected zigzag scan (e.g., mid-range DCT coefficients), other coefficients are set to zero;
3) Apply the IDCT to the 2-D pseudo random sequence p_i' to produce P_i ;
4) Apply the same rendering operations during the generation of an arbitrary view to P_i , in order to generate a rendered watermark P_i^r (assuming the position and rotation of the virtual camera is known);
5) Decode the watermark message bit-by-bit using a correlation detector. That is, the *i*th bit of the watermark message is decoded as

$$\hat{w}_i = \begin{cases} 1, & corr(I_w^r, P_i^r) \geq 0 \\ -1, & corr(I_w^r, P_i^r) < 0 \end{cases}$$

where $corr(\bullet)$ is the correlation of two vectors. The extracted watermark message is $\hat{w} = [\hat{w}_1 \ \hat{w}_2 \cdots \hat{w}_i \cdots \hat{w}_M], \hat{w}_i \in \{-1,1\}$.

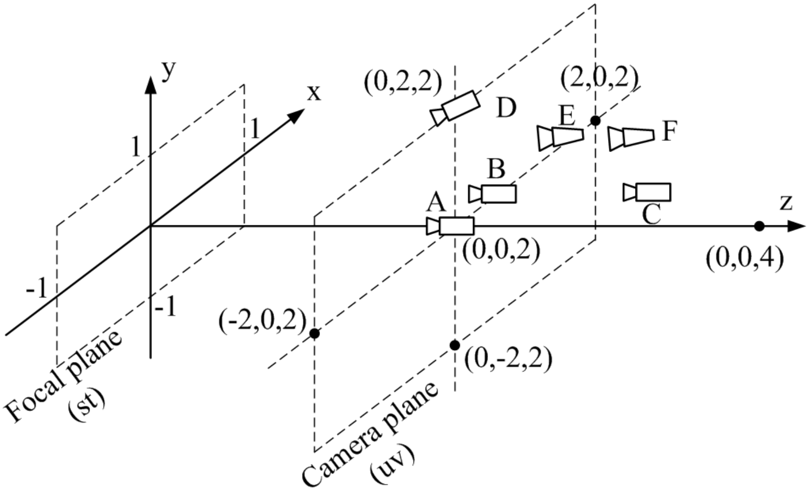


Fig. 5. Location of camera & focal plane for *Dragon* light field

4 Experimental Results

A common light field, *Dragon* [11], is used in the simulations. The parameterization of the focal and camera plane for *Dragon* light field is shown in Fig. 5. The size of *Dragon* light field image is 256×256 pixels. The watermarking strength α is set to 1.2. The length of the pseudo random sequence p_i is set as $N = 30000$. The length of the watermark message M is 50 bits. The watermark message is only embedded into the brightness component of the color image in the simulation. The capacity of the watermarking scheme should triple, if watermark message is embedded into three components (i.e. RGB channels). The decoding bit-error rate (BER), defined as the ratio between the number of incorrectly decoded bits and the total number of embedded bits, is used to evaluate the robustness of the watermarking scheme. 20 different randomly generated watermark sequences are tried and the BER is taken as the average of the 20 cases.

4.1 Imperceptibility Test

Typical rendered views for the original and watermarked Dragon light field are presented in Fig.6. Virtual camera is located at [0 0 2] with the normal direction of [0 0 -1] (Position-A in Fig. 5). The peak signal to noise ratio (PSNR) value between Fig. 6 (a) and Fig. 6 (b) is 32.6. Fig. 6 (c) shows the difference between Fig. 6 (a) and Fig. 6 (b) which has been multiplied by 2 for the purpose of better display. Another example is given in Fig.7, the virtual camera is located at [1.5 0 2] with the direction of [-1 0 -1] (Position-E in Fig. 5). The PSNR value between Fig. 7 (a) and Fig. 7 (b) is 34.2. From Fig.6 and Fig.7, we can see that the fidelity of the watermark is very high.

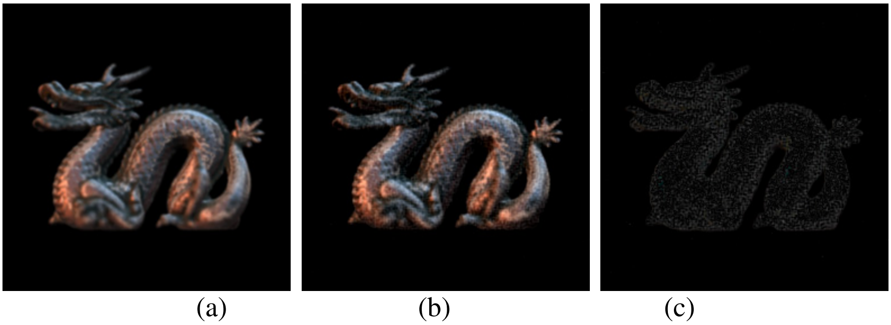


Fig. 6. (a) Rendered view of virtual camera at Position-A in *Dragon* light field, (b) watermarked view at Position-A in *Dragon* light field, (c) The difference between (a) and(b), multiplied by 2 for the purpose of better display.

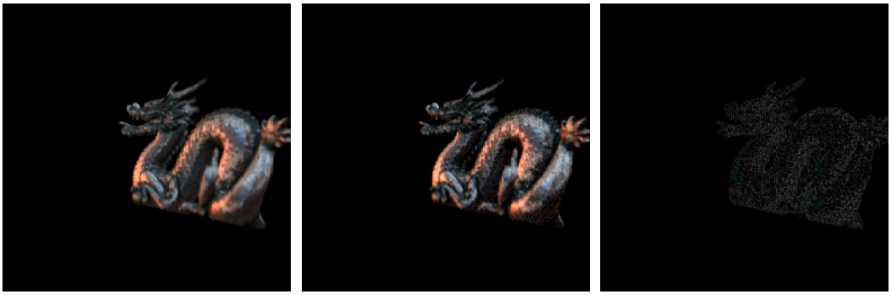

(a) (b) (c)

Fig. 7. (a) Rendered view of virtual camera at Position-E in *Dragon* light field, (b) watermarked view at Position-E in *Dragon* light field, (c) The difference between (a) and (b), multiplied by 2 for the purpose of better display

4.2 Robustness Test for Rendering

In the robustness tests, the extraction scheme is applied for different imagery views based upon the virtual camera position and orientation. In *Dragon* light field, Position-A in Fig. 5 is taken as a reference, in order to describe translation and rotation in the results. The camera position of Position-A is [0 0 2] and normal direction is [0 0 -1]. Six cases are considered in the simulations as shown in Table 1. These cases cover the translation, rotation and scaling type of processing for the rendered views. The robustness tests of the six cases evaluated with average BER of 20 random watermark sequences are shown in Table 2.

From Table 2 we can see that the proposed watermarking scheme performs very well on different imagery views. For Case I, II, IV and V, BER values are all lower than 1%. Especially, the BER value is zero in Case I and II. For Case III and VI, the energy of watermark reduces seriously due to shrinking of the rendered view. However, BER=3.3% is a satisfactory value for Case III where the shrink is very severe. So the proposed watermarking scheme is successful.

Table 1. Six cases for the creation of rendered views in the *Dragon* light field

	Translation on uv-plane?	Translation on z-axis?	Rotation?	Position	Direction	Label
Case I	No	No	No	[0 0 2]	[0 0 -1]	A
Case II	Yes	No	No	[0.5 0 2]	[0 0 -1]	B
Case III	Yes	Yes	No	[0.5 0 3]	[0 0 -1]	C
Case IV	Yes	No	Yes	[0 2 2]	[0 -1 -1]	D
Case V	Yes	No	Yes	[1.5 0 2]	[-1 0 -1]	E
Case VI	Yes	Yes	Yes	[1.5 0 2.5]	[-1 0 -1]	F

Table 2. Robustness test for six cases of light field rendering

	Case I	Case II	Case III	Case IV	Case V	Case VI
BER	0	0	0.033	0.005	0.002	0.015

Table 3. Robustness against various attacks of the proposed watermarking scheme

	Case I	Case II	Case III	Case IV	Case V	Case VI
No attacks	0	0	0.033	0.005	0.002	0.015
Medina filter 2×2	0.015	0.014	0.170	0.14	0.094	0.137
Medina filter 3×3	0.007	0.029	0.192	0.117	0.121	0.155
Mean filter 2×2	0.013	0.013	0.165	0.142	0.090	0.125
Mean filter 3×3	0.034	0.061	0.229	0.180	0.143	0.208
Gaussian filter 3×3	0	0	0.056	0.023	0.014	0.033
Uniform noise ($\beta = 0.01$)	0	0	0.032	0.006	0.002	0.015
Uniform noise ($\beta = 0.02$)	0	0	0.033	0.005	0.002	0.015
Uniform noise ($\beta = 0.03$)	0	0	0.033	0.005	0.002	0.015
Salt & peppers noise (scale = 0.05)	0.056	0.068	0.211	0.125	0.134	0.162
Salt & peppers noise (scale = 0.08)	0.113	0.133	0.271	0.189	0.172	0.204
Gaussian noise (var. = 0.01)	0.011	0.009	0.138	0.076	0.052	0.089
Gaussian noise (var. = 0.02)	0.026	0.040	0.188	0.136	0.101	0.152
Gaussian noise (var. = 0.04)	0.083	0.099	0.278	0.213	0.183	0.227
JPEG 80	0	0	0.041	0.007	0.003	0.019
JPEG 60	0	0	0.060	0.029	0.016	0.043
JPEG 40	0.001	0.007	0.107	0.052	0.033	0.067
JPEG 30	0.006	0.012	0.125	0.070	0.057	0.087
Cropping 5	0	0	0.033	0.005	0.003	0.026
Cropping 20	0	0	0.035	0.005	0.007	0.054
Cropping 30	0	0	0.044	0.005	0.019	0.076

4.3 Robustness Test against Other Attack

We also evaluate the robustness of the watermarking method against common signal processing attacks, because they could occur in the transmission chain of FTV. The performance of the proposed scheme under various common signal processing attacks is shown in Table 3. These attacks might include Median filtering with size 2×2 and 3×3, Mean filtering, Gaussian filtering with size 3×3, adding uniform noise, adding salt & peppers noise, JPEG compression and center cropping. The Gaussian filter matrix is

$$\begin{bmatrix} 0.0113 & 0.0838 & 0.0113 \\ 0.0838 & 0.6193 & 0.0838 \\ 0.0113 & 0.0838 & 0.0113 \end{bmatrix}$$

The attacked image with adding uniform noise is

$$I'(x, y) = I(x, y) \cdot (1 + \beta \cdot n(x, y))$$

where $I(x, y)$ is the pixel grayscale value of an input image at (x, y), β is a parameter that controls the strength of the additive noise, $n(x, y)$ is noise with uniform distribution, zero mean and unit variance, and $I'(x, y)$ is the pixel grayscale value of the attacked image.

From Table 3 we can see that the proposed watermarking scheme is not only resistant to common signal processing but also robust against combined signal processing attacks and light field rendering in six cases. Especially, the robustness against Gaussian filter, adding uniform noise, JPEG compression and cropping of the watermarking scheme performs very well.

5 Conclusion

In the emerging FTV system, there are more challenging requirements, compared to well-studied mono-view video watermarking. The ownership of the multi-view video should be proved not only on the original views of the multi-view video, but also on any virtual view generated for an arbitrary view. Apler Koz *et al.* propose a watermarking approach for the free-view video. However, it is only a one-bit watermarking scheme. In this paper, a multi-bit watermarking scheme for free-view video is proposed. The watermark message is embedded into every frames of multiple views using DS-CDMA embedding method. The watermarking extraction is carried out in the DCT domain of virtual frame generated for an arbitrary view with a correclation detector. Experimental results show that the watermark for FTV can be detected from virtual views generated for an arbitrary view. Moreover, the proposed scheme is resistant to common signal processing including lowpass filtering, adding noise, JPEG compression and cropping. More exhilaratingly, the watermarking scheme is robust against combined signal processing attacks and light field rendering operation.

Acknowledgments. This work was supported in part by 973 Program (2011CB302204), the National Science Foundation of China for Distinguished Young Scholars (61025013), Sino-Singapore JRP (2010DFA11010), National NSF of China (61073159) and Fundamental Research Funds for the Central Universities (2009JBZ006, 2011YJS292).

References

1. Barni, M., Bartolini, F., Checcacci, N.: Watermarking of MPEG-4 Video Objects. J. IEEE Trans. Multimedia 7(1), 23–32 (2005)
2. Hsu, C., Wu, J.: Digital Watermarking for Video. In: Proc. IEEE Int. Conf. Digital Signal Processing, vol. 1, pp. 217–220. IEEE Press, Santorini (1997)

3. Tian, H., Zhao, Y., Ni, R., Cao, G.: Geometrically robust image watermarking by sector-shaped partitioning of geometric-invariant regions. J. Optics Express 17(24), 21819–21836 (2009)
4. Zhang, C., Chen, T.: A Survey on Image Based Rendering-representation, Sampling and Compression. J. EURASIP Signal Process.: Image Commun. 19(1), 1–28 (2004)
5. Koz, A., Cigla, C., Alatan, A.A.: Watermarking of Free-view Video. J. IEEE Tran. Image Proc. 19(7), 1785–1797 (2010)
6. Koz, A., Cigla, C., Alatan, A.A.: Free-View Watermarking for Free-View Television. In: 2006 IEEE International Conference on Image Processing, Atlanta, pp. 1405–1408 (2006)
7. Cox, I.J., Kilian, J., Leighton, F.T., Yang, Y., Shamoon, T.: Secure spread spectrum watermarking for multimedia. J. IEEE Trans. Image Process. 6(12), 1673–1687 (1997)
8. Levoy, M., Hanrahan, P.: Light field rendering. In: Proc. ACM Siggraph 1996, New Orleans, pp. 31–42 (1996)
9. Tanimoto, M.: FTV (Free-viewpoint Television) creating ray-based image engineering. In: Proc. IEEE Int. Conf. Image Proce., Genova, vol. 2, pp. 25–28 (2005)
10. Dong, P., Brankov, J.G., Galatsanos, N.P., Yang, Y., Davoine, F.: Digital Watermarking Robust to Geometric Distortions. J. IEEE Trans. Image Process. 14(12), 2140–2150 (2005)
11. The Stanford Light Field Archive, `http://graphics.stanford.edu/software/lightpack/lifs.html`

Three Novel Algorithms for Hiding Data in PDF Files Based on Incremental Updates

Hongmei Liu, Lei Li, Jian Li, and Jiwu Huang

School of Information Science and Technology,
Sun Yat-sen University, Guangzhou, China, 510006
lilei27@student.sysu.edu.cn

Abstract. PDF is a widely used document format. By studying the structure of PDF file, we notice that incremental updates method used by PDF file can be used to embed information for covert communication. So in this paper, we present three novel data hiding algorithms based on incremental updates which can provide large enough embedding capacity without any change of file display. These algorithms embed information by different covert channels and the logistic chaotic map is used to enhance the security of the embedded data. The experimental results show that the algorithms are also robust to PDF files annotating, marking and interactive forms editing.

Keywords: Incremental Updates, Security, Embedding Capacity, Robust.

1 Introduction

PDF (Portable Document Format) is a file format which relies on the same imaging model as the PostScript page description language to describe text and graphics in a device-independent and resolution-independent manner. It enables users to exchange and view electronic documents easily and reliably, independently of the environment in which they were created [1]. Thus, it has become a widely used electronic document format. It is advantageous to use PDF files as cover media to hide data for secret message transmission and other uses. Although PDF files are popular now, there are yet not many researches on data hiding in them. So it is useful to propose some novel data hiding algorithms in PDF files for various application purposes.

The existing PDF-based algorithms of information hiding can be summarized in two categories. The algorithms in first category hide information by varying the line, word, character spacing or other certain display attributes slightly[2,3,4,5,6,7]. Nevertheless, they have the obvious defects that the effect of page display is disturbed and that information security is relatively low . The algorithms in second category embed information by adding or changing the content of data streams, for example, software wbStego4.3 [8] writes 2 bytes information between two indirect objects, the algorithm in [9] adds irrelevant indirect objects to the body of a file, and the algorithm in [10] embeds information

Y.Q. Shi, H.J. Kim, and F. Perez-Gonzalez (Eds.): IWDW 2011, LNCS 7128, pp. 167–180, 2012.

by changing the order of the dictionary object's entries. These algorithms have done well in visual imperceptibility, but have disadvantages in guaranteeing large capacity, high security and robustness to some degree. Thus, novel algorithm is needed to improve the overall performance.

By studying the structure of PDF file, we notice that the incremental updates method used by PDF file can be used to hide data. We proposed three data hiding algorithms based on incremental updates. All of these algorithms ensure the visual imperceptibility of display, security of covert information and large enough embedding capacity as well. Furthermore, it has strong robustness to both common reading operations (including annotating, marking, etc) and interactive forms editing.

The rest of this paper is organized as follows. Because the algorithms heavily depend on the structure of PDF files and incremental updates, we introduce them in section 2 and 3 respectively. The three proposed algorithms are presented in section 4. Section 5 demonstrates the experimental results and analysis. Conclusions and future work are given in section 6.

2 The Structure of PDF Files

PDF file structure can be classified into file structure(physical structure) and document structure(logic structure): the file structure includes the header, the body which contains a lot of objects, the cross-reference table containing information about the indirect objects in the file and the trailer. Fig.1 shows the Initial file structure of PDF file. It determines how the objects are stored in a PDF file. A PDF document can be regarded as a hierarchy of objects contained in the body section of a PDF file[1]. The document structure of PDF file is organized in the shape of an object tree topped by Catalog as root and five subtrees named Page tree, Outline hierarchy, Article thread, Named destinations

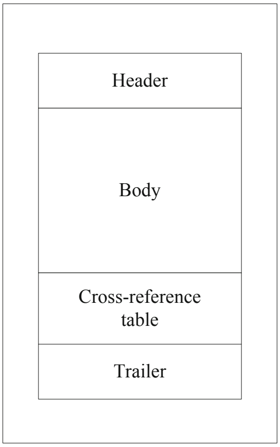


Fig. 1. Initial file structure of a PDF file [1]

and Interactive form. Fig.2 shows the document structure of a PDF file. The tree structure allows PDF consumer applications, using only limited memory, to quickly open a document containing thousands of pages [10]. On the page tree, there are many page objects, which involve the file contents, typeface applied, format, pictures, and so on.

An object is the basic element in PDF files. PDF supports eight basic types of objects: Boolean Object, Numeric Object, String Object, Name Object, Array Object, Dictionary Object, Stream Object and Null Object. Objects may be labeled so that they can be referred to by other objects. A labeled object is called an indirect object [1]. In this paper, the indirect object is an important carrier to hide information.

The content stream belongs to Page tree and contains almost all information about PDF text contents and display attributes. Each page's contents will be cut into blocks and each block's contents are saved in a stream object named Contents

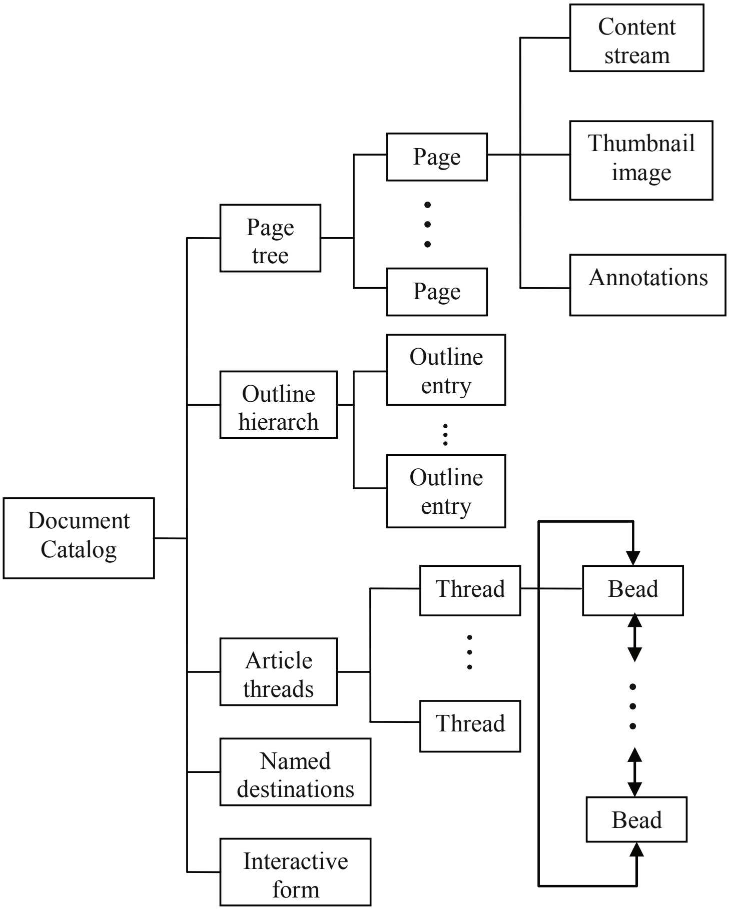


Fig. 2. Document structure of a PDF document [1]

Object. The stream of Contents Object is often encoded. The filter or filters for the stream are specified by the Filter entry in the streams dictionary. By using the corresponding filter to uncompress the stream, complete Contents Object will be obtained. It includes text object and text state. The text object describes the text contents and the text state is a collection of page display attributes.

3 Incremental Updates

The contents of PDF file can be updated incrementally without rewriting the entire file. Changes are appended to the end of the file, leaving its original contents intact. This is a particular update way for PDF files. The main advantage to updating a file in this way is that small changes to a large document can be saved quickly. In an incremental update, any new or changed objects are appended to the file, which constitute the updated body at the end of the file, a cross-reference section and a new trailer are appended followed. The resulting file has the structure shown in Figure 3. The cross-reference section contains the information of the objects in the updated body[1]. For more information, the readers can refer to [1].

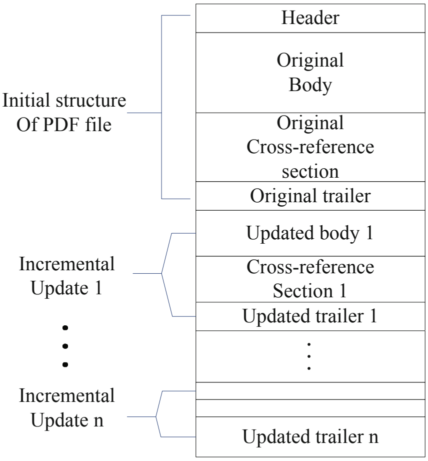


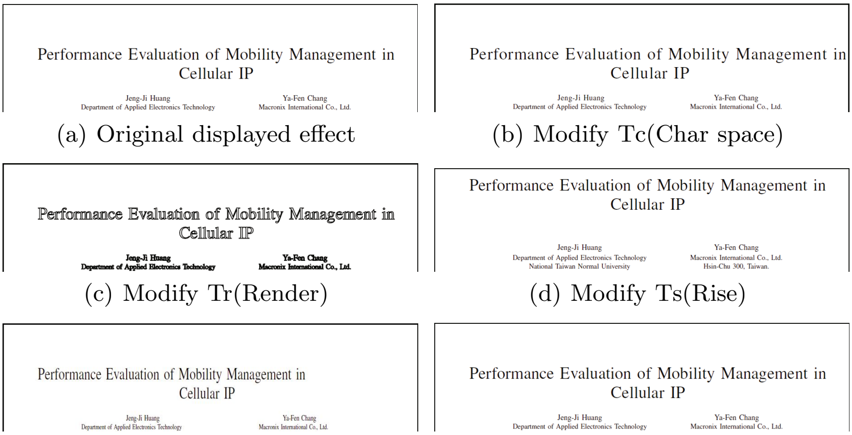
Fig. 3. File structure of an incremental updated PDF file [1]

4 The Proposed Algorithms

4.1 Algorithm 1: A Compensated Version of Modifying Display Attributes Algorithms

As shown in Fig.2, in the document structure tree, there is a subtree named Page tree. Under Page tree, there are Contents Objects which involve the text contents and text states. The logical relationship from root-Catalog to Content stream is shown in Fig.2. We can find all of the Contents Objects of a PDF file by this structure.

Text state in Contents Object indicates the attributes of text display. Every attribute has a operator key word to mark it, such as Char Space: Tc, Word Space: Tw, Scale: Tz, Leading: TL, Font size: Tf, Render: Tr, Rise: Ts etc. The values of these attributes in the content stream can be modified to hide information. In the literature, there are some algorithms which hide data in PDF files by altering the line, word, character spacing or other certain display attributes such as [2,3,4,5,6,7]. But these algorithms affect the display of the PDF file. The effects of modifying text attributes are shown in Fig.4.



Performance Evaluation of Mobility Management in Cellular IP

Jeng-Ji Huang
Department of Applied Electronics Technology

Ya-Fen Chang
Macronix International Co., Ltd.

(a) Original displayed effect

Performance Evaluation of Mobility Management in Cellular IP

Jeng-Ji Huang
Department of Applied Electronics Technology

Ya-Fen Chang
Macronix International Co., Ltd.

(b) Modify Tc(Char space)

Performance Evaluation of Mobility Management in Cellular IP

Jeng-Ji Huang
Department of Applied Electronics Technology

Ya-Fen Chang
Macronix International Co., Ltd.

(c) Modify Tr(Render)

Performance Evaluation of Mobility Management in Cellular IP

Jeng-Ji Huang
Department of Applied Electronics Technology
National Taiwan Normal University

Ya-Fen Chang
Macronix International Co., Ltd.
Hsin-Chu 300, Taiwan.

(d) Modify Ts(Rise)

Performance Evaluation of Mobility Management in Cellular IP

Jeng-Ji Huang
Department of Applied Electronics Technology

Ya-Fen Chang
Macronix International Co., Ltd.

(e) Modify Tz(Scale)

Performance Evaluation of Mobility Management in Cellular IP

Jeng-Ji Huang
Department of Applied Electronics Technology

Ya-Fen Chang
Macronix International Co., Ltd.

(f) Compensated display effect

Fig. 4. The effects of modifying some text attributes to display and recovering files display by incremental update ((a) is the source file display; (b)-(e) are the display after varying a text attribute; (f) is the display by making an incremental update to each file from (b) to (e))

In this paper, we can compensate the side effect of the above data hiding algorithms using incremental updates of PDF files. The principle is that after altering the text states of contents objects to embed information, the original contents objects are written in updated body. Thus, the display effect can be removed because when the PDF file is displayed, the latest updated body will be used instead. The effect is shown in Fig.4 (f). Encrypt method can enhance the security of hiding data, so we will encrypt the embedded data by certain method (for example, by logistic chaotic map).

Fig.5 shows the overview of the embedding algorithm, and the following is the text description of the embedding steps:

Step1: Read the PDF file stream;

Step2: Find out all of the contents objects according to the page subtree;
Step3: Decode the stream contents according to the corresponding filter, then find out the text state entries and estimate the capacity;

Step4: Read embedded binary data, then scramble it by logistic chaotic map.

Step5: Modify the lowest order's parity of the text state value based on the bit in embedded data;

Step6: Record the entries and contents objects which have been modified as a part of stego-key. Rewrite the original contents objects by incremental update;

Step7: Output the embedded PDF file and stego-key.

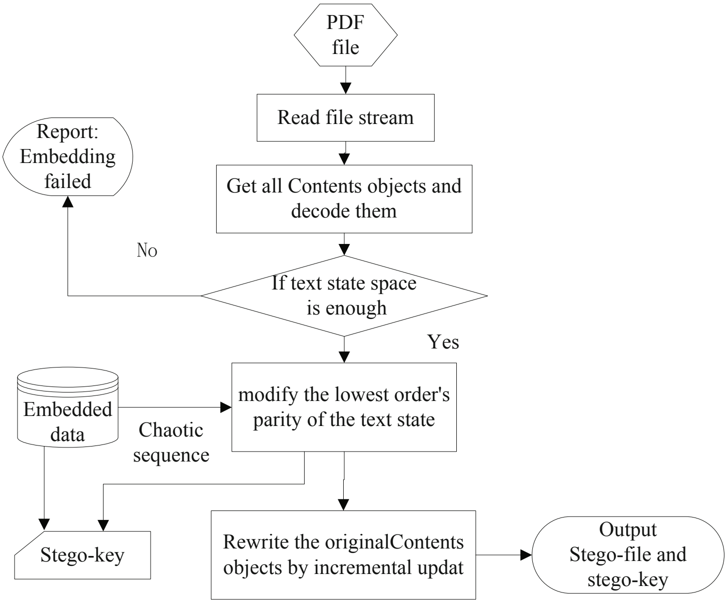

PDF file
Read file stream
Report: Embedding failed
Get all Contents objects and decode them
No
If text state space is enough
Yes
Embedded data
modify the lowest order's parity of the text state
Chaotic sequence
Stego-key
Rewrite the originalContents objects by incremental updat
Output Stego-file and stego-key

Fig. 5. An overview of data embedding steps of Algorithm 1

4.2 Algorithms Based on Updated Body and Cross-Reference Section

Algorithm 2 Based on Updated Body. Fig.3 shows the structure of an incrementally updated PDF file, we can see that each incremental update will insert an updated body following original file stream. The new body includes the objects that need to be modified, added or deleted. It can be regarded as the most important part of an incremental update. And this update way and structure provide us a theoretically infinite space to embed information.

In the updated body, the actual covert information carrier is indirect objects. Numeric Object, String Object, Array Object and Stream Object can be used as carriers. But considering the complexity of inserting objects, content security, capacity and other factors, we select stream object as the embedded carrier. The steps of the embedding algorithm is as follows:

Step1: Read the PDF file stream;

Step2: Read embedded binary string, then scramble it by logistic chaotic map. Divide the scrambled string into segments(For example: we can choose 32 bits as the length of each segment to hide 128 bits information. Thus we need to add 4 new objects);

Step3: Find the max object number of original file in the trailer, in order to determine the number of new objects(each new object has a number);

Step4: Create new stream object whose content is the scrambled data in each segment. Record the new object number as a part of stego-key;

Step5: If all of the information has been embedded, write the new cross-reference and new trailer; otherwise go to Step4;

Step6: Output the PDF file and stego-key.

Algorithm 3 Based on New Cross-Reference Section. The cross-reference section contains a one-line entry for each indirect object, specifying the location of that object within the body of file. A 10-digit number in each entry, padded with leading zeros if necessary, is the byte offset. It gives the number of bytes from the beginning of the file to the beginning of the object. When make an incremental update to a PDF file, a cross-reference section will be added following the updated body to specifying the information of updated object. In this algorithm , we select the new cross-reference section as covert information carrier. We can embed information by controlling the 10-digit byte offset in cross-reference section's entry. Use the difference of adjacent entries' offset to represent the covert information. The following is the text description of the embedding algorithm.

Step1: Read the PDF file stream;

Step2: Read embedded binary string, then scramble it by logistic chaotic map. Divide the scrambled string into segments(considering the size of decimal number transformed from binary segment, we choose 6 bits as the length of each segment);

Step3: Transform the scrambled binary segment to decimal number, and add 100 to each number(make sure the decimal number has a suit size to create a new object). Then get an ordered decimal number sequence;

Step4: Insert the new stream objects with the length based on decimal number sequence in order. The stream contents is random and the number of them is n+1(n is the number of binary segment);

Step5: Insert the new cross-reference section and new trailer if the information has been embedded. Record the characteristic value of the new cross-reference section(such as offset) as the other part of stego-key;

Step6: Output the PDF file and stego-key.
An example is as follows to explain the embedding procedure of algorithm 3:

We suppose a 32 bits embedded binary string which has been scrambled: 01010110110011011111000001110001. We divide the string into segments with the length of 6, padded with following zeros if necessary:{010101,101100,110111, 110000,011100,010000}. Then we transform each segment to decimal number, and add 100 to each one. Thus, we get an ordered decimal number sequence:{121, 144,155,148,128,116}. Next, we need to insert seven stream objects in the updated body and the lengths of first six one are: 121, 144, 155, 148, 128 and 116. The last object's length is random(such as 130). We suppose the first new object's offset is 300000, then the new cross-reference section will be:

```
0000300000 00000 n
0000300121 00000 n
0000300265 00000 n
0000300420 00000 n
0000300568 00000 n
0000300696 00000 n
0000300812 00000 n
```

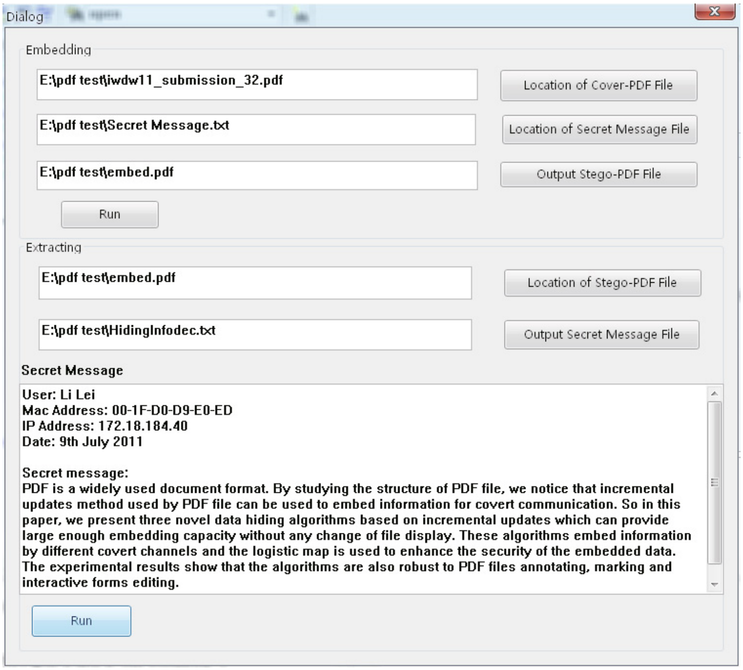
5 The Experimental Results and Analysis

In our experiment, the cover PDF files are downloaded from Internet randomly. These files use different PDF versions from 1.2 to 1.6 but have the same canonical file structure. That is to say, they all use cross-reference table instead of cross-reference stream to save the information of indirect objects. We use all of the three algorithms proposed above to embed covert data. As the three algorithms have the similar experimental results, we just choose one of them randomly to show the algorithms' performance except data embedding capacity.

5.1 Data Embedding Capacity and Perceptual Transparency

The data embedding capacity of the first data hiding algorithm is limited by the number and size of usable Contents Objects. And the embedding capacity of the second and third algorithm are unlimited so that we can embed a large amount of secret data. In order to implement the proposed methods for covert communication, we designed a user interface written in the language of C++. It uses the algorithm 2 and 3 to embed and extract secret message. The experimental result is shown in Fig.6.

The comparison of displays before and after embedding data is shown in Fig.7. From the effects chart, we can't see any change. The algorithm 1 can recover the


Dialog
Embedding
Location of Cover-PDF File
Location of Secret Message File
Output Stego-PDF File
Run
Extracting
Location of Stego-PDF File
Output Secret Message File
Secret Message
Run

E:\pdf test\iwdw11_submission_32.pdf

E:\pdf test\Secret Message.txt

E:\pdf test\embed.pdf

E:\pdf test\embed.pdf

E:\pdf test\HidingInfodec.txt

User: Li Lei
Mac Address: 00-1F-D0-D9-E0-ED
IP Address: 172.18.184.40
Date: 9th July 2011

Secret message:
PDF is a widely used document format. By studying the structure of PDF file, we notice that incremental updates method used by PDF file can be used to embed information for covert communication. So in this paper, we present three novel data hiding algorithms based on incremental updates which can provide large enough embedding capacity without any change of file display. These algorithms embed information by different covert channels and the logistic map is used to enhance the security of the embedded data. The experimental results show that the algorithms are also robust to PDF files annotating, marking and interactive forms editing.

Fig. 6. User interface of embedding and extracting covert data(using algorithm 2 and 3)

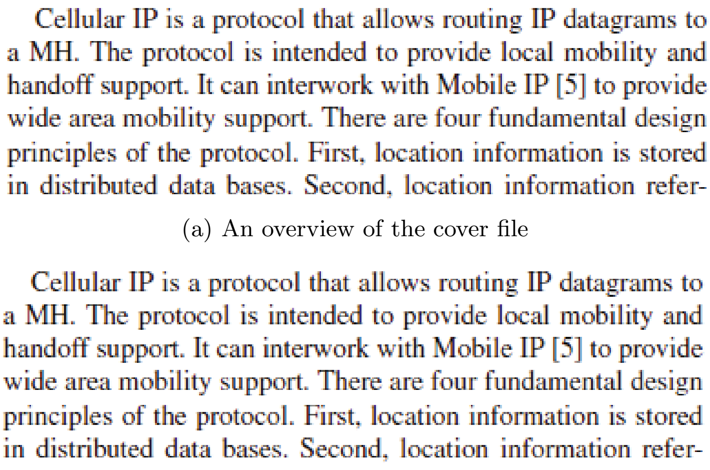
display by rewriting original Contents objects. The new objects embedded by the algorithm 2 and 3 have nothing to do with PDF file display. So our hiding methods are invisible.

5.2 The Robustness to Reading and Editing Operations

Generally speaking, existing PDF reading softwares such as Foxit reader, Adobe reader, CAJ reader, etc., fail to edit the content and format of the PDF files, which is greatly unlike the Microsoft Office documents. On the other hand, annotating and marking as functions provided by these softwares are prevailing in our daily use. In order to examine whether our proposed algorithms have good robustness to annotating and marking operations in various forms, we add comments and marks to the files embedded with hidden data.

The PDF file shown in Fig.8 has been embedded covert data. After adding the various comments and marks to this file, we try to extract the covert information from it. And the experiment result shows that the accuracy of extracted data is 100%.

Interactive forms is a special type in PDF. These files are convenient to collect users' information. They permit users to edit some information in some specified location in PDF files. Users can choose, write, modify or delete the content in the form. We test whether the interactive forms editing operations will damage the embedded information.



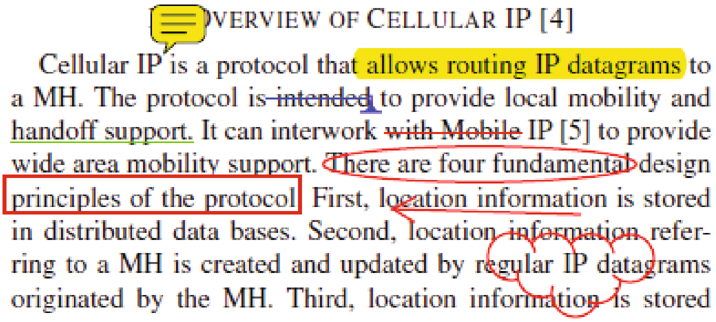
Cellular IP is a protocol that allows routing IP datagrams to a MH. The protocol is intended to provide local mobility and handoff support. It can interwork with Mobile IP [5] to provide wide area mobility support. There are four fundamental design principles of the protocol. First, location information is stored in distributed data bases. Second, location information refer-

(a) An overview of the cover file

Cellular IP is a protocol that allows routing IP datagrams to a MH. The protocol is intended to provide local mobility and handoff support. It can interwork with Mobile IP [5] to provide wide area mobility support. There are four fundamental design principles of the protocol. First, location information is stored in distributed data bases. Second, location information refer-

(b) An overview of the stego file

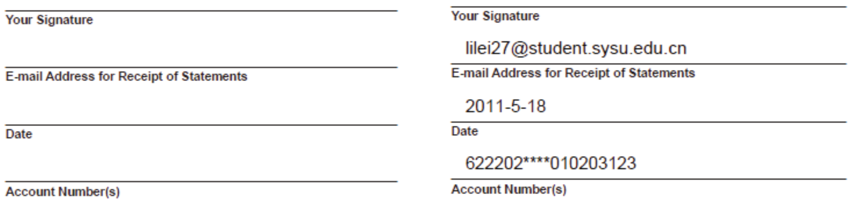
Fig. 7. The cover file and the stego file



VERVIEW OF CELLULAR IP [4]

Cellular IP is a protocol that allows routing IP datagrams to a MH. The protocol is ~~intended~~ to provide local mobility and handoff support. It can interwork ~~with Mobile~~ IP [5] to provide wide area mobility support. There are four fundamental design principles of the protocol. First, location information is stored in distributed data bases. Second, location information referring to a MH is created and updated by regular IP datagrams originated by the MH. Third, location information is stored

Fig. 8. Apply Adobe Acrobat 9 Pro to annotate and mark the embedded PDF file in various ways



Your Signature

E-mail Address for Receipt of Statements

Date

Account Number(s)

(a) Original interactive forms file with embedded data

Your Signature

lilei27@student.sysu.edu.cn

E-mail Address for Receipt of Statements

2011-5-18

Date

622202****010203123

Account Number(s)

(b) Writing some contents in the interactive forms file

Fig. 9. Make editing operations to an interactive forms file with embedded data

Table 1. The increase in size of carrier-object by algorithm 1

File	Original size	Page number	Embedded size	Increase rate
1	149KB	4	153KB	2.7%
2	237KB	4	245KB	3.4%
3	271KB	4	272KB	0.4%
4	298KB	4	306KB	2.7%
5	303KB	6	304KB	0.3%
6	349KB	7	350KB	0.3%
7	413KB	2	415KB	0.5%
8	543KB	5	544KB	0.2%
9	663KB	4	664KB	0.15%
10	801KB	10	803KB	0.2%

We download an interactive forms file from Internet then embedded covert data in it by our method. In Fig.9, (a) is the stego file without any editing and (b) is the file been written some contents to (a). We try to extract the covert information from (b), and the experiment result shows that the accuracy of extracting test is 100%.

From experimental results, we can see that the proposed algorithms are robust to reading and editing operations. The analysis is as follows:

1.Reading: Annotating and marking operations will add some new objects in the original PDF file body. the new objects are appended behind original objects, and the cross-reference table and trailer will be updated. This way of file update is different from incremental updates. But these operations won't influence both the original objects and the new objects generated by incremental updates, even if the information embedded in the difference of offset. Therefore, the proposed algorithm has good robustness to such editing operations.

2.Editing: The editing operations will insert some new objects or add some new contents in original objects which won't influence our embedded data. Therefore, the proposed algorithm has good robustness to such editing operations.

5.3 Increase in the Size of Cover File

Our algorithms are based on incremental updates. Each algorithm will add some stream content in original file when embedding information. So the embedded file will have a larger size than original file inevitably. But if the increased percentage is too big, the performance of algorithm will be affected. In the following experiment, we want to test the increase percentage of carrier-object when embed information in PDF files by our algorithms.

First, we test the algorithm 1's performance because the size increase of algorithm 1 is based on carrier file. We select 10 PDF files which have the standard file structure. The files have 2 to 10 pages and with size between 149KB to 801KB. We use algorithm 1 to embed 128 bits information into these ten files and table 1 shows the experiment results.

Analyzing the embedding process, we can learn that basically, one contents object is enough to embed 128 bits information. Rewriting a Contents object by incremental update will increase the size of the original file by 1 to 8 KB (depending on the size of the original Contents object which is used to embed information). For a file with a size of 200 KB, the rate of size increase is below 5%. The real experimental result shows that the average rate of files' size increase is around 1%, which is acceptable in our experiments. Algorithm 1 selects the first Contents object which is viable to embed information. Although we just embed 128 bits, a large object which is 8KB may be selected and rewriting this object back to the PDF file will increase the size by 8KB. We can improve it by choosing Contents object based on the embedded information size instead of the first object.

The increase of the size caused by algorithm 2 is irrelevant to the original files. Suppose we divide the embedded information into four segments and hide them into four indirect objects. Each new object will be around 100 bytes. The size of the added new cross reference section (each entry is twenty bytes long) and new trailer is less than 200 bytes. The embedding operation of algorithm 2 will add no more than 1KB to original PDF file. For a file with a size of 200 KB, the rate of size increase can be keep around 0.5%.

The increase of the size caused by algorithm 3 is also irrelevant to the original files. It is depending on the 128 bits binary string. If we choose 6 bits as the length of segment, we will get 22 segments. Transform each binary segment into decimal number and add 100 to each one. We can get 22 decimal numbers between 100 to 163. The maximal size increase will be around 4 to 5 KB. For a file with size of 200 KB, the rate of files' size increase keep within 2.5%. And the average increase percentage is 1.5%.

5.4 The Performance Comparison

In table 2, performance comparison among the methods proposed in this paper and other three existing ones is shown from four main aspects. In spite of

Table 2. The result of the performance comparison

Performance	Incremental updated methods	wbStego4.3[8]	The method based on varying display attributes[2,3,4,5,6,7]	The method based on changing entries' order[10]
Perceptual transparency	No changed	No changed	Slightly changed	No changed
Embedding capacity	Large enough	Small	Based on file	Based on file
Security	Relatively high	low	Relatively low	High
Robustness	Strong	Relatively Strong	Relatively Strong	Medium

increasing the size of document, which may affect its performance slightly such as unable to prevent the general statistical attack effectively, our algorithms have better performance than the others in main aspects. Our algorithms not only guarantee the original display effects of the cover file, but also realize the big enough embedded capacity and strong robustness to many editing operations.

Incremental updates is widely used in PDF files like saving the modified PDF file, updating some PDF file information such as title, author, theme and key words. Although the size of cover file will increase by hiding data based on incremental updates, it's hard to distinguish the embedded information from the useful update information. Therefore, the security of embedded information is high enough.

6 Conclusion and Future Work

This paper proposes three novel PDF files data hiding algorithms based on incremental updates, which is a particular method used by PDF file. The covert channel lies in file's physical and logical structure. The logistic chaotic map is used to enhance the security of the embedded data. The proposed algorithms are invisible, robust and high payloaded. The experimental results and analysis support them.

Different versions of PDF files are being used at present. Some higher versions of PDF files have used cross-reference streams to store the information of indirect objects. How to advance our algorithms to be compatible with different PDF versions is our future work.

References

1. Adobe Systems Incorporated. PDF Reference, 5th edn., version 1.6 (2006), `http://www.adobe.com/devnet/pdf/pdfs/PDFReference16.pdf`
2. Low, S.H., Maxemchuk, N.F.: Performance comparison of two text marking methods. IEEE Journal on Selected Areas in Communications 16(4), 561–572 (1998)
3. Brassil, J.T., et al.: Electronic marking and identification techniques to discourage document copying. IEEE Journal on Selected Areas in Communications 13(8), 1495–1504 (1995)
4. Zhong, S., Chen, T.: Information Steganography Algorithm Based on PDF Documents. Computer Engineering 32(3), 161–163 (2006)
5. Low, S.H., et al.: Document marking an identification using both line and word shifing. In: Proceedings INFOCOM 1995, Boston, MA, pp. 853–860 (April 1995)
6. Maxemchuk, N.F., Low, S.H.: Marking text documents. In: Proceedings, International Conference Image Processing, Boston, Santa Barbara, CA, October 1997, pp. 13–17 (1997)
7. Franz, E., Pfitzmann, A.: Steganography Secure against Cover-Stego-Attack. In: Pfitzmann, A. (ed.) IH 1999. LNCS, vol. 1768, pp. 29–46. Springer, Heidelberg (2000)

8. wbStego Studio. The steganography tool wbStego4 (2007), http://www.wbailer.com/wbstego
9. Liu, Y., Sun, X., Luo, G.: A Novel Information Hiding Algorithm Based on Structure of PDF Document. Computer Engineering 32(17), 230–232 (2006)
10. Liu, X., Zhang, Q., Tang, C., Zhao, J., Liu, J.: A Steganographic Algorithm for Hiding Data in PDF Files Based on Equivalent Transformation. In: 2008 International Symposiums on Information Processing (ISIP), May 23-25, pp. 417–421 (2008)

Use of "Emergable Watermarks" as Copy Indicators for Securing Video Content

Takaaki Yamada and Yoshiyasu Takahashi

Hitachi, Ltd., Yokohama Research Laboratory, Japan
{takaaki.yamada.tr,yoshiyasu.takahashi.gq}@hitachi.com

Abstract. We propose using emergable watermarks as copy indicators that become visible in video content that has been copied under designated conditions. A watermark pattern is designed that is sensitive to image processing specific to illegal copying. This is done on the basis of the aliasing effect, which causes image distortion when the re-sampling frequency is low. Watermarks using such patterns are embedded in the original video. They become visible only if the copy had been illegally generated by re-encoding with designated scaling. They show, for example, copy indicators, warning the viewer that the video had been illegally copied, and distort the image quality. Testing of a prototype implementation demonstrated the effectiveness of this method. The peak-signal-to-noise-ratio in watermarked video was adjustable by more than 28 dB. Use of this method should help deter the illegal copying and/or distributing of copyrighted videos.

Keywords: emergable watermark, copy indicator, content security, video watermark, copyright protection, authentication.

1 Introduction

The growing availability of video content online is exacerbating the problem of copyright violation. The copyright of video content can easily be violated because the Internet facilitates copying and redistribution of content. Moreover, video-sharing services, which provide a huge amount of the video content worldwide, enable illegal copies to be distributed anonymously [1].

Typical video content is protected by using proven methods such as encryption and authentication. Methods for securing content such as digital rights management (DRM) are being developed [2, 3]. For instance, a method preventing the illegal re-recording of images displayed on a cinema screen has been developed [4]. It is based on the differences in sensory perception between the human eye and CCD devices and adds noise to the re-recorded images to that their quality is greatly degraded. However, this method requires installation of dedicated devices behind the screen. While complete copyright protection of video content is technically possible, strong security mechanisms can cause usability problems and increase costs.

To avoid such problems, the copying of video is sometimes technically allowed, such as copying video through an analog channel. In such a case, the illegalness

Y.Q. Shi, H.J. Kim, and F. Perez-Gonzalez (Eds.): IWDW 2011, LNCS 7128, pp. 181–195, 2012.

(whether copying is illegal or not) depends on the copyright condition of the video content. For instance, even if content is recorded in analog format under a specific allowed condition, a copyright violation may be flagged if it is redigitized and redistributed through the Internet as this goes beyond the allowed condition. Although conscientious users may not intentionally violate a copyright, they may do so if they are unaware of the copyright condition of the video content.

A common approach to deterring users from making illegal copies is to embed copyright information into the content in the form of digital watermarks. Various digital watermarking algorithms have been developed for securing content, including ones for copyright protection and for authentication [5–10]. Methods using robust digital watermarks are widely used for common copyright protection [9, 10]. The watermarks in video content are almost imperceptible, so they do not disturb the legal viewing of the content. Illegal copying can be prevented by detecting the watermarks and enforcing the corresponding copy function in accordance with the watermark detection result. However, digital watermarks themselves do not directly prevent malicious users from making illegal copies. Moreover, invisible watermarks do not directly work for conscientious (non-malicious) users because they are invisible. While visible watermarks such as a small logo in the corner of each image help deter conscientious users from making illegal copies, their visibility degrades the image quality.

To deal with these problems, we propose a new approach, the use of “emergable" watermarks. A pattern is designed that is sensitive to image processing specific to illegal copying. This is done on the basis of the aliasing effect, which causes image distortion when the re-sampling frequency is low. Watermarks using such patterns are embedded in video content as noise that does not degrade the legal viewing the video. The pattern pops up in the copied video content in the form of a copy indicator under designated conditions, such as re-encoding with scaling. We evaluated the effectiveness of our methods by implementing it in a prototype.

In section 2, related work is quickly reviewed to highlight the uniqueness of our method. Our method is described in section 3. Results obtained in its experimental evaluation are then presented in section 4. We summarize the main points, discuss a possible limitation, and mention future work in section 5.

2 Application Image

2.1 Deterrence to Illegal Video Copying

Consider a video archive that stores video files captured by surveillance cameras, as shown in Fig. 1. These files may contain scenes showing personal information, copyrighted materials, or trade secrets that need to be protected. When an authorized organization requests particular files, an official at the video archive packages the files and sends them to the recipient. The size of the data files might be reduced due to the limited capacity of the delivery media. If contents from the files were found by chance on a video-sharing site, serious problems such as privacy violation, copyright piracy, and/or contract infringement would arise. It would be difficult, however, to determine whether the information leak occurred in the office of the organization to which the file was sent or in the video archive office. Quickly identifying the source of an information leak is therefore an important issue.

One way to identify the site of an information leak is to use a digital watermarking technique that embeds the recipient's ID into the video file before it is sent [11]. A watermark embedding process can be incorporated into the video package creation process. If a problematic video is found by chance, the embedded ID can be used to identify the party who redistributed it.

One potential application of emergable watermarks is deterrence to illegal video copying as mentioned above. Various encoders and decoders are available to users, enabling them to easily convert video files from one compression format to another. Users can re-encode video images for sending samples by scaling them (e.g. by resizing them from VGA format to QVGA). In such cases, the recipient's ID embedded in the copied content can be made to pop up as a copy indicator. Such emerged watermarks would help deter users from copying and distributing copyrighted materials.

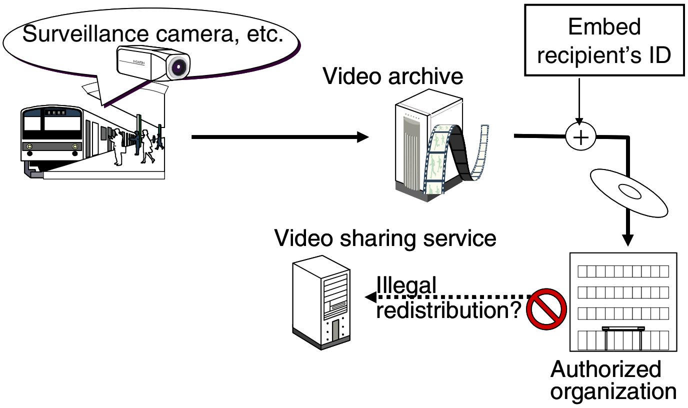


Fig. 1. Deterrence to illegal video copying using digital watermarks

2.2 Copy Protection in Printed Documents

There are many techniques for authenticating paper documents [12, 13]. Many, such as microprinting, using pearl ink, intaglio printing, hologramming, using luminescent ink, latent imaging, watermarking, and using a copy indicator, assume that dedicated devices or dedicated materials (such as paper and ink) are used.

Copy indicators, which are hidden in the original document, emerge in a copy of the document, as shown in Fig. 2. The emergence of these indicators in print documents is due to the difference in resolution between the printer used and scanners [13]. When the printer creates the original document, it adds hidden copy indicators. The scanners are equipped within common copy machines. Since the smallest printable dots are natively smaller than the smallest copyable dots, specific dots (intentionally selected in advance) are not copied due to this difference in resolutions. The dots that are copied result in the copy indicators in the copied document. Our approach applies this idea to video content.

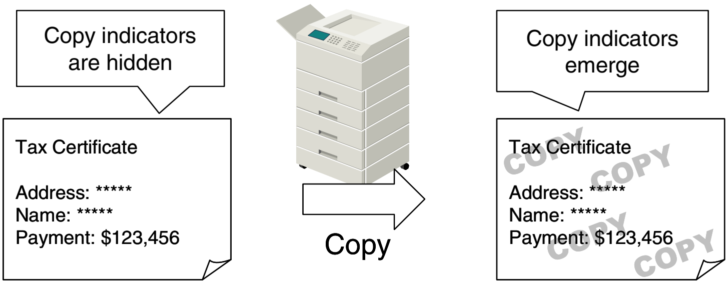


Fig. 2. Copy protection in printed document

3 Proposed Method

3.1 Research Approach

Although common copy function generates the same content as the original video content, so a difference in the copied video cannot be made on a digital basis, there are many use cases for copying with image processing. For instance, video files are often transcoded in order to shrink the data size. Moreover, thumbnail images are often made in order to overview multiple video files quickly. When such image processing is used to make illegal copies, the size of the original frame images is often scaled down. We can thus design special patterns in the original images that are sensitive to specific image processing. If we embed such patterns in video content, the patterns will expectedly change significantly due to the designated image processing. That is, we can use the pattern as a copy indicator for copyright protection, tracing, traitor tracking, content authentication, and so on. See Fig. 3.

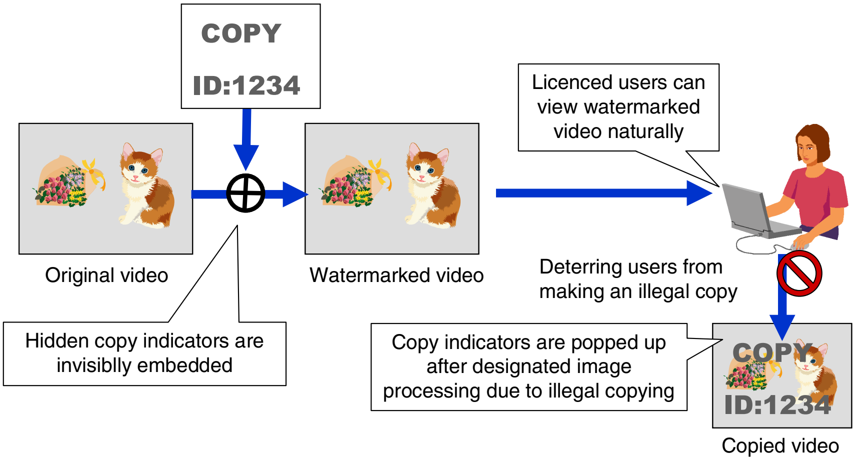

COPY
ID:1234
Licenced users can view watermarked video naturally
Original video
Watermarked video
Deterring users from making an illegal copy
Hidden copy indicators are invisiblly embedded
Copy indicators are popped up after designated image processing due to illegal copying
COPY
ID:1234
Copied video

Fig. 3. Emergable watermarking

Emergable watermarks as copy indicators are embedded in the original video. The watermarked video contains seemingly similar video content as that of the original video.

3.2 Principle

We superimpose a hidden image into the original image on the basis of the aliasing effect. The aliasing effect causes distortion or an artifact when the signal reconstructed from samples differs from the original continuous signal. In an image, the aliasing can be observed as a moiré pattern. It occurs when the original signal contains a signal above the Nyquist frequency. In the sampling theorem, the Nyquist frequency is given by.

$$\Delta x_{\text{Nyquist}} = \frac{1}{2\xi_{\max}} = \frac{\lambda_{\xi_{\max}}}{2}, \tag{1}$$

where $\Delta x_{\text{Nyquist}}$ is the Nyquist interval,

$\xi_{\max}$ is the maximum spatial frequency of the sampled object, and

$\lambda_{\xi_{\max}}$ is the spatial wavelength of the sine curve of frequency $\xi_{\max}$.

According to this equation, it is generally required to sample the target signal at a frequency at least twice that of the sampled object. If the sampling is done at frequency lower than the Nyquist frequency, the aliasing effect will occur and a moiré pattern will be evident in the images [16].

To embed a “message”, such as a copy indicator into video content, we use this aliasing effect. If the frame size is scaled down by image processing, the number of sampled points decreases. That is, the Nyquist frequency also decreases. Therefore, if we add a high frequency image (one with a high spatial frequency) onto each frame image in the video content, the modified frame images will expectedly be affected by "aliasing" if their size is scaled down. An image for a message to be superimposed is divided in small regions. We can control the spatial frequency of each region so that the hidden messages can be clearly seen under designated conditions. That is, both high- and low-frequency parts of the image are superimposed on each frame image as a form of not-annoying random noises. After designated scaling down of the frame image, the high-frequency parts cause aliasing while the low-frequency parts do not. If a hidden message is placed in the high-frequency parts, it remains in the copied image and emerge as a moiré pattern. Those regions having more impact on the copied image are called "residue regions" in that their pixels can better survive the designated image processing than those of other regions.

Note that watermarks may appear even if video content do not have any problem. Preventing such false alarm is a question which we want to keep beyond the scope of this present discussion.

3.3 System Architecture for Embedding Watermarks

The system architecture for embedding watermarks is shown in Fig. 4. The system has three kinds of inputs (original video file and message text, and message image) and one output (watermarked video file). The system has seven components; an input controller, a demultiplexer, a video decoder, a superimposer, a video encoder, a multiplexer, and an output controller.

- The input controller reads the message data (image or text). If text is input, it is converted into document image form. The input and converted image are both called the "cipher image". The copy indicators are created in the cipher image, which is then used in the superimposer. After processing the cipher image data, the input controller opens the original video file and sends it to the demultiplexer.
- The demultiplexer separates the multiplexed data into the video stream and the other streams (audio stream, subtitles stream, etc.).
- The video decoder decodes the video stream into uncompressed frame images frame by frame.
- The superimposer sequentially reads each frame image output from the video decoder. It creates a pattern using the cipher image data and adds it to the frame image.
- The video encoder re-encodes the superimposed frame images into a video stream.
- The multiplexer combines the video stream and the other streams (audio, subtitles, etc.) and places the results in the video stream buffer.
- The output controller writes that data to a watermarked file, the system output.

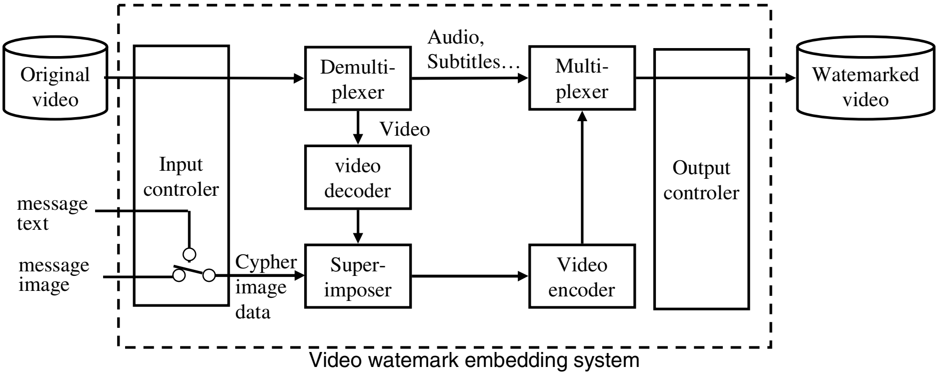


Fig. 4. System architecture for generating watermarked video

3.4 Process Flow for Superimposing

The key component of the watermark embedding system is the superimposer. The process flow for superimposing consists of six steps.

1. Read the cipher image and write it into an array for binary values: $\mathbf{b} = \{b_i \mid 0 \le i \le n\}$.
2. Allocate memories for writing the frame image buffer data that will be sent to the encoder. The size (width and height) of the output image is identical to that of the input images (the frame images in the original video and cipher image).
3. Repeat the following steps for each frame image until the input frames are exhausted.
4. Read the *j*-th frame image decoded from the original video file as a set of pixel values: $\mathbf{c} = \{c_i \mid 0 \le i \le n\}$.
5. Repeat the following steps, (a), (b), and (c), for each pixel in the original image.
   (a) Check if the selected *i*-th pixel (x, y) in the original image is in a residue region by using equation (2). Pixels in a residue region survive much better against the designated image processing, as described in section 3.2. To simplify the discussion, we suppose that the nearest-neighbor algorithm is used for scaling images. In addition, we suppose that the aspect ratio is maintained after scaling. A set of coordinates (u, v) indicates the corresponding location in the *j*-th frame image in the copied video content. The estimated scaling rate in the copying process is a_j. For instance, $a_j = 0.5$ indicates that a VGA image is resized to a QVGA image. The function $\mathrm{int}(z)$ gives the maximum integer below z. If both integers, u and v, are found to satisfy equation (2), then (x, y) is considered to be in the residue region.

$$\mathrm{int}\left(\frac{u}{a_j}\right) = x, \qquad \mathrm{int}\left(\frac{v}{a_j}\right) = y \tag{2}$$

   (b) Calculate the strength needed to modify the *i*-th pixel value in the original image. The strength is calculated using $f_j(\mathbf{b}, \mathbf{c}, i)$.
   (c) If the *i*-th pixel is in a residue region, set its value in the watermarked image to an estimated value by adding the corresponding pixel value and its strength in the original image.
6. After all pixels in the original image have been processed, send the watermarked image as a frame image to the encoder.

3.5 Prototype

We implemented our method in a prototype system. For step 5(b) above, it simply uses $f_j(\mathbf{b}, \mathbf{c}, i) = s \cdot b_i$, where b_i indicates the *i*-th binary value, and s is constant. In step 5(c), it sets the *i*-th pixel value in the original image to that of the watermarked image if the pixel is not in a residue region. Watermarks are embedded by modifying the luminance values of the pixels.

A graphical user interface (GUI) image of the prototype is shown in Fig. 5**.** A user can embed copy indicators into a video stream by selecting a function from the pull-down menu. A simulation function for resizing images is also available. The image in the figure is from a sample video in the standard video set and is called "airplane landing (No. 36)" [12]. The text message “test” was repeatedly embedded into the original image and is barely visible in the image immediately after watermark embedding (larger left window in Fig. 5). However, if the image is resized using a designated scaling factor (50% here; i.e., a scaling rate of 0.5), the message clearly emerges (smaller right window).

The data types for system input are shown in Table **1**. We implemented still image watermarking for additional system input as original data. The prototype system is able to read two kinds of input as original data (still images and video), and two kinds of input for the cipher message (still images and text). Therefore, it has four (two by two) input functions. If the user inputs a text message, the prototype converts it into image form before embedding it as a watermark.

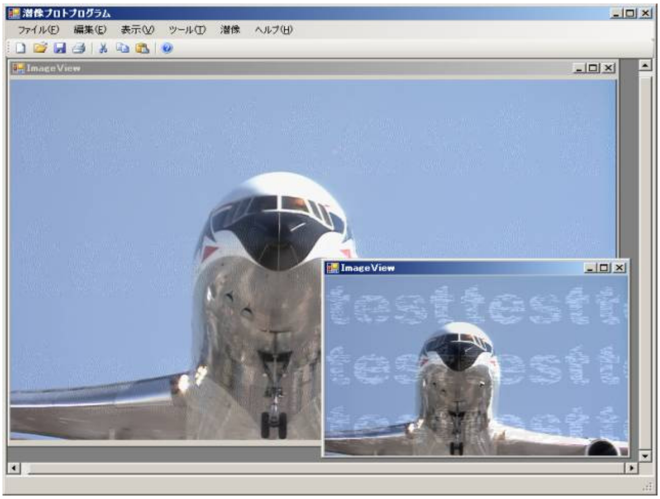


Fig. 5. Graphical user interface of the prototype and sample images showing embedded text message before (left) and after (right) scaling by a designated rate

Table 1. Data types for system input

Input as original data	Input as cipher message
Still image	Still image
Still image	Text
Video	Still image
Video	Text

4 Evaluation

We evaluated our prototype system with three experiments. The requirements for the experiments are summarized in Table 2. The sample video used in each one was a 15s SDTV-format (720 × 480 pixels) video, the "airplane landing **(No. 36)**" video [12].

Table 2. Requirements for three experiments

Exp.	Test type	Requirements
1	Re-encoding test	• Copy indicators are embedded into video content. • Watermarked video is then re-encoded with scaling to simulate illegal copying. • Embedded message should emerge in copied video.
2	Subjective tests for image quality and legibility	• Image quality of watermarked video should be maintained. • Hidden messages should be clearly visible in video copied under designated conditions.
3	Peak-signal-to noise ratio (PSNR) test	• Image quality of watermarked video should be maintained. • PSNR of watermarked video is quantitatively calculated for various scaling rates.

4.1 Re-encoding Test

The copy indicators should become visible after re-encoding a video file using a designated scaling rate. The experimental data flow is shown in Fig. 6. A cipher image converted from input text is embedded into the original video file. The resulting encoded watermarked-video simulates video that would be distributed in actual applications. The watermarked video file is re-encoded using a common MPEG4-based encoder with scaling from 720 × 480 to 360 × 240 (i.e., a scaling rate of 0.5) resulting in an encoded copied video. The parameters for the encoder were set to the default values except for the output size (width and height). This procedure simulates the illegal uploading of a video file. Next, the encoded copied video was checked to see whether the copy indicator was visible.

Fig. 7 shows frames with the same frame number (and thus at the same position on the time line) in the original video (a), watermarked video (b), and copied video (c). Because the watermarked video image had minute embedded dots, it is darker than the original video image. However, the minute dots were undetectable with the naked eye. The embedded message clearly emerged after re-encoding with designated scaling, as shown in Fig. 7 (c). Conscientious users would be expectedly deterred from distributing illegal copies by these copy indicators visible in the copied video content.

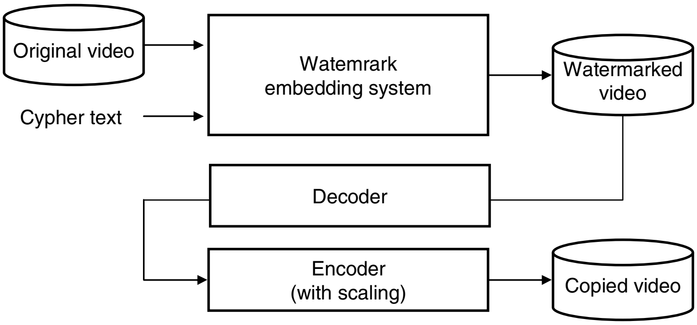


Fig. 6. Experimental data flow for re-encoding test

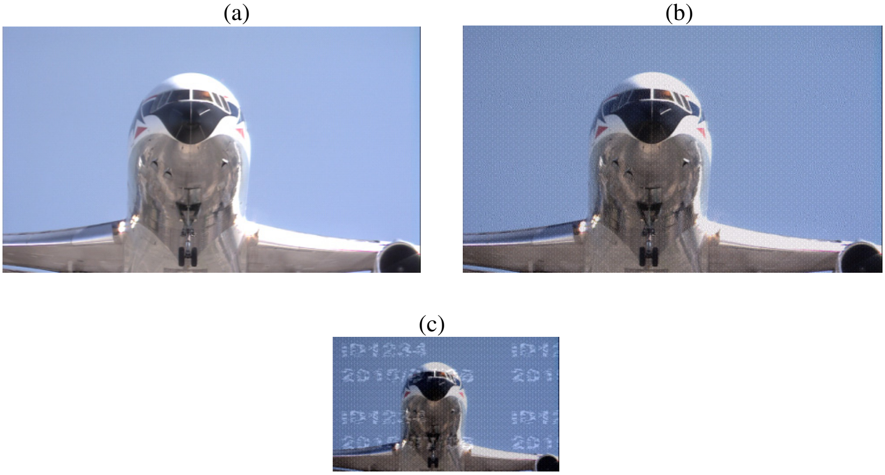


Fig. 7. Frames at the same position in time line from (a) original video, (b) watermarked video, and (c) copied video (which was re-encoded with scaling down)

4.2 Subjective Tests

The more, watermarked video is modified to embed strong watermarks, the more, its image quality is affected. However, the more, it is modified, the more, legible the copy indicators in copied video become. There is thus a trade-off relationship between image quality and indicator legibility. This is analogous to the trade-off between image quality and watermark robustness with common watermarking methods. Therefore, our method should be evaluated from two points of view; (1) the image quality of the watermarked video and (2) the legibility of copy indicators in copied video.

Image Quality of Watermarked Video

We subjectively evaluated the quality of images watermarked with our prototype using a procedure based on Recommendation ITU-R BT.500-7 [13]. That is, watermarked videos were displayed on a monitor by using a procedure based on the double-stimulus impairment scale method and evaluated by participants who rated the image quality by scoring the level of disturbance due to watermarks. The levels used are summarized in Table 3. Five participants evaluated the same sample as used in the re-encoding test.

The average scores are shown in Fig. 8. The horizontal axis represents the scaling rate used in making the watermarked video. The vertical axis represents the average subjective score. The image quality should be high for a scaling rate of 1.0 (original size) and low for the other scaling rates due to the aliasing effect. The experimental results meet this expectation. When the scaling rate was 1.0, the average score was the highest 4.4. Watermarks are designed to be most visible in copied video content when the scaling rate is 0.5. When the scaling rate was 0.5, the average score was the lowest 1.4. This score means that all participants felt annoyed or very annoyed with the copied video content. Interview to the participants revealed that they paid attention on reading messages emerged under designated conditions.

Table 3. Levels for image quality

Disturbance	Score
Imperceptible	5
Perceptible but not annoying	4
Slightly annoying	3
Annoying	2
Very annoying	1

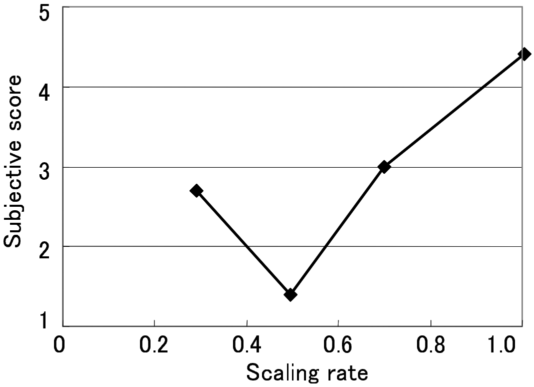


Fig. 8. Subjective image quality of watermarked video

Legibility of Copy Indicators in the Copied Video

If the copy indicators that pop up in the copied video are clear enough to be read, they are considered to be legible. Copied videos were displayed on a monitor by using a procedure based on the single-stimulus impairment scale method and evaluated by participants who rated the legibility of the indicators by scoring on a five-point scale.

The levels used are summarized in Table 4. Note that we used an original scale as we are unaware of a proven scale for such evaluation. The same five participants evaluated the same sample as used in the re-encoding test. The average scores are shown in Fig. 9.

The legibility should be low for a scaling rate of 1.0 (original size) and high for the other scaling rates. When the scaling rate was 0.5, which is under the designated condition, the average score was 3.8, meaning that the copy indicator could be read clearly. The legibility scores for the other scaling rates were 1 (very illegible) for all participants. For those scaling rates, the copy indicator could not be read, meaning that the image quality of copied video is only distorted (emerged watermarks will slightly annoys users, as shown in Fig. 8). For instance, if another pattern sensitive to a different condition (such as a scaling rate of 0.6) was also applied in the watermarked video, the legibility score would expectedly be higher for this condition as well. If various patterns according to different designated conditions were applied, the legibility of the copy indicator would expectedly be higher for this condition as well. If various patterns corresponding to different designated conditions were applied, the legibility of the copy indicator would be improved.

Table 4. Levels for legibility

Legibility	Score
Very legible	5
Legible	4
Slightly legible	3
Illegible	2
Very illegible	1

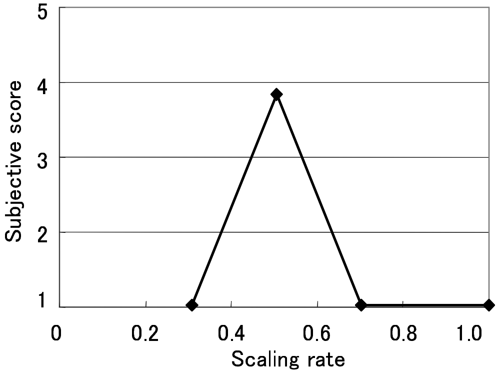


Fig. 9. Subjective legibility of copy indicators in copied video

4.3 PSNR Test

A frame image in the sample video was taken into an uncompressed still image format. That is, it was the original image. A watermarked image is generated by embedding watermarks in the original image. We resized both the original and watermarked images at each scaling rate and then calculated the peak-signal-to-noise ratio (PSNR):

$$\mathrm{PSNR} = 20\log_{10}\left(255 \Big/ \sqrt{\frac{\sum_{p\in\Omega}(\mathrm{stego}(p) - \mathrm{cover}(p))^2}{w\cdot h}} \right) \tag{3}$$

where Ω is set of valid integer coordinates, i.e., $(0,0)(0,0),\ldots,(0,h-1)\,(1,0),\ldots,(w-1,h-1)$, $\mathrm{stego}(p)$ is the luminance value of pixel p in the watermarked image, $\mathrm{cover}(p)$ is the luminance value of pixel p in the original image, *w* is width of the images, and *h* is height of the images.

Fig. 10 shows the PSNR for various scaling rates. We can divide the set of measured PSNRs into two groups, those around 27.85 and those around 28.15. When the sample image was resized using each scaling rate corresponding to the PSNRs around 28.15, the hidden message in the resized image was hard to read. On the other hand, the embedded messages could easily be read when the sample images was resized using each scaling rate corresponding to the PSNRs around 27.85 (scaling rate = 0.1 or 0.5). Those results demonstrate that our method distorts the image quality of watermarked video when resizing is done at designated scaling rates.

Ideally, the PSNR should be higher only when the scaling rate is 1.0 and lower at the other scaling rates to deter copying at any scaling rate. We do not consider this to be an inherent limitation of our prototype. Our prototype embeds a cipher image in the video content that becomes visible at the designated scaling rate (a_j in equation (2)). The peak-signal-to-noise-ratio in watermarked video was adjustable by more than 28 dB. For actual application, we can prepare cipher images corresponding to the possible scaling rates by using spatial and temporal video components and thereby deter copying at any scaling rate.

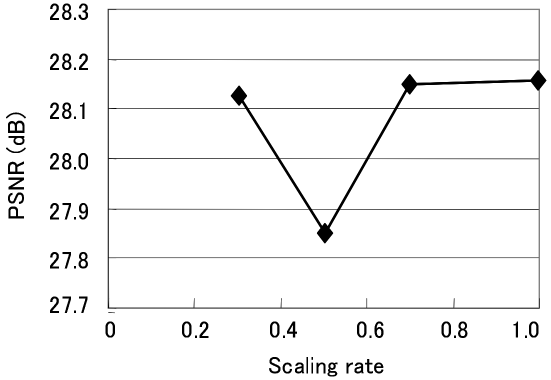


Fig. 10. Peak-signal-to-noise ratios in copied videos for various scaling rates

5 Summary and Future Works

We have developed a novel approach to deterring the illegal copying of video content: copy indicators emerge in video content that has been copied under designated conditions. A specially designed pattern in the original content becomes visible due to the aliasing effect. This pattern then degrades the image quality of the copied video and/or warns the user that the video has been illegally copied by presenting messages in the video content. We hypothesize that watermarked video is re-encoded with resizing at a particular scaling rate when it is illegally copied. Implementation of this method in a prototype demonstrated that it works well: hidden warning messages emerged when the hypothesis was met. Use of this method should help deter users from illegally copying and/or distributing copyrighted videos.

This work has confirmed the validity of our hypothesis and the feasibility of our approach. Although the presentation here was mainly based on the use of nearest neighbor interpolation, we believe that proposed method will be effective when other scaling algorithms are used. Because watermark strength is adjustable, we should be able to improve the PSNR by using weaker watermarks. Further evaluation (subjective and objective tests), application development, and system security establishment remain for future works.

References

1. George, C., Scerri, J.: Web 2.0 and user-generated content: legal challenges in the new frontier. Journal of Information, Law and Technology 2 (2007)
2. Bloom, J.A., Cox, I.J., Kalker, T., Linnartz, J.-P.M.G., Miller, M.L., Traw, C.B.S.: Copy protection for DVD video. Proc. IEEE 87(7), 1267–1276 (1999)
3. Hartung, F., Ramme, F.: Digital rights management and watermarking of multimedia content for M-commerce applications. IEEE Communication Magazine 38(11), 78–84 (2000)
4. Yamada, T., Gohshi, S., Echizen, I.: Re-shooting prevention based on difference between sensory perceptions of humans and devices. In: Proc. of Int'l Conf. on Image Processing (ICIP 2010), pp. 993–996 (2010)
5. Cox, I.J., Miller, M.L., Bloom, J.A., Fridrich, J., Kalker, T.: Digital watermarking and steganography, 2nd edn. Morgan Kaufmann (2007)
6. Wu, et al.: Watermarking for image authentication. In: Proc. of IEEE Int'l Conf. on Image Processing, vol. 2, pp. 437–441 (1998)
7. Lin, C.Y., Chang, S.F.: A robust image authentication method surviving JPEG lossy compression. In: Proc. SPIE, vol. 3312, pp. 296–307 (1998)
8. Lin, C.Y., Chang, S.F.: Issues and solutions for authenticating MPEG Video. In: Proc. SPIE, vol. 3657, pp. 54–65 (1999)
9. Jeong, Y.J., Kim, W.H., Moon, K.S., Kim, J.N.: Implementation of watermark detection system for hardware based video watermark embedder. In: Proc. of Int'l Conf. on Convergence and Hybrid Information Technology, pp. 450–453 (2008)
10. Atomori, Y., Echizen, I., Dainaka, M., Nakayama, S., Yoshiura, H.: Robust video watermarking based on dual plane correlation for immunity to rotation, scale, translation, and random distortion. Journal of Digital Information Management 6(2), 161–167 (2008)

11. Yamada, T., Takashima, K., Yoshioka, H.: Development of Transcoder in Conjunction with Video Watermarking to Deter Information Leak. In: de Leon F. de Carvalho, A.P., Rodríguez-González, S., De Paz Santana, J.F., Rodríguez, J.M.C. (eds.) Distributed Computing and Artificial Intelligence. AISC, vol. 79, pp. 229–236. Springer, Heidelberg (2010)
12. van Renesse, R.L.: Paper based document security – a review. In: Proc. of European Conference on Security and Detection (ECOS 1997), vol. (437), pp. 75–80 (1997)
13. Huang, S., Wu, J.K.: Optical watermarking for printed document authentication. IEEE Trans. on Information Forensics and Security 2(2), 164–173 (2007)
14. The Institute of Image Information and Television Engineers (ITE) : Standard video
15. Recommendation ITU-R BT.500–11: Methodology for the subjective assessment of the quality of television pictures (2002)
16. Hersch, R.D., Chosson, S.: Band Moire Images. Proc. SIGGRAPH 2004, ACM Trans. on Graphics 23(3), 239–248 (2004)

Authenticating Visual Cryptography Shares Using 2D Barcodes

Jonathan Weir and WeiQi Yan

Queen's University Belfast, Belfast, BT7 1NN, UK

Abstract. One of the problems pertinent with many visual cryptography (VC) schemes is that of authentication. VC provides a way of sharing secrets between a number of participants. The secrets are in the form of an image that is encoded into multiple pieces known as shares. When these shares are physically superimposed, the secret can be instantly observed. A known problem is that of authentication. How is it possible to know that the secret being recovered is genuine? There has been some work devoted to this using so called cheating prevention schemes which attempt to provide a means of traceability or authentication via a set of additional shares that are used to check authenticity. This paper proposes a scheme that attempts to alleviate this suspicion by using 2D barcodes as a means of authentication which may have more practicality in terms of real world usage. Results are provided using an application that is available on mobile devices for portable barcode reading.

1 Introduction

As far as secret sharing goes, visual cryptography [9] provides a very effective method for accomplishing this. One of the common problems that arise when designing VC schemes is whether or not the scheme can be cheated such that the set of known participants that are allowed to recover the secret can be cheated into recovering a secret of a different type without knowing they have been compromised.

Specific schemes that have been designed for cheat prevention focus on the probability that an attacker successfully cheats a scheme is negligible, that is, the known participants suspect that the shares or recovered secret is not genuine [4]. There are two types of approaches used when constructing these cheating prevention techniques. The first type is an authentication based method whereby each known participant is given an additional share that is used to authenticate the recovered secret. This provides the participants with the ability to verify the integrity of the shares before secret reconstruction takes place. The other authentication method uses a blind authentication technique that uses the properties of the reconstructed secret image. Blind authentication attempts to make it more difficult for the cheaters to predict the structure of a valid share that is in the possession of the qualified participants.

Despite the obvious advantages of having an additional share for verification purposes, the fact that an additional share is required is rather cumbersome and

Y.Q. Shi, H.J. Kim, and F. Perez-Gonzalez (Eds.): IWDW 2011, LNCS 7128, pp. 196–210, 2012.

impractical. This paper attempts to embed the verification information inside the recovered secret in the form of a 2D barcode. This way, no additional share is required and if cheating is suspected, the 2D barcode will not verify the invalid share after it is used to recover the secret. This is due to the fact that the barcode cannot be successfully guessed or that after a cheater has used his share, a barcode may not even be recovered as part of the secret.

Additionally, an extended form of this secret sharing is presented which embeds the 2D barcode as part of the cover image for each participant. This allows share verification before the secret recovery has even been attempted. This type of early verification can take place using mobile devices such as iPhones, which support software barcode readers that use the mobile devices onboard camera.

The main contributions discussed within this paper are:

- Using a 2D barcode to authenticate a VC share.
- Any VC scheme can benefit from this type of authentication, depending on the practical requirements.
- The 2D barcode can be used as the secret transport mechanism. That is, a long string of alphanumeric characters can be embedded inside the barcode.
- This increases the overall capacity of sharing a large amount of data within a small manageable set of shares.
- The practical usage involving mobile devices with an onboard camera is very simple and effective.

The remainder of the paper is organized accordingly, Section 2 provides a brief background on cheating and cheat prevention schemes in VC and the related work accomplished in this area, Section 3 highlights the proposed idea and how it will be achieved, while Section 5 provides the conclusion.

2 Related Work

Despite visual cryptography's secure nature, many researchers have experimented with the idea of cheating the system. Methods for cheating the basic VC schemes have been presented, along with techniques used for cheating extended VC schemes [7,10,18].

Prevention of cheating via authentication methods [10] has been proposed which focus on identification between two participants to help prevent any type of cheating taking place. Yang and Laih [18] presented two types of cheating prevention, one type used an online trust authority to perform the verification between the participants. The second type involved changing the VC scheme whereby the stacking of two shares reveals a verification image, however this method requires the addition of extra pixels in the secret.

Another cheating prevention scheme described by Horng et al. [7], through which if an attacker knows the exact distribution of black and white pixels of each of the shares of honest participants then they will be able to successfully attack and cheat the scheme. Horng's method prevents the attacker from obtaining this distribution. Since then, there have been numerous efforts devoted to designing cheating prevention schemes within visual cryptography [5,8,12].

Successfully cheating a visual cryptography scheme (VCS) however, does not require knowledge of the distribution of black and white pixels. Hu and Tzeng [8] where able to present numerous cheating methods, each of which where capable of cheating Horng et al.'s cheating prevention scheme. Hu and Tzeng also present improvements on Yang and Laih's scheme and finally present their own cheating prevention scheme which attempts to minimize the overall additional pixels which may be required. No online trust authority is required and the verification of each image is different and confidential. The contrast is minimally changed and the cheating prevention scheme should apply to any VCS. Hu and Tzeng where also able to prove that both a malicious participant (**MP**), that is $\mathbf{MP} \in P$, and a malicious outsider (**MO**), $\mathbf{MO} \notin P$, can cheat in some circumstances, where P is the set of participants.

The **MP** is able to construct a fake set of shares using his genuine share. After the fake share has been stacked on the genuine share, the fake secret can be viewed. The second cheating method involving an **MO** is capable of cheating the VC scheme without having any knowledge of any genuine shares. The **MO** firstly creates a set of fake shares based on the optimal $(2, 2)$-VCS. Next, the fake shares are required to be resized to that of the original genuine shares size. However, an assumption is to be made on the genuine shares size, namely that these shares where printed onto a standard size of paper, something like A4 or A3. Therefore, shares of those sizes are created, along with fractions of those sizes. Management of this type of scheme would prove to be problematic due to the number of potential shares created in order to have a set of the correct size required to cheat a specific scheme, but once that size is known, cheating is definitely possible as an **MO**.

A traceable model of visual cryptography [1] was also examined which endeavours to deal with cheating. It deals with the scenario when a coalition of less than k traitors who stack their shares and publish the result so that other coalitions of the participants can illegally reveal the secret. In the traceable model, it is possible to trace the saboteurs with the aid of special markings. The constructions of traceable schemes for both (k, n) and (n, n) problems were also presented. Furthermore, other practical applications have also been examined including one involving biometrics [3,6,14,15,16,17]. This paper further adds to this list of practical applications involving visual cryptography.

3 The Proposed Scheme

With the previous work in mind, a number of areas should be focused on during the creation of the new scheme. A suitable barcode must be selected to perform the authentication along with a VC scheme capable of processing and handling this type of information. Using this type of technique with existing VC schemes should also be considered. This would allow an authentication mechanism to be used with current techniques. The actual problem of authentication and the issues facing it should be acknowledged as well.

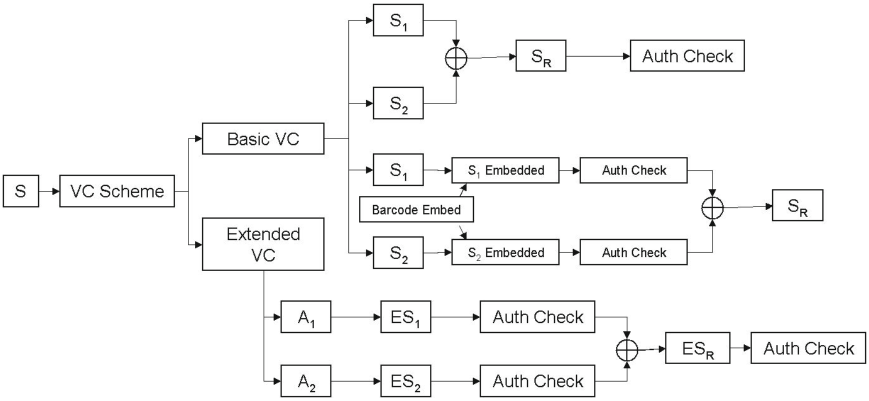


Fig. 1. Flowchart of the proposed VC authentication system

Figure 1 provides the flowchart for the proposed scheme. Two separate VC schemes are used, a traditional basic VC scheme and an extended VC scheme.

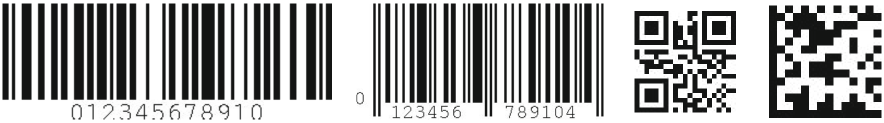
The secret S is selected. For the traditional scheme, this would be the barcode that is used for authentication. The extended scheme can use this type of secret as well, or another secret such as a PIN number or code number for a safe or bank vault. After the secret has been input, the choice of VC scheme is next. The basic VC scheme will go on to process the secret and generate two shares S_1 and S_2. Physically superimposing these shares will recover the secret image S_R. This results in the recovery of the barcode which can then be used for authentication. The basic scheme also offers the ability to authenticate each share for traditional secret recovery. The barcodes disappear after the secret has been recovered.

The extended scheme functions quite differently, in that more authentication checks are possible during the process due to the nature of the extended VC scheme. Cover images A_1 and A_2 are generated. These are barcodes which contain unique authentication information. Using these authentication images, two shares ES_1 and ES_2 are created which when combined can recover the original secret ES_R. It can be noticed that further authentication checks are now possible before secret recovery takes place. The final secret that is recovered can also be checked for authenticity.

3.1 Determining a Suitable Barcode

A barcode is an optical machine-readable representation of data, which shows data about the object to which it attaches. Traditionally, barcodes represent data by using parallel lines with varying widths and spacing. Other geometric patterns, such as rectangles, dots and hexagons have also be used in the creation of barcodes. Barcode scanners are used to read the barcode and decode the information within it.

There are a vast array of barcode types to choose from, but the main examples used within this paper include the more common types: Code-128 (1D), EAN-13 (1D), QR code (2D), Datamatrix (2D). An example of these types of barcode are shown in Figure 2.



(a) Code-128 (b) EAN-13 (c) QR code (d) Datamatrix

Fig. 2. Example barcode types. Each of which contain the same information string: 012345678910

However, for this type of application, alphanumeric characters are more suited to the type of secret verification that will be used, so EAN-13 will not be considered during the testing. It is included here merely as an example.

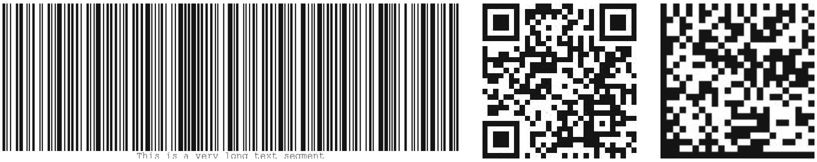
Barcodes are highly robust when it comes to extracting the information contained within them. Using mobile devices equipped with a camear, barcodes, in the form of QR codes are a common way to read information from a magazine or advertisement [11]. Such mobile devices have limited processing power and low resolution cameras. This requires the barcodes to be machine readable on limited devices. This is why they are useful for the work combining visual cryptography.

Traditionally, visual cryptography deals with binary images, this is another great advantage for combining it with barcodes. No colour image processing is required and the share generation can use many types of existing VC techniques to generate the required shares. Ranging from schemes that expand the pixels into a 2 × 2 block to size invariant schemes which maintain the original aspect ratio of the secret. Barcode readers are still capable of recovering the information from the barcode, even after such a distortion may have taken place.

Another issue which crops up time and again is that of capacity. Clearly hiding and recovering quantities of text within an image, especially one that has been encoded using visual cryptography has been problematic. Using such methods can be cumbersome and difficult to read. Recovering barcodes accurately which store long types of textual information would be a much more useful and easier. Figure 3 illustrates a number of barcode schemes which contain a long string of text. Each barcode can be accurately read by a barcode reader.

Share size is another issue that researchers face when designing visual cryptography schemes. Management of large shares is unwieldy and the smaller the share, the better it is for the application. This goes hand in hand with secret recovery. If the shares are small, the secret will be unclear, unless the secret is in the form of a barcode, such as a QR code, which can contain a large amount of textual data inside a small image.

From the barcodes displayed in Figure 2 and 3 it can be observed that the Code-128 barcode is the most effected by the change in information. The final image size jumped from 303 × 106 to 1161 × 392. The QR code and Datamatrix



(a) Code-128 (b) QR code (c) Datamatrix

Fig. 3. Textual data within each of the barcodes that support that type of information. The information string: “This is a very long text segment”.

Table 1. Table of resolutions depending on the barcode type used along with the amount and type of data represented

Barcode type	Resolution (digits only)	Resolution (text, including spaces)
Code-128	303 × 106	1161 × 392
QR code	63 × 63	87 × 87
Datamatrix	70 × 70	120 × 120

both fared much better, with less of a size increase. Both of which remained at a very manageable size of 87 × 87 and 120 × 120 respectively. These resolution increases are tabulated in Table 1.

Based on this, QR code and Datamatrix representation will be used during each of the test phases when combining these barcode types with visual cryptography as an authentication medium. Improving efficiency in terms of processing power, when generating shares, favours smaller secret images to begin with. So we can successfully ignore the Code-128 type of barcode from the remainder of the testing.

3.2 Visual Cryptography Scheme Selection

Now that the type of barcode has been chosen, the next step involves using a VC scheme that allows sufficient secret recovery such that a barcode scanner can be used to read the recovered barcode. This is extremely important, if the barcode cannot be read correctly, the shares or the participants cannot be authenticated. Due to the small nature of the QR code image that is produced, many VC schemes would be well suited to sharing this type of information. A number of tests will be performed using a variety of schemes to illustrate this.

The QR code image that will be used as an authentication image can be viewed in Figure 4. The authentication message that goes along with it is also included. The process in itself will be one of a digital nature at first, using a computer to recover the secret and then test the barcode for authentication. The tests will then be extended into the physical form of testing, so that tests can be done using traditional means with a camera on a mobile device.

The application used to read the barcode is an open source software suite known as ZBar [2]. This package is used because it supports a multitude of operating systems which includes the iPhone and other embedded devices. Mobile

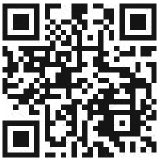
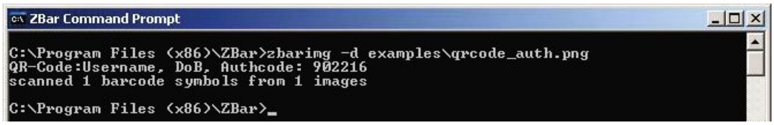


Fig. 4. QR code (87 × 87) with personal details and the authentication number. Embedded text: “Username, DoB, Authcode: 902216”



```
C:\Program Files (x86)\ZBar>zbarimg -d examples\qrcode_auth.png
QR-Code:Username, DoB, Authcode: 902216
scanned 1 barcode symbols from 1 images

C:\Program Files (x86)\ZBar>_
```

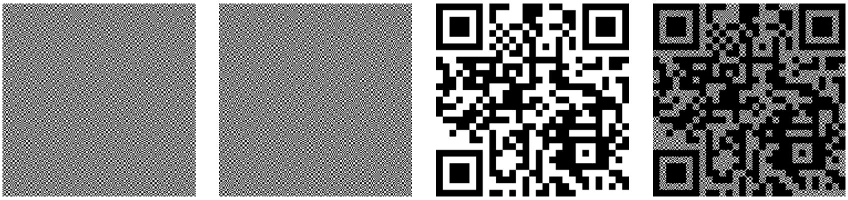
Fig. 5. ZBar reading the QR code and confirming the correct details

devices such as iPhones are becoming evermore popular, this is a good indication as to whether this research has merit in a real world application. Figure 5 provides an example of ZBar reading the corresponding QR code in Figure 4 and returning the correct details.

Essentially there are two different types of secret recovery, the first method will use the XOR binary operation to combine the shares, the later uses the OR operation [13]. XOR can be used when dealing primarily in a digital environment, as it removes any grey shading that may be observed when black and white pixels are arranged together one after another, leaving either solid white or solid black sections of the image.

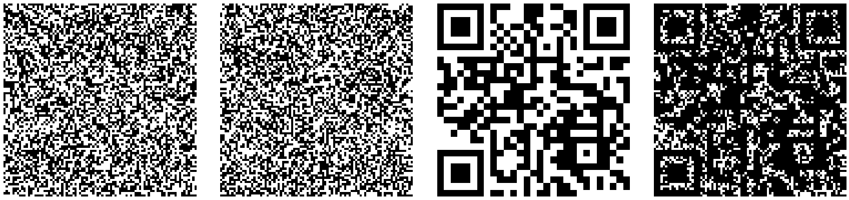
A number of VC schemes will be tested throughout. A basic $(2, 2)$ VC scheme which involved 2×2 pixel expansion and a size invariant scheme, both $(2, 2)$ and $(2, 3)$ to show that k out of n sharing is possible. Figure 6 provides an example of the shares based on (2,2) VC scheme, with a pixel expansion of 2×2. Figure 7 gives an example based on a (2,2) size invariant scheme. Included within the figures are the XOR and OR secret recovery images.

Figure 8 provides the results of a (2,3) size invariant scheme which shows that the same results are possible using a k out of n secret sharing scheme.



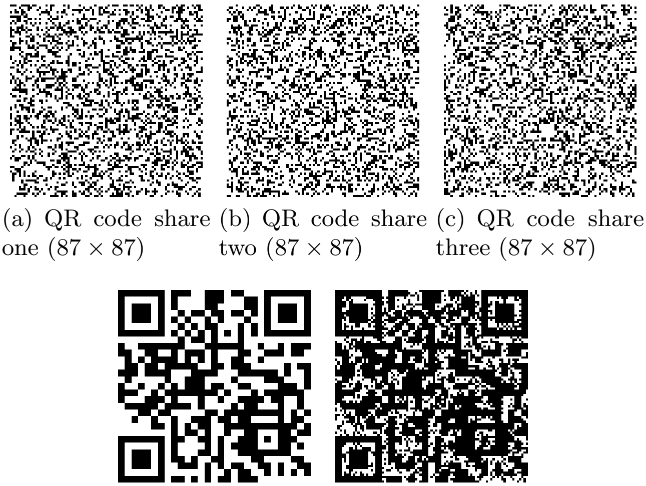
(a) QR code share one (174 × 174) (b) QR code share two (174 × 174) (c) XOR secret recovery (174 × 174) (d) OR secret recovery (174 × 174)

Fig. 6. A $(2, 2)$ basic visual cryptography secret recovery process involving the XOR and OR binary operations



(a) QR code share one (87 × 87) (b) QR code share two (87 × 87) (c) XOR secret recovery (87 × 87) (d) OR secret recovery (87 × 87)

Fig. 7. A $(2, 2)$ size invariant visual cryptography secret recovery process involving the XOR and OR binary operations



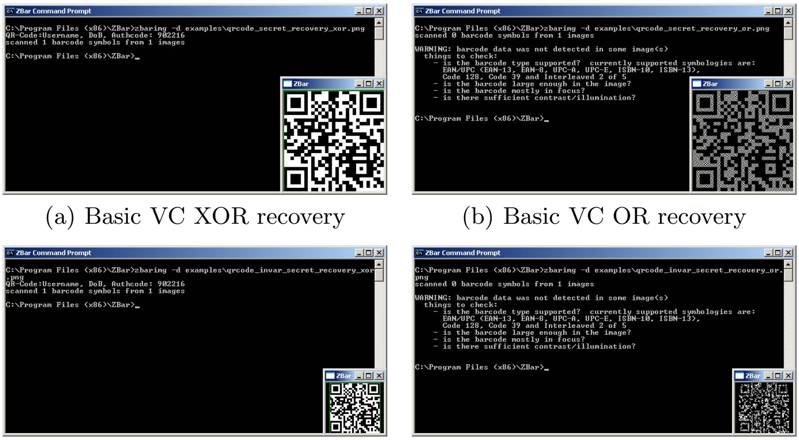
(a) QR code share one (87 × 87) (b) QR code share two (87 × 87) (c) QR code share three (87 × 87)

(d) XOR secret recovery (87 × 87) (e) OR secret recovery (87 × 87)

Fig. 8. A $(2, 3)$ size invariant visual cryptography secret recovery process involving the XOR and OR binary operations

Both the recovered secrets using the XOR methods can be verified very easily by the barcode reader application. However, the barcode recovered using the OR operation presents a problem, both barcodes cannot be successfully read. There are a number of reasons why this is the case. The barcode could possibly be too small. This can be ruled out, as the barcode is twice as large, or the same size as the original barcode. The focus on the image is another issue. This is not the case either, as the testing so far has been digital only. The last reason why the barcode may not be read is that there may not be sufficient contrast or illumination on the image.

Figure 9 shows the results of attempting to recover the authentication data from each of the secrets. It can be observed that the barcodes recovered using the XOR operation can be perfectly read, whereas the secrets recovered using the OR operation cannot be detected. This helps to reinforce the point about



```
C:\Program Files (x86)\ZBar>zbarimg -d examples\qrcode_secret_recovery_xor.png
QR-Code:Username, DoB, Authcode: 902216
scanned 1 barcode symbols from 1 images

C:\Program Files (x86)\ZBar>_
```

(a) Basic VC XOR recovery

```
C:\Program Files (x86)\ZBar>zbarimg -d examples\qrcode_secret_recovery_or.png
scanned 0 barcode symbols from 1 images

WARNING: barcode data was not detected in some image(s)
  things to check:
    - is the barcode type supported?  currently supported symbologies are:
      EAN/UPC (EAN-13, EAN-8, UPC-A, UPC-E, ISBN-10, ISBN-13),
      Code 128, Code 39 and Interleaved 2 of 5
    - is the barcode large enough in the image?
    - is the barcode mostly in focus?
    - is there sufficient contrast/illumination?


C:\Program Files (x86)\ZBar>_
```

(b) Basic VC OR recovery

```
C:\Program Files (x86)\ZBar>zbarimg -d examples\qrcode_invar_secret_recovery_xor
.png
QR-Code:Username, DoB, Authcode: 902216
scanned 1 barcode symbols from 1 images

C:\Program Files (x86)\ZBar>_
```

(c) Size invariant VC XOR recovery

```
C:\Program Files (x86)\ZBar>zbarimg -d examples\qrcode_invar_secret_recovery_or.
png
scanned 0 barcode symbols from 1 images

WARNING: barcode data was not detected in some image(s)
  things to check:
    - is the barcode type supported?  currently supported symbologies are:
      EAN/UPC (EAN-13, EAN-8, UPC-A, UPC-E, ISBN-10, ISBN-13),
      Code 128, Code 39 and Interleaved 2 of 5
    - is the barcode large enough in the image?
    - is the barcode mostly in focus?
    - is there sufficient contrast/illumination?


C:\Program Files (x86)\ZBar>_
```

(d) Size invariant VC OR recovery

Fig. 9. Attempted authentication of each of the barcodes based on the type of secret recovery used

share tampering. If the shares have been altered, then reading of the authentication information from the barcode becomes problematic. This will give a good indication as to whether the shares have been altered or not.

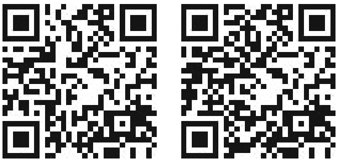
This is an important point as many VC schemes rely on a difference in contrast in order to display the secret after recovery. This is why the XOR operation is much better suited to this type application. From Figure 9 it can be seen due to the dark nature of the secrets recovered using the OR operation, authentication becomes difficult. The barcode reader has difficulty in determining where the barcode is on the image. Additionally, size is not an issue, as the barcode can be successfully read after the XOR recovery on the original barcode secret that was encoded using a size invariant VC scheme.

3.3 The Authentication Problem

After observing the results obtained with the previous example, the authentication problem can be viewed from a simpler stance. Slight alterations create or generate barcodes that are completely unreadable or could possibly generate valid barcodes with invalid data.

Figure 10 illustrates this with a simple example, which increments the last digit of the authentication string by one. Viewed side by side, it can be seen that a superficial change in the data generates a vastly different barcode. This is important in terms of authentication. If the shares are tampered with and a barcode is generated that is not exact, no authentication will be possible. This renders the shares useless. This is a great property to have for authentication purposes.

This favours VC a lot from the point of view that if additional noise is added to the tampered image then a false or non-genuine authcode will be extracted,

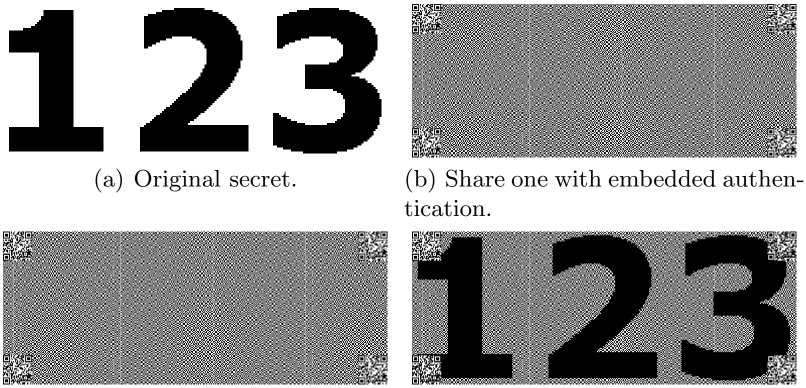


(a) Authcode: 1111 (b) Authcode: 1112

Fig. 10. Two QR codes, both containing a similar authcode. A slight difference in authcode produces a vastly different QR code.

highlighting the fact that the shares cannot be trusted for that participant. Minor disruptions to the shares which result in spurious or completely invalid barcodes being generated is a good way to keep track of authenticate shares.

Figure 11 shows how the shares can be authenticated using the 2D barcode embedded into the corner of the share. The original image included along with its corresponding shares and recovered secret. The barcode share is embedded with a similar density and distribution as the share it is placed into. It also disappears when the secret is recovered. Allowing for more exact recovery.



(a) Original secret.

(b) Share one with embedded authentication.

(c) Share two with embedded authentication.

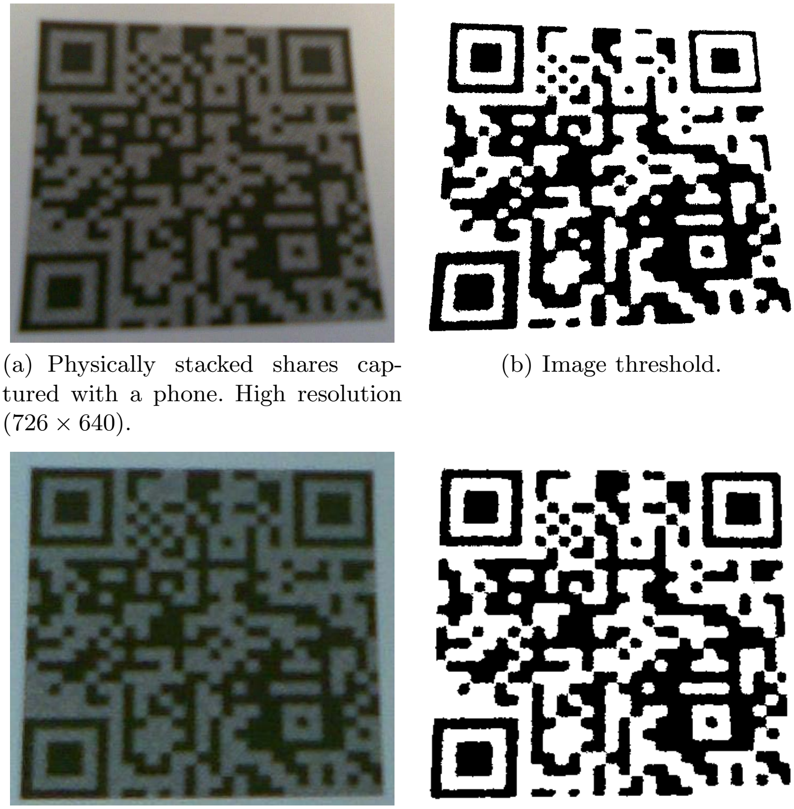
(d) Recovered secret.

Fig. 11. Using a 2D barcode to authenticate each share

3.4 Authenticating the Shares

Authenticating the shares at a practical level using the techniques described can be achieved using a standard mobile device or smart phone. Many of these devices support programming platforms such as Python. This is very useful when it comes to simple image processing that may be required to process photographs of the shares when it comes to authenticating them.

Figure 12 provides an example of a photograph of the same barcode taken at different resolutions (Figure 12(a) and 12(c)). The angles at which the photographs were taken also differs in each of the figures. This helps to reinforce the robustness of using a barcode for this type of authentication. Barcodes are very resilient to changes in angle such as in this figure. It also removes the onus from the user of having to take a photograph at a specific angle or to carefully align the camera. The thresholds of each of the barcode photographs have been taken, these are visible in Figure 12(b) and 12(d) respectively.



(a) Physically stacked shares captured with a phone. High resolution (726 × 640).

(b) Image threshold.

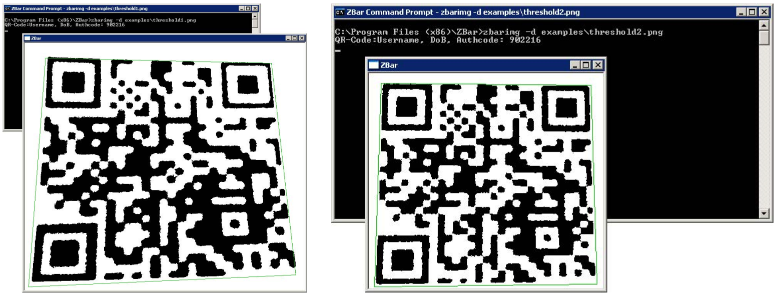
(c) Physically stacked shares captured with a phone. Low resolution (354 × 325).

(d) Image threshold.

Fig. 12. Capturing the physically stacked barcode images and thresholding them so that the binary barcode reader can process them correctly for authorization

Due to the noisy and grainy nature of the photographs, taking a threshold of the original photo is necessary when it comes to accurately reading the barcode. Despite the fact that barcodes are very robust to many types of image manipulation, contrast and brightness are also important factors in recovery and reading of a barcode. If contrast and brightness conditions are suboptimal, the code will not be successfully read.

The Zbar application can then be used to read the barcode from each of the threshold images that are processed. Figure 13 illustrates this recovery process, displaying clearly each of the authentication messages. The application also correctly outlines the area that it is reading and has recognized as the barcode itself. The resolution and angle of the photograph are not overly important to the barcode reader as is illustrated by the successful recovery of the barcode after the shares have been physically stacked.


ZBar Command Prompt - zbarimg -d examples\threshold1.png
ZBar
ZBar Command Prompt - zbarimg -d examples\threshold2.png
ZBar

```
C:\Program Files (x86)\ZBar>zbaring -d examples\threshold1.png
QR-Code:Username, DoB, Authcode: 902216
```

(a) Successful barcode recovery 1.

```
C:\Program Files (x86)\ZBar>zbarimg -d examples\threshold2.png
QR-Code:Username, DoB, Authcode: 902216
```

(b) Successful barcode recovery 2.

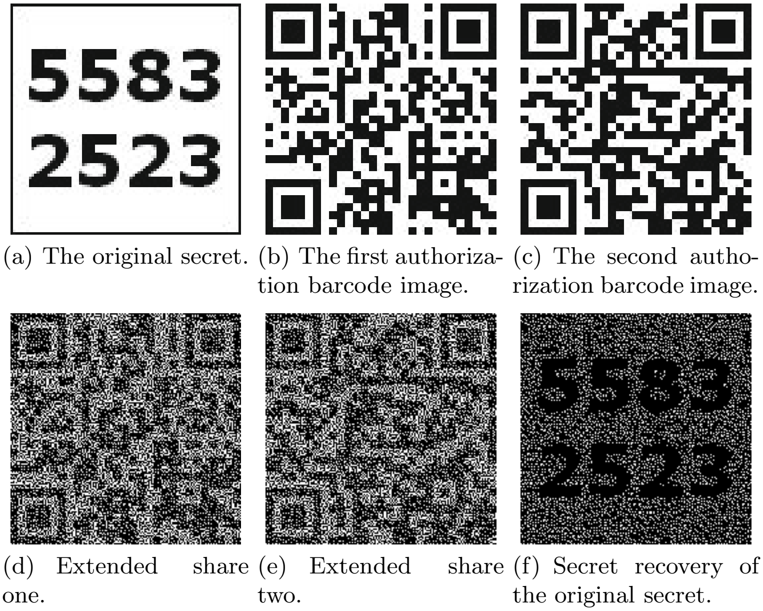
Fig. 13. Recovering the barcode information from each of the images processed by the phone

Furthermore, the authcode that has been extracted from each of the share can then be checked against a database on the mobile device or against a remote database on the network or internet for an additional check for the correct details.

Including a verification process for each individual share is also achievable. Using an extended form of visual cryptography, the authorization barcodes can be used as the secrets for each share. The shares can be verified and checked using a mobile device in the same manner as the previous example. Each share can have the same authcode or a unique code, depending on the type of authentication required by each person. Another advantage of this is that after verification, the main secret (a safe combination for example) can be completely recovered, while having no part of the original authorization barcode obscuring it.

Figure 14 provides an example of how the extended type of visual cryptography shares can be used to achieve this. The secret can be viewed in Figure 14(a). The authorization images are shown within the Figure 14(b) and 14(c). The shares used for the secret recovery can be observed in Figure 14(d) and 14(e). These shares contain the verification barcodes from Figure 14(b) and 14(c) respectively. The recovered secret is displayed in Figure 14(f). The recovery occurs when each of the shares physically superimposed.

One issue with this type of authorization is that changes in light intensity and contrast have a big impact on reading the barcode embedded within the share.



(a) The original secret. (b) The first authorization barcode image. (c) The second authorization barcode image.

(d) Extended share one. (e) Extended share two. (f) Secret recovery of the original secret.

Fig. 14. Extended VC with an authentication mechanism built into the share images

This is an area where improvement can be difficult. If too much of the barcode is visible, reconstruction of the secret could be obscured by this.

4 Security Analysis

The security of the scheme rests entirely with VC and its construction. Firstly, the presented scheme is secure in that given any amount of sub-pixels from a single share, it is impossible to tell if the corresponding shares sub-pixels represent a black or a white pixel after superimposing them. This can be illustrated using a probabilistic proof. For a random secret, it cannot be assumed that the pixel values selected to represent those from the secret are uniformly distributed. This is down to the size invariant scheme, in that one pixel from the secret has to be represented by one pixel from one of the shares, while the second share must also contain just a single pixel while keeping the value of the secret pixel hidden.

If a black pixel is to be represented from the secret, then its corresponding pattern is always black. Conversely, if a white pixel is to be represented then it can have two possible representations. A white pixel in each share is possible, or a white pixel in share one with a black pixel in the second share, which ultimately ends up as a black pixel, but does indeed represent a white pixel from the secret. So if a pixel is examined and found to be black in one share, the probability that it is black in the second share is 0.5. However, if the pixel in the share is white then there is also a probability of 0.5 that the pixel will be either black or white.

This makes it very difficult to analyze the secret based on these probabilities due to the nature of the pixel representations.

5 Conclusion

The principal idea from this paper is to use barcodes as an authentication means by which visual cryptography shares can be verified. The type of applications that can make use of this secret sharing are many. Using the scheme as a verification for opening bank vaults or other security related schemes is an important issue and can be achieved with relative simplicity in terms of checking that the barcode is accurate and untampered. Simple, manageable methods of share verification are quite difficult and require other methods as previous described, such as other shares.

This paper differs in that it presents the verification data within the share in the form of a barcode which helps to remove the issues that plague other schemes. Especially since such common devices such as mobile telephones and smart phones (iPhone) can be used as a means to facilitate this, the additional hardware requirement is not something of an issue.

From the results it can be observed that while the barcode application may not be able to read the shares that have been digitally superimposed, the shares that are physically stacked and captured with a camera can be recovered and identified successfully. This would be a good way of verifying and authenticating a set of shares to confirm the identity of a particular party or individual.

Along with that, storing a larger amount of textual data inside the barcode and read using a mobile device has also been accomplished. Storing a long, easily readable and easily recoverable secret using traditional VC techniques becomes very problematic, as the share size increases dramatically as more text is added. Using the scheme presented within this paper, the problem of share size is removed completely in terms of the amount of data can be held inside the embedded barcode. Essentially, smaller shares with more information are a great improvement.

References

1. Biehl, I., Wetzel, S.: Traceable Visual Cryptography. In: Han, Y., Quing, S. (eds.) ICICS 1997. LNCS, vol. 1334, pp. 61–71. Springer, Heidelberg (1997)
2. Brown, J.: ZBar bar code reader, `http://zbar.sourceforge.net/`
3. Chan, C.-W., Lin, C.-H.: A New Credit Card Payment Scheme Using Mobile Phones Based on Visual Cryptography. In: Yang, C.C., Chen, H., Chau, M., Chang, K., Lang, S.-D., Chen, P.S., Hsieh, R., Zeng, D., Wang, F.-Y., Carley, K.M., Mao, W., Zhan, J. (eds.) ISI Workshops 2008. LNCS, vol. 5075, pp. 467–476. Springer, Heidelberg (2008)
4. Chang, C.C., Chen, T.H., Liu, L.J.: Preventing cheating in computational visual cryptography. Fundamenta Informaticae 92, 27–42 (2009)
5. De Prisco, R., De Santis, A.: Cheating immune threshold visual secret sharing. The Computer Journal 53, 1485–1496 (2010), `http://dx.doi.org/10.1093/comjnl/bxp068`

6. Hegde, C., Manu, S., Shenoy, P., Venugopal, K., Patnaik, L.: Secure authentication using image processing and visual cryptography for banking applications. In: 16th International Conference on Advanced Computing and Communications, pp. 65–72 (December 2008)
7. Horng, G., Chen, T., Tsai, D.S.: Cheating in visual cryptography. Designs, Codes and Cryptography 38(2), 219–236 (2006)
8. Hu, C.M., Tzeng, W.G.: Cheating prevention in visual cryptography. IEEE Transactions on Image Processing 16(1), 36–45 (2007)
9. Naor, M., Shamir, A.: Visual Cryptography. In: De Santis, A. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 1–12. Springer, Heidelberg (1995)
10. Naor, M., Pinkas, B.: Visual Authentication and Identification. In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 322–336. Springer, Heidelberg (1997)
11. Rouillard, J.: Contextual qr codes. In: ICCGI 2008. The Third International Multi-Conference on Computing in the Global Information Technology, July 27-August 1, pp. 50–55 (2008)
12. Tsai, D.S., Chen, T.H., Horng, G.: A cheating prevention scheme for binary visual cryptography with homogeneous secret images. Pattern Recognition 40, 2356–2366 (2007), http://portal.acm.org/citation.cfm?id=1240339.1240571
13. Tuyls, P., Hollmann, H.D.L., Lint, J.H.V., Tolhuizen, L.: XOR-based visual cryptography schemes. Designs, Codes and Cryptography 37, 169–186 (2005), 10.1007/s10623-004-3816-4
14. Tuyls, P., Kevenaar, T., Schrijen, G.-J., Staring, T., van Dijk, M.: Visual Crypto Displays Enabling Secure Communications. In: Hutter, D., Müller, G., Stephan, W., Ullmann, M. (eds.) Security in Pervasive Computing. LNCS, vol. 2802, pp. 271–284. Springer, Heidelberg (2004)
15. Weir, J., Yan, W.: Resolution variant visual cryptography for street view of google maps. In: IEEE International Symposium on Circuits and Systems, ISCAS 2010 (May 2010)
16. Weir, J., Yan, W.-Q.: Dot-Size Variant Visual Cryptography. In: Ho, A.T.S., Shi, Y.Q., Kim, H.J., Barni, M. (eds.) IWDW 2009. LNCS, vol. 5703, pp. 136–148. Springer, Heidelberg (2009)
17. Weir, J., Yan, W.: Image hatching for visual cryptography. In: Proceedings of the International Machine Vision and Image Processing Conference, pp. 59–64. IEEE Computer Society Press, Los Alamitos (2009)
18. Yang, C., Laih, C.: Some new types of visual secret sharing schemes. In: National Computer Symposium (NCS 1999), vol. III, pp. 260–268 (December 1999)

Flexible Visual Cryptography Scheme without Distortion

Feng Liu[1], Teng Guo[1,2], ChuanKun Wu[1], and Ching-Nung Yang[3]

[1] State Key Laboratory of Information Security,
Institute of Information Engineering,
Chinese Academy of Sciences, Beijing 100190, China
[2] Graduate University of Chinese Academy of Sciences, Beijing 100190, China
[3] Department of Computer Science and Information Engineering,
National Dong Hwa University Shoufeng, Hualien 974, Taiwan
{liufeng,guoteng,ckwu}@is.iscas.ac.cn, cnyang@mail.ndhu.edu.tw
http://iscas.ac.cn/~liufeng/

Abstract. For visual cryptography scheme (VCS), normally, the size of the recovered secret image will be expanded by $m(\geq 1)$ times of the original secret image. In most cases, m is not a square number, hence the recovered secret image will be distorted. Sometimes, m is too large that will bring much inconvenience to the participants to carry the share images. In this paper, we propose a visual cryptography scheme which simulated the principle of fountains. The proposed scheme has two advantages: non-distortion and flexible (with respect to the pixel expansion). Furthermore, the proposed scheme can be applied to any VCS that is under the pixel by pixel encryption model, such as VCS for general access structure, color VCS and extended VCS, and our VCS does not restrict to any specific underlying operation. Compared with other non-distortion schemes, the proposed scheme is more general and simpler, real flexible and has competitive visual quality for the recovered secret image.

Keywords: Visual Cryptography, Secret Sharing, Non-Distortion, Flexible.

1 Introduction

The basic principle of visual cryptography scheme (VCS) was first introduced by Naor and Shamir [1]. In the VCS, there is a secret image which is encrypted into some share images. The secret image is called the *original secret image* for clarity, and the share images are the encrypted images (and are called the transparencies if they are printed). When a qualified set of share images (transparencies) are stacked together, it gives a visual image which is almost the same as the original secret image, we call it the *recovered secret image*. In the case of black and white images, the original secret image is represented as a pattern of black and white pixels. Each of these pixels is divided into subpixels which themselves are encoded as black and white to produce the share images. The recovered secret

Y.Q. Shi, H.J. Kim, and F. Perez-Gonzalez (Eds.): IWDW 2011, LNCS 7128, pp. 211–227, 2012.

image is also a pattern of black and white subpixels which should visually reveal the original secret image if a qualified set of share images are stacked.

For most VCS's in the literature, the scheme has to be applied on each secret pixel in the image respectively, in this paper we call such a way of encryption the *pixel by pixel encryption model.*

Many studies focused on enhancing the visual quality or reducing the pixel expansion of VCS, such as [2, 3, 4, 5, 6, 7, 8, 9, 10, 11] Besides, Ateniese et al. extended the threshold VCS to the general access structure [12], and Droste et al. proposed extended VCS which could have meaningful share images [2, 13, 14]. Many researchers also consider the novel applications of VCS [15, 16, 17]. Recently, a book covering an extensive range of topics related to VCS is published [18].

In general, the recovered secret image will be expanded by $m(\geq 1)$ times over the size of the original secret image i.e. the pixel expansion is m. However, in most cases, m is not a square number, hence the recovered secret image will be distorted. An example of distorted VCS can be found in Figure 1.

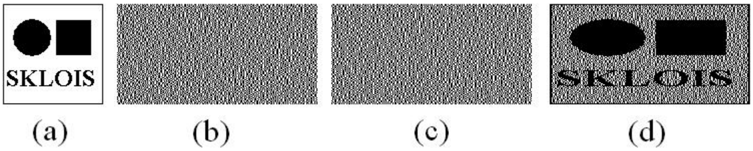


Fig. 1. An example of traditional VCS with pixel expansion 2, (a) is the original secret image with image size 100×100, (b) and (c) are the share images with image size 200×100, (d) is the recovered secret image with image size 200×100

In Figure 1, the circle and square are compromised to an oval and a rectangle respectively and hence lead to the loss of information. This will not be allowed, especially when the aspect ratio is viewed as important information of the secret image. To avoid distortion, many methods have been proposed. Naor and Shamir [1] recommended adding extra subpixels to retain the value of m as a square number. In such a case, the pixel expansion of the scheme will increase significantly for some m and meanwhile may degrade the visual quality of the scheme. Yang et al. [19, 20] proposed some aspect ratio invariant VCS's which relied on adding dummy subpixels to the shares, such methods also increase the overall pixel expansion. Beside, their method is complicated, how to design a mapping pattern that reduces the number of dummy subpixels to the minimum is, as they said, a huge challenge, especially for some pixel expansions and secret image sizes.

Sometimes, m is so large that will bring much inconvenience to the participants to carry them. Some other studies, hence, consider size invariant VCS, i.e. VCS with no pixel expansion [21, 22, 23, 24]. For such schemes, the recovered secret image will have no distortion. The size invariant VCS's are usually called probabilistic visual cryptography schemes (PVCS) for the reason that a secret pixel can only be recovered with a certain probability. In contrast to PVCS, the traditional VCS's are called deterministic visual cryptography schemes (DVCS), which means that a secret pixel can be recovered deterministically. Because of PVCS's probabilistic nature, the recovered secret images of PVCS often have bad visual quality. Usually, better visual quality of the recovered secret image requires larger pixel expansion [23].

In this paper, we propose a visual cryptography schemes which simulated the principle of fountains (see Section 3). The proposed scheme has two advantages: Non-distortion and flexible (with respect to the pixel expansion). The proposed scheme can be applied to any VCS that is under the pixel by pixel encryption model, such as VCS for general access structure, color VCS and extended VCS, and our VCS does not restrict to any specific underlying operations (OR or XOR). For larger pixel expansion, the recovered secret image of our scheme will have better visual quality, and smaller pixel expansion will compromise poorer visual quality. Hence, our scheme is flexible, the dealer can tradeoff the visual quality and pixel expansion of the recovered secret image according to different scenarios. Compared with other non-distortion schemes [19, 20, 25], the proposed scheme is more general, simpler and real flexible, while having competitive visual quality. Compared with the size invariant VCS [21, 22, 23, 24], our scheme can have flexible overall pixel expansion, i.e. the dealer can choose the overall pixel expansion at will, even less than 1.

The paper is organized as follows: In Section 2, we give some preliminary definitions about VCS. In Section 3 we give the basic fountain algorithm. In Section 4, we improve the basic algorithm with respect to the visual quality. In Section 5, we give some comparisons with some well-known schemes. Finally, the paper is concluded in Section 6.

2 Definitions about VCS

By a (k, n)-VCS we mean a scheme where the original secret image is divided into n shares, which are distributed to n participants. Any subgroup of k out of these n participants can get a recovered secret image, but any subgroup consisting of less than k participants does not have any information other than the size about the original secret image. More precisely, we give the formal definitions of (k, n)-DVCS and (k, n)-PVCS as follows:

Definition 1 (Deterministic VCS [26]). *Let k, n, m, l and h be nonnegative integers satisfying $2 \leq k \leq n$ and $0 \leq l < h \leq m$. The two collections of $n \times m$ binary matrices, (C_0, C_1), constitute a visual cryptography scheme (k, n)-VCS if the following properties are satisfied:*

1. *(Contrast) For any $s \in C_0$, the OR of any k out of n rows of s is a vector v that satisfies $w(v) \leq l$, where $w(v)$ is the Hamming weight of v.*
2. *(Contrast) For any $s \in C_1$, the OR of any k out of n rows of s is a vector v that satisfies $w(v) \geq h$.*
3. *(Security) For any $i_1 < i_2 < \cdots < i_t$ in $\{1, 2, \cdots, n\}$ with $t < k$, the two collections of $t \times m$ matrices D_j, $j = 0, 1$, obtained by restricting each $n \times m$ matrix in C_j, $j = 0, 1$, to rows $i_1, i_2, \cdots i_t$, are indistinguishable in the sense that they contain the same matrices with the same frequencies.*

Note: in the above definition,

1. m is called the pixel expansion of the scheme. A pixel of the original secret image is represented by m subpixels in the recovered secret image. h is called the whiteness level and l is called the darkness level.
2. Define the value $\alpha = \frac{h-l}{m}$ to be the contrast of the scheme. Usually, the visual quality of the recovered secret image is better for larger contrast. Note, however, that there are other definitions of the contrast of VCS. We use this definition to establish our result. Proves of our results of the paper will be similar for other definitions of contrast.

Definition 2 (Probabilistic VCS [26, 24, 23]). *Let k, n and m' be nonnegative integers, $\bar{l}$ and $\bar{h}$ be positive numbers, satisfying $2 \leq k \leq n$ and $0 \leq \bar{l} < \bar{h} \leq m'$. The two collections of $n \times m'$ binary matrices (C_0, C_1) constitute a probabilistic Visual Cryptography Scheme, (k, n)-PVCS, if the following properties are satisfied:*

1. *(Contrast) For the collection C_0 and a share matrix $s \in C_0$, by v a vector resulting from the OR of any k out of the n rows of s. If $\overline{w(v)}$ denotes the average of the Hamming weights of v, over all the share matrices in C_0, then $\overline{w(v)} \leq \bar{l}$*
2. *(Contrast) For the collection C_1, the value of $\overline{w(v)}$ satisfies $\overline{w(v)} \geq \bar{h}$.*
3. *(Security) For any $i_1 < i_2 < \cdots < i_t$ in $\{1, 2, \cdots, n\}$ with $t < k$, the two collections of $t \times m'$ matrices D_j, $j = 0, 1$, obtained by restricting each $n \times m'$ matrix in C_j, $j = 0, 1$, to rows $i_1, i_2, \cdots i_t$, are indistinguishable in the sense that they contain the same matrices with the same frequencies.*

The definition of PVCS in [24] only considers the case with $n \times 1$ share matrices, we extend this definition to the $n \times m'$ case. And the definition of PVCS in [23] used the factor β to reflect the contrast, we use the values $\bar{l}$ and $\bar{h}$ to reflect the contrast. The common point of the three definitions of PVCS is that, for a particular pixel in the original secret image, the qualified participants can only correctly represent it in the recovered secret image with a certain probability. Because the human eyes always average the high frequency black and white dots into grey areas, so the average value of the Hamming weight of the black dots in the area reflects the greyness of the area. The PVCS does not require the satisfaction of the difference in greyness for each pixel in the recovered secret image as the DVCS does. It only reflects the difference in greyness in the overall view.

The contrast of the DVCS is fulfilled for each pixel (consisting of m subpixels) in the recovered secret image, however, this is quite different in the PVCS. The application of the *average contrast*, denoted by $\bar{\alpha}$, first appeared in [27]. This term is often used in the PVCS, see [23, 24, 22, 28], where the traditional contrast of the PVCS does not exist. Here we define the *average contrast* to be the average value of the overall contrast of the recovered secret image, i.e. the mean value of the contrast of all the pixels in the recovered secret image. According to our definition of the contrast $\alpha = \frac{h-l}{m}$, the average contrast can be calculated by the formula $\bar{\alpha} = \frac{\bar{h}-\bar{l}}{m'}$, where $\bar{h}$ and $\bar{l}$ are the mean values of $w(v)$ for the black and white pixels in the overall recovered secret image respectively, and m' is the pixel expansion of the PVCS. Because the number of pixels is large in the recovered secret image, the values $\bar{h}$ and $\bar{l}$ are equivalent to the mean values of the $w(v)$ in the collections C_1 and C_0 respectively. Note that, the DVCS also has the average contrast, and many proposed DVCS's in the literature have $\bar{\alpha} = \alpha$, see examples in [1, 2, 12] etc. When comparing DVCS that has $\bar{\alpha} = \alpha$ then, in the overall view, the visual quality of the recovered secret image of the PVCS is the same as the visual quality of the recovered secret image of a DVCS. However, because of the probabilistic nature, a PVCS is disadvantaged in displaying the details of the original secret image, especially for the white background areas in the recovered secret image.

A simple construction of PVCS based on a given DVCS (we will call it the original DVCS hereafter) can be as follows:

Construction 1 (Construction of PVCS based on an original DVCS [23]) *Denote (C_0, C_1) as the share matrix collections of a (k, n)-DVCS with pixel expansion m. The $n \times m'$ share matrix collections of a (k, n)-PVCS, denoted by (C_0', C_1'), can be generated by restricting each share matrix in C_0 and C_1 to its first m' columns respectively.*

According to the Construction 1, we have the following lemma:

Lemma 1. *The Construction 1 generate a (k, n)-PVCS based on an original (k, n)-DVCS, where the average contrast of (k, n)-PVCS equals to the contrast of (k, n)-DVCS, i.e. $\bar{\alpha} = \alpha$.*

Proof: First, for the contrast condition, let h and l be the whiteness level and darkness level of the (k, n)-DVCS respectively, and let $\bar{h}$ and $\bar{l}$ be the average whiteness level and average darkness level of the (k, n)-PVCS respectively. According to Construction 1, it is easy to verify that $\bar{h} = \frac{m'}{m}h$ and $\bar{l} = \frac{m'}{m}l$. Hence, we have that $\bar{\alpha} = \frac{\bar{h}-\bar{l}}{m'} = \frac{\frac{m'}{m}h - \frac{m'}{m}l}{m'} = \frac{h-l}{m} = \alpha > 0$, i.e. $\bar{h} > \bar{l}$.

Second, for the security condition, note that according to the security of DVCS, the two collections of matrices D_0 and D_1, obtained by restricting each $n \times m$ matrix in C_0 and C_1 to any less than k rows, contain the same matrices with the same frequencies. It is clear that, the two collections of matrices D_0' and D_1', obtained by restricting the above D_0 and D_1 to $m'(\leq m)$ columns, contain the same matrices with the same frequencies. □

A similar discussion about the average contrast and security properties of Construction 1 can be found in [23].

3 The Fountain Algorithm

The main idea of our scheme can be reflected by Figure 2. Imagine a pool with several water nozzles as depicted in Figure 2. The nozzles spray water with the same speed. In such a case the water will fill up the pool. Think of a blank image as a pool which has no distortion to the shape of original secret image(only differs in the size), think of the secret pixels of the original secret image as water injection nozzles that are evenly distributed in the pool, think of the subpixels of each secret pixel as water drops. As a result, the pool will be filled up by subpixels of secret pixels, and hence becomes a share image. Note that, each water nozzle sprays water with the same speed, hence, each nozzle will spray almost the same number of subpixels into the pool. We do the same process to all the share images, we get a VCS with no distortion. Certainly, the stacking of the share images will recover the secret image visually.

For the case of Figure 2, the size of the secret image is 6×6, where each secret pixel is a water nozzle. The size of the share image can be flexible and its size equals to the size of the pool. The water nozzles (secret pixels) spray water (subpixels) and fill up the pool (secret image). Clearly, the generated share images will have no distortion with the secret image.

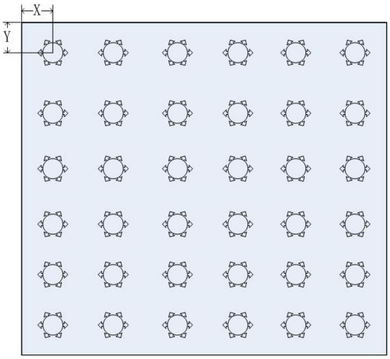


Fig. 2. A pool with 36 water injection nozzles

Formally, we give the following construction:

Construction 2

Input: *The original secret image S_I, overall pixel expansion (pool expansion) m_N, an original DVCS with pixel expansion m_o.*

Output: *The non-distortion share images $S_1, S_2, \cdots, S_n$.*

Step 1. *Generate a blank image (pool), M, that is m_N times of the size of the original secret image and has no distortion, i.e. the length (resp. width) of M is $\sqrt{m_N}$ times of that of S_I. Generate n blank share images $S_1, S_2, \cdots, S_n$, which have the same size of M.*

Step 2. *For a secret pixel at position (p, q) in the original secret image, initialize an empty list $L_{p,q}$ which is used to store the positions of subpixels in M (or $S_1, S_2, \cdots, S_n$).*

Step 3. *Distribute the secret pixels (water injection nozzles) of the original secret image evenly into the blank image M. Note that the corresponding coordinates of a pixel (p, q) of the original secret image is (p', q') in M now.*

Step 4. *For each subpixel in the blank image M, find the nearest secret pixel (water injection nozzle), suppose the position of the secret pixel is (p', q'). Add the position of the subpixel to list $L_{p,q}$.*

Step 5. *Sort each list $L_{p,q}$ with ascending order with respect to the distance to the secret pixel (water injection nozzle) (p', q').*

Step 6. *Denote $|L_{p,q}|$ as the number of positions in $L_{p,q}$. Encrypt the secret pixel (p, q) by applying the original DVCS in order, by $\lceil \frac{|L_{p,q}|}{m_o} \rceil$ times and distribute the subpixels of the shares in order, to the positions of $L_{p,q}$ in $S_1, S_2, \cdots, S_n$ respectively, while discarding the redundant subpixels.*

In the above construction, the new position (p', q') of a pixel at position (p, q) in the original secret image can be calculated as follows:

$$p' = p\sqrt{m_N} + X \quad and \quad q' = q\sqrt{m_N} + Y$$

where X and Y are shown in Figure 2.

Denote the length (resp. width) of the secret image as e (resp. f), then the length (resp. width) of the pool will be $e\sqrt{m_N}$ (resp. $f\sqrt{m_N}$). If $e\sqrt{m_N}$ (resp. $f\sqrt{m_N}$) is not an integer, then we will use $\lceil e\sqrt{m_N} \rceil$ (resp. $\lceil f\sqrt{m_N} \rceil$) instead.

By saying "applying the original DVCS in order", we mean applying the DVCS by several times and concatenate the output shares (subpixels) in order, for each participants respectively.

Note that the overall pixel expansion, m_N, of our scheme is not necessarily equals to the pixel expansion of the original DVCS m_o, and it can be any value larger than 0.

In order to make things clear, we give an example for the $(2, 2)$-VCS, where the share matrix collections are as follows.

$$C_0 = \left\{ \begin{bmatrix} 10 \\ 10 \end{bmatrix}, \begin{bmatrix} 01 \\ 01 \end{bmatrix} \right\} \text{ and } C_1 = \left\{ \begin{bmatrix} 10 \\ 01 \end{bmatrix}, \begin{bmatrix} 01 \\ 10 \end{bmatrix} \right\}$$

Example 1. The recovered secret images of the proposed scheme (Construction 2) can be found in Figure 3.

As depicted in Figure 3, by comparing the three recovered secret images (b), (c) and (d), we can observe that, larger pixel expansion will result in better visual

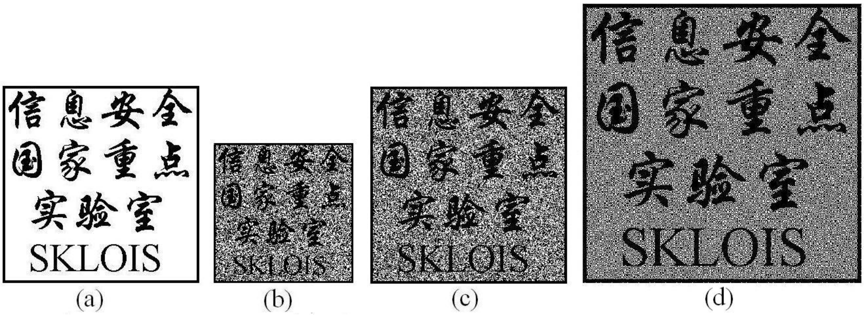


Fig. 3. (a) is the original secret image with size 300×300, (b) is the recovered secret image with overall pixel expansion $m_N = 0.5$ and image size 213×213, (c) is the recovered secret image with overall pixel expansion $m_N = 1$ and image size 300×300, (d) is the recovered secret image with overall pixel expansion $m_N = 2$ and image size 425×425

quality, and smaller pixel expansion will compromise poorer visual quality. Our scheme is flexible with respect to the compromise between the visual quality and overall pixel expansion of the recovered secret image.

Formally, we give the following Theorem 1 for Construction 2.

Theorem 1. *The Construction 2 generates a PVCS with no distortion and the size of its share images and recovered secret image can be flexible.*

Proof: It is clear that the final share images and recovered secret image have no distortion with respect to the original secret image since the nuzzles are evenly distributed in the pool. Besides, the size of the share images and recovered secret image can be flexible since the size of the pool is flexible. We only need to prove that Construction 2 satisfies the contrast condition and the security condition of PVCS.

First, for the contrast condition, according to Construction 2, we have that the shares of, for each secret pixel, the proposed VCS are formed by concatenating the shares of $\lfloor\frac{|L_{p,q}|}{m_o}\rfloor$ original DVCS and $(\lceil\frac{|L_{p,q}|}{m_o}\rceil - \lfloor\frac{|L_{p,q}|}{m_o}\rfloor)$ original PVCS with pixel expansion $m' = |L_{p,q}| - \lfloor\frac{|L_{p,q}|}{m_o}\rfloor m_o$. Let h and l be the whiteness level and darkness level of the original DVCS respectively, and let $\bar{h}$ and $\bar{l}$ be the average whiteness level and average darkness level of the original PVCS respectively. According to the contrast condition of Definition 1, Definition 2 and Lemma 1, it is easy to verify that $\bar{h} = \frac{m'}{m_o}h$ and $\bar{l} = \frac{m'}{m_o}l$ hold. Denote h_N, l_N and α_N as the average whiteness level, average darkness level and average contrast of the proposed VCS respectively. Then we have $h_N = \lfloor\frac{|L_{p,q}|}{m_o}\rfloor h + (\lceil\frac{|L_{p,q}|}{m_o}\rceil - \lfloor\frac{|L_{p,q}|}{m_o}\rfloor)\bar{h}$ and $l_N = \lfloor\frac{|L_{p,q}|}{m_o}\rfloor l + (\lceil\frac{|L_{p,q}|}{m_o}\rceil - \lfloor\frac{|L_{p,q}|}{m_o}\rfloor)\bar{l}$. The overall pixel expansion is $m_N = \lfloor\frac{|L_{p,q}|}{m_o}\rfloor m_o + (\lceil\frac{|L_{p,q}|}{m_o}\rceil - \lfloor\frac{|L_{p,q}|}{m_o}\rfloor)m'$.

Hence, the average contrast satisfies

$$\begin{aligned}\alpha_N &= \frac{h_N - l_N}{m_N}\\ &= \frac{\lfloor\frac{|L_{p,q}|}{m_o}\rfloor(h-l) + (\lceil\frac{|L_{p,q}|}{m_o}\rceil - \lfloor\frac{|L_{p,q}|}{m_o}\rfloor)(\bar{h}-\bar{l})}{\lfloor\frac{|L_{p,q}|}{m_o}\rfloor m_o + (\lceil\frac{|L_{p,q}|}{m_o}\rceil - \lfloor\frac{|L_{p,q}|}{m_o}\rfloor)m'}\\ &= \frac{\lfloor\frac{|L_{p,q}|}{m_o}\rfloor(h-l) + (\lceil\frac{|L_{p,q}|}{m_o}\rceil - \lfloor\frac{|L_{p,q}|}{m_o}\rfloor)\frac{m'}{m_o}(h-l)}{\lfloor\frac{|L_{p,q}|}{m_o}\rfloor m_o + (\lceil\frac{|L_{p,q}|}{m_o}\rceil - \lfloor\frac{|L_{p,q}|}{m_o}\rfloor)m'}\\ &= \frac{h-l}{m_o}\\ &= \alpha\end{aligned}$$

We have that the proposed scheme satisfies the contrast condition of PVCS.

Second, for the security condition, according to Construction 2, we have that the shares of the proposed VCS are formed by concatenating the shares of $\lfloor\frac{|L_{p,q}|}{m_o}\rfloor$ original DVCS and $(\lceil\frac{|L_{p,q}|}{m_o}\rceil - \lfloor\frac{|L_{p,q}|}{m_o}\rfloor)$ original PVCS. Hence, the security of the proposed scheme follows from the security of the original DVCS and PVCS. □

4 Improvements on the Visual Quality

Recall that, the probabilistic subpixels degrade the visual quality of the recovered secret image, especially for the details in the recovered secret image. In this section, we improve the visual qualities of the recovered images of Construction 2. The main idea of our improvement is to remove the probabilistic subpixels in the pool.

Suppose that the pixel expansion of the original DVCS is m_o and the pool expansion is m_N. When the pool expansion m_N is not a multiple of the pixel expansion m_o, the pool expansion subpixels can be divided into two parts: The multiple part and the remaining part. Denote $d = \lfloor \frac{m_N}{m_o} \rfloor$, $m_N = d \times m_o + t$, $0 < t < m_o$, the multiple part contains $d \times m_o$ subpixels and the remaining part contains t(resp. $0 < t < m_o$) subpixels. The multiple part can be filled by repeating the original DVCS for d times. The remaining part can be filled by choosing t columns from the basis matrices(resp. the remaining part is filled by a PVCS with pixel expansion t). So when m_N is not a multiple of m_o, pool expansion subpixels will be filled by $d \times m_o$ subpixels from the original DVCS and t subpixels from a PVCS. The probabilistic subpixels will add some visual-noise to the recovered image, which will blur the details in the recovered image. Thus the visual quality of the recovered image will be degraded. So we would like to remove the PVCS part. Our strategy is: The remaining part is assigned by m_o subpixels with probability $\frac{t}{m_o}$ or assigned by no subpixels with probability $\frac{m_o - t}{m_o}$. On average, the remaining part is assigned by t subpixels. From an overall view, a pixel of the original secret image(a water nozzle) is assigned by $\lfloor \frac{m_N}{m_o} \rfloor \times m_o$ subpixels with probability $\frac{m_o - t}{m_o}$, and is assigned $\lceil \frac{m_N}{m_o} \rceil \times m_o$ subpixels with probability $\frac{t}{m_o}$. Suppose there is a Boolean matrix the same size as the original secret image, then there is a one-to-one mapping between a secret pixel and an entry in the Boolean matrix. If the secret pixel is assigned by $\lfloor \frac{m_N}{m_o} \rfloor \times m_o$ subpixels, we denote the corresponding entry as 0, else if the secret pixel is assigned by $\lceil \frac{m_N}{m_o} \rceil \times m_o$ subpixels, we denote the corresponding entry as 1. Then we will get a Boolean matrix for which $\frac{t}{m_o}$ proportion of its entries are 1, and the entries of 1 are evenly distributed. Meanwhile the entries of 0 are evenly distributed in the Boolean matrix too. For example, for a (2,2)-DVCS with pixel expansion 2. Suppose the pool is three times as large as the original secret image. We distribute two subpixels for 50% water nozzles and four subpixels for the remaining 50% water nozzles, where there will be three subpixels for each water nozzle on average. And the two cases(two subpixels for a water nozzle, four subpixels for a water nozzle) are evenly distributed in the pool.

Formally, we give the following construction:

Construction 3

Input: *The original secret image S_I, overall pixel expansion m_N, an original DVCS with pixel expansion m_o.*

Output: *The non-distortion shares $S_1, S_2, \cdots, S_n$.*

Preprocess. *Let $s = \lfloor \frac{m_N}{m_o} \rfloor$, $t = \lceil \frac{m_N}{m_o} \rceil$ where s and t satisfy $s \times m_o \leq m_N \leq t \times m_o$. Let a and b be two non-negative real numbers satisfying $a + b = 1$ and $a \times (s \times m_o) + b \times (t \times m_o) = m_N$. Suppose the size of S_I is $m \times n$. Then* *we generate an $m \times n$ random Boolean matrix D, in which 0 appears with probability a and 1 appears with probability b. Then there is a one-to-one mapping between the pixels of the original secret image and the entries of D. If the entry in D is 0, we distribute $s \times m_o$ subpixels for the corresponding pixel of the original secret image. If the entry in D is 1, we distribute $t \times m_o$ subpixels for the corresponding pixel of the original secret image.*

Step 1-3. *The same as that of Construction 2.*

Step 4. *For each secret pixel (water injection nozzle) in the blank image M, if the entry of D is 0, find $s \times m_o$ nearest and undistributed subpixels, else if the entry of D is 1, find $t \times m_o$ nearest and undistributed subpixels. Suppose the position of the secret pixel is (p', q'). Add the positions of the subpixels to list $L_{p,q}$.*

Step 5. *Encrypt the secret pixel (p, q) by applying the original DVCS in order, by s or t times and distribute the subpixels of the shares in order, to the positions of $L_{p,q}$ in $S_1, S_2, \cdots, S_n$ respectively. The undistributed subpixels in the pool are simply set to black.*

In the above construction, if the pool expansion m_N is a multiple of the pixel expansion m_o, then every water nozzle will be assigned by m_N subpixels. If the pool expansion m_N is smaller than the pixel expansion of the original DVCS m_o, then each water nozzle will be assigned by m_o subpixels with probability $\frac{m_N}{m_o}$ or assigned by no subpixels with probability $\frac{m_o - m_N}{m_o}$, which implies that $\frac{m_o - m_N}{m_o}$ of the secret pixels in the original secret image are lost in the recovered secret image on average.

In the following, we give a comparison for Construction 2 and Construction 3 for $(2, 2)$-VCS, where the original DVCS is the same as that of Example 1.

Example 2. Suppose that the pool is 1.37311 (this value can be arbitrarily chosen) times as large as that of the original secret image. Thus the length (resp. width) of the pool is 1.1718 times the length (resp. width) of the original secret image. The parameters in the stage of Preprocess of Construction 3 are m_N=1.37311, m_0=2, s=0 and t=1. In Construction 2, we assign one or two subpixel for each secret pixel (water injection nozzle), for which about 37.311% secret pixels are assigned with two subpixels (filled by a $(2, 2)$-DVCS) and about 62.689% secret pixels are assigned with one subpixel (filled by a $(2, 2)$-PVCS with pixel expansion 1). In Construction 3, we assign two subpixels for 68.6555% secret pixels(water injection nozzles) and assign no subpixel for 31.3445% secret pixels (water injection nozzles).

We make use of two types of secret images: Characters and Human face. The original secret images are in the first column. The visual quality of Construction 2 can be found in the second column of Figures 4 and 5. The visual quality of Construction 3 can be found in the third column of Figures 4 and 5.

As depicted in Figures 4 and 5, by comparing the recovered secret images (generated by Construction 2) and that of Construction 3, we can observe that, the recovered secret images for both constructions are clear and one can easily identify the contents of the original secret image. One also can observe that Construction 3 results in better visual quality than Construction 2 with respect

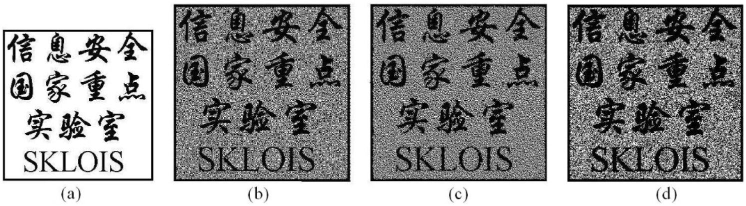


Fig. 4. (a) is the original secret image Characters with image size 300×300. (b) and (c) are the recovered secret images of Construction 2 and Construction 3 with image size 352×352 respectively. (d) is the recovered secret image of Yang et al.'s VCS proposed in [25] with image size 352×352.

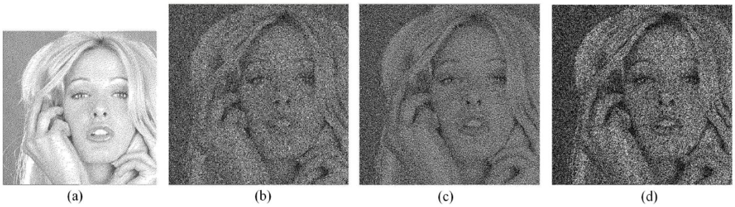


Fig. 5. (a) is the original secret image Human face with image size 512×512. (b) and (c) are the recovered secret images of Construction 2 and Construction 3 with image size 600×600 respectively. (d) is the recovered secret image of Yang et al.'s VCS proposed in [25] with image size 600×600.

to the evenness. Particularly, the recovered secret image is much more even at the white background areas.

Because of the page limit, we cannot provide more experimental results. If more experimental results for different pixel expansion are given, one also can observe that, larger pixel expansion will result in better visual quality, and smaller pixel expansion will compromise poorer visual quality, i.e. Construction 3 is also flexible with respect to the compromise between the visual quality and overall pixel expansion of the recovered secret image.

5 Comparisons with Some Well-Known VCS's

In this section, we give some comparisons with some well-known schemes that also consider non-distortion.

5.1 Comparison with Hou et al.'s Scheme [21]

Hou et al. proposed a multi-pixel encryption visual cryptography scheme (MPEVCS) [21], where multiple secret pixels are encrypted at a time. For example, for a (2,2)-MPEVCS, two secret pixels are encrypted at a time, and the original DVCS is the same as that of Example 1. To share two white secret pixels, the dealer randomly chooses a share matrix from C_0. To share two black secret pixels, the dealer randomly chooses a share matrix from C_1. To share a white and a black pixel, the dealer chooses a share matrix from C_0 and C_1 in turn.

Unfortunately, Hou et al.'s MPEVCS may result in some obvious errors in the recovered secret image. An example of (2,2)-VCS for comparing MPEVCS and the proposed schemes can be found in Figure 6.

For the secret image of Figure 6(a), the recovered secret image of (2,2)-MPEVCS will be Figure 6(b). One can clearly observe that, one of the thin lines is missing. Meanwhile, one can clearly identify both the thin lines in the recovered secret image of Construction 2 and Construction 3 (Figures 6(c) and (d)).

According to the above discussion, the MPEVCS is not suitable for encrypting secret images consisting of thin lines, such as maps and geometry figures, while Construction 2 and Construction 3 are both competent.

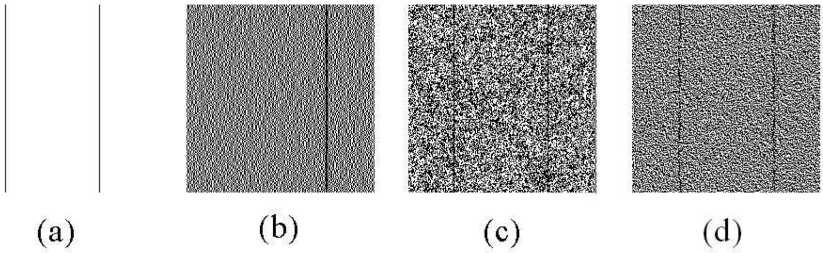


Fig. 6. (a) is the original secret image with image size 200×200. (b) is the recovered secret image of Hou et al.'s MPEVCS [21] with image size 200×200. (c) and (d) are the recovered secret images of Construction 2 and Construction 3 with image size 200×200.

5.2 Comparison with Yang et al.'s Scheme [25]

Yang et al. proposed a non-distortion VCS in [25]. Their scheme is also flexible. Unfortunately, the overall pixel expansion of their scheme can only range from 1 to m_o (the pixel expansion of the original DVCS), i.e. their scheme is not real flexible. The proposed Construction 2 and Construction 3 both can have arbitrary overall pixel expansion even smaller than 1. This means that our scheme is more general than Yang et al.'s scheme with respect to the overall pixel expansion.

Yang et al. also tried to improve the visual quality of the recovered secret image. They divided the secret pixels in the original secret image into two categories: More important secret pixels (the edge information of the original secret image) and less important secret pixels (the remaining secret pixels other than the edges). They assign more subpixels to the more important secret pixels and assign less subpixels to the less important secret pixels. However, it should be pointed out that they still employ probabilistic subpixels, hence, the visual quality of the recovered secret image will be degraded compared with that of Construction 3.

Experimental results show that our scheme has competitive visual quality with that of Yang et al.'s scheme in [25]. According to Figures 4 and 5, one can easily observe that the recovered secret images of Construction 3 is more even than that of Yang et al.'s scheme, especially for the white background areas. The advantage of Yang et al.'s scheme is that, the edges in the recovered secret image are enhanced.

5.3 Comparison with Yang et al.'s Scheme [20, 19]

Yang et al. proposed several non-distortion VCS that are not flexible with respect to the pixel expansion, such as [20, 19]. Take Yang et al.'s scheme proposed in [20] as an example, the scheme depends on designing proper mapping patterns. The mapping patterns indicate the positions of the subpixels in the shares for a block of secret pixels. The mapping patterns should be different for different secret image sizes and different pixel expansions. For example, a $M_{4,7}$ mapping pattern maps the subpixels of a block of 4×4 secret pixels to a pattern of 7×7 block for a (2,3)-VCS with pixel expansion 3. To retain the aspect ratio, dummy subpixels are added. In the best case, one dummy subpixel is added. Furthermore, the length and width of the secret image should be multiple of 4, otherwise, extra dummy subpixels are required. Note that the dummy subpixels will degrade the visual quality and increase the pixel expansion of their scheme. Their method is complicated, how to design a mapping pattern that reduces the number of dummy subpixels to the minimum is, as they said, a huge challenge, especially for some pixel expansions and secret image sizes. This may be a bottleneck for their methods to be used practically.

In contrast, our scheme is more general and simpler. It does not add any extra subpixels and can be applied to any VCS that is under the pixel by pixel encryption model, such as VCS for general access structure, color VCS and extended VCS, and our VCS does not restrict to any specific underlying operations.

Our method can get competitive visual quality with Yang et al.'s schemes proposed in [20, 19]. In order to make things clear, we give an example for $(2,2)$-VCS and $(2,3)$-VCS, where the share matrix collections of the $(2,2)$-VCS are the same as that in Example 1, and the share matrix collections of the $(2,3)$-VCS are as follows:

$$C_0 = \left\{ \begin{bmatrix} 110 \\ 110 \\ 110 \end{bmatrix}, \begin{bmatrix} 101 \\ 101 \\ 101 \end{bmatrix}, \begin{bmatrix} 011 \\ 011 \\ 011 \end{bmatrix} \right\} \text{ and}$$

$$C_1 = \left\{ \begin{bmatrix} 110 \\ 101 \\ 011 \end{bmatrix}, \begin{bmatrix} 110 \\ 011 \\ 101 \end{bmatrix}, \begin{bmatrix} 101 \\ 110 \\ 011 \end{bmatrix}, \begin{bmatrix} 101 \\ 011 \\ 110 \end{bmatrix}, \begin{bmatrix} 011 \\ 110 \\ 101 \end{bmatrix}, \begin{bmatrix} 011 \\ 101 \\ 110 \end{bmatrix} \right\}$$

For Yang et al.'s schemes proposed in [20, 19], we make use of the mapping patterns of Fig. 3(b) in [19] and Fig. 8(a) in [20] for $(2,2)$-VCS and $(2,3)$-VCS respectively. The original secret image is the same as that of Figure 4.

According to Figure 7, the image sizes of the recovered secret images for our method are slightly smaller than that of Yang et al.'s schemes. The reason is that,

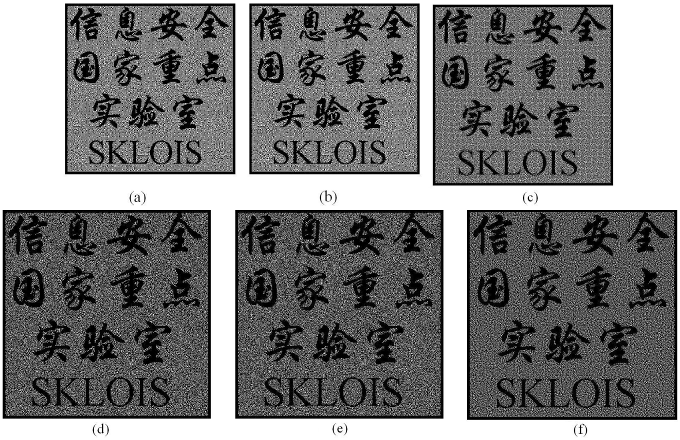


Fig. 7. (a) and (b) are the recovered secret images of Construction 2 and Construction 3 for (2,2)-VCS with image size 425×425 respectively, (c) is the recovered secret image of Yang et al.'s scheme proposed in [19] for a (2,2)-VCS with image size 450×450, (d) and (e) are the recovered secret images of Construction 2 and Construction 3 for (2,3)-VCS with image size 520×520, (f) is the recovered secret image of Yang et al.'s scheme proposed in [20] for a (2,3)-VCS with image size 525×525

the scheme in [20, 19] contains a dummy subpixel in each mapping pattern of their two schemes respectively, while the proposed schemes do not. We set the dummy subpixel of Yang et al.'s scheme to be black, otherwise there will be noise like pixels appear in the recovered secret images. According to Figure 7, we can observe that the proposed schemes and Yang et al.'s schemes have competitive visual quality for the recovered secret images. First, the images (a) and (b) (resp. (d) and (e)) are lighter than (c) (resp. (f)) i.e. the average contrasts of (a) and (b) (resp. (d) and (e)) are larger than that of (c) (resp. (f)). Second, the image (c) (resp. (f)) is more even than (a) and (b) (resp. (d) and (e)).

5.4 Comparison on Effectiveness with Some Well-Known Non-distortion VCS

We compare the effectiveness of our scheme with that of some well-known non-distortion schemes and size invariant schemes in the literature (Note that, the size invariant schemes also have no distortion) in Table 1.

In Table 1, the word "Depends" means "No if m is square number, otherwise Yes". And the word "Yes*" means the ranges of the overall pixel expansion of the VCS are limited. For example, the overall pixel expansion of VCS's proposed in [23] and [25] can only range from 1 to m_o (the pixel expansion of the original DVCS), and the overall pixel expansion of VCS proposed in [29] should be a multiple of m_o. In one word, these VCS's are not real flexible.

According to Table 1, it is clear that only our schemes satisfy both the real flexible and non-distortion properties simultaneously.

Table 1. Comparisons on effectiveness with well-known non-distortion VCS's

Schemes Criteria	Construction 2	Construction 3	[20]	[19]	[21]	[22]
Flexible	Yes	Yes	No	No	No	No
Distortion	No	No	No	No	No	No
Schemes Criteria	[23]	[24]	[28]	[29]	[30]	[25]
Flexible	Yes*	No	No	Yes*	No	Yes*
Distortion	Depends	No	Depends	Depends	Depends	No

6 Conclusions

In this paper, we propose a visual cryptography schemes which simulated the principle of fountains. The proposed scheme has two properties: Non-distortion and flexible. The proposed scheme can be applied to any VCS that is under the pixel by pixel encryption model, such as VCS for general access structure, color VCS and extended VCS, and our VCS does not restrict to any specific underlying operations. We show that our scheme is flexible with respect to the compromise between the pixel expansion and visual quality of the recovered secret image. For larger pixel expansion, the recovered secret image of our scheme will have better visual quality, and smaller pixel expansion will compromise poorer visual quality.

We give comparisons with some well-known non-distortion VCS's [19, 20, 25] and size invariant schemes [21, 22, 23, 24]. The comparisons show that our scheme has many advantages on generality, simplicity, real flexible and effectiveness, besides, our scheme has competitive visual quality of the recovered secret image with that of many well-known non-distortion VCS's.

Acknowledgements. Many thanks to the anonymous reviewers for their valuable comments. This work was supported in part by China NSFC grant No. 60903210, China national 973 project No.2007CB807902 and was supported in part by the Testbed@TWISC, National Science Council under the Grant NSC 100-2219-E-006-001.

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] Droste, S.: New Results on Visual Cryptography. In: Koblitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 401–415. Springer, Heidelberg (1996)

[3] Blundo, C., De Santis, A., Stinson, D.R.: On the contrast in visual cryptography schemes. Journal of Cryptology 12(4), 261–289 (1999)

[4] Cimato, S., De Prisco, R., De Santis, A.: Optimal colored threshold visual cryptography schemes. Designs, Codes and Cryptography 35, 311–335 (2005)

[5] Krause, M., Simon, H.U.: Determining the optimal contrast for secret sharing schemes in visual cryptography. Combinatorics, Probability & Computing 12(3), 285–299 (2003)

[6] Viet, D.Q., Kurosawa, K.: Almost Ideal Contrast Visual Cryptography with Reversing. In: Okamoto, T. (ed.) CT-RSA 2004. LNCS, vol. 2964, pp. 353–365. Springer, Heidelberg (2004)

[7] Koga, H.: A General Formula of the (t,n)-Threshold Visual Secret Sharing Scheme. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 328–345. Springer, Heidelberg (2002)

[8] Bose, M., Mukerjee, R.: Optimal (k,n) visual cryptographic schemes for general k. Designs, Codes and Cryptography 55, 19–35 (2010)

[9] Liu, F., Wu, C.K.: Embedded meaningful share visual cryptography schemes. IEEE Transactions on Information Forensics & Security 6(2), 307–322 (2011)

[10] Liu, F., Wu, C.K., Lin, X.J.: Step construction of visual cryptography schemes. IEEE Transactions on Information Forensics & Security 5(1), 27–38 (2010)

[11] Liu, F., Wu, C.K., Lin, X.J.: A new definition of the contrast of visual cryptography scheme. Information Processing Letters 110, 241–246 (2010)

[12] Ateniese, G., Blundo, C., De Santis, A., Stinson, D.R.: Visual cryptography for general access structures. Information and Computation 129, 86–106 (1996)

[13] Ateniese, G., Blundo, C., De Santis, A., Stinson, D.R.: Extended capabilities for visual cryptography. ACM Theoretical Computer Science 250(1-2), 143–161 (2001)

[14] Zhou, Z., Arce, G.R., Di Crescenzo, G.: Halftone visual cryptography. In: Proceedings of 2003 International Conference on Image Processing, vol. 1, pp. I–521–I–524 (2003)

[15] Surekha, B., Swamy, G., Rao, K.S.: A multiple watermarking technique for images based on visual cryptography. Computer Applications 1, 77–81 (2010)

[16] Monoth, T., Anto P., B.: Tamperproof transmission of fingerprints using visual cryptography schemes. Procedia Computer Science 2, 143–148 (2010)

[17] Weir, J., Yan, W.: Resolution variant visual cryptography for street view of google maps. In: Proceedings of the ISCAS, pp. 1695–1698 (2010)

[18] Cimato, S., Yang, C.N.: Visual cryptography and secret image sharing. CRC Press, Taylor & Francis (2011)

[19] Yang, C.N., Chen, T.S.: Aspect ratio invariant visual secret sharing schemes with minimum pixel expansion. Pattern Recognition Letters 26, 193–206 (2005)

[20] Yang, C.N., Chen, T.S.: Reduce shadowsize in aspect ratio invariant visual secret sharing schemes using a square block-wise operation. Pattern Recognition 39, 1300–1314 (2006)

[21] Hou, Y.C., Tu, C.F.: Visual cryptography techniques for color images without pixel expansion. Journal of Information, Technology and Society 1, 95–110 (2004) (in Chinese)

[22] Ito, R., Kuwakado, H., Tanaka, H.: Image size invariant visual cryptography. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Science E82-A(10), 2172–2177 (1999)

[23] Cimato, S., De Prisco, R., De Santis, A.: Probabilistic visual cryptography schemes. The Computer Journal 49(1), 97–107 (2006)

[24] Yang, C.N.: New visual secret sharing schemes using probabilistic method. Pattern Recognition Letters 25, 481–494 (2004)

[25] Yang, C.N., Chen, T.S.: Visual secret sharing scheme: prioritizing the secret pixels with different pixel expansions to enhance the image contrast. Optical Engineering 46(9), 097005 (2007)

[26] Liu, F., Wu, C.K., Lin, X.J.: The alignment problem of visual cryptography schemes. Designs, Codes and Cryptography 50, 215–227 (2009)

[27] Biham, E., Itzkovitz, A.: Visual cryptography with polarization. In: The Dagstuhl seminar on Cryptography, and in the RUMP session of CRYPTO 1998 (September 1997)

[28] Kuwakado, H., Tanaka, H.: Size-reduced visual secret sharing scheme. IEICE Transactions on Fundamentals E87-A(5), 1193–1197 (2004)

[29] Yang, C.N., Chen, T.S.: Size-adjustable visual secret sharing schemes. IEICE Transactions on Fundamentals E88-A(9), 2471–2474 (2005)

[30] Yang, C.N., Chen, T.S.: New size-reduced visual secret sharing schemes with half reduction of shadow size. IEICE Transactions on Fundamentals E89-A(2), 620–625 (2006)

An Extended Visual Cryptography Scheme for Continuous-Tone Images

Yasushi Yamaguchi[1,2]

[1] The University of Tokyo, Graduate School of Arts and Sciences
[2] Japan Science and Technology Agency, CREST,
3-8-1 Komaba, Meguro-ku, Tokyo 153-8902, Japan
yama@graco.c.u-tokyo.ac.jp
http://www.graco.c.u-tokyo.ac.jp/yama-lab/

Abstract. Visual cryptography is a kind of cryptography that can be decoded directly by the human visual system when transparencies are stacked. Extended visual cryptography allows to print meaningful images on transparencies which can conceal the existence of "secret" in the transparencies. A lot of studies have tried to incorporate continous-tone images into extended visual cryptography. However, most of them suffer from deterioration of the resulting images. This paper proposes a new scheme for extended visual cryptography for continuous-tone images. It mainly consists of two techniques, namely parallel error diffusion and optimum tone mapping and can quickly encrypted images with no pixel expansion and high contrast. Some experimental results are shown to examine its effectiveness as well as limitation.

Keywords: extended visual cryptography, halftoning, error diffusion, tone mapping.

1 Introduction

Visual cryptography is a kind of cryptography that can be decoded directly by the human visual system without any computation for decryption. It usually prints certain images on transparencies and the secret image is reconstructed by simply stacking the transparencies together. Extended visual cryptography allows to print meaningful images on transparencies so that it can conceal the very existence of "secret" in the transparencies. There have been a lot of studies to incorporate continous-tone images into extended visual cryptography. However, most of them suffer from deterioration of the resulting images caused by pixel expansion and low contrast. This paper proposes a new scheme for extended visual cryptography for contunuous-tone images. It mainly consists of two techniques, namely parallel error diffusion and optimum tone mapping, and can encrypt images with no pixel expansion and high contrast in an instant.

Section 2 illustrates the fundamentals of conventional extended visual secret sharing scheme. The issues of extended visual secret sharing schemes to incorporate continuous-tone images are discussed in Section 3. Our proposed scheme

Y.Q. Shi, H.J. Kim, and F. Perez-Gonzalez (Eds.): IWDW 2011, LNCS 7128, pp. 228–242, 2012.

for continuous-tone images which consists of parallel error diffusion and optimum tone mapping is introduced in Section 4. Section 5 shows the experimental results of our proposed method. The related works of the proposed method are explained and makes comparisons in Section 6. Section 7 concludes this study.

2 Extended Visual Cryptography Scheme

Naor and Shamir proposed (k, n) *Visual Secret Sharing Scheme* (VSSS) in 1994 [10]. This scheme generates n transparencies from an original *secret image*. The transparencies are usually shared by n participants so that each participant is expected to keep one transparency. The secret image can be observed if any k or more of them are stacked together. However, the secret image is totally invisible if fewer than k transparencies are stacked. The images on transparencies are called *shadow image*s. All the shadow images consist of uniformly random pattern of black and white subpixels. Naor and Shamir pointed out an extension of this scheme for concealing the very existence of the secret image.

Ateniese et al. extended the VSSS in the sense of a *General Access Structure* (GAS) [1] and extended capability [2]. A General Access Structure allows to control the qualified set of transparencies with which one can recover the secret image, while any k or more transparencies can reconstruct the secret image in (k, n) VSSS. An extended capability is able to introduce a meaningful image as a shadow image. In the *Extended Visual Cryptography Scheme* (EVCS), for an *access structure* $(\Gamma_{\mathsf{Qual}}, \Gamma_{\mathsf{Forb}})$ on a set of n participants, the shared (secret) image can be recovered by any *qualified set* $X \in \Gamma_{\mathsf{Qual}}$ with no trace of the shadow images, but any *forbidden set* $X \in \Gamma_{\mathsf{Forb}}$ has no information on the secret image. Moreover, the shadow images are meaningful so that each participant can recognize the image on ones transparency.

An EVCS can be constructed in a pixel-wise manner. An original secret pixel will be transformed to n patterns of pixels for shadow images. These pixels on shadow images are called *share*s. A share consists of m black and white subpixels. Human visual system observes the average of subpixels, because they exist in close proximity. This structure is usually described by an $n \times m$ Boolean matrix $M = [m_{ij}]$. Here $m_{ij} = 0$ or 1 if the jth subpixel in the ith shadow is black or white, respectively. [1] If a set of transparencies X are stacked in a way that properly aligns the subpixels, each combined share can be represented by the Boolean "AND" of the corresponding set of rows X in the Boolean matrix M. Let M_X denote the m-D vector obtained by taking the Boolean "AND" of a set of row vectors X. The gray level of a pixel combined by the shares is obtained by the Hamming weight $H(M_X)$ of the "AND"ed m-D vector M_X. A human observer interprets this gray level as white if $H(M_X) \geq t_X$ and as black if $H(M_X) < t_X - \alpha_{S/R}m$. Here, $t_X \in \{1, \cdots, m\}$ is called *threshold*, while the

[1] A black (white) pixel is usually represented by 1 (0) in most of visual cryptography studies. This paper uses 0 (1) for a black (white) pixel, because it would be suitable for discussing brightness of a pixel and consequently image processing techniques.

value $\alpha_{S/R} > 0$ and the number $\alpha_{S/R}m \geq 1$ are called *relative difference* and *contrast*, respectively.

Since n participants share one secret image and have their own n shadow images, we have to consider $n+1$ colors, $c, c_1, \cdots, c_n \in \{b, w\}$ where b and w stands for black and white, respectively. The value c denotes the color of the secret image pixel and c_i denotes the color of the original image pixel for i-th participant's shadow image. In order to realize an EVCS which obtains a secret pixel of color c when transparencies associated to a set $X \in \Gamma_{\mathsf{Qual}}$, we need 2^n pairs of collections of $n \times m$ Boolean matrices, $(\mathcal{C}_b^{c_1 \cdots c_n}, \mathcal{C}_w^{c_1 \cdots c_n})$, one for each possible combination of black and white pixels in the n original images for the shadow images.

An EVCS for an access structure $(\Gamma_{\mathsf{Qual}}, \Gamma_{\mathsf{Forb}})$ for n participants is valid if it fulfills the following conditions.

1. For any $X \in \Gamma_{\mathsf{Qual}}$ and for any $c_1, \cdots, c_n \in \{b, w\}$, the threshold t_X and the relative difference α_R exist which satisfy $H(M_X) \leq t_X - \alpha_R m$ for any $M \in \mathcal{C}_b^{c_1 \cdots c_n}$ and $H(M_X) \geq t_X$ for any $M \in \mathcal{C}_w^{c_1 \cdots c_n}$. Here M_X denotes the m-D vector obtained by taking Boolean "AND" of the row vectors of M corresponding to the participants in X and $H(M_X)$ denotes the Hamming weight of the vector M_X.
2. For any $X = \{i_1, \cdots, i_q\} \in \Gamma_{\mathsf{Forb}}$ and for any $c_1, \cdots, c_n \in \{b, w\}$, the two collections of $q \times m$ matrices, $\mathcal{D}_b^{c_1 \cdots c_n}$ and $\mathcal{D}_w^{c_1 \cdots c_n}$, obtained by extracting rows $i_1, \cdots, i_q$ from each $n \times m$ matrix in $\mathcal{C}_b^{c_1 \cdots c_n}$ and $\mathcal{C}_w^{c_1 \cdots c_n}$, respectively, are indistinguishable so that the collections contain the same matrices with the same frequencies.
3. For any $i \in \{1, 2, \cdots, n\}$ and any $c_1, \cdots, c_{i-1}, c_{i+1}, \cdots, c_n \in \{b, w\}$, it results that
$$\min_{M \in \mathcal{M}_w} H(M_i) - \max_{M \in \mathcal{M}_b} H(M_i) \geq \alpha_S m,$$
where
$$\mathcal{M}_b = \mathcal{C}_b^{c_1 \cdots c_{i-1} b c_{i+1} \cdots c_n} \cup \mathcal{C}_w^{c_1 \cdots c_{i-1} b c_{i+1} \cdots c_n},$$
$$\mathcal{M}_w = \mathcal{C}_b^{c_1 \cdots c_{i-1} w c_{i+1} \cdots c_n} \cup \mathcal{C}_w^{c_1 \cdots c_{i-1} w c_{i+1} \cdots c_n},$$
and $H(M_i)$ denotes the Hamming weight of the i-th row vector M_i of a matrix M.

The values $\alpha_R > 0$ and $\alpha_S > 0$ are referred as *relative difference of the reconstructed image* and *relative difference of shadow images*, respectively. The number $\alpha_R m \geq 1$ and $\alpha_S m \geq 1$ are called *contrast*s of the reconstructed image and the shadow images in visual cryptography studies. People would like both α_R and α_S to be as large as possible.

The first condition is the *contrast* condition which indicates any qualified set $X \in \Gamma_{\mathsf{Qual}}$ can recover the secret image. The secret image can be recovered by stacking the transparencies of a qualified set, belonging to Γ_{Qual}. The second condition is the *security* condition which states any forbidden set $X = \{i_1, \cdots, i_q\} \in \Gamma_{\mathsf{Forb}}$ has no information on the secret image. People cannot get any information on the secret image by inspecting the shadow images of a forbidden set. The third condition is the *extended* condition which implies that the shadows images are still meaningful after the original images are encoded. Any participant can recognize the shadow image on ones transparency.

Here we show how to accomplish a 2 out of 2 EVCS. Each pixel consists of 4 subpixels. However, it contains either a 1 or two 1's depending on the colors of pixels of the corresponding original image, black or white, respectively. The scheme is given by the 4 pairs of collections $(\mathcal{C}_b^{c_1c_2}, \mathcal{C}_w^{c_1c_2})$, namely 8 collections $\mathcal{C}_c^{c_1c_2}$, where $c, c_1, c_2 \in \{b, w\}$. The collections are obtained by permuting the columns of the following 8 basic matrices, $S_c^{c_1c_2}$:

$$S_b^{bb} = \begin{bmatrix} 0\,0\,0\,1 \\ 1\,0\,0\,0 \end{bmatrix}, S_b^{bw} = \begin{bmatrix} 0\,0\,0\,1 \\ 1\,1\,0\,0 \end{bmatrix}, S_b^{wb} = \begin{bmatrix} 0\,0\,1\,1 \\ 1\,0\,0\,0 \end{bmatrix}, S_b^{ww} = \begin{bmatrix} 0\,0\,1\,1 \\ 1\,1\,0\,0 \end{bmatrix},$$
$$S_w^{bb} = \begin{bmatrix} 0\,0\,0\,1 \\ 0\,0\,0\,1 \end{bmatrix}, S_w^{bw} = \begin{bmatrix} 0\,0\,0\,1 \\ 0\,1\,0\,1 \end{bmatrix}, S_w^{wb} = \begin{bmatrix} 0\,0\,1\,1 \\ 0\,0\,0\,1 \end{bmatrix}, S_w^{ww} = \begin{bmatrix} 0\,0\,1\,1 \\ 0\,1\,0\,1 \end{bmatrix}.$$

The reconstructed pixel has one or zero white subpixel if the original secret pixel is white or black, respectively. In this scheme, the relative contrasts are given as $\alpha_R = \alpha_S = 0.25$. Figure 1 shows an example of resulting shadow images and reconstructed secret image. The size of all images are 128×128 pixels, where all the original shadow and secret images have 64×64 pixels.

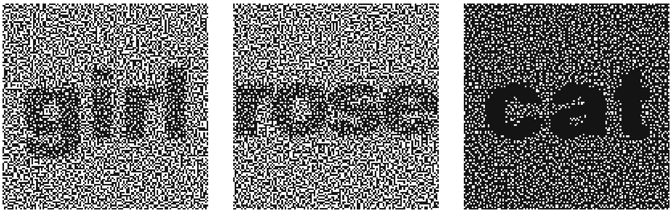


Fig. 1. An example of EVCS. Two resulting shadow images (left and middle) and reconstructed secret image (right).

Ateniese et al. also pointed out some important aspects of the extended capability [2]. One is related to the contrasts of images. A trade-off between two relative differences exists, α_R and α_S, in any (k, k) EVCS as below:

$$2^{k-1}\alpha_R + \frac{k}{k-1}\alpha_S \leq 1.$$

This means we cannot increase both contrasts of a reconstructed image and shadow images, $\alpha_R m$ and $\alpha_S m$, simultaneously. They also specified the lower bound of the pixel expansion m in (k, k) EVCS as below:

$$m \geq 2^{k-1} + 2.$$

This indicates we need more pixels to obtain EVCS. Although people would like contrasts to be as large as possible and pixel expansion as small as possible, there exist certain limits of them.

3 EVCS for Continuous-Tone Images

Digital cameras have become very popular and people can easily obtain digital image data of continuous tone. However the scheme explained in the last section accepts binary images as input. Thus a continuous-tone image must be converted to a binary image which can be observed similar to the original image by human visual system. The process which can achieve such conversion is referred as *digital halftoning* or *halftoning* in short [11,6].

Error diffusion, commonly-used halftoning algorithm, was proposed by Floyd and Steinberg [3]. It is an adaptive algorithm that uses the threshold error feedback to produce patterns having different spatial frequency content. A single pass is carried out over the input image each pixel of which is processed sequentially. A single pixel process consists of a binary thresholding of the input pixel and an error computation caused by the binarization. This error is distributed to the neighboring pixels that have not been processed according to an error filter. An example of error filter is shown in Figure 2 proposed by Jarvis et al. [5], where X indicates the current pixel. In other words, the values of neighboring pixels are corrected to keep the total tone of a local region. The schematic diagram of the algorithm is illustrated in Figure 3.

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

Fig. 2. A sample of error filters for error diffusion proposed by Jarvis et al. [5]

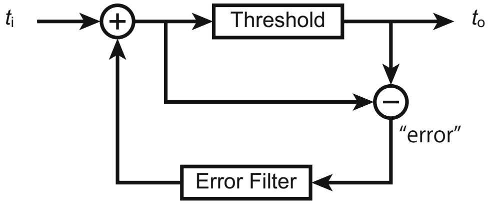


Fig. 3. A diagram of error diffusion process

The straightforward way to incorporate continuous-tone images into visual cryptography is as below:

1. Convert continuous-tone images to binary images by halftoning.
2. Encrypt a secret image by EVCS explained in Section 2.

The resulting shadow images to be printed on transparencies and reconstructed secret image by stacking shadow images are binary images. The pixel expansion of $(2, 2)$ EVCS is $m = 4$. A tradeoff between relative differences of shadow image and reconstructed image exists. If we restrict both relative differences to be the same, the maximum relative differences are $\alpha_S = \alpha_R = 0.25$.

Let us discuss quality of resulting images in terms of parameters m and α.

Pixel expansion (m) Pixel expansion is an important parameter that affects quality of images as well as its data size. The resulting image requires m times more subpixels which means that subpixels must be m times smaller than the original pixel if the image size is fixed to that of the original image.

Relative difference (α) It is obvious that contrast is also one of the most important parameters related to image quality. An image with low contrast is obscure and difficult to see its details. Furthermore, there exists a certain tradeoff between contrasts of shadow and secret images in case of extended visual cryptography.

4 Proposed Scheme of Extended Visual Cryptography

This paper proposes a new scheme of extended visual cryptography based on *parallel error diffusion*, which can encrypt images without any pixel expansion in an instant. Furthermore the resulting images may have large contrasts comparing with the conventional cryptography schemes by *optimum tone mappping*. The actual encrypting process first applies tone mapping to input images, followed by parallel error diffusion. However, due to the ease of explanation, parallel error diffusion is presented first in this section, and the optimum tone mapping will be explained next. Here we focus on a $(2, 2)$ EVCS for simplifying our discussion.

4.1 Basic Encrypting Process – Parallel Error Diffusion

Parallel error diffusion [12] can produce the encrypted images with no pixel expansion by considering secret information as extra noise to the images and taking into account with binarization error. Figure 4 illustrates the encryption process. The parallel error diffusion takes three grayscale images as input and generates two encrypted (binary) images by processing three corresponding pixels simultaneously. Let us call the three corresponding shadow and secret pixels as *triplet.* The encryption process is performed triplet by triplet. Here, $t^i_0, t^i_1, t^i_r \in [0, 1]$ stand for the pixel values of a triplet, i.e., shadow 0, shadow 1, and secret pixels, while $t^o_0, t^o_1, t^o_r \in \{0, 1\}$ represent the pixel values of the resulting triplet. The pixel value of a resulting secret pixel t^o_r is simply determined by the corresponding shadow pixels, t^i_0 and t^i_0, as below:

$$t^o_r = t^o_0 \cdot t^o_1 \qquad t^o_0, t^o_1, t^o_r \in \{0, 1\} \, .$$

Thus the possible combinations of three pixel values (t^o_0, t^o_1, t^o_r) are $(0, 0, 0)$, $(0, 1, 0)$, $(1, 0, 0)$, and $(1, 1, 1)$. The binarization errors are distributed to the local unprocessed pixels like usual error diffusion algorithm. We must note that a result of parallel error diffusion is no more secret sharing, because one of the resulting shadow images is affected by both the other shadow and secret images. However, it is almost impossible to detect a secret image from an encrypted

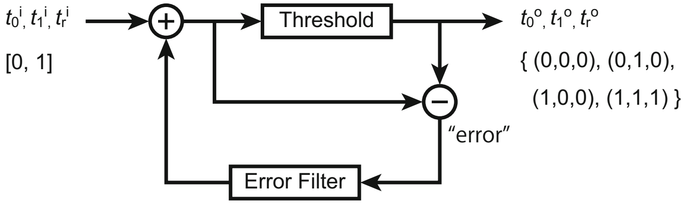


Fig. 4. A diagram of parallel error diffusion process

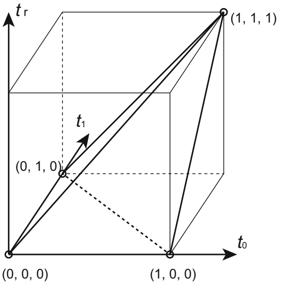


Fig. 5. The region of possible pixel values

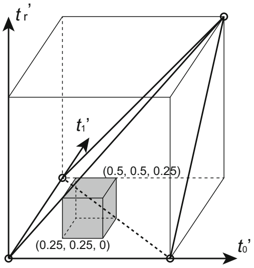


Fig. 6. The region obtained by the conservative tone mapping

shadow image. A shadow is influenced not only by the secret image but also by the other shadow image, and their influence is very faint. Furthermore, image-signal separation is almost impossible from a single image.

4.2 Optimal Tone Mapping for Encryption

It is obvious that there are some constraints among pixel values of a triplet which can be encrypted. For instance, no white secret pixel can be obtained if one or more shadow pixels are black. No black pixel can be accomplished by completely-white shadow pixels. Figure 5 shows the space of pixel values of a triplet, t_0, t_1, and t_r. The four points indicated by circles are possible pixel values of a encrypted triplet. The encrypted shadow images and resulting secret image are implemented with the four possible combinations which limit the dynamic ranges of resulting images into the tetrahedron spanned by the four points of possible combinations. The input images must be tone-mapped so that the all combinations of pixel values lie inside of the tetrahedron, which guarantees the encryption.

The most conservative tone mapping is given as below:

$$t_0' = 0.25t_0 + 0.25, \quad t_1' = 0.25t_1 + 0.25, \quad t_r' = 0.25t_r.$$

Here, t_0, t_1, and t_r represent the original pixel values while t_0', t_1', and t_r' stand for the tone-mapped pixel values. The entire dynamic range of a unit cube is

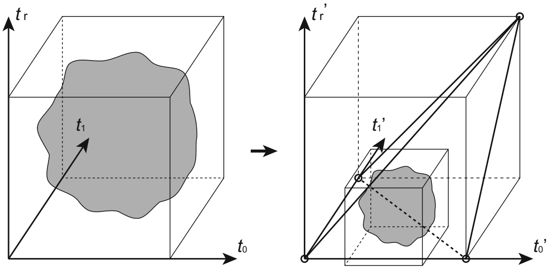


Fig. 7. The tone-mapped region taking into account of the actual pixel values of images

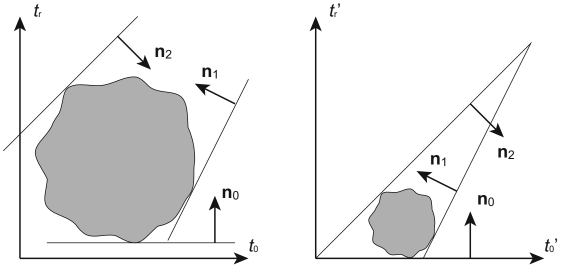


Fig. 8. The tone mapping of affine transformation which inscribes the region in a tetrahedron

mapped to a inscribed cube of the tetrahedron as shown in Figure 6. The resulting dynamic range is exactly the same as that of the conventional extended visual cryptography.

An image usually contains limited numbers of completely black or white pixels. Furthermore, it is quite rare that those extreme pixels are located at the same position. The contrast may be extended more than a quarter, because the extreme triplet values such as $(0, 0, 1)$, $(1, 1, 0)$, etc. do not exist. Suppose a shaded region on the left side of Figure 7 depicts an example of actual triplet region. An expected result of tone mapping is shown in Figure 7 right so that the shaded region tightly fits the tetrahedron. A set of parameters of the tone mapping that inscribes the given region in the tetrahedron should be calculated. Let us assume the tone mapping is determined by an affine transformation as below:

$$t'_0 = at_0 + b_0, \quad t'_1 = at_1 + b_1, \quad t'_r = at_r + b_r,$$

so that the resulting three images have the same contrast a. The degree of freedom of this affine transformation is four which is the same as the number of constraints given by the tetrahedron, i.e., the number of faces. Since this transformation is an isotropic scaling with a translation, the contacting points can be determined by the planes having the same normals as those of the tetrahedron's faces. Figure 8 illustrates this property in 2D space. Therefore, we can compute parameters of the required affine transformation by detecting contacting points and making them on the tetrahedron's faces.

Let us consider the meanings of tetrahedron's faces. The tetrahedron is bounded by four faces which are represented by the following inequalities:

$$t'_r \le t'_0, \quad t'_r \le t'_1, \quad t'_r \ge t'_0 + t'_1 - 1, \quad t'_r \ge 0. \tag{1}$$

The first two inequalities give the upper bounds of reconstructed secret pixel values. The secret pixels cannot be brighter than the corresponding shadow pixels as shown in Figure 9 top. The last two inequalities stand for the lower bounds of reconstructed secret pixel values. The secret pixel cannot be so dark if both of the shadow pixels are bright (Figure 9 bottom left). Of course, a secret pixel value cannot be negative (Figure 9 bottom right). These inequalities constrain the ratio of black and white areas. However, each pixel of the resulting images is completely black or white. So the inequalities represent constraints on corresponding local regions of three images rather than a single triplet. This type of constraints can be handled by applying a blur filtering to the input images before computing tone-mapping parameters. The blur filtering may result in a larger contrast, because it makes a triplet region smaller.

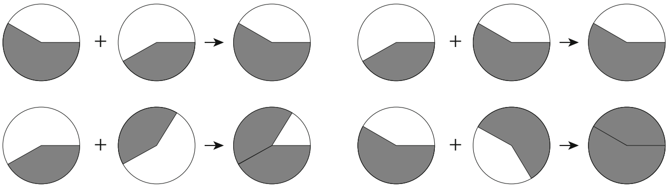


Fig. 9. The constraints on the pixel values of a triplet

The entire process of proposed EVCS is illustrated in Figure 10.

1. A gaussian blur filtering is applied to the input images.
2. A set of tone-mapping parameters are computed by solving constraints.
3. The input images are tone-mapped with the obtained parameters.
4. The tone-mapped images are encrypted by parallel error diffusion.

The more blurred filtering leads to the larger contrast as discussed above. However, too much blurring may results in distortion of local regions. We will discuss this issue in the next section.

5 Experiments

We made some experiments to examine our proposed scheme and the effect of a gaussian filtering. Six images in Figure 11 were used for the experiments. All images have the same size of 512 × 512 pixels.

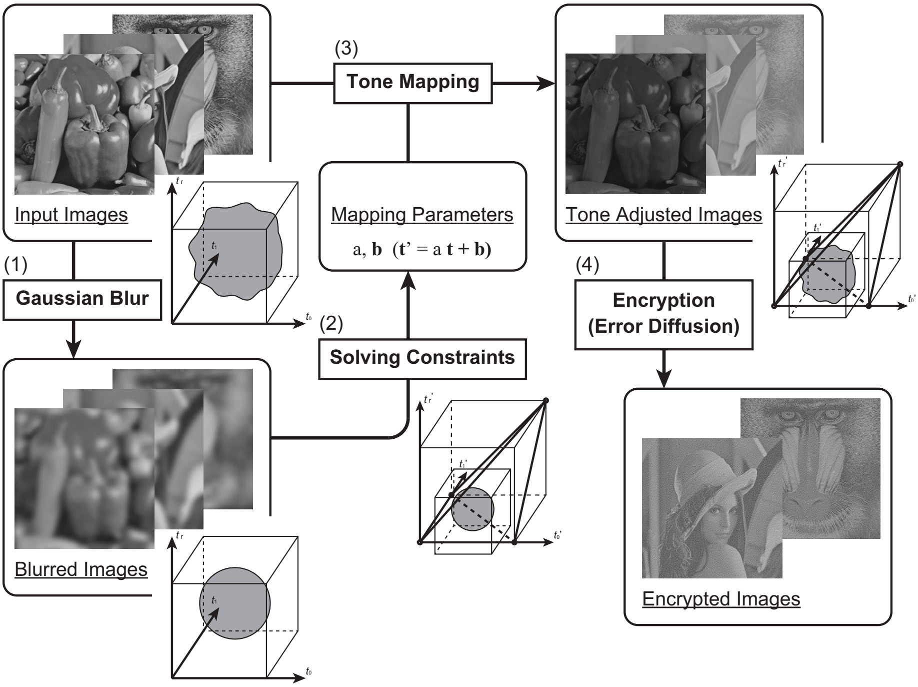


Fig. 10. The entire algorithm of proposed scheme

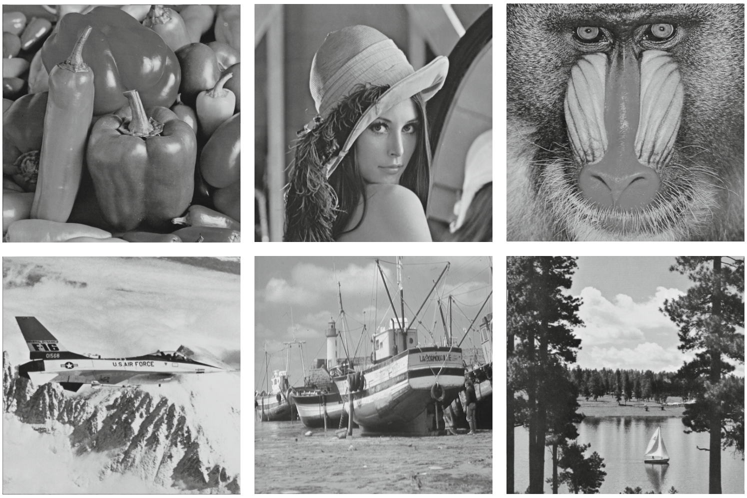


Fig. 11. Six images used in the experiments, "Peppers" (top left), "Lena" (top center), "Mandrill" (top right), "Airplane" (bottom left), "Fishing Boat" (bottom center), and "Sailboat on Lake" (bottom right)

Figure 12 shows the encrypted results of three different ways, namely, straightforward implementation based on the conventional EVCS, parallel error diffusion with the conservative tone-mapping, and parallel error diffusion with the optimum tone-mapping. The image of ”Peppers” was used as a secret image and the images of ”Lena” and ”Mandrill” were used as shadow images. The left image are the resulting shadow images of ”Lena” while the right images are the reconstructed secret images. Only one shadow image is depicted because of the space limitation. Every image consists of 512 × 512 pixels. The top result lacks details of images because they were once scaled down into 256 × 256 pixels due to the pixel expansion. The middle result preserves some details, though it has the same contrast, 0.25, as the top. The bottom result was generated with a gaussian filtering of kernel size $k = 21$ (pixels) and standard deviation $\sigma = 10$ (pixels). The resulting contrast is 0.44 which improves the image quality.

Because of the nature of parallel error diffusion, relatively big errors may occur during the encryption process. Figure 13 indicates the regions where the big errors are detected. Green pixels specify white pixels which should be black by the ordinary error diffusion, while red pixels are black pixels to be white. The top images point out the errors of conservative tone-mapping. Although the conservative tone-mapping maps the entire dynamic range into the inside of the tetrahedron as illustrated in Figure 6, the binarizing errors diffused from nearby pixels may cause relatively big errors. This type of errors do not deteriorate the resulting images, because they occurs seldom and separately. However, the error occurs more often if contrasts of input images are enhanced. The bottom images in Figure 13 show the errors during the parallel error diffusion of higher contrast. The tone-mapping parameters are computed from the blurred images whose standard deviation was $\sigma = 20$. The resulting contrast was 0.56 in this case. The yellow (bright) circles indicate the regions where one of shadow images is too dark comparing with the secret image. The pink (dark) circles indicate the region where two shadow images are too bright to achieve the dark secret image. This type of errors may cause serious deteriorations of encrypted shadow images as well as reconstructed secret image.

In order to examine the effect of a blur size, i.e., standard deviation σ of a gaussian filter, we have computed contrasts and numbers of big-error pixels. For each standard deviation value, ${}_6\mathrm{C}_3 \times {}_3\mathrm{C}_1 = 30$ combinations of images chosen from six in FIgure 11 were encrypted, because contrasts and errors depend on the combination of images. Figure 14 shows a graph illustrating the variation of contrast with the standard deviation σ of a gaussian kernel. The graph indicates that the average contrast is 0.32 without a gaussian blur. The average contrast increased as the blur size increases, however the increasing rate gradually decreases. Figure 15 is a graph showing the variation of big errors with the standard deviation σ. According to this graph, number of errors stays low while the blur size is very small, $\sigma \leq 5$. It gradually increases if the blur size is relatively small, $5 \leq \sigma \leq 10$, and increases very rapidly if the blur size exceeds a certain size, $\sigma > 10$. Significant deteriorations cannot be found if σ is smaller than 10 in our experiments.

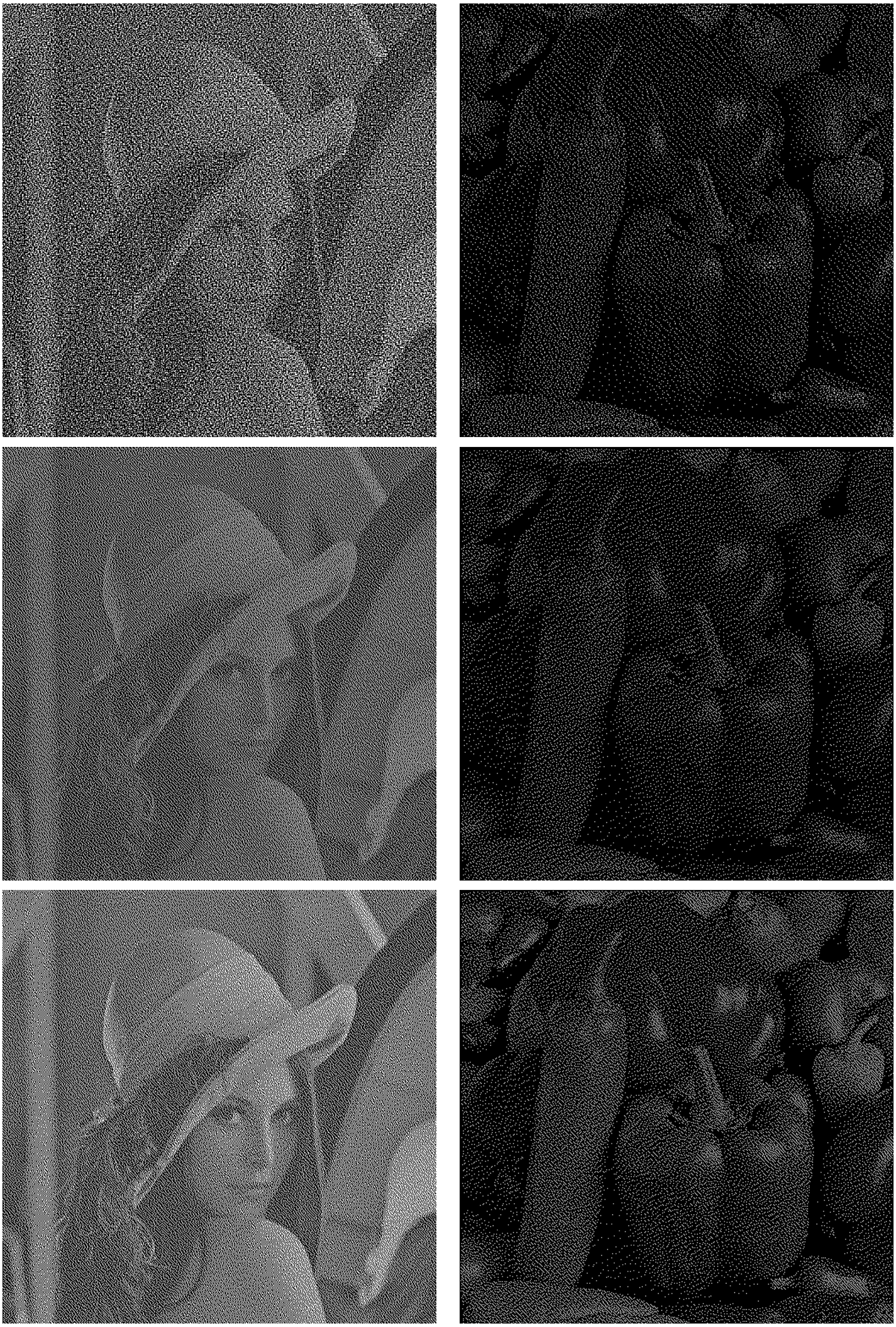


Fig. 12. Comparison of encryption results. The straightforward encryption (top), proposed method with conservative tone-mapping (middle), and proposed method with enhanced tone-mapping (bottom).

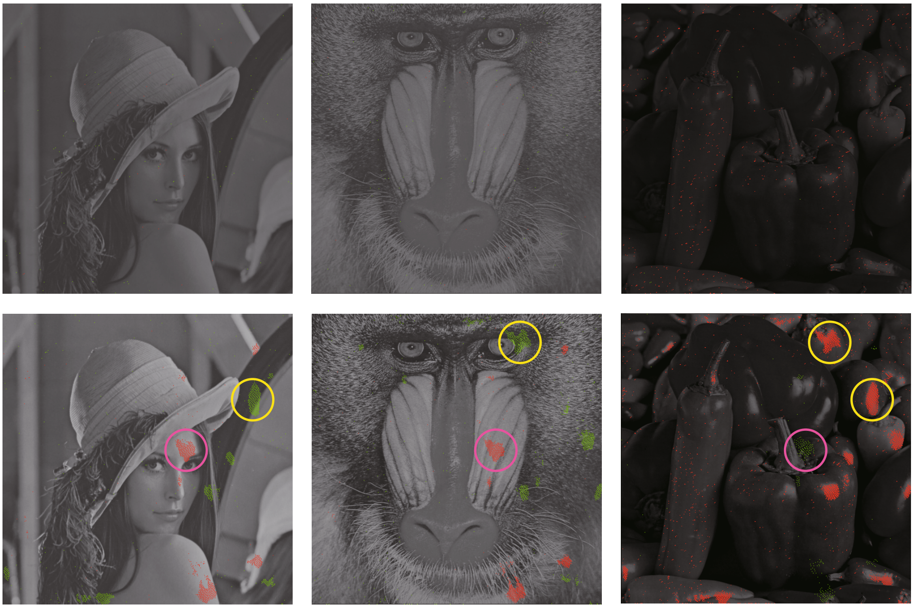


Fig. 13. The relatively big errors detected during the encryption process with $\sigma = 0$ (top) and $\sigma = 20$ (bottom)

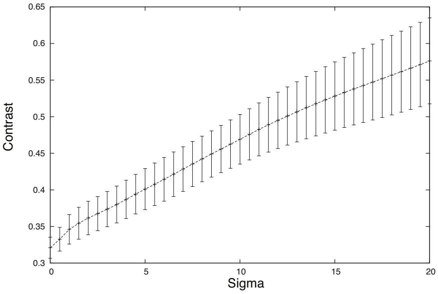


Fig. 14. The variation of the resulting contrast with standard deviation σ of a gaussian filter

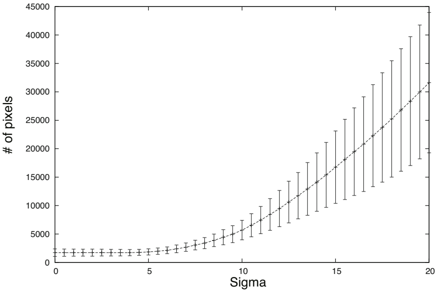


Fig. 15. The variation of big-error pixels with standard deviation σ of a gaussian filter

6 Discussions

The parallel error diffusion, which was explicitly proposed by Wang et al. in 2009 [12], considers secret information as extra noise to the shadow image and takes into account with binarizing error. This type of approach was first mentioned by Fu et al. in 2003 [4]. Their basic algorithm is as below:

1. The first shadow image is halftoned by the error diffusion algorithm.
2. The second shadow image is also halftoned by the conjugate error diffusion algorithm . The "Noise" is added according to the corresponding pixels of the secret image and the first halftoned shadow image. This noise causes a pixel-wise distortion to the second shadow image which is controlled by some appropriate threshold T_1. A large T_1 allows more pixels to hide secret image but results in a large distortion of the second shadow image.

The approach is free from pixel expansion, $m = 1$. However, [4] was only applied to a logo image as a secret image in order to avoid a huge distortion in the second shadow image. It could reconstruct only a faint logo with the traces of both shadow images. Myodo et al. extended this approach to be able to handle a continuous-tone image as a secret image [7]. Wu et al. [13] implied to use vector error diffusion for the encryption, but they actually proposed an iteration-based search method to obtain encrypted hailftoned-images which might cost much more computation time.

Due to the nature of EVCS, a contrast reduction is inevitable. It is preferrable to enhance contrasts of the resulting images as much as possible. The conventional visual cryptography studies consider relative differences, α_S and α_R, which represent a limitation of possible pixel values. Nakajima and Yamaguchi precisely examined the interactions of pixel values in $(2, 2)$ EVCS [9] and presented the same conditions as Equation (1). The tones of given images must be properly adjusted which makes the contrast as large as possible. Affine transformation or piecewise linear transformation is most commonly used for this tone adjustment [9,13,14,7,8]. Wu et al. suggested to calculate optimum parameters [13]. However, [13] does not explain any details how to obtain optimum parameters. Myodo et al. proposed a method that can determine optimum parameters in an instant [8]. They claimed that their method can enhance contrasts to 0.28 on average without any violation. Our experimental results indicate better average contras 0.32 even if $\sigma = 0$, but this difference might be caused by the set of images to be examined.

7 Conclusions

This paper proposed a new scheme of extended visual cryptography. The scheme consists of parallel error diffusion and optimum tone mapping. Parallel error diffusion provides a very simple and quick way to produce extended visual cryptography with no pixel expansion. The optimum tone mapping can be accomplished by solving linear equation systems. A gaussian blur filtering can enhance contrasts by relaxing the constraints. We made some experiments to examine effectiveness and limitation of the scheme.

Unfortunately the scheme does not accomplish secret sharing, because the encrypted shadow images are affected by the secret image. However it is virtually impossible to reconstruct a secret image from a single shadow. A shadow is influenced by the other shadow as well as the secret image and their influence is very faint. Furthermore, image-signal separation is still a tough problem in image processing studies.

This paper focused on a $(2, 2)$ scheme of extended visual cryptography because of the space limitation. However, the proposed scheme consisting of optimum tone mapping and parallel error diffusion can be extended to other cases.

Acknowledgement. This work was partially supported by JSPS Grant-in-Aid for Scientific Research (B) (22300030).

References

1. Ateniese, G., Blundo, C., Santis, A.D., Stinson, D.R.: Visual cryptography for general access structure. Information and Computation 129, 86–106 (1996)
2. Ateniese, G., Blundo, C., Santis, A.D., Stinson, D.: Extended capabilities for visual cryptography. Theoretical Computer Science 250, 143–161 (2001)
3. Floyd, R., Steinberg, L.: An adaptive algorithm for spatial grayscale. Society for Information Display 17, 75–77 (1976)
4. Fu, M., Au, O.: A novel method to embed watermark in different halftone images: data hiding by conjugate error diffusion (dhced). In: Intl Conference on Acoustics, Speech, and Signal Processing, vol. III, pp. 529–532 (2003)
5. Jarvis, J., Judice, C., Ninke, W.: A survey of techniques for the display of continuous tone pictures on bilevel displays. Computer Graphics and Image Processing 5(1), 13–40 (1976)
6. Kang, H.: Digital Color Halftoning. SPIE Optical Engineering Press, IEEE Press (1999)
7. Myodo, E., Takagi, K., Miyaji, S., Takishima, Y.: Halftone visual cryptography embedding a natural grayscale image based on error diffusion technique. In: Intl Conference on Multimedia and Expo., pp. 2114–2117 (2007)
8. Myodo, E., Takagi, K., Yoneyama, A.: Deterministic tone mapping to gamut area of halftone visual cryptography. IEICE Trans. on Fundamentals J93-A(12), 805–820 (2010)
9. Nakajima, M., Yamaguchi, Y.: Extended visual cryptography for natural images. Journal of WSCG 10(2), 303–310 (2002)
10. Naor, M., Shamir, A.: Visual Cryptography. In: De Santis, A. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 1–12. Springer, Heidelberg (1995)
11. Ulichney, R.: Digital Halftoning. The MIT Press (1987)
12. Wang, Z., Arce, G.R., Crescenzo, G.D.: Halftone visual cryptography via error diffusion. IEEE Trans. on Information Forensics and Security 4(3), 383–396 (2009)
13. Wu, C., Thompson, G., Stanich, M.: Digital watermarking and steganography via overlays of halftone images. Electrical Engineering IBM Research Report RC23267 (W0407-013), IBM (2004)
14. Yang, C.N., Chen, T.S.: Extended visual secret sharing schemes: improving the shadow image quality. International Journal of Pattern Recognition and Artificial Intelligence 21, 879–898 (2007)

A Comprehensive Study on Third Order Statistical Features for Image Splicing Detection

Xudong Zhao[1], Shilin Wang[2], Shenghong Li[1], and Jianhua Li[1]

[1] Department of Electronic Engineering, Shanghai Jiao Tong University
[2] School of Information Security Engineering, Shanghai Jiao Tong University
Shanghai, P.R. China 200240
{zxd_1220,wsl,shli,lijh888}@sjtu.edu.cn

Abstract. Second order statistical features (e.g. Markov transposition probability matrix and gray level co-occurrence matrix) have been proved to be effective for passive image forgery detection in the past few years. In this paper, third order statistical features are proposed for image splicing detection. We model the thresholded adjacent difference block DCT coefficient array of an image as conditional co-occurrence probability matrix, second order Markov transition probability matrix and second order co-occurrence matrix. Since the dimensionality exponentially depends on the order, dimensionality of the third order features is much larger than that of second order features, principal component analysis (PCA) is therefore introduced in our work to overcome the high dimensionality introduced computational complexity and the possible overfitting for a kernel based supervised classifier. Experimental results show that conditional co-occurrence probability matrix outperforms second order features and PCA is proved to be an effective dimensionality reduction tool for image splicing detection. We also test the robustness of third order statistical features, despite higher dimensionality, third order statistical features demonstrate the same robustness as that of second order features.

Keywords: image splicing detection, third order statistical feature, PCA.

1 Introduction

We are living in a digital world, the constantly upgraded hardware and software bring great convenience to our lives. However, every coin has two sides, forging an image is becoming easier and easier, even an unskilled person is able to forge an eye-deceiving image without much time. Researchers have made effort on digital image forensics to regain trust on digital images. Image forensics on the whole can be concluded as active methods and passive methods. Digital signature and watermarking have been proposed as active detection methods, however, the signature or watermark must be inserted in the imaging process which limits its application. In contrast, passive image forgery detection methods do not need any prior knowledge, they work on the assumption that the image

Y.Q. Shi, H.J. Kim, and F. Perez-Gonzalez (Eds.): IWDW 2011, LNCS 7128, pp. 243–256, 2012.

to be detected has no signature or watermark. In this paper, we concern with passive image forgery detection method.

In the past few years, several passive detection methods have been proposed [1]. T. T. Ng et al. in [2] proposed bi-coherence based features for image splicing detection, and the detecting accuracy over image dataset [3] is 71%. In the imaging process, the digital camera introduces some artifacts which can be specifically modeled. In consistencies of these artifacts can be used as features for image forgery detection. Popescu et al. in [4] proposed an interpolation based method to model the color filter array (CFA) interpolation process of digital camera, CFA interpolation introduced correlations are likely to be destroyed by image tampering, therefore this model can be used to detect image forgery, but this method failed in detecting images with high compression rate. When composing an image, it is difficult for the forger to match the lighting conditions. Lighting inconsistencies can therefore be used as evidence to reveal the tampering [5]. But, the proposed methods failed in detecting the splicing part with the same lighting conditions. The tampered image may be quite eye deceiving, but the tampering may change the underlying statistics of the original image, some statistical-feature based methods were proposed to expose the image forgeries. Phase congruency and statistical moments of characteristic functions of wavelet sub-bands were proposed in [6] to catch the splicing introduced artifacts and the experimental results over [3] is 82.3%. SIFT was proposed in [7] [8] to detect image region duplications and the proposed method can handle cases when a region is scaled or rotated before pasted to a new location. Yun Q. Shi et al. in [9] proposed a natural image model for image splicing detection, the statistical features consists of moments of characteristic functions of wavelet sub-bands and first order Markov transition probabilities of difference block DCT array (1^{st} Markov). Detecting accuracies over [3] showed that moments features achieved 86.8% and 1^{st} Markov achieved 88.3%. In [10] and [11], gray level co-occurrence matrix features and Run-length Run-number features extracted in chroma spaces (Cb and Cr) were employed to detect image splicing, experimental results showed that features in chroma spaces demonstrate much better performance than that in luma space. However, both methods do not get satisfying performance over image dataset [3].

Image splicing introduces sharp edges in a tampered image, capturing the artifacts is the key for the detection work. Second order statistical features have been proved as effective features, higher order statistics in [2] and [6] were proposed to capture the splicing artifacts, however, the detection rate is not satisfying. In this paper, we model the relationships among neighboring three elements in block DCT domain as third order statistical features, that is, conditional co-occurrence probability matrix (CCPM), second order Markov transition probability matrix (2^{nd} Markov), second order co-occurrence probability matrix (2^{nd} CPM). Third order features are more informative than lower order features, however, the dimensionality of features exponentially depends on order, for modern supervised machine learning algorithms, high dimensionality usually causes computational complexity and overfitting [12]. Therefore, PCA is employed in our work to reduce the higher order introduced problems. In the experimental work, we investigate the detecting performance of CCPM, 2$^{\text{nd}}$ Markov and 2$^{\text{nd}}$ CPM over image dataset [3], and we find that CCPM outperforms 2$^{\text{nd}}$ Markov, 2$^{\text{nd}}$ CPM, 1$^{\text{st}}$ Markov and first order co-occurrence probability matrix (1$^{\text{st}}$ CPM) features, PCA is able to reduce the dimensionality of proposed features without losing discriminative information.

The rest of this paper is organized as follow. The proposed method is described in section 2. In section 3, the experimental work and performance analysis are reported. Finally, conclusions and future works are given in section 4.

2 Proposed Method

2.1 Preprocessing

Image contents are usually deemed as noise in the splicing detection work, that is, image contents introduce interferential information in the detection process. Therefore it is necessary to eliminate the influence of image contents before feature extraction work. DCT has very good information packing properties for most of the images, it concentrates most of the image energy in a few coefficients. In this paper, we extract features in the adjacent difference 8*8 block DCT array which proved to be effective in the work [9] and it is given as

$$E_h(i,j) = |X(i,j)| - |X(i+1,j)| \tag{1}$$

$$E_v(i,j) = |X(i,j)| - |X(i,j+1)|, \tag{2}$$

where X is the $8*8$ non-overlapping block DCT of the image to be detected and it is defined as

$$X = \begin{pmatrix} X_{11} & X_{12} & \cdots & X_{1m} \\ X_{21} & X_{22} & \cdots & X_{2m} \\ \vdots & \vdots & \cdots & \vdots \\ X_{m1} & X_{m2} & \cdots & X_{mm} \end{pmatrix}, \tag{3}$$

where $X_{ij}(1 \leq i,j \leq m)$ is $8*8$ block DCT coefficient array of the image, and it is given by

$$X_{ij} = U^T X_{ij}^s U, \tag{4}$$

where X_{ij}^s is the corresponding $8*8$ image block and U is defined as

$$\begin{cases} U(n,k) = \frac{1}{2\sqrt{2}}, & k = 0, 0 \leq n \leq 7 \\ U(n,k) = \frac{1}{2}\cos(\frac{\pi(2n+1)k}{16}), & 1 \leq k \leq 7, 0 \leq n \leq 7 \end{cases} \tag{5}$$

E_h and E_v are then rounded and thresholded in the range $[-T,T]$, i.e. there are totally $2T+1$ states to be modeled. Larger T means more information taken into considertation, however it may also brings disturbing information and high computational complexity. A properly selected T should be a compromise between detction performance and computing complexity. We give the detection performance of proposed mehtod with different T in the experimental work.

2.2 Third Order Statistical Feature Extraction

Markov chain is commonly used to characterize the underlying dependences between neighboring states. For states $\omega_1, \omega_2, \cdots, \omega_N$ (N is the total number of states and $N = 2T + 1$), the first order Markov model assume that

$$P(\omega_{i_k}|\omega_{i_{k-1}}, \omega_{i_{k-2}}, \cdots, \omega_{i_1}) = P(\omega_{i_k}|\omega_{i_{k-1}}), \tag{6}$$

that is, the dependence is limited within two neighboring states. First order Markov features have been proved to be one of the most effective features for image splicing detection [9]. In this paper we expand first order Markov features to third order statistical features, i.e. we consider the relationships among three adjacent states. Second order Markov features proposed in [13] verified its effectiveness in steganalysis, inspired by the work of [13], we model our features for splicing detection as follows

$$CCPM \equiv \begin{bmatrix} P(\omega_1, \omega_1|\omega_1) & P(\omega_2, \omega_1|\omega_1) & \cdots & P(\omega_N, \omega_N|\omega_1) \\ P(\omega_1, \omega_1|\omega_2) & P(\omega_2, \omega_1|\omega_2) & \cdots & P(\omega_N, \omega_N|\omega_2) \\ \vdots & \vdots & \cdots & \vdots \\ P(\omega_1, \omega_1|\omega_N) & P(\omega_2, \omega_1|\omega_N) & \cdots & P(\omega_N, \omega_N|\omega_N) \end{bmatrix} \tag{7}$$

$$2^{nd}Markov \equiv \begin{bmatrix} P(\omega_1|\omega_1, \omega_1) & P(\omega_1|\omega_1, \omega_2) & \cdots & P(\omega_1|\omega_N, \omega_N) \\ P(\omega_2|\omega_1, \omega_1) & P(\omega_2|\omega_1, \omega_2) & \cdots & P(\omega_2|\omega_N, \omega_N) \\ \vdots & \vdots & \cdots & \vdots \\ P(\omega_N|\omega_1, \omega_1) & P(\omega_N|\omega_1, \omega_2) & \cdots & P(\omega_N|\omega_N, \omega_N) \end{bmatrix} \tag{8}$$

$$2^{nd}CPM \equiv \begin{bmatrix} P(\omega_1, \omega_1, \omega_1) & P(\omega_1, \omega_1, \omega_2) & \cdots & P(\omega_1, \omega_N, \omega_N) \\ P(\omega_2, \omega_1, \omega_1) & P(\omega_2, \omega_1, \omega_2) & \cdots & P(\omega_2, \omega_N, \omega_N) \\ \vdots & \vdots & \cdots & \vdots \\ P(\omega_N, \omega_1, \omega_1) & P(\omega_N, \omega_1, \omega_2) & \cdots & P(\omega_N, \omega_N, \omega_N) \end{bmatrix}, \tag{9}$$

where $P(\omega_{i_k}, \omega_{i_{k-1}}|\omega_{i_{k-2}})$ is the co-occurrence probability of $(\omega_{i_k}, \omega_{i_{k-1}})$ given state $\omega_{i_{k-2}}$, $P(\omega_{i_k}|\omega_{i_{k-1}}, \omega_{i_{k-2}})$ indicates the second order Markov transition probability, i.e. the future sate is determined by the current state and previous state, $P(\omega_{i_k}, \omega_{i_{k-1}}, \omega_{i_{k-2}})$ is the joint probability of three states.

Third order statistical feature matrices are directional, that is, they can be used to depict the relationships of adjacent three states along eight directions. In our work, CCPM, 2^{nd} Markov, and 2^{nd} CPM are employed to model the horizontal right and vertical down directional relationships of neighboring three states which are shown in Fig. 1. Gray lattices in Fig. 1 indicate the adjacent three states to be made statistics and arrow denotes the direction along which probabilities mentioned above are computed. We extract third order statistical features from the directional thresholded adjacent difference DCT array. Probabilities in CCPM, 2^{nd} Markov and 2^{nd} CPM are stretched into a vector and fed into classifier for classification. To view the class separability of third order statistical features, linear discriminant analysis (LDA) is employed to map the original features extracted from [3] onto the 1 D feature space, and the projections are shown in Fig. 2. Note that for the convenience of viewing, we scatter plot the transformed 1 D features on the 2 D space whose horizontal axis indicates the number of samples and vertical axis indicates the value of the transformed 1 D features. It can be seen from Fig. 2 that the third order statistical features belonging to two different classes are well clustered in the transformed space.

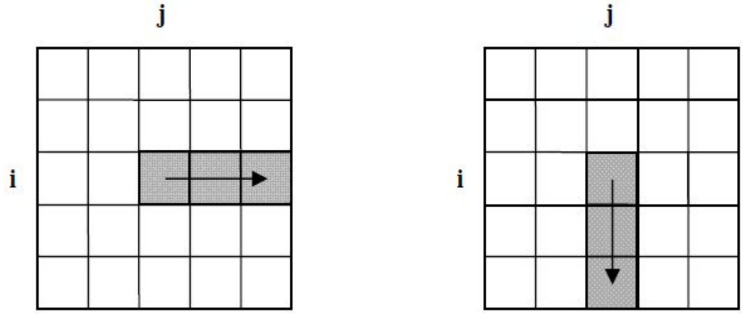


Fig. 1. Relationships of adjacent three states along horizontal right and vertical down directions. (i, j) is the spatial coordinate of initial state.

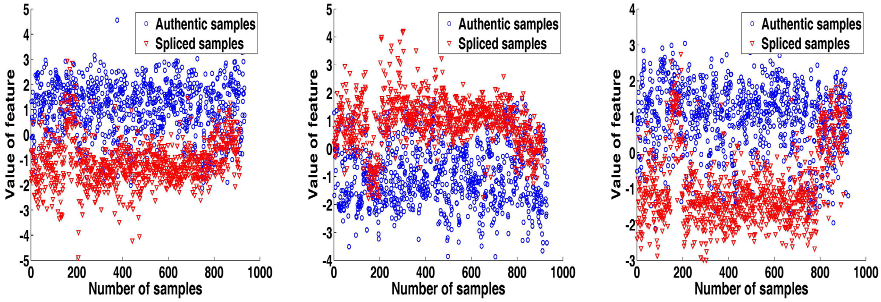


Fig. 2. LDA projections of third order statistical features over image dataset [3]. CCPM, 2$^{\text{nd}}$ Markov and 2$^{\text{nd}}$ CPM are shown from left to right.

2.3 Dimensionality Reduction

CCPM, 2$^{\text{nd}}$ Markov and 2$^{\text{nd}}$ CPM are all with size $N * N^2$, that is, $2N^3$ dimensional features (N^3 for each direction) are used to describe the third order relationships of different states, thus dimensionality of third order features is much larger than that of lower order ones. High dimensionality introduces computational complexity during the training and testing phases of a supervised classifier, furthermore although single feature carry discriminative information respectively, when combined together there is little gain if they are highly correlated, finally, curse of dimensionality should also be taken into consideration. PCA is therefore introduced in our work for dimensionality reduction.

Let

$$Y = A^T(X - \bar{X}), \tag{10}$$

where X is original feature vector set with D rows (feature dimensionality) and K columns (number of samples), Y is the new feature vector set with d rows ($d <= D$) and K columns, $\bar{X}$ denotes the mean value of X. Correlation matrix R_y is defined as

$$R_y \equiv E(YY^T) = E(A^T(X - \bar{X})(X - \bar{X})^T A) = A^T cov_{X-\bar{X}} A, \tag{11}$$

where $cov_{X-\bar{X}}$ is the covariance matrix of $X - \bar{X}$ and it is symmetric, hence its eigenvectors are mutually orthogonal. When A is comprised of eigenvectors of $cov_{X-\bar{X}}$, R_y is then a diagonal matrix, i.e. features in Y are uncorrelated. Therefore A can be formulated as $A = [v_1, v_2, \cdots, v_d]$, $v_i(i = 1, 2, \cdots, d)$ is the eigenvector corresponding to λ_i of $cov_{X-\bar{X}}$ and $\lambda_1 > \lambda_2 > \cdots > \lambda_d$ which makes $var(v_1) > var(v_2) > \cdots > var(v_d)$. Y is therefore the projection of X onto the subspace spanned by the eigenvectors and the significance of features decreases with the growth of dimensionality.

There exist high correlations between third order statistical features, PCA can be used to avoid the information redundancies, and put most of discriminative information in the first few dimensionalities of features. Fig. 3 shows the correlation coefficient images and standard deviation distribution of original feature set and the PCA transformed feature set on image dataset [3]. Note that we employ gray image to signify the correlation coefficients between different pair of features, high gray value indicates high correlation and vice versa. It can be seen from Fig. 3 that original third order statistical features are highly correlated, while the features after PCA are uncorrelated, standard deviations of original feature set spread over a wide range, in contrast, standard deviations of PCA transformed feature set concentrate around the first few features and decrease dramatically with the increasing of dimensionality.

3 Experimental Results and Performance Analysis

3.1 Image Dataset

Columbia Image Splicing Detection Evaluation Dataset [3] is used in our experimental work. This image dataset consists of 933 authentic and 912 spliced images. It covers a variety of images. Images in this dataset are all in BMP format with fixed size of $128 * 128$. Spliced images are manipulated via two types of operations, crop and past along object boundaries and crop and past of horizontal or vertical strips. The spliced parts can be from the same image or from another image. Some of the images are given in Fig. 4.

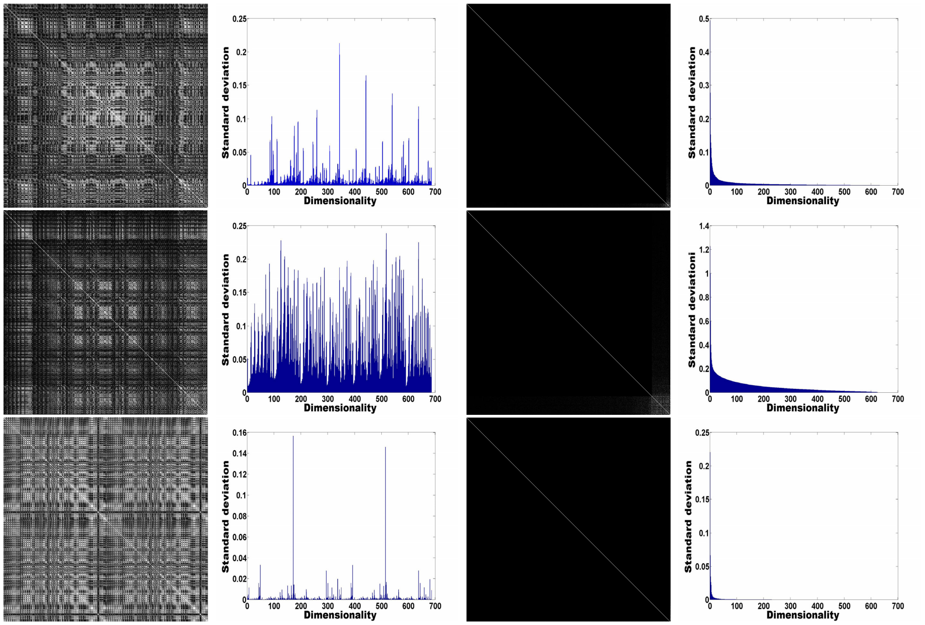


Fig. 3. Correlation coefficient images and standard deviation distributions of CCPM, 2nd Markov and 2nd CPM feature sets are shown from top to bottom. The first two columns indicate correlation coefficient images and standard deviation distributions of original features, the last two columns show the correlation coefficient images and standard deviation distributions of PCA transformed features.

3.2 Classifier

Support vector machine (SVM) is employed in our work to test the effectiveness of proposed method. LIBSVM [14] which is a library for SVM is used in our experimental work and radial basis function (RBF) is selected as kernel of SVM. Since SVM is a supervised machine learning method, for each experiment, 1/2 authentic images and 1/2 spliced images are randomly selected to train the SVM, and the left authentic and spliced images are used for test. Grid searching is employed to select the best parameters C (positive constant that controls the relative influence of the competing terms) and γ (variance of RBF kernel) for SVM. This procedure is repeated thirty times to eliminate the effect of randomness. Experimental results are evaluated by the average detecting accuracies over thirty times and receiver operating characteristics curves (ROC).

3.3 Comparisons and Performance Analysis

Detecting results of CCPM, 2nd Markov and 2nd CPM with different T values are given in Table 1 where TP and TN denote true positive rate and true negative rate respectively, Accuracy is the average detection rate over thirty runs.

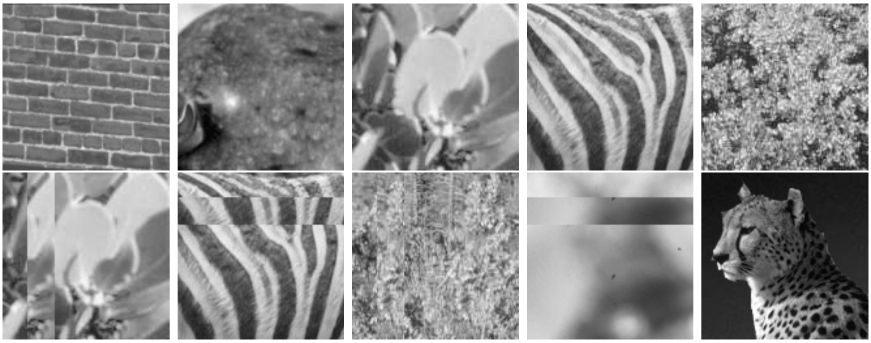


Fig. 4. Some examples in image dataset [3]. Authentic images are shown in the first row, spliced ones are given in the second row.

Standard deviations over thirty random tests are given in the parentheses. Detecting accuracies of the three proposed features increase with the value of T, however, the dimensionality of features increases dramatically with T. Larger T means more information taken into consideration, while it may also brings disturbing features and additonal computational complexity for classification. As can be seen in the table, detecting results of the three kinds of features with $T = 4$ is almost as good as that with $T = 3$. For above reasons we set $T = 3$ for the following experimental work.

To test the effectiveness of proposed features, third order statistical features with dimensionality 686 ($T = 3$) and second order statistical features [9] [10] (both of which are extracted in the 8*8 block DCT domain) with dimensionality 98 are given in Table 2. For third order statistical features, CCPM outperforms 2^{nd} Markov and 2^{nd} CPM. For second order statistical features, the average detecting accuracy of 1^{st} Markov is 2.7% higher than that of 1^{st} CPM. CCPM achieves 1.9% higher average detecting accuracy than 1^{st} Markov does. 2^{nd} Markov which is superior to 1^{st} Markov in steganalysis [13] doesn't demonstrate its superiority in image splicing detection.

The dimensionality of third order statistical features proposed in our work is 686 which is time consuming in training or testing phase of SVM. PCA is introduced for dimensionality reduction, first, we compute the mean value $\bar{X}_{train}$ of the training set X_{train} , then, A_{train} is formulated as the eigenvectors of $cov_{X_{train}-\bar{X}_{train}}$, finally we get the PCA features Y_{test} of the original test set X_{test} by $Y_{test} = A^T_{train}(X_{test} - \bar{X}_{train})$. Fig. 5 indicates comparisons of detecting performance among CCPM, 2^{nd} Markov, 2^{nd} CPM, 1^{st} Markov and 1^{st} CPM features after PCA dimensionality reduction. Note that we use the first 98 dimensional PCA features for the convenience of comparison.

From Fig. 5 we can see that

(1) Detection performance increases dramatically for the first few features, after that it vibrates on a small scale, therefore, we can use the first d dimensional ($d << D$) PCA features for classification without discriminative information loss.

Table 1. Detecting results of third order statistical features with different T value

		T=1, 54 D	T=2, 250 D	T=3, 686 D	T=4, 1458 D
CCPM	TP	85.4% (2.130)	85.2% (1.855)	86.6% (1.396)	85.9% (1.489)
	TN	78.5% (1.941)	81.4% (1.684)	90.9% (1.252)	91.9% (1.166)
	Accuracy	82.0% (0.885)	83.3% (0.952)	88.8% (0.805)	88.9% (0.918)
2$^{\text{nd}}$ Markov	TP	84.5% (1.872)	85.1% (1.439)	84.4% (1.639)	85.0% (2.020)
	TN	81.3% (1.865)	86.9% (1.618)	89.1% (1.498)	87.0% (1.615)
	Accuracy	82.9% (1.109)	86.0% (0.849)	86.8% (0.603)	85.9 (1.094)
2$^{\text{st}}$ CPM	TP	83.7% (1.663)	86.5% (1.786)	85.5% (1.788)	85.1% (1.955)
	TN	79.7% (1.976)	80.3% (1.878)	85.6% (2.085)	86.2% (1.892)
	Accuracy	81.7% (0.865)	83.5% (1.216)	85.5% (1.023)	85.6% (1.011)

Table 2. Detecting results of third and second order statistical features over image dataset [3]

	Third order features			Second order features	
	CCPM	2$^{\text{nd}}$ Markov	2$^{\text{nd}}$ CPM	1$^{\text{st}}$ Markov	1$^{\text{st}}$ CPM
TP	86.6% (1.396)	84.4% (1.639)	85.5% (1.788)	84.9% (1.617)	85.6% (1.742)
TN	90.9% (1.252)	89.1% (1.498)	85.6% (2.085)	89.0% (1.769)	82.8% (1.803)
Accuracy	88.8% (0.805)	86.8% (0.603)	85.5% (1.023)	86.9% (0.976)	84.2% (0.958)

(2) CCPM demonstrates the best detecting performance among the five features, 2$^{\text{nd}}$ CPM performs slightly better than 1$^{\text{st}}$ CPM does, 2$^{\text{nd}}$ Markov's detecting performance is as good as 1$^{\text{st}}$ Markov's.

(3) Compared with Table 2, PCA features with dimensionality larger than 30 can perform as well as original features do, in some cases, even better than the original features.

To testify the effectiveness of PCA for dimensionality reduction, boosting feature selection method proposed in [10] [15] is employed in our work for comparison, and the comparison results are given in Fig. 6. It can be seen from Fig. 6 that the performance of PCA dimensionality reduction outperforms that of boosting feature selection dramatically. Features in CCPM, 2$^{\text{nd}}$ Markov, 2$^{\text{nd}}$ CPM are highly correlated which have been proved in Fig. 3, that is, every single

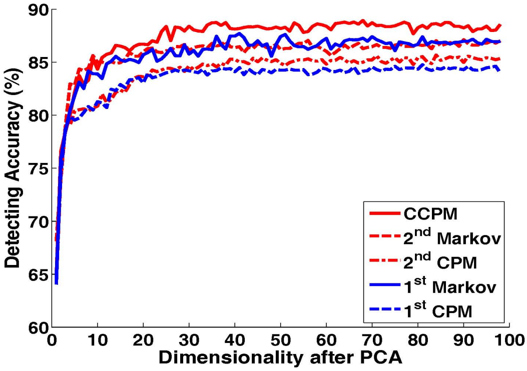
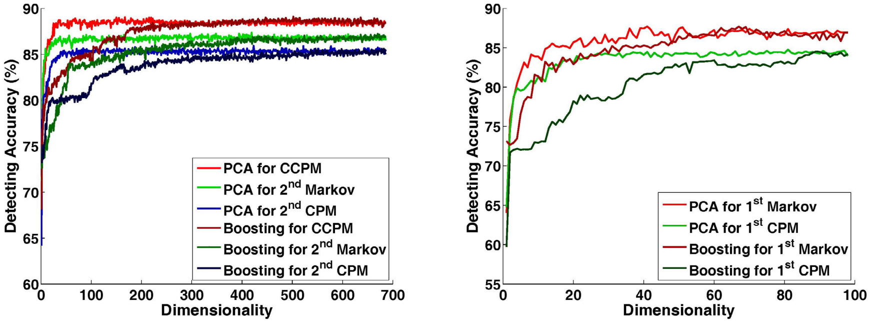


Fig. 5. Detecting performance comparisons among CCPM, 2nd Markov, 2nd CPM, 1st Markov and 1st CPM after PCA dimensionality reduction



(a) (b)

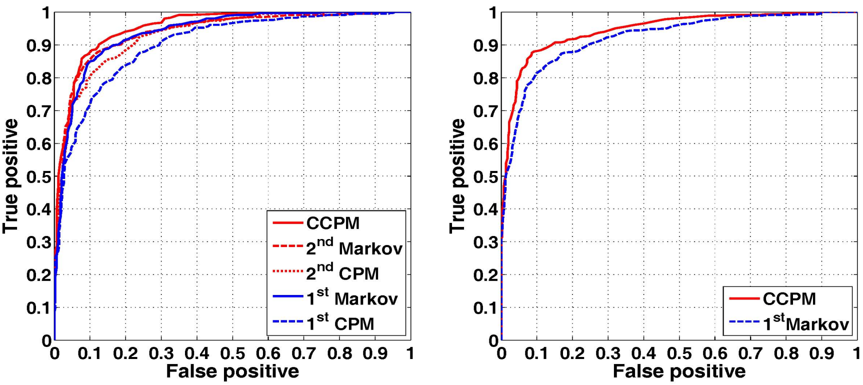
Fig. 6. Detection performance comparisons between PCA dimensionality reduction and boosting feature selection method. (a) Third order statistical features, (b) second order statistical features.

feature contains discriminative information but when combined together there is only a little gain which can be seen from the boosting selection procedure. In contrast, PCA maps the highly correlated features onto a new orthogonal coordinate system, the new gotten features are mutually uncorrelated, and the first few features can be considered as dominant features.

Finally, ROC curves of original features are given in Fig. 7 (a), ROC curves of the first 30 dimensional PCA features of CCPM and the first 30 dimensional PCA features of 1st Markov are presented in Fig. 7 (b).

3.4 Robustness Tests

Jpeg compression, Gaussian low pass filtering and image scaling are applied to verify the robustness of proposed method. Jpeg is a lossy compression method



(a) (b)

Fig. 7. ROC curves. (a) Comparisons among original features, (b) comparisons between the first 30 dimensional PCA features of CCPM and 1st Markov.

and the degree of compression can be adjusted via quality factors (usually in the range [1, 100]). To test the robustness of proposed features under Jpeg compression, we first compress the image dataset [3] with quality factor Q, then CCPM, 2nd Markov and 2nd CPM are extracted from the compressed images, finally the extracted features are fed into SVM for classification. The detecting results of the third order and second order statistical features under Jpeg quality factor Q are shown in the left part of Table 3, and the detecting results of the corresponding features after PCA are shown in the right part of Table 3. Gaussian low pass filtering is a commonly used to conceal the tampering traces. we work on the Gaussian low pass filtered image dataset, that is, Gaussian low pass filter with standard deviation σ is applied on all of the images in [3] and features are extracted from the filtered image dataset. The detecting results of the third order and second order statistical features under Gaussian low pass filter with standard deviation σ are given in the left part of Table 4, and the detecting results of the corresponding features after PCA are given in the right part of Table 4. Images are usually scaled in use, we also test the robustness of proposed features under image scaling attack. All of the images in [3] are scaled S times the size of original ones, and we test the effectiveness of the proposed features over the new image dataset. The detecting results of the third order and second order statistical features are given in the left part of Table 5, and the detecting results of the corresponding features after PCA are given in the right part of Table 5. As can be seen in Table 3, Table 4 and Table 5, Jpeg compression, Gaussian low pass filtering and image scaling degrade the detecting accuracies of both third order features and second order features. CCPM which demonstrates its superiority over image dataset [3] does not show better detecting performance than 2nd Markov, 2nd CPM and second order features. The features after PCA work as good as the original ones. Robust features will be further studied in our future work.

Table 3. Detecting results over Jpeg compressed image dataset [3]

	Original Features				Features after PCA			
	Q=95	Q=85	Q=75	Q=65	Q=95	Q=85	Q=75	Q=65
CCPM	82.8% (1.064)	74.7% (0.913)	73.3% (1.070)	67.1% (0.850)	82.1% (1.103)	73.8% (1.113)	72.4% (1.008)	67.3% (0.870)
2$^{\text{nd}}$ Markov	80.8% (0.985)	69.0% (1.500)	67.5% (1.188)	62.3% (1.198)	80.4% (0.881)	67.5% (1.232)	69.9% (1.119)	64.5% (1.192)
2$^{\text{nd}}$ CPM	79.7% (1.029)	74.8% (1.071)	73.9% (1.169)	69.5% (1.253)	79.4% (1.114)	74.5% (1.133)	73.1% (0.950)	69.5% (1.329)
1$^{\text{st}}$ Markov	82.8% (0.968)	73.5% (1.127)	72.5% (1.109)	67.0% (0.886)	82.1% (1.030)	74.1% (1.160)	72.6% (1.097)	67.2% (1.208)
1$^{\text{st}}$ CPM	80.7% (0.903)	75.2% (1.173)	74.4% (1.206)	69.1% (1.232)	80.5% (0.907)	75.3% (1.110)	73.7% (1.032)	68.6% (1.212)

Table 4. Detecting results over Gaussian low pass filtered image dataset [3]

	Original Features				Features after PCA			
	σ=0.5	σ=1	σ=1.5	σ=2	σ=0.5	σ=1	σ=1.5	σ=2
CCPM	86.4% (0.855)	75.1% (1.117)	73.8% (0.795)	75.0% (0.873)	85.6% (0.909)	75.2% (0.827)	75.3% (0.905)	73.9% (0.816)
2$^{\text{nd}}$ Markov	84.1% (1.259)	73.3% (1.189)	73.9% (0.931)	72.9% (1.131)	83.1% (1.192)	72.2% (1.584)	71.1% (1.328)	72.3% (0.977)
2$^{\text{nd}}$ CPM	82.6% (1.015)	76.2% (1.265)	75.4% (1.578)	75.5% (1.339)	82.2% (0.847)	76.2% (1.204)	75.1% (1.017)	75.3% (1.157)
1$^{\text{st}}$ Markov	86.3% (1.154)	75.3% (1.145)	75.0% (1.076)	73.8% (0.987)	86.2% (0.901)	75.3% (0.994)	75.0% (0.955)	75.9% (1.000)
1$^{\text{st}}$ CPM	82.8% (0.985)	72.1% (0.982)	71.4% (1.094)	72.5% (1.033)	82.8% (1.016)	71.8% (1.103)	70.9% (1.317)	71.9% (1.211)

Table 5. Detecting results over scaled image dataset [3]

	Original Features				Features after PCA			
	S =0.6	S =0.8	S =1.2	S =1.4	S =0.6	S =0.8	S =1.2	S =1.4
CCPM	68.0% (1.363)	72.0% (0.964)	79.3% (0.973)	79.5% (0.901)	69.2% (1.010)	72.5% (0.960)	79.8% (1.305)	79.2% (0.991)
2$^{\text{nd}}$ Markov	66.3% (0.876)	70.1% (1.173)	78.6% (0.968)	78.2% (0.793)	69.1% (1.262)	73.4% (1.175)	79.7% (0.987)	78.9% (1.269)
2$^{\text{nd}}$ CPM	69.4% (1.048)	71.9% (1.360)	78.7% (1.222)	78.1% (1.124)	69.1% (1.169)	72.5% (1.199)	78.8% (0.885)	77.3% (1.130)
1$^{\text{st}}$ Markov	67.8% (1.307)	71.0% (1.149)	77.1% (1.338)	75.9% (0.907)	67.9% (1.404)	72.3% (1.102)	77.1% (1.044)	76.0% (1.053)
1$^{\text{st}}$ CPM	68.6% (1.132)	72.0% (1.250)	75.9% (1.155)	76.3% (1.297)	68.3% (1.438)	71.9% (1.192)	76.3% (1.044)	76.2% (1.109)

4 Conclusions and Future Works

Passive image splicing detection is becoming a hot research topic. Different kinds of features have been proposed in the past few years, and 1st Markov features have been verified as one of the most effective features. In this paper, we model the thresholded adjacent difference 8*8 block DCT coefficient matrix of image to be tested as CCPM, 2nd Markov and 2nd CPM. That is we consider the states dependences within three neighboring states and the conditional co-occurrence probability, second order Markov transition probability and joint probability of three adjacent states are treated as discriminative feature. Higher order statistical features contain more discriminative information while the high dimensionality usually leads to computational complexity and overfitting for modern supervised classifier. PCA is therefore proposed for dimensionality reduction. To test the effectiveness of proposed method, image dataset [3] and LIBSVM [14] are employed for classification. Experimental results have shown that CCPM outperforms the other two third order statistical features, CCPM achieves 1.9% higher average detecting accuracy than 1st Markov does. PCA is verified as an effective tool for image splicing detection, it can reduce the dimensionality of original features greatly without losing discriminative information. We also test the robustness of proposed features under Jpeg compression, Gaussian low pass filtering and image scaling. We find that all of the above operations degrade the detecting performance of both third order features and second order ones. Despite higher dimensionality, the robustness of third order features is as good as that of second order ones. Robust higher order statistical features integrated with dimensionality reduction methods will be further studied in our future work.

Acknowledgements. This research work is funded by the National Science Foundation of China (61071152, 60702043), 973 Program (2010CB731403, 2010CB731406) of China, Shanghai Educational Development Foundation and Beijing Key Laboratory Program of Communication and Information System of Beijing Jiaotong University. Credits for the use of the Columbia Image Splicing Detection Evaluation Dataset are given to the DVMM Laboratory of Columbia University, CalPhotos Digital Library and the photographers listed in http://www.ee.columbia.edu/ln/dvmm/downloads/AuthSplicedDataSet/photographers.htm.

References

1. Farid, H.: A survey of image forgery detection. IEEE Signal Processing Magazine 26(2), 16–25 (2009)
2. Ng, T.T., Chang, S.-F., Sun, Q.: Blind detection of photomontage using higher order statistics. In: IEEE International Symposium on Circuits and Systems (2004)
3. Columbia DVMM Research Lab: Columbia Image Splicing Detection Evaluation Dataset, http://www.ee.columbia.edu/ln/dvmm/downloads/AuthSplicedDataSet/AuthSplicedDataSet.htm

4. Popescu, A.C., Farid, H.: Exposing digital forgeries in color filter array interpolated images. IEEE Transactions on Signal Processing 53(10), 3948–3959 (2005)
5. Johnson, M.K., Farid, H.: Exposing digital forgeries in complex lighting environments. IEEE Transactions on Information Forensics and Security 2(3), 450–461 (2007)
6. Chen, W., Shi, Y.Q., Su, W.: Image splicing detection using 2-d phase congruency and statistical moments of characteristic function. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, vol. 6505 (February 2007)
7. Huang, H., Guo, W., Zhang, Y.: Detection of copy-move forgery in digital images using SIFT algorithm. In: 2008 Pacific-Asia Workshop on Computational Intelligence and Industrial Application (2008)
8. Pan, X., Lyu, S.: Detecting image region duplication using SIFT features. In: Acoustics Speech and Signal Processing, ICASSP 2010 (2010)
9. Shi, Y.Q., Chen, C., Chen, W.: A natural image model approach to splicing detection. In: ACM Proceedings of the 9th Workshop on Multimedia & Security (2007)
10. Wang, W., Dong, J., Tan, T.: Effective image splicing detection based on image chroma. In: International Conference on Image Processing, ICIP 2009 (2009)
11. Zhao, X., Li, J., Li, S., Wang, S.: Detecting Digital Image Splicing in Chroma Spaces. In: Kim, H.-J., Shi, Y.Q., Barni, M. (eds.) IWDW 2010. LNCS, vol. 6526, pp. 12–22. Springer, Heidelberg (2011)
12. Bengio, Y., Delalleau, O., Le Roux, N.: The curse of dimensionality for local kernel machines. Technical report TR 1258 (2005)
13. Pevny, T., Bas, P., Fridrich, J.: Steganalysis by subtractive pixel adjacency matrix. IEEE Transactions on Information Forensics and Security 5(2), 215–224 (2010)
14. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines (2001), http://www.csie.ntu.edu.tw/~cjlin/libsvm
15. Dong, J., Chen, X., Guo, L., Tan, T.: Fusion Based Blind Image Steganalysis by Boosting Feature Selection. In: Shi, Y.Q., Kim, H.-J., Katzenbeisser, S. (eds.) IWDW 2007. LNCS, vol. 5041, pp. 87–98. Springer, Heidelberg (2008)

Blind Copy-Paste Detection Using Improved SIFT Ring Descriptor

Lin-na Zhou[1,2], Yun-biao Guo[2], and Xin-gang You[2]

[1] Department of Electronic Engineering, Tsinghua University, Beijing 100084, China
[2] Beijing Institute of Electronic Technology Application, Beijing 100091, China
zhoulinna@tsinghua.edu.cn

Abstract. Recent studies on digital watermarking and forensics techniques for multimedia security have been focused on digital multimedia forensics & anti-forensics. The digital forensic to detect image content tampering is a typical application of multimedia security. Based on improved scale invariant feature transform(SIFT), a new copy-move detection method using ring and sequence of each feature vector to ensure rotation invariance is proposed in this paper, which reduces the operator dimension of description. Experimental results show that the improved algorithm is more stable and faster when copy-paste fake combination in post-processing such as rotation, scaling and intensity adjustment.

Keywords: forgeries detection, copy-move forgery, SIFT(Scale Invariant Feature Transform), ring descriptor.

1 Introduction

Nowadays, the advent of low-cost and high-resolution digital cameras, and sophisticated photo-editing software make it remarkably easy to manipulate and alter digital images. The saying "seeing is believing" is no longer true in this digital world, and one would naturally ask whether the photo he receives is a real one [1]. A common manipulation when altering an image is to copy and paste portions of the image to conceal a person or an object in the scene. Region duplication is a simple and effective operation to create digital image forgeries, where a continuous portion of pixels in an image, after possible geometrical and illumination adjustments, is copied and pasted to a different image or a different location in the same image. Illegally duplication and tampering of the distributed content in the Internet may cause some troubles or even great economic loss to the digital image providers. This also becomes a serious issue when it comes to photographic evidence presented in the court or for insurance claims. Due to these problems, we need a reliable way to examine the authenticity of images content, even in a situation where the images look real and unsuspicious to human. The detection of digital tampering has become a crucial requirement.

Most existing region duplication detection methods are based on watermarking technology with data hiding in images or blind detection without data hiding . Several techniques based on data hiding in images have been designed as means for detecting

Y.Q. Shi, H.J. Kim, and F. Perez-Gonzalez (Eds.): IWDW 2011, LNCS 7128, pp. 257–267, 2012.

tampering. However, in practice, very few images are created with watermarks. Under most circumstances active approaches fail because there is no watermark to detect. This gives rise to research activities in passive blind image authentication that handle images with no previously added hidden information.

There are many previous works to detect and locate forged regions in an image without the help of data hiding [2-8]; for example, a novel method that could detect duplicate regions, which are produced by copy and paste operations, is proposed by Fridrich [4]. Alin C. Popescu.[5] present a technique that that employs a principal component analysis (PCA) on fixed-size image blocks, and lexicographic sorting to efficiently detect the presence of duplicated regions even in noisy or lossy compressed images. A similar method for detecting duplicated regions based on lexicographic sorting of DCT block coefficients was proposed in [8]. These approaches could detect simple duplicate regions in fake image.

2 Related Work

Several approachs in digital image forensics may be applied to detect duplicated regions considering the scenario in which a digital forgery is created by splicing together two or more images. In order to create a convincing match, it is often necessary to re-size, rotate, or stretch the images, or portions of them. Many existing region duplication detection methods are based on directly matching blocks of image pixels or transform coefficients, and are not effective when the duplicated regions have geometrical or illumination distortions. In contrast to these approaches, another alternative to the block matching based detecting feature is Scale invariant feature transform(SIFT) keypoints. SIFT can be used to perform reliable matching between different views of an object or scene. The features are invariant to image scale and rotation, and are shown to provide robust matching across a substantial range of affine distortion, change in 3D viewpoint, addition of noise, and change in illumination. Several previous works to detect and locate forged regions in an image based on SIFT feature were presented recently [11-14]. Huang et al [11] proposed an effective and robust method based on SIFT feature matching to detect Copy-Move forgery. The BBF (Best-Bin-First) search and detection threshold to improve the detecting accuracy were discussed in detail in ref [11]. Pan et al [12] gave a similar scheme for detecting Copy-Move tampering and the robustness of SIFT keypoints and features to image distortions is fully exploited. These work paid more attention on robustness and accuracy of detection however the detecting efficiency was considered little.

We propose a new copy-paste detection scheme based on improved scale invariant feature transform(SIFT) using ring and sequence of each feature vector to ensure rotation invariance in this paper. The scheme calculated all the eigenvectors of the tested image and performed division and matching operations on the set of eigenvectors. All the matching points were linked with lines located between the two regions obviously in a copy-paste falsified image in the absence of any digital watermark or signature. It can determine whether an image is forgery or not and locate the copy-move regions in post processing such as rotation. Experimental results demonstrate the effectiveness of the proposed scheme.

The rest of the paper is organized as follows: we analysis the copy-paste operation and its characteristic in section 3 and present our scheme in section 4; experimental results are given in section 5; finally, we conclude the paper in section 6.

3 Copy-Paste Operation and Characteristic

A common manipulation in removing an unwanted person or object from an image, is to copy and paste portions of the same image over the desired region. If the splicing is imperceptible, little concern is typically given to the fact that identical (or virtually identical) regions are present in the image. The sketch map of copy-paste fake in the same image and its effects are illustrated in Fig.1. The sketch map of copy-paste fake in two different image and its effects are illustrated in Fig.2.

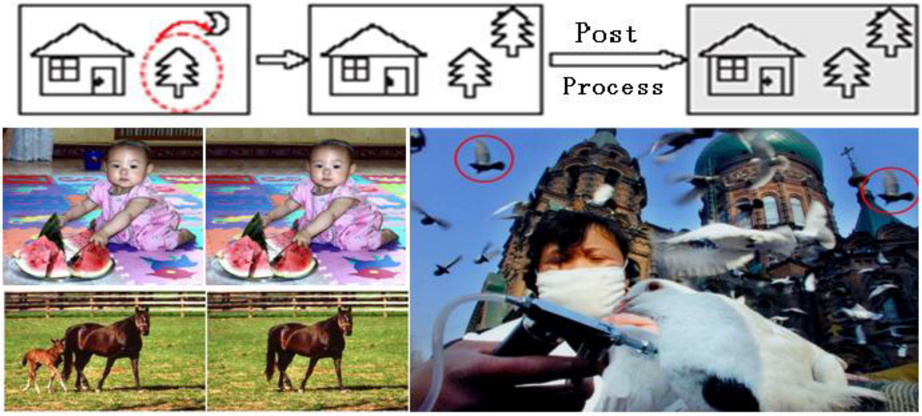


Fig. 1. The sketch map of copy-paste fake in the same image and its effects

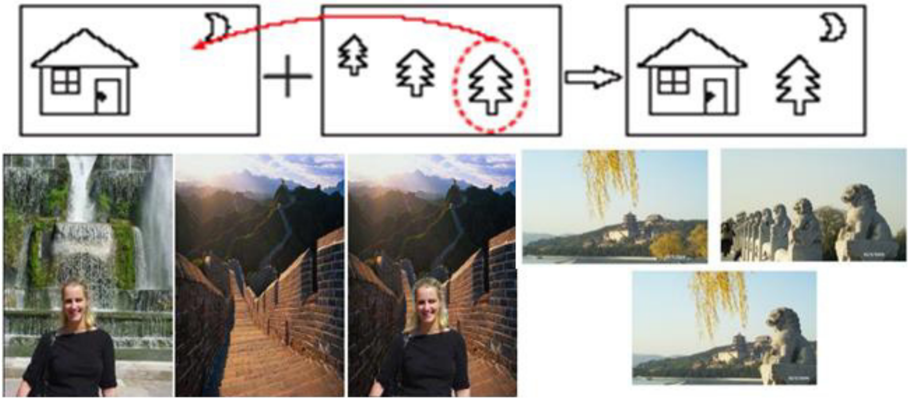


Fig. 2. The sketch map of copy-paste fake in two different image and its effects

In order to make a seamless and plausible fake image, it is often necessary to re-size, rotate, or stretch the images, or portions of them, considering the scenario in which a digital forgery is created by splicing together two or more images. The example of re-size, rotate, or stretch the images are illustrated in Fig.3.

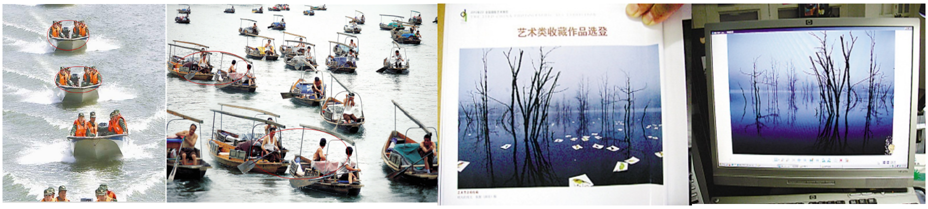


Fig. 3. Image copy-paste fake combination in post-processing such as resize, rotate

The main purpose of copy-move forgery detection methods is to find blocks of copied pixels. Figure 4 shows an overview of the copy-move forgery detection algorithm pipeline. Normally the given image is preprocessed first. After a possible preprocessing step, the image is tiled in small overlapping blocks. For every block a discriminating feature vector is computed. The main difference between the various methods lies in feature extraction. So the feature extraction is the key part in the designed scheme.

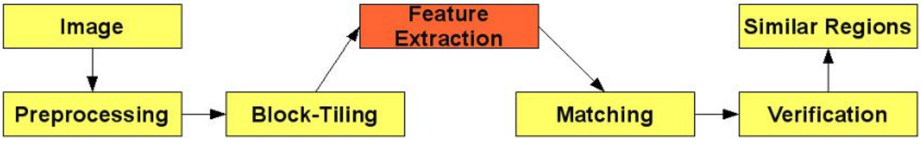


Fig. 4. The copy-move forgery detection algorithm pipeline

4 Detect Copy-Paste Using Improved SIFT Ring Descriptor

Lowe's patented Scale Invariant Feature Transform (SIFT) method [10] in International Journal of Computer Vision 2004,which can robustly identify objects even among clutter and under partial occlusion, because the SIFT feature descriptor is invariant to scale, orientation, and affine distortion, and partially invariant to illumination changes. SIFT algorithm for extracting distinctive invariant features from images that can be used to perform object or scene recognition, image index and matching. In this paper, we improved SIFT ring descriptor and used it to detect copy-paste fake.

The improved copy-paste detection using SIFT ring descriptor method approach is composed of five main steps: Scale-space extrema detection, keypoint localization, keypoint feature descriptor, dimensions of feature vector and feature matching.

4.1 Scale-Space Extrema Detection

The first stage of computation searches over all scales and image locations. It is implemented efficiently by using a Difference-of-Gaussian(DoG, see Figure 5) function to identify potential interest points that are invariant to scale and orientation.

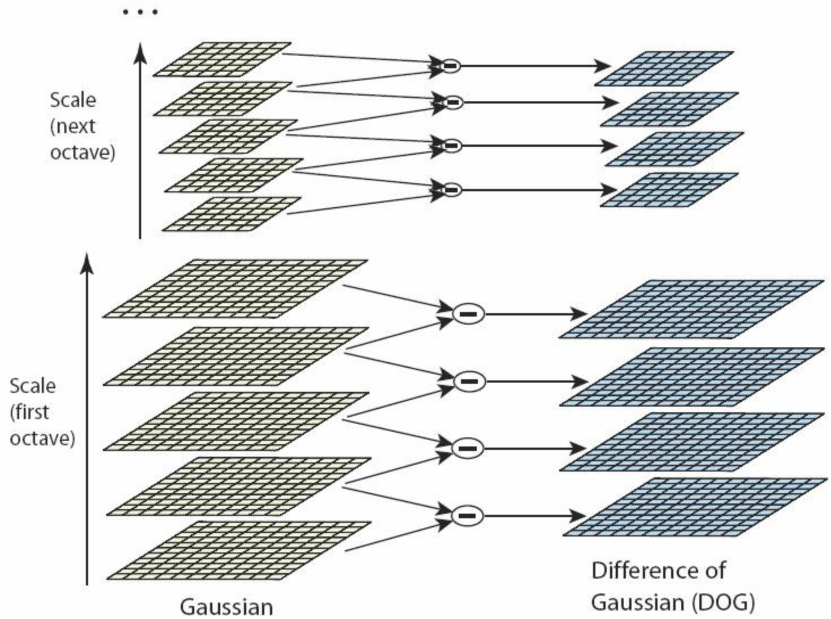


Fig. 5. DoG images are taken from adjacent Gaussian-blurred images per octave

To efficiently detect stable keypoint locations in scale space, Lowe [10] have proposed using scale-space extrema in the difference-of-Gaussian function convolved with the image, $D(x, y, \sigma)$, which can be computed from the difference of two nearby scales separated by a constant multiplicative factor k:

$$D(x,y,\sigma) = (G(x,y,k\sigma) - G(x,y,\sigma)) * I(x,y) = L(x,y,k\sigma) - L(x,y,\sigma) \quad (1)$$

Where $L\,(x, y, k\sigma)$ is the convolution of the original image $I\,(x, y)$ with the Gaussian blur $G\,(x, y, k\sigma)$at scale kσ.

4.2 Keypoint Localization

This is the stage where the interest points, which are called keypoints in the SIFT framework, are detected. In order to detect the local maxima and minima of $D(x, y, \sigma)$, each sample point is compared to its eight neighbors in the current image and nine neighbors in the scale above and below (see Figure 6). It is selected only if it is larger than all of these neighbors or smaller than all of them. The cost of this check is reasonably low due to the fact that most sample points will be eliminated following the first few checks. Maxima and minima of the difference-of-Gaussian images are detected by comparing a pixel (marked with X) to its 26 neighbors in 3x3 regions at the current and adjacent scales (marked with circles).

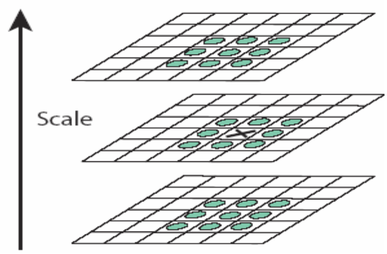


Fig. 6. Keypoint Localization in DoG scale space

4.3 Keypoint Feature Descriptor

Previous steps found keypoint locations at particular scales and assigned orientations to them. This ensured invariance to image location, scale and rotation. Now we want to compute a descriptor vector for each keypoint such that the descriptor is highly distinctive and partially invariant to the remaining variations such as scale, orientation, and affine distortion, etc.

Earlier work described a local descriptor to detect copy-paste [3,11-14]. In previous work, each keypoint is assigned one or more orientations based on local image gradient directions. Figure 6 illustrates the computation of the keypoint descriptor. A keypoint descriptor is created by first computing the gradient magnitude and orientation at each image sample point in a region around the keypoint location, as shown on the left. These are weighted by a Gaussian window, indicated by the overlaid circle. These samples are then accumulated into orientation histograms summarizing the contents over 4x4 subregions, as shown on the right, with the length of each arrow corresponding to the sum of the gradient magnitudes near that direction within the region. This figure shows a 2x2 descriptor array computed from an 8x8 set of samples, whereas the experiments in this paper use 4x4 descriptors computed from a 16x16 sample array. Therefore, the experiments in previous work use a 4x4x8=128 element feature vector for each keypoint.

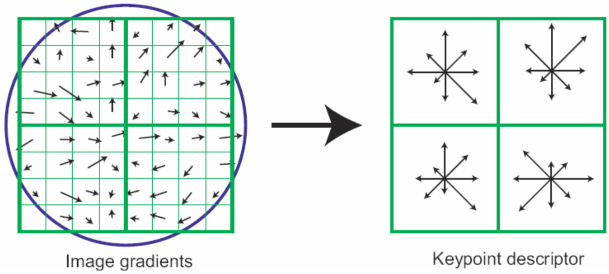


Fig. 7. Keypoint descriptor formed in [3]

Because of 128-dimensional description of the feature point reduces the efficiency of the Scale Invariant Feature Transform(SIFT) algorithm, we presents an improved SIFT algorithm in this paper, which uses ring and sequence of each feature vector to ensure rotation invariance, to improve algorithm in stability and speed when there are different levels of image geometric distortion, radiation distortion and noise.

It is well known that the content in cirque region will be invariant before and after rotation and scale distortion, so we construct the keypoint descriptor use circle, the keypoint descriptor is computed 12-dimensional gradient magnitude and orientation at each concentric circle in a region around the keypoint location, as shown in figure 8. The processes reduced the description of operator dimension from 128-dimensional to 24-dimensional.

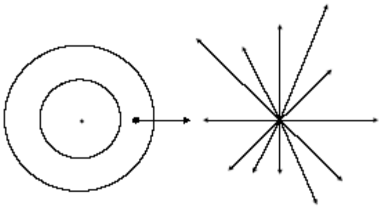


Fig. 8. Improved Keypoint descriptor formed in this paper

4.4 Dimensions of Feature Vector

The key parameter of circle keypoint descriptor is ascertain the dimensions of feature vector N, dimensions N determined by matching efficiency. We definition: Matching Efficiency(%)= Matching Right Rate(%)÷Matching Time(s), Figures 9 illustrate the relationship of Matching Efficiency, Matching Right Rate and Matching Time, now we select dimensions of feature vector N=12,while the circle keypoint descriptor have best matching efficiency.

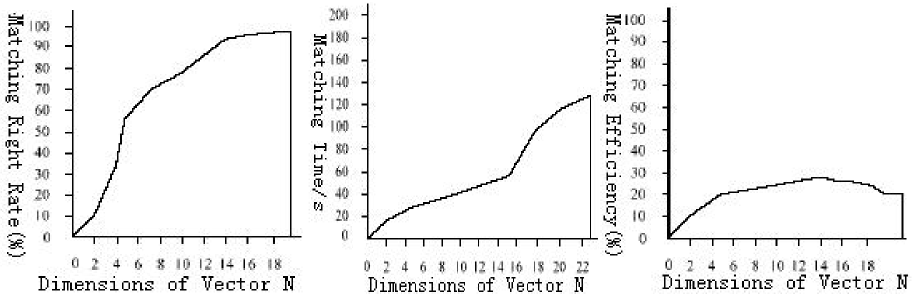


Fig. 9. The relationship of Matching Efficiency, Matching Right Rate and Matching Time

Now, it is the key step in achieving invariance to rotation as the keypoint descriptor can be represented relative to this orientation and therefore achieves invariance to image rotation. First, the Gaussian-smoothed image $L(x,y,\sigma)$at the keypoint's scale σ is taken so that all computations are performed in a scale-invariant manner. For an image sample $L(x,y)$at scale σ, the gradient magnitude, $m(x,y)$, and orientation, $\theta(x,y)$, are precomputed using pixel differences:

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2} \tag{2}$$

$$\theta(x,y) = \tan^{-1}\left(\frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)}\right) \tag{3}$$

We can get feature vector(d_1,d_2,···,d_{12}) through compute the grads cumulate plus values from inner cirque using grads histogram, and also get feature vector(d_{13},d_{14},···,d_{24}) from exterior cirque. Normalize the feature vector can reduce illumination influence of feature descriptive:

$$\overline{D} = \frac{D}{\sqrt{\sum_{i=1}^{12} d^2{}_i}} = \left(\overline{d1}, \overline{d2}, \cdots, \overline{d12}\right) \tag{4}$$

To ensure rotation invariance , we need reorder the sequence of each feature vector in terms of their ranks in an array sorted according measurement values, for example, in inner cirque $\overline{d5}$ =max{ $\overline{d1}, \overline{d2}, \cdots, \overline{d12}$ },and the final feature vector is $\left(\overline{d5}, \overline{d6}, \cdots, \overline{d12}, \overline{d1}, \cdots, \overline{d4}\right)$.

4.5 Feature Matching to Detect Copy-Paste

Feature matching introduces the nearest neighbor (NN) algorithm. In NN algorithm the best candidate match for each keypoint is found by identifying its nearest neighbor in test images. The nearest neighbor is defined as the keypoint with minimum Euclidean distance for the invariant descriptor vector. For each keypoint, we search all over the image to get the nearest neighbor point and the second-closest neighbor point, comparing the distance of the nearest neighbor to that of the second-closest neighbor. If the ratio is less than the previous appointed threshold, matching of the pair of points is determinant .

In Figure 10 we describe our feature matching flow to detect copy-paste. Six major steps make up the pipeline of a copy-paste detect system base on feature matching:

5 Experimental Results

The performance of the SIFT copy-paste detection scheme has been tested by experimental using Visual Studio Platform. Different from other prior studies, some of our test images are collected from the Internet (News photo), and some test pictures were taken by digital camera. More real-life images were used to achieve a more practical method for digital image forgery detection. The experimental results are listed in Table1. Some examples demonstrating the effectiveness of the proposed scheme is shown in Fig.11.

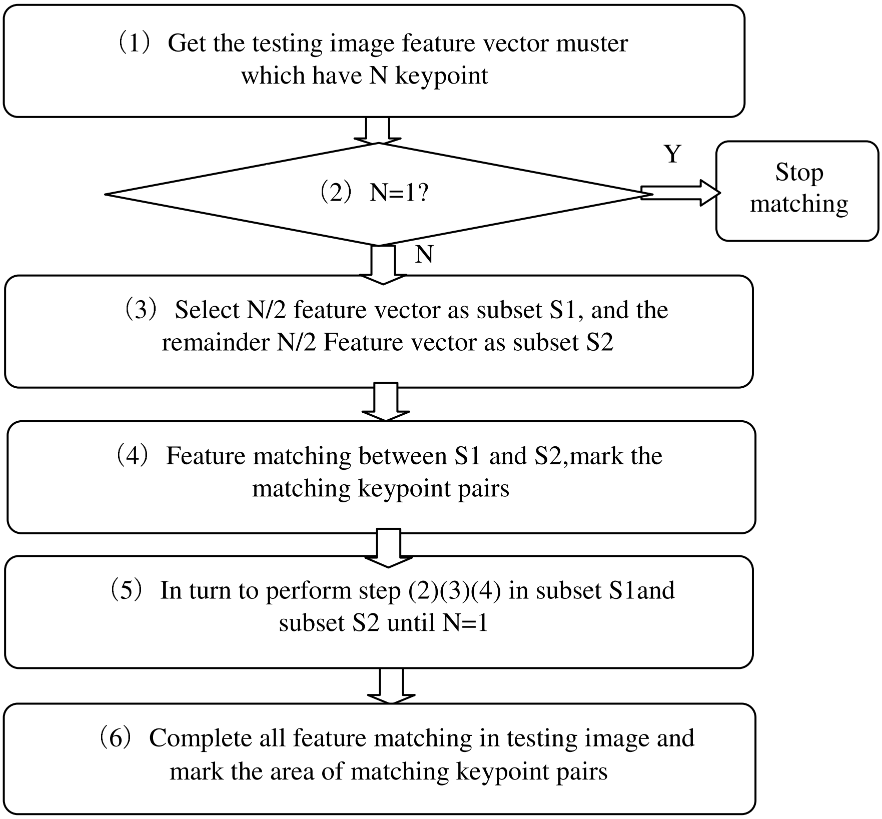

（1） Get the testing image feature vector muster which have N keypoint
（2） N=1?
Y
Stop matching
N
（3） Select N/2 feature vector as subset S1, and the remainder N/2 Feature vector as subset S2
（4） Feature matching between S1 and S2,mark the matching keypoint pairs
（5） In turn to perform step (2)(3)(4) in subset S1and subset S2 until N=1
（6） Complete all feature matching in testing image and mark the area of matching keypoint pairs

Fig. 10. Feature matching to detect copy-paste flow

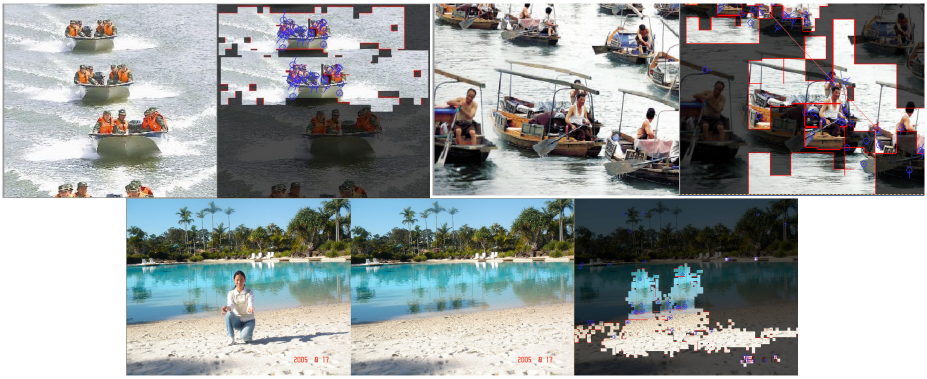


Fig. 11. Examples of detection in post-processing(detection time is 15s)

For various image transformations post-processing applied to a sample of 256 images(800×640 pixels) , table1 gives detection rate and calculation time(the average test time of every image) compared with other current copy-paste detection algorithm.

Table 1. Copy-paste detection of combination in post-processing

Image post-processing	Farid H method in [5]		Fridrich J method in [4]		Li method in [3]		Our method in this paper	
	Detection Rate(%)	Detection time(s)	Detection Rate(%)	Detection time(s)	Detection Rate(%)	Detection time(s)	Detection Rate(%)	Detection time(s)
A. Rotate by 20 degrees	0	×	0	×	85	68	99	15
B. Scale by 0.7	0	×	0	×	81	65	98	15
C. Increase contrast by 1.2	28	320	22	380	80	60	90	16
D. Decrease intensity by 0.2	25	320	21	380	82	65	95	18
E. Stretch by 1.5	20	320	25	380	88	70	92	15
F. Add 10% pixel noise	81	320	45	380	85	60	96	16
G. Gauss blur in 0.5 pixel width	30	320	20	380	81	65	88	16
H. All of A,B,C,D,E,G.	0	×	0	×	70	80	95	20

6 Conclusion and Future Direction

In this paper, we presented a fast copy-paste detection scheme which outperforms the current copy-paste detection algorithm, both in speed and accuracy. Aiming at the problems that 128-dimensional description of the feature point reduces the efficiency of the Scale Invariant Feature Transform(SIFT) algorithm, an improved SIFT algorithm using ring and sequence of each feature vector to ensure rotation invariance is discussed in this paper, which reduces the operator dimension of description. Experimental results show that when there are different levels of image geometric distortion, radiation distortion and noise, the improved algorithm is more stable and faster.

Digital tampering may affect image characteristics in many aspects, and the work based on a single feature described in this paper reveals just a small fraction of the image forensics detection. With the development of digital forgery, digital detection could hardly keep pace with digital tampering only depending on single digital forensic tool. Though still having a lot of room for improvement, the scheme we propose can serve as a well-posed starting point for future exploration in this direction. The future digital forensic direction would be multiplex forensic tools in conjunction with awareness and sensible policy and law to create convincing digital forgeries.

Acknowledgments. This research was supported by a grant of the National Natural Science Foundation of China (No. 60970148, 61170175, 61003284), the National Foundation Theory Research of China (973 Program, No. 2007CB310806) and China Postdoctoral Science Foundation (No. 20090450429).

References

1. Farid, H.: Creating and Detecting Doctored and Virtual Images: implications to The Child Pornography Prevention Act [EB/OL], `http://www.ists.dartmouth.edu/library/tr-2004-518.pdf`
2. Linna, Z.: Study of Digital Forensics Based on Image Content, PhD thesis, Beijing University of Posts and Telecommunications, Beijing, China (2007)
3. Li, S., Zhang, A., Zheng, Y., et al.: Detection of copy-move image forgeries based on SIFT. Journal of PLA University of Science and Technology(Natural Science Edition) 10(4), 339–343 (2009)
4. Fridrich, J., Soukal, D., Lukáš, J.: Detection of copy-move forgery in digital images. In: Proceedings of Digital Forensic Research Workshop, pp. 5–8 (2003)
5. Popescu, A.C., Farid, H.: Exposing digital forgeries by detecting duplicated image regions. TR2004-515, Hanover, NH, USA: Department of Computer Science, Dartmouth College (2004)
6. Wu, Q., Li, G., Sun, S., et al.: A sorted neighborhood approach for detecting duplicated regions in image forgeries based on DWT and SVD. Acta Automatica Sinica 34(12), 1458–1466 (2008)
7. Luo, W.Q., Huang, J.W., Qiu, G.P.: Robust detection of region-duplication forgery in digital image. In: Proceedings of 18th International Conference on Pattern Recognition, pp. 746–749. IEEE (2006)
8. Dybala, B., Jennings, B., Letscher, D.: Detecting filtered cloning in digital images. In: Proceedings of the 9th Workshop on Multimedia and Security, pp. 43–50. IEEE (2007)
9. Luo, W.: Study on Passive Multimedia Forensics PhD thesis, Sun Yat-sen University, Guangzhou, China (2008)
10. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2), 91–110 (2004)
11. Huang, H., Guo, W., Zhang, Y.: Detection of copy-move forgery in digital images using SIFT algorithm. In: Proc. IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application, Wuhan, China (2008)
12. Pan, X., Lyu, S.: Region Duplication Detection Using Image Feature Matching. IEEE Trans. Information Forensics and Security 5(4), 857–867 (2010)
13. Amerini, I., Ballan, L., Caldelli, R., Del Bimbo, A., Serra, G.: Geometric tampering estimation by means of a sift-based forensic analysis. In: ICASSP (2010)
14. Pan, X., Lyu, S.: Detecting image region duplication using SIFT features. In: ICASSP, Dallas, USA (2010)

Camera Model Identification Based on the Characteristic of CFA and Interpolation

Shang Gao[1], Guanshuo Xu[2], and Rui-Min Hu[1,*]

[1] National Engineering Research Center for Multimedia Software, Wuhan University, China
email.nancy.g@gmail.com, hrm1964@public.wh.hb.cn
[2] Department of Electrical and Computer Engineering, NJIT, New Jersey
gx3@njit.edu

Abstract. In this paper, we propose a camera-model classification method based on characteristics of color filter array (CFA) and interpolation. As CFA patterns and interpolation algorithms are different among different camera models, the artifacts introduced by CFA and interpolation can reflect model-specific to some extent. To capture the artifacts, we design a 69-D feature set and perform camera-model classification. Images from seven camera models in the Dresden Image Database are chosen as our experiment database. Experiment results show that in seven models detection, our method can do the classification with high detection accuracy from 98.39% to 99.88%.

Keywords: CFA, interpolation, camera model identification, demosaicing.

1 Introduction

Recent years, analog multi-media format is replaced by digital counterpart rapidly because of its low-cost, real-time and easy to store or transfer. These excellent characteristics make more and more digital multi-media come into our life, such as digital camera, video surveillance, etc. These applications entertain us and also make it possible for us to collect digital copy version as evidence in courts. However, every coin has its two sides. Except for its convenience, they also can be easily changed, even faked. For the purpose of court usage, however, evidence is needed to be proved as authentic. The source and content of it needs to be credible. Source camera model identification tries to address the problem of image source verification and is the topic we will discuss in this paper.

There are two approaches of camera source identification in today's digital forensic field. One is to identify various camera brands and models; the other is to recognize individual properties in each camera.

The first approach can be achieved by finding out the difference of hardware component and digital image processing (DIP) technologies which vary from camera-model to model. To find out these differences, the detection works often focus on some special processing procedures, such as optical distortions in lens systems, sensor

[*] Corresponding author.

Y.Q. Shi, H.J. Kim, and F. Perez-Gonzalez (Eds.): IWDW 2011, LNCS 7128, pp. 268–280, 2012.

resolutions, CFA patterns and interpolation algorithms, quantization process, and other post processing, etc.

Some associated works have been published in recent years. For instance, J. Lucas et al. [1], M. Chen et al. [2] and Chang-Tsun Li [3] average multiply noise images, which is extracted from images by wavelet de-noising filter, to build a reference fixed noise pattern. The identification is achieved by calculating correlation between test noise image and reference pattern. This method is sensitive to geometrical transformation, such as re-sample, crop, etc. M. Kharrazi et al. [4] approach this issue by machine learning. They first extract three kinds of statistics from images, such as color features, wavelet statistics and image quality metrics, and then combine them as the input of SVM classifier for camera model identification. It can achieve 97% correct detection rate on four different cameras. But for different modes in one brand, like 5 cameras with 3 same brands, result drops to 88%. K. S. Choi et al. [5] also use machine learning to do model detection. They use radial distortion parameters as input of SVM for classification. But it is highly affected by focal length of lens. Guan-Shuo Xu et al. [20] propose a camera model identification algorithm by extracting the Markov transition probability matrix as features, and the method can achieve 92.5% correct detection rate under multi-models. Beyond the above mentioned approaches, some specific applications are presented in this field in recent years, such as E. Dirik et al. [10] used dust model to discriminate individual DSLR cameras. O. Celiktutan et al. [11] combined several previous forensics features as input of SVM classifier to identify source cell phones.

There are many proposed detection methods focus on CFA and interpolation in camera image processing. In [6-9], the demosaicing algorithm in camera is modeled as a linear filter. The weights of the linear filters are estimated by EM algorithm after considering various CFA patterns, the estimated weights and estimated error are combined to be features. The result is good but also costs lots of time for computation.

In this paper, a camera-model detection method is proposed. We present 69-D features to capture the artifacts introduced by CFA and interpolation. The 'Dresden Image Database' is used as our experiment database. The results show that our method can do camera-model detection with low dimension features and high detection accuracy.

This paper is organized as follow: Section 2 describes CFA pattern and interpolation in camera pipeline, then the artifacts caused by CFA and its interpolation. Section 3 introduces the features we propose in this paper, and also includes feature extraction steps. The experiment and results are presented in section 4. Section 5 concludes this paper. Section 6 is acknowledgements.

2 Artifact Introduced by CFA and Interpolation

2.1 CFA and Its Interpolation in Camera Imaging

Taking a photo involves a series of processing inside a camera, which can generate a final digital image from initially captured scene light. Sensor is an indispensable unit in these processing, which transfer the light into a digital image. It will be good if each sensor pixel can respond to all color components of light beam. But the fact is most commercial camera manufactures use single color pixel sensor to reduce cost. So, before CCD sensor, a color filter array which is composed of filters with three to four color disciplinary hybrid, like RGB (read, green, blue), will be used. This array will make sure to recover approximately real scene image from 'mosaicking' image (the output of sensor) become possible. One of recovery work is recovering the lacked color, called CFA interpolation, or demosaicking. Fig.1. shows common camera pipeline. Red dotted part is units which do CFA and interpolation processing.

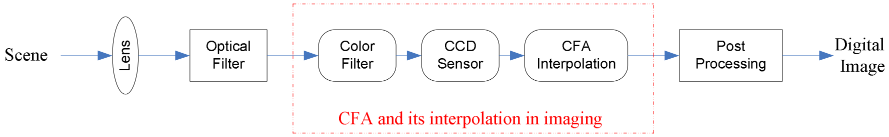


Fig. 1. Camera Pipeline

2.2 Artifacts Brought by CFA and Its Interpolation

Common consumer-level digital cameras are single-sensor and thus require color sampling for each pixel location. The color filter arrays (CFA) which are placed before sensors perform the sampling. For simple implementation, CFAs are usually periodic and form certain pattern. Bayer pattern is the most popularly implemented CFA pattern. After CFA color sampling, the lost color components are recovered by interpolation. While certain camera models use fixed CFA pattern and interpolation algorithms, those sampling and interpolation could be different in different camera models, especially among cameras produced by different manufacturers. Hence, artifacts introduced by CFA and interpolation should reflect model-specific difference to some extent, which can be utilized for model classification.

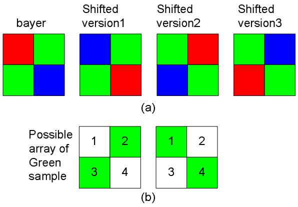


Fig. 2. (a) Bayer pattern and its shifted versions; (b) its possible arrays of Green sample

Due to the CFA sampling and interpolation process, parts of the pixel color values are originally recorded and the rest are estimated or calculated. For example, if we consider Bayer pattern, half of the green values are original and the other half are interpolated.

The process of interpolation is to estimates lack color component by its existing neighbor color components. It can be regarded as a similarly weighted average processing and has low-pass nature [8, 9]. Theoretically, it leads to the fact that interpolated color components are smoother than the original part statistically.

After camera imaging, photos can be considered as two parts, real scene data and noise. Real scene data is the true image information that we take and expect. It is the main information in photo. Noise is generated from nearly everywhere during camera imaging, such as shot noise [12] [13], fixed pattern noise (FPN) and photo-response non-uniformity noise (PRNU) [13]. The artifacts introduced by interpolation are small, which is covered by real scene data. Hence, the artifact is hard to observe using relative statistics of photo directly. Noise also has two parts after interpolation, real noise obtained by sensor and 'fake' noise generated by interpolation in terms of its neighbor noise. We call raw noise and estimated noise here for short. Since the energy of noise is small, low-pass nature of interpolation can be reflected more obviously by the statistic of noise part without image content.

The definition of raw and estimated noise sequences is, for example, according to the Bayer pattern and its shift versions showed in Fig. 2 (a), we can get two possible sample arrays of green channel denoted in Fig. 2 (b). In this case, for green channel, if an image is divided into many non-overlapping 2×2 blocks, and the CFA pattern is Bayer or shift Bayer pattern version 1, then its raw noise sequence here is collection of noise at pixel position 2 and 3 in these 2×2 blocks. Its estimated noise sequence here is that at pixel position 1 and 4 in these 2×2 blocks. If the CFA pattern is shift Bayer pattern version 2 or 3, the situation is the opposite.

Variance is the second order of statistics. It can reflect smoothness of an image signal. Theoretically, while the image is photography, variance of raw noise and estimated noise should be different in each color component.

As different interpolation algorithms have different smoothing effect, the ratio of variance of raw noise sequence and estimated noise sequence will vary from camera to camera. Hence CFA and interpolation artifacts can be measured by taking the ratio of noise variances at raw and estimated pixel positions [16]. To ensure that the statistics we calculate is greater than 1, the calculation of maximum is introduced after calculating the ratio of variance. Hence, the basic estimation statistic is determined as equation (1).

$$\max\left(\frac{\mathrm{var}(N^r)}{\mathrm{var}(N^e)},\frac{\mathrm{var}(N^e)}{\mathrm{var}(N^r)}\right) \tag{1}$$

Where N^r and N^e denote raw and estimated noise sequences respectively, while 'var' and 'max' present variance and maximum calculation respectively.

In practice, though advantage interpolation algorithms are more complex and highly signal-adaptive and linearity assumption may be oversimplified, it has been successfully applied to several other CFA based forensic approaches [9].

3 Features

3.1 Feature Set 1: Variance Feature

More than one kind of CFA patterns are employed in different camera brand or model. By only considering one pattern, one cannot get enough statistics to discriminate a couple of camera models. Besides, interpolations in green channels and red/blue channels are usually different. Hence, for discrimination feature response to camera identification, we choose four simple CFA patterns as reference possible patterns [15]. In view of the fact that red and blue channels have same patterns and most likely they share same interpolation algorithm, only green and red channels are used.

Fig3 (a), (b), (c), (d) is the four CFA patterns (Bayer, modified Bayer, Diagonal strip, and vertical striped patterns) we use respectively; Fig3 (a1-a2), (b1-b2), (c1-c6), (d1-d6) denote the possible green sample array of four CFA patterns respectively; Fig3 (a3-a6), (b3-b6), (c7-c12), (d7-d12) denote the possible red sample array of four CFA patterns respectively. For every possible pattern, an image can be divided into two position collections, raw and estimated positions. Approximate noise image can be obtained from photo using de-noising filter. According to the raw and estimated positions, we can get raw noise sequence and estimated noise sequence. From every possible block, we can find statistics by equation (1). There are 36 possible blocks for green and red sample arrays as shown in Fig 3, which means we can get 36 statistics here as features to capture the artifacts caused by interpolation. Among these features, in mathematical sense, the four green possible arrays for Bayer and modified Bayer pattern lead to same result. So we get rid of three cases and finally get 33 features calculated by equation (2).

$$\max\left(\frac{\mathrm{var}(N_{p_i}{}^{r})}{\mathrm{var}(N_{p_i}{}^{e})}, \frac{\mathrm{var}(N_{p_i}{}^{e})}{\mathrm{var}(N_{p_i}{}^{r})}\right), i = 1, \ldots, 33 \tag{2}$$

Where $N_{p_i}{}^{r}$ and $N_{p_i}{}^{e}$ denote raw and estimated noise sequences under the i^{th} possible sample array.

A) Effect on same CFA pattern, different interpolation.

Different interpolation algorithms use different interpolation coefficients, which lead to different ratio. It can be seen as in fig 4, fig.4 (a-c) show the original image, and the re-interpolated images by bilinear and bicubic respectively; Fig.4 (d) shows 33 variance ratio statistics from these three images under Bayer CFA pattern. We can see that even under same condition, different interpolation algorithm do affect representation of these statistics. In other words, the character of different interpolation can be captured by these statistics even under same CFA pattern.

B) Effect on various CFA patterns by limited reference CFAs

In this paper, we only consider calculating statistics from four reference CFA patterns, which may not cover all the patterns in reality. Even if the reference patterns does not hit the CFA pattern of test camera, the statistical features we extract can still

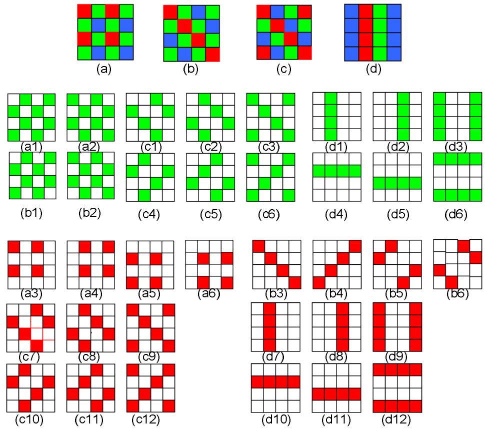
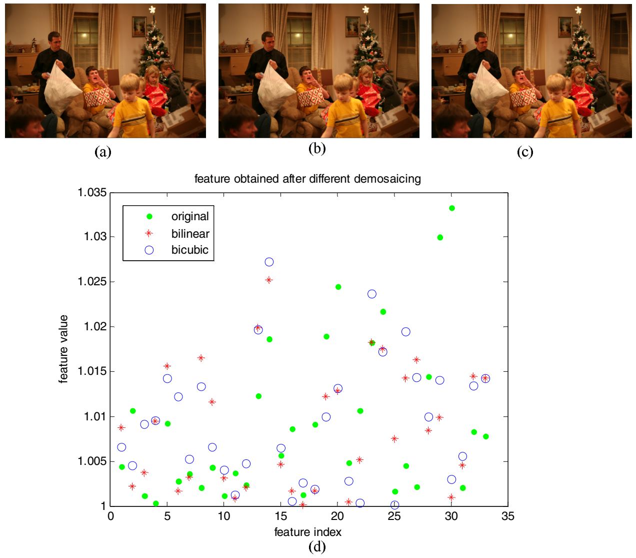


Fig. 3. (a) Bayer Pattern [15]; (b) modified Bayer Pattern [15]; (c) Diagonal Strip Pattern [15]; (d) Vertical Striped Pattern [15]


(a)

(b)

(c)

(d)

Fig. 4. (a) Original image; (b) re-interpolated image by bilinear; (c) re-interpolated image by bicubic; (d) variance features extracted from above image in (a), (b), (c) respectively

reflect the difference between camera and camera to some extent due to the fact that most CFA patterns we know are periodic, which means that the percentage of raw and estimated noise pixel numbers at reference hypothesis CFA sampled and interpolated pixel positions are certain.

3.2 Feature Set 2: Inter-color Statistics

A popular color difference interpolation scheme utilizes inter-channel correlation between colors to do interpolation. According to constant hue assumption, the color difference components are perfectly smooth within image objects [18]. If one color channel is fully available, the others can be recovered by low-pass filtering the difference plane [19]. In color interpolation, if we assume that the full G is available by some interpolation process, R can be recovered via

$$R = I\{R_s - G_s\} + G \tag{3}$$

Where $I\{\bullet\}$ denotes linear low-pass filtering [19]; R_s , G_s denotes red and green color pixel values at the neighbor red sample position. Equation (3) can be written as equation (4).

$$R - G = I\{R_s - G_s\} \tag{4}$$

According to equation (4), we find that in this kind of interpolation, the difference of inter color channels after interpolation can be obtained from linearly low-pass filtering on difference of its neighbor sample inter channels. In fact, the linear low-pass nature is part of interpolation kernel filter's characters. Hence, the difference of inter color channels after interpolation is smoothed, and the variances of difference between every two different noise images will be different because of this kind of smoothing process. Feature set 2 is designed to capture this kind of process and can be presented as equation (5-7).

$$\text{var}(N^R - N^G) \tag{5}$$

$$\text{var}(N^B - N^G) \tag{6}$$

$$\text{var}(N^R - N^B) \tag{7}$$

Where N^R , N^G , N^B denote noise image of red, green, and blue channel respectively.

3.3 Feature Set 3: Kurtosis Statistics

Apart from the features that calculated from variance of noise images in Section 3.1, we also use kurtosis as one of our statistical measure. Kurtosis is a statistic can distinguish the 'peakedness' of probability distribution of a real-value data, which is also sensitive to low-pass nature of interpolation algorithms. Minor difference in two series of data may have close variance but different kurtosis. Hence, based on the first feature set, we replace the variance statistics by kurtosis statistics and get 33 features as our third feature set. For the 33 possible sample arrays, kurtosis can reflect minor difference of interpolation. Hence, feature set 3 can be calculated as equation (8).

$$\max\left(\frac{kurtosis(N_{p_i}{}^{r})}{kurtosis(N_{p_i}{}^{e})}, \frac{kurtosis(N_{p_i}{}^{e})}{kurtosis(N_{p_i}{}^{r})}\right), \quad i = 1, \ldots, 33 \tag{8}$$

3.4 Feature Extraction Steps

The Feature Extraction steps are as follows.

Step 1: List 33 possible red and green sample arrays mentioned in section 3.1;

Step 2: Get noise image from photo by de-noising filter [22];

Step 3: According to 33 possible sample arrays in step 1, find out 33 possible combinations of raw noise sequence and estimated noise sequence;

Step 4: In terms of the 33 possible combinations in step 3, obtain 33 features calculated by equation (2) as feature set 1;

Step 5: In terms of the 33 possible combinations in step 3, obtain 33 features calculated by equation (8) as feature set 3;

Step 6: According to noise image in step2, obtain 3 features calculated by equation (5-7) as feature set 2;

Step 7: Combine feature set 1, set 2, and set 3 obtained by step 4, step 6 and step5 as 69-D feature set, this 69-D feature set is the features we proposed in this paper.

The feature extraction flow diagram is showed in Fig.5.

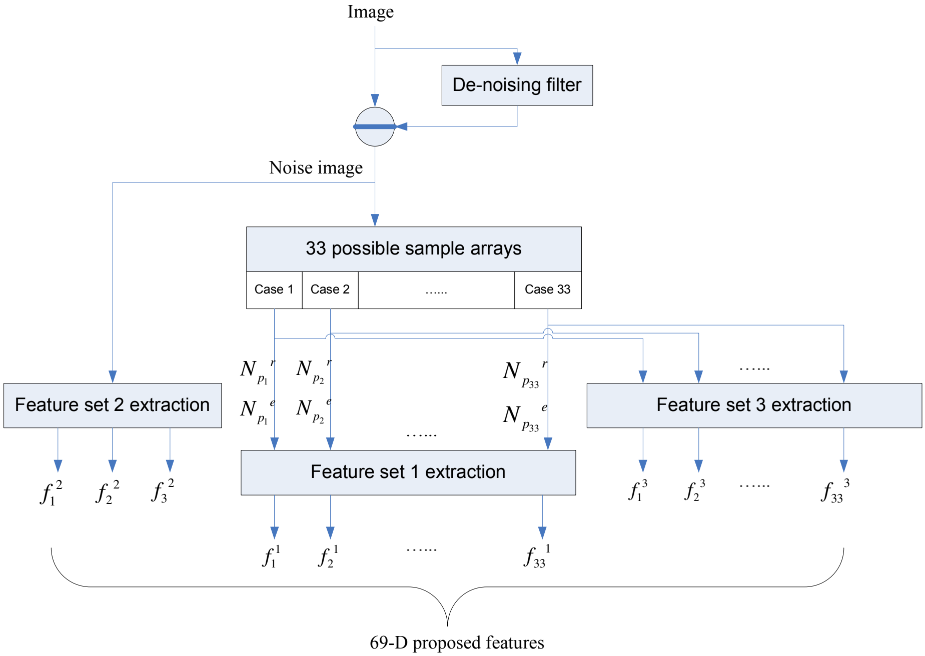


Fig. 5. Feature extraction flow diagram

Where f_1^1 - f_{33}^1 denote 33 features in feature set 1; f_1^3 - f_{33}^3 denote 33 features in feature set 3; f_1^2 , f_2^2 and f_3^2 denote 3 features in feature set 2;

4 Experiment

4.1 Image Database

We use the 'Dresden Image Database' as our experiment image database. The Dresden image database is a public database designed for benchmarking algorithms in the area of digital image forensics. Most images are taken under same or similar acquisition procedure, such as at the same scenes, same taken positions, and same up to two motives with tripods in Dresden, and photographed with each camera of one set with systematically varying camera setting (flash, focal length and interchanging lens, if possible) [21]. The cameras used to build the dataset are different, which can be categorized in term of devices, models, bands. Therefore, the Dresden image database is collection of same or similar scene images taken by different camera, which can be categorized to do manufactory, model, or device detection or etc. Dresden database hasn't been finished while experiment. Only the images taken from more than one device per model are suitable for model detection. Hence, images from seven models have been chosen. Some details are shown in Table 1.

Table 1. Camera model

No.	model	Device num/model	Image num/model	Image resolution	Image format
1	CanonIxus70	3	567	3072×2304	JPEG
2	CasioEXZ150	5	925	3264×2448	JPEG
3	FujiFirmFinePixJ50	3	630	3264×2448	JPEG
4	NikonCoolPixS710	5	925	4352×3264	JPEG
5	NikonD70s	2	367	3008×2000	JPEG
6	NikonD200	2	752	3872×2592	JPEG
7	KodakM1063	5	2391	3664×2748	JPEG

4.2 Experiment Method

There are three portions in our experiment, image blocking, feature extraction, and model classification.

To increase the number of classifying samples and unify sample size, we extract four 512x512 sub-blocks from center of each image. The sub-block position is showed in Fig. 6.

Although the original images are JPEG images, our features are extracted from spatial domain (RGB color space). Each sub-block is a small three color channels RGB image with size of 512×512. There is no JPEG compression after we get the sub-blocks.

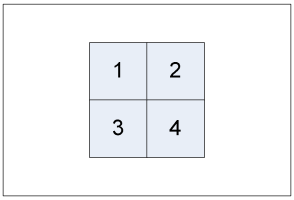


Fig. 6. Sub-block array in an image for detection

The feature generation process is showed in section 3.4 and 69-D features are extracted from each sub-block. We use SVM classifier [17] to do model detection. 90% of the images are randomly chosen for training the classifier and the rest of them are used for testing. The random choosing is controlled to make sure the sub-blocks in training part and testing part are not from same image.

As a loss compression, JPEG compression and its quantization do affect the detection based on CFA and interpolation. But in common case, it can not ‘erase’ the artifacts introduced by CFA and interpolation. Besides, most of the commercial cameras take and store images as JPEG format. It is make more sense for us to use JPEG image sets as testing database.

4.3 Experiment Result

We use images from seven models mentioned above to do model detection. The random selection of training and testing image sets and classification is performed 20 times, the average result shows in Table 2. The diagonal values are correct detection percentage rate for each model. According to Table 2, our method can identify seven models with high detection rate from 98.39% to 99.88%. The average detection accuracy of our proposed method is 99.32%

Table 2. Model detection percentage rate (values below 0.05 are denoted as * for instead, blank denotes 0)

	1	2	3	4	5	6	7
CanonIxus70	99.65			0.35			
CasioEXZ150		99.88	*		0.11		
FujiFirmFinePixJ50		1.43	98.39		0.12	0.06	
NikonCoolPixS710	0.22			99.76		*	
NikonD70s		0.07	0.31	*	98.98	0.58	*
NikonD200	*		*	*	1.23	98.70	
KodakM1063			0.07	*	*	*	99.87

To measure the effectiveness of our method, we use 324-D Markov features extracted from Y component [20] to do model detection using exactly same image datasets for training and testing. Fig.7. shows detection accuracy using two algorithms.

Blue bars denote correct detection rate of our method. Brown bars denote that of Markov method. From model 1 to model 7 denotes seven camera models as Table 1 respectively. Except for model 3 (FujiFirmFinePixJ50), the rest six models correct detection rates are all higher than that of Markov method. The average detection accuracy of our proposed method is 99.32%, For Markov method, the average detection accuracy is 98.78%.

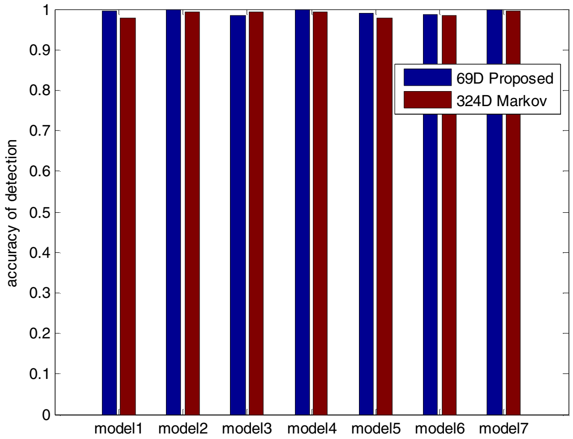


Fig. 7. Model Detection accuracy using 69-D our method and 324-D Markov method [20]

In our experiment, samples are from the Dresden image database. According to the above experiment result, both our proposed method and Markov method works well on seven camera models classification. The detection accuracy of our proposed method is from 98.39% to 99.88%, even slight higher than that of Markov method.

5 Conclusion

This paper presents an algorithm for camera-model identification. Since CFA pattern and interpolation algorithm are same in each model, the artifacts introduced by these two processing can be considered as differences between models. Three feature sets are designed to catch the artifacts. Combining them together, 69-D features are obtained to do model detection. We use images from seven models of the Dresden image database as our sample resource. The experiment result shows that the detection accuracy of our proposed method works well on seven camera models. The average detection accuracy is 99.32%.

Acknowledgements. Thanks for Gloe, T. and Böhme, R, to build and maintain the Dresden image database is a huge load of work. They offer it to all of us. This work is supported in part by: The major national science and technology special projects

(2010ZX03004-003-03), National Nature Science Foundation of China (No. 60832002), Nature Science Foundation of Hubei Province (No. 2009CDB222, No. 2010CDB08602), and Doctoral Fund of Ministry of Education of China (Grant No.200910141110054).

References

1. Lucas, J., Fridrich, J., Goljan, M.: Digital camera identification from sensor pattern noise. IEEE Trans. Inf. Forensics Security 1(2), 205–214 (2006)
2. Chen, M., Fridrich, J., Goljan, M., Lukáš, J.: Determining image origin and integrity using sensor noise. IEEE Trans. Inf. Security Forensics 3(1), 74–90 (2008)
3. Li, C.-T.: Source Camera Identification Using Enhanced Sensor Pattern Noise. IEEE Trans. Inf. Forensics Security 5(2), 280–287 (2010)
4. Kharrazi, M., Sencar, H.T., Memon, N.: Blind source camera identification. In: Proc. Int. Conf. Image Processing, vol. 1, pp. 709–712 (2004)
5. Choi, K.S., Lam, E.Y., Wong, K.Y.: Automatic source identification using the intrinsic lens radial distortion. Opt. Express 14(24), 11551–11565 (2006)
6. Bayram, S., Sencar, H.T., Memon, N.: Improvements on source camera-model identification based on CFA interpolation. In: Proc. Working Group 11.9 Int. Conf. Digital Forensics, FL (2006)
7. Swaminathan, A., Wu, M., Liu, K.J.R.: Non-intrusive component forensics of visual sensors using output images. IEEE Trans. Inf. Forensics Security 2(1), 91–106 (2007)
8. Cao, H., Kot, A.C.: Accurate detection of demosaicing regularity for digital image forensics. IEEE Transactions on Information Forensics and Security 4(4), 899–910 (2009)
9. Kirchner, M.: Efficient Estimation of CFA Pattern Configuration in Digital Camera Images. In: Media Forensics and Security II. Proc. SPIE, vol. 754110 (2010)
10. Dirik, E., Sencar, H.T., Memon, N.: Source camera identification based on sensor dust characteristics. In: Proc. Signal Processing Applications Public Security Forensics, April 11–13, pp. 1–6 (2007)
11. Celiktutan, O., Sankur, B., Avcibas, I.: Blind identification of source cell-phone model. IEEE Trans. Inf. Forensics Security 3(3), 553–566 (2008)
12. Holst, G.C.: CCD Arrays, Cameras, and Displays, 2nd edn. JCD & SPIE, Winter Park, FL, and Bellingham (1998)
13. Janesick, J.R.: Scientific Charge-Coupled Devices, vol. PM83. SPIE, Bellingham (2001)
14. Dark frame subtraction, Qimage Help, `http://www.ddisoftware.com/qimage/qimagehlp/dark.htm`
15. Lukac, R., Plataniotis, K.N.: Color filter arrays: Design and performance analysis. IEEE Transactions on Consumer Electronics 51, 1260–1267 (2005)
16. Dirik, A.E., Memon, N.: Image tamper detection based on demosaicing artifacts. In: ICIP, Cairo, Egypt, vol. (09), pp. 429–432 (November 2009)
17. Chang, C.-C., Lin, C.-J.: LIBSVM: a library for support vector machines (2001), `http://www.csie.ntu.edu.tw/~cjlin/libsvm`
18. Gunturk, B.K., Goltzbach, J., Altunbasak, Y., Schafer, R.W., Mersereau, R.M.: Demosaicking: Color filter array interpolation. IEEE Signal Process. Mag. 22, 44–54 (2005)
19. Ho, J.S., Au, O.C., Zhou, J., Guo, Y.: Inter-channel demosaicking traces for digital image forensics. In: ICM 2010, pp. 1475–1480 (2010)

20. Xu, G., Gao, S., Shi, Y.Q., Hu, R., Su, W.: Camera-Model Identification Using Markovian Transition Probability Matrix. In: Ho, A.T.S., Shi, Y.Q., Kim, H.J., Barni, M. (eds.) IWDW 2009. LNCS, vol. 5703, pp. 294–307. Springer, Heidelberg (2009)
21. Gloe, T., Böhme, R.: The 'Dresden Image Database' for benchmarking digital image forensics. In: Proceedings of SAC, pp. 1584–1590 (2010)
22. Magiera, P., Löndahl, C.: ROF Denoising Algorithm (2008), http://www.mathworks.com/matlabcentral/fileexchange/22410-rof-denoising-algorithm/content/ROFdenoise.m

Detecting Re-captured Videos Using Shot-Based Photo Response Non-Uniformity

Dae-Jin Jung[1], Dai-Kyung Hyun[1], Seung-Jin Ryu[1],
Ji-Won Lee[1], Hae-Yeoun Lee[2], and Heung-Kyu Lee[1]

[1] Department of CS, Korea Advanced Institute of Science and Technology (KAIST),
291 Daehak-ro, Yuseong-gu, Daejeon, Republic of Korea
{djjung,dkhyun,sjryu,jwlee,hklee}@mmc.kaist.ac.kr

[2] Department of Computer Software Engineering, Kumoh National Institute of Technology, Sanho-ro 77, Gumi, Gyeongbuk, Republic of Korea
haeyeoun.lee@kumoh.ac.kr

Abstract. With advances in digital camcorders, re-capturing commercial videos called camcorder theft is getting a big problem. In this paper, we propose an automatic detection method for re-captured videos based on the photo response non-uniformity (PRNU). To discern a re-captured video, a given video is divided into shots first. Several usable shots are selected and PRNU is estimated from each of the shots. Using peak-to-correlation energy (PCE), a connection matrix, which indicates which shots were recorded with a specific camcorder, is constructed. Then, false negative connections are corrected by using Warshall's algorithm. With the number of connections from connection matrix, the given video is determined whether it was the re-captured or not. The experimental results show that the proposed method performs well even with compressed and scaled re-captured videos.

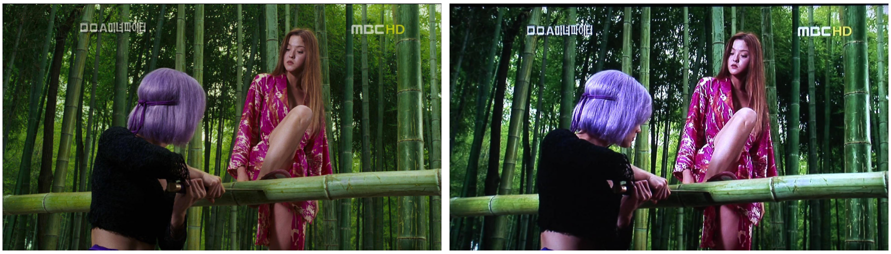
Keywords: Forensics, Photo Response Non-Uniformity (PRNU), Re-captured video.

1 Introduction

With highly sophisticated IT technologies, digital camcorders that are capable of producing high quality footage with low prices and easy usage have been developed. Those advantages of using digital camcorders make many people use digital camcorders more common. Furthermore, traditional analog videos in the movie industry are also replaced by digital videos since digitally recorded movies are cheap and easy to be edited and stored compared with the traditional ones.

Digital camcorders come into wide use due to their great benefits, however, increase in digital camcorder use brought many misuses. The most common abuse is re-capturing the commercial videos, called camcorder theft. Approximately 90% of newly released movies are re-captured in the theater with digital camcorders. The illegally re-captured videos are the largest source of fake DVDs and unauthorized copies distributed through the Internet [1]. As a result, the camcorder theft causes a great loss on movie industry and becomes a big problem.

Y.Q. Shi, H.J. Kim, and F. Perez-Gonzalez (Eds.): IWDW 2011, LNCS 7128, pp. 281–291, 2012.



(a) (b)

Fig. 1. An example of captured shots from a movie : (a) captured shot from original video, (b) captured shot from re-captured video

In early days, re-captured videos had low quality so they could be easily detected by naked eyes. However, with the highly functional optical device technology, the quality of re-captured videos is improved. As shown in Fig. 1, the re-captured video is now comparable to the original video. Therefore, we need an automatic technique which can detect re-captured videos.

Some studies were proposed for protecting videos using watermarking techniques against camcorder theft. The representative study was introduced by Lee *et al.* [2]. Their scheme was designed to be robust to camcorder theft and showed robustness. However, the watermark degrades the quality of videos. Also, the watermarking way requires an embedding process during movie playback.

Cao *et al.* proposed a method that identifies re-captured images on LCD screens [3]. Forensic features such as local binary pattern, multi-scale wavelet statistics, and color features were extracted from image sets. By using the extracted features, a probability support vector machine classifier was trained and then tested. Their scheme could discriminate re-captured images with good qualities from original images with equal error rate lower than 0.5%. However, their method took too much time and could not be applied to video directly.

The re-projected video detection by estimating a skew parameter was proposed by Wang *et al.* [4]. Their method could detect the re-projected video with some frames and could have much lower false positive by extracting more feature points. However, the feature points needed to be positioned in ridged body geometry. In this step, some feature points not on the ridged body geometry should be removed manually since it is hard to check those points automatically.

In this study, we propose a method to discriminate the re-captured video based on the shot-based photo response non-uniformity (PRNU). The proposed method can discriminate re-captured vieo without any additive information and it is designed for videos. Moreover, the entire procedure of the proposed method performs automatically.

The rest of this paper is structured as follows. The differences between original videos and re-captured videos are analyzed in Sec. 2. Then, the detail of the proposed method is explained in Sec. 3. Experimental results are exhibited in Sec. 4 and Sec. 5 concludes.

2 Differences between Original and Re-captured Videos

In this chapter, we describe the differences between original and re-captured videos. These differences are caused by the following factors:

1) *Different recording devices*: The original videos can be recorded by analog cameras or digital camcorders. Even though digital camcorders provide several benefits such as editing efficiency, reducing film cost, easy process to insert CGs, and etc., analog film cameras are still used because of their own characteristics such as high quality, soft shades of colors, and so on. On the other hand, the re-captured videos are mostly recorded by digital camcorders. Compact size, light weight, and easy manipulation make easier for pirates to handle digital camcorders in theaters without being observed.
2) *The number of cameras used in recording*: In the original videos, multiple cameras are used to record shots. For example, two or more cameras are used to shoot talking two actors; one for one actor, another for another actor, and the other for both actors. It means that each shot in the original videos has high probability to be recorded by different cameras. On the contrary, only a single digital camcorder is used to re-capture the original videos because pirates do not need multiple camcorders to re-capture videos.
3) *Different post-processing*: Original videos are edited by huge amount of post-processing in general. As discussed above, original videos are recorded by multiple cameras. Each camera has unique characteristics such as color tone, contrast, brightness and so on. Thus, post-processing for each shot is essential to harmonize the whole content. Furthermore, it is usual to insert CGs and other visual effects into shots. However, re-captured videos are not edited by much post-processing. Only some of them are re-compressed or resized for convenience.

Above three differences can affect PRNU of the original and re-captured video. The PRNU is pixel variation under illumination. It was proposed to identify the source digital camera by Lukas *et al.* [5]. Digital camera has a charge coupled device or complementary metal-oxide-semiconductor sensor, and the PRNU is caused by sensor imperfection which is introduced in sensor manufacturing process. Since the PRNU is unique for each sensor, it is considered as a fingerprint of a digital camera. Also, the PRNU can be used to identify source digital camcorders. Therefore, three differences between original and re-captured videos and the characteristics of the PRNU, we can infer some properties for the re-captured video detection as follows:

- Spcifically, the shot-based PRNU has low correlation with each other if we estimate them from original shots. First, the shots from analog films do not have their own PRNU because analog cameras do not include any digital sensor. Therefore, the PRNU estimated from alnalog shots cannot be used to identify source analog camera. Second, even though PRNU is estimated from digitally recorded shots, their source camcorders would vary and the estimated PRNU would be damaged from heavy post-processing. There might be several original shots which are recorded by digital camcorders and edited by little post-processing. It may give high correlation among shots. However, those shots are still not be correlated with other shots which are taken from other digital camcorders. Thus, those correlated shots will be grouped, consequently the number of groups will be greater than one. This factor would be an evidence that the given video is original.

- In contrast, the shot-based PRNU of re-captured videos is highly correlated with each other. All shots in the re-captured video are taken from the same digital camcorder and they are edited by little and same post-processing for each shot. These conditions let the PRNU from the re-captured video be correlated each other.

By exploiting these properties, we can differentiate the re-captured videos from original videos.

3 Proposed Method

We propose a method that can discriminate re-captured videos from original videos. Fig. 2 depicts the proposed method. Once a suspected video is given, the shot change detection process is performed to find suitable shots for PRNU estimation. After dividing the given video into shots, we estimate PRNU from each shot. Then peak-to-correlation energy (PCE) values between PRNU is calculated as a measure to find out whether those shots are taken from the same digital camcorder or not. With results of PCE values, we decide whether the given video is a re-captured video or not.

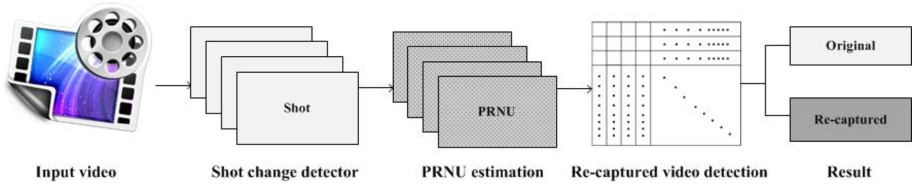


Fig. 2. An overview of proposed re-captured videos detection

3.1 Shot Change Detection

We first divide a given video into numbers of shots. A shot can be defined as a continuous strip of motion picture film recorded with a single camera. Accurate shot change detector, which divides a given video into shots, is important since wrong shot change declaration can affect the result of re-captured video detection. If two or more shots are declared as a single shot by a shot change detector, PRNU estimated from that shot will be mixed PRNU from plural cameras so that the false positive rate in PRNU comparison will be increased. In addition, if one shot is declared as two or more shots by a shot change detector, it can also increase false positive rate in re-captured video detection.

A histogram comparison method is used for shot change detection because it has good performance and it is relatively fast [6]. Let $H_i(j)$ denote a histogram value for ith frame, where j is one of G possible gray levels and SD_i is the sum of absolute differences between ith frame and $(i+1)$th frame. Then the sum of absolute differences, SD_i, is given by the following formula:

$$SD_i = \sum_{j=1}^{G} |H_i(j) - H_{i+1}(j)| \tag{1}$$

To use SD_i for shot change detection with any size of video, SD_i is normalized by frame size. If the normalized SD_i is larger than a given threshold, the shot change is declared. Note that the operations in the equations appeared in this paper are element-wise.

3.2 PRNU Estimation

To find out whether the shots are taken from the same digital camcorder or not, PRNU is estimated from those shots and compared each other. PRNU estimation method for digital camcorders are proposed by Chen *et al.* [7]. The PRNU for digital camcorders is modeled as follow:

$$\mathbf{I} = g^{\gamma} \cdot [(\mathbf{1} + \mathbf{K})\mathbf{Y} + \mathbf{\Lambda} + \mathbf{\Theta}_s + \mathbf{\Theta}_r]^{\gamma} + \mathbf{\Theta}_q \tag{2}$$

where $\mathbf{I}$ denotes the sensor output compromised by numerous in-camcorder processing, g does the color channel gain, γ is the gamma correction factor, $\mathbf{K}$ is PRNU multiplicative factor which can be used as a fingerprint of digital camcorder, Y is the light intensity, and $\mathbf{\Lambda}$, $\mathbf{\Theta}_s$, $\mathbf{\Theta}_r$, $\mathbf{\Theta}_q$ denote dark current, shot noise, read-out noise, and quantization noise, respectively. Using first order Taylor expansion, simple form of this model can be obtained:

$$\mathbf{I} = \mathbf{I}^{(0)} + \gamma \mathbf{I}^{(0)}\mathbf{K} + \mathbf{\Theta} \tag{3}$$

Here, $\mathbf{I}^{(0)}$ is the noise-free sensor output(frame) from one channel before demosaicing is applied. $\mathbf{\Theta}$ is a noise component including above noises.

We use simplified model in Eq (3) to estimate PRNU from each shot. To suppress the influence of the noise-free frame $\mathbf{I}^{(0)}$, an estimate $\hat{\mathbf{I}}^{(0)}$ of $\mathbf{I}^{(0)}$ is subtracted from both sides of Eq (3). $\hat{\mathbf{I}}^{(0)}$ can be estimated by using denoising filter which is a wavelet based filter [8].

$$\mathbf{W} = \mathbf{I} - \hat{\mathbf{I}}^{(0)} = \mathbf{IK} + (\mathbf{I}^{(0)} - \hat{\mathbf{I}}^{(0)}) + [(\mathbf{I}^{(0)} - \mathbf{I})\mathbf{K}] + \mathbf{\Theta} \tag{4}$$

PRNU factor $\mathbf{K}$ can be estimated by using Maximum Likelihood Estimation (MLE) method as

$$\gamma \hat{\mathbf{K}} = \frac{\sum_{k=1}^{N} \mathbf{W}_k \hat{\mathbf{I}}_k^{(0)}}{\sum_{k=1}^{N} (\hat{\mathbf{I}}_k^{(0)})^2} \tag{5}$$

where $\mathbf{W}_k$ is noise residual of kth frame.

After MLE process, codec noise is removed by using denoising filter. Usually a video undergoes DPCM-block DCT transform which causes block artifacts [7]. The block artifact should be removed since it causes false correlations between uncorrelated PRNU. Wiener filter in frequency domain is used in our method to suppress the codec noise [9].

Then, PCE is calculated for a pair of PRNU to decide whether two shots are taken from same digital camcorder. To calculate PCE, we calculate normalized correlation first:

$$NCC[\mathbf{X},\mathbf{Y}] = \frac{(\mathbf{X}-\overline{\mathbf{X}}) * (\mathbf{Y}-\overline{\mathbf{Y}})}{\|\mathbf{X}-\overline{\mathbf{X}}\|\|\mathbf{Y}-\overline{\mathbf{Y}}\|} \tag{6}$$

where $\mathbf{X}$, $\mathbf{Y}$ are estimated PRNU, $\overline{\mathbf{X}}$ is mean of $\mathbf{X}$, $\mathbf{X} * \mathbf{Y}$ is dot product and $\|\mathbf{X}\|$ is the norm of $\mathbf{X}$. Then PCE is calculated as follow [10]:

$$PCE[\mathbf{X},\mathbf{Y}] = \frac{|NCC[\mathbf{X},\mathbf{Y}](u_0,v_0)|^2}{\mathbf{E}_{NCC[\mathbf{X},\mathbf{Y}]}} \tag{7}$$

where (u_0, v_0) denotes the center location of correlation plane and $\mathbf{E}_{NCC[\mathbf{X},\mathbf{Y}]}$ is the correlation plane energy of $NCC[\mathbf{X},\mathbf{Y}]$. If PCE of given two PRNU from two different shots is higher than certain threshold, then we decide that those two shots are taken from same source digital camcorder.

3.3 Detecting Re-captured Videos

To decide whether a given video is re-captured or not, we investigate every PRNU from the video is related with each other. For this purpose, we use Warshalls algorithm which calculates the connectivity of a given graph [11]. Let the X_i be the PRNU of selected shot, when $i = 1, \ldots, N$. And we can consider the X_i as a vertex. Then, a connection between two vertexes $(X_i,\ X_j)$ is decided by the value of $PCE[X_i, X_j]$. If the $PCE[X_i, X_j]$ has greater value than predefined threshold T, X_i and X_j have connection to each other. In contrast, lower $PCE[X_i, X_j]$ value than threshold T implies X_i and X_j have no connection to each other. As a consequence a symmetric $N \times N$ connection matrix is created after calculating the connectivity for every possible pair of PRNU.

By using Warshalls algorithm, we can correct false negative connections. Fig. 3 depicts a simple case of the false negative connection correction by Warshall's algorithm. After processing Warshall's algorithm, we can decide the origin of a given video from the connection matrix. If the $N \times N$ connection matrix has N^2 connections, we decide the given video is re-captured video because N^2 connections from N PRNU mean that the entire shots have same source digital camcorder. Otherwise, the given video is decided as original video since less than N^2 connections from N PRNU mean the given video has two or more source digital camcorders.

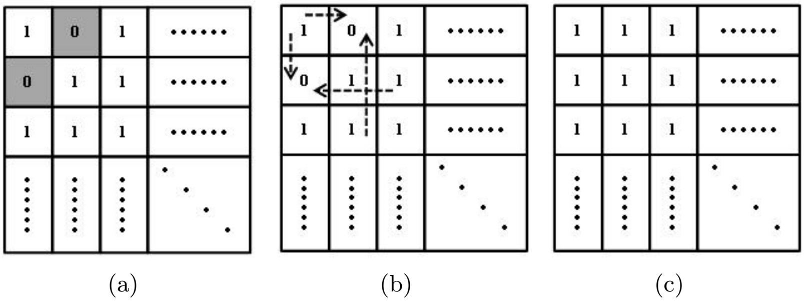


Fig. 3. A simple example of false negative connections correction : (a) Matrix with false negative connections, (b) Correcting false negative connections (c) Corrected false negative connections

Table 1. Information about original videos used in experiments(resolutions and their main cameras)

	Resolution	**Main Camera(Digital/Analog)**
#1	1280 x 720	Sony PMW-F3(Digital)
#2	1280 x 720	Sony PMW-F3(Digital)
#3	1280 x 720	Sony PMW-F3(Digital)
#4	1920 x 1080	Red One(Digital)
#5	1920 x 1080	Red One(Digital)
#6	1280 x 720	unknown(unknown)
#7	1920 x 1080	unknown(unknown)
#8	1280 x 720	Panavision camera(Analog)
#9	1280 x 720	Panavision Panaflex Platinum(Analog)
#10	1280 x 720	unknown(unknown)

4 Experimental Results

In this section, we examine the proposed re-captured video detection method. We used 4 digital camcorders (Samsung HMX-H205BD, Sony HDR-CX500, Sony HDR-CX550, and Sony HDR-SR10) to re-capture the original videos. We used 10 original videos and 5 of them were fully or partially recorded by digital video camcorders [12]. 40 videos were created by re-capturing 10 original videos with 4 digital camcorders. The resolution of the original videos varied from 1280x720 to 1920x1080 and the resolution of re-captured video was set as 1920x1080. Specific information about original videos used in experiments is in Table 1. To estimate PRNU, we divided each video into shots using the shot change detector. In shot change detection, frames are converted into gray frames to calculate the histogram differences. Before estimating PRNU from divided shots, some shots unsuitable for PRNU estimation are excluded. More specifically, shots

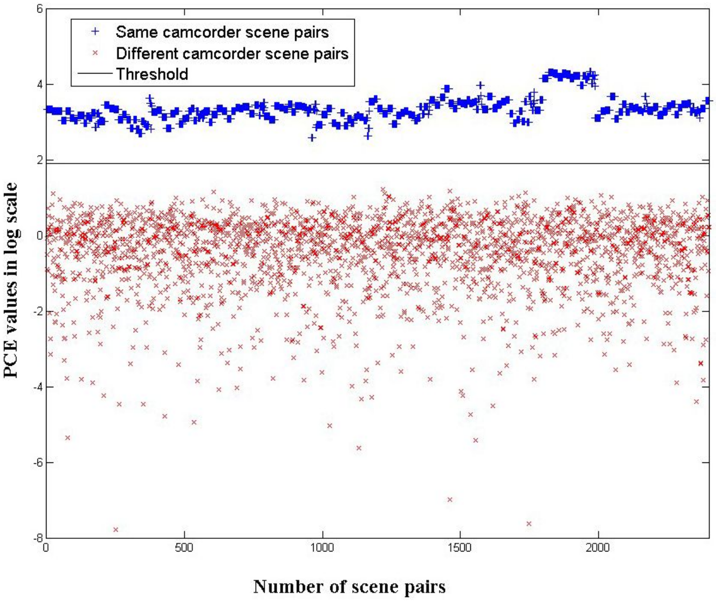


Fig. 4. PCE values in log scale of shots from the same camcorders shots from the different camcorders.

constructed with small number of frames or dark frames are excluded since those shots can increase false negative rate in PRNU comparison. We extracted 200 successive frames from a shot in the PRNU estimation step.

Before testing proposed method, the threshold T for PRNU comparison needs to be set. To decide the adequate threshold T, 2400 pairs of PRNU from same camcorders and 2400 pairs of PRNU from different camcorders are prepared. Using PCE measurement, a scatter plot of PCE values in log scale for those pairs was obtained. As shown in Fig. 4, PCE values can be divided by simple straight line whose values is 80. Thus, we set 80 as a threshold T.

4.1 Re-captured Video Detection Experiment

We tested re-captured video detection for 10 original videos and 10 videos for each digital camcorder, totally 40 re-captured videos. 20 shots are collected from each video($N = 20$). Table 2 shows the result of re-captured video detection. Items in the table is the ratio of connections in $N \times N$ connection matrix. Every video had at least 20 connections in diagonal line in connection matrix since each shot was correlated with itself. Original videos had lower number of connections than N^2 since it was not recorded by single digital camcorder. On the contrary, every re-captured video had N^2 connections because it had only single source digital camcorder. In this experiment, the detection ratio of re-captured videos was 100% even before applying Warshall's algorithm.

Table 2. Connection ratio for original videos and re-captured video (20 shots were used)

	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
Original movie	0.05	0.38	0.33	0.16	0.13	0.05	0.07	0.05	0.05	0.05
Samsung HMX-H205BD	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Sony HDR-CX500	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Sony HDR-CX550	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Sony HDR-SR10	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

4.2 Compression Experiment

We also tested the robustness to compression. Re-captured videos were compressed with different quality factors (QFs) while the resolution was not changed. MPEG4 (AVC/H.264) was used in re-encoding. Fig. 5 shows the result for compressed re-captured videos. For QF 100∼70, the proposed method showed 100% detection ratio. A few false negative connections appeared in QF 70, but all of false negative connections are corrected by using Warshall's algorithm. For QF 60, the detection ratio dropped to 35% because some shots which had no connection to other shots had appeared. Those shots were not able to be corrected by Warshall's algorithm. However, QF 60 is not commonly used in video compression due to severe quality degradation such as block artifacts.

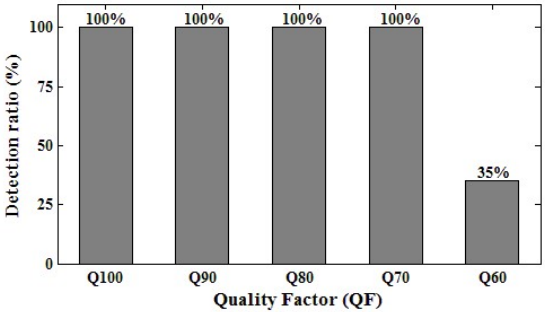


Fig. 5. Detection ratio for compressed videos with different quality factors

4.3 Scaling Experiment

Re-captured videos were scaled with various scale factors (SFs) while QF was set as 100. Since up-scaling is rare for videos, we only tested for SFs lower than 1. MPEG4 (AVC/H.264) was used for re-encoding. Fig. 6 shows the result for scaled re-captured videos.

The proposed method showed low detection ratio for SF 0.3 which is not parameter for common video resizing. However, the proposed method detected most of re-captured videos which were scaled with SF 0.9∼0.4.

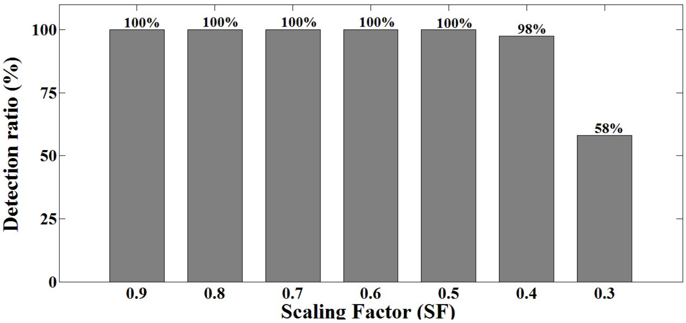


Fig. 6. Detection ratio for scaled videos with different scaling factors

4.4 Combinational Experiment

Combinational experiment was also conducted. Usually, re-captured videos are re-encoded before being redistributed. The common options for re-encoding are QFs higher than 80% and SFs higher than 0.5. Thus, we tested proposed method for re-captured videos which were re-encoded with parameters of QF 80 and SF 0.5. And the proposed method detected them 100%. This result is meaningful since those parameters are common for re-encoding videos.

We did not conduct further geometric distortions such as affine transform because they are not common for videos. Even if any geometric distortion is proceeded for a re-captured video, every PRNU estimated from shots will be synchronized since all frames in the video are manipulated by the same distortion. Eventually, the re-captured video which has undergone any geometric distortion will be detected by the proposed method if the distortion does not ruin PRNU information severely.

5 Conclusion

In this paper, we have investigated to detect the re-captured videos. The proposed method operates automatically for a given video and does not use any additive information such as watermarks. This proposed method is based on the photo-response non-uniformity (PRNU), which is unique fingerprint of digital image sensors. The proposed method consists of 3 steps. First, a given video is divided into shots. Then, PRNU is estimated from collected N shots. At last, an $N \times N$ connection matrix is created by evaluating PCEs for each pair of N shots. Finally, we can decide the given video is re-captured or not with the result of the connection matrix. Experimental results show that proposed method performs excellent in detecting re-captured videos. The proposed method performs well even a given video is re-compressed and re-scaled. However, the proposed method is still weak against severe attacks. Therefore, our future work is to detect re-captured videos even they are re-compressed with low quality and scaling factors.

Acknowledgments. This research project was supported by Ministry of Culture, Sports and Tourism(MCST) and from Korea Copyright Commission in 2011.

References

1. Motion Picture Association Of America, `http://www.mpaa.org`
2. Lee, M.J., Kim, K.S., Lee, H.K.: Digital Cinema Watermarking for Estimating the position of the Pirate. IEEE Transactions Multimedia, 605–621 (2010)
3. Cao, H., Kot, A.C.: Identification of recaptured photographs on LCD screens. In: Acoustics Speech and Signal Processing (ICASSP), pp. 1790–1793. IEEE press, Texas (2010)
4. Wang, W., Farid, H.: Detecting Re-projected Video. In: Solanki, K., Sullivan, K., Madhow, U. (eds.) IH 2008. LNCS, vol. 5284, pp. 72–86. Springer, Heidelberg (2008)
5. Lukas, J., Fridrich, J., Goljan, M.: Digital camera identification from sensor pattern noise. IEEE Trans. Information Forensics and Security, 205–214 (2006)
6. Zhang, H.J., Kankanhalli, A., Smoliar, S.W.: Automatic partitioning of full-motion video. Multimedia Syst. 1, 10–28 (1993)
7. Chen, M., Fridrich, J., Goljan, M., Lukas, J.: Source Digital Camcorder Identification Using Sensor Photo Response Non-Uniformity. In: The International Society for Optical Engineering, SPIE (2008)
8. Mzhqak, M.K., Kozintsev, I., Ramchandran, K.: Spatially Adaptive Statistical Modeling of Wavelet Image Coefficients and its Application to Denoising. In: Acoustics, Speech, and Signal Processing (ICASSP), vol. 6, pp. 3253–3256. IEE press, Arizona (1999)
9. Chen, M., Fridrich, J., Goljan, M., Lukas, J.: Digital Imaging Sensor Identification (Further Study). In: The International Society for Optical Engineering, SPIE (2007)
10. Kumar, B.V.K.V., Hassebrook, L.: Performance measures for correlation filters. Applied Optics 29, 2997–3006 (1990)
11. Warshall, S.: A theorem on boolean matrices. Journal of the ACM (JACM) 9, 11–12 (1962)
12. The Internet Movie Database, `http://www.imdb.com`

Distinguishing Photographic Images and Photorealistic Computer Graphics Using Visual Vocabulary on Local Image Edges

Rong Zhang[1], Rang-Ding Wang[1], and Tian-Tsong Ng[2]

[1] College of Information Science and Engineering, Ningbo University, Zhejiang, China 315211
{zhangrong,wangrangding}@nbu.edu.cn
[2] Institute for Infocomm Research, A*STAR, Singapore 138632
ttng@i2r.a-star.edu.sg

Abstract. Differentiating computer graphics from natural images remains a representative problem of digital image forensics because the two categories of images reflect typical different aspects of generation and forgery of digital images. This paper aims to address this problem through analyzing the statistical property of local edge patches in digital images. First, we preprocess image edge patches and project them into a 7-dimensional sphere as in [7]. Then, a visual vocabulary is constructed via determining the key sampling points in accordance with Voronoi cells. The proposed approach to constructing visual vocabulary avoids troubles in traditional partitioning algorithms such as *k*-means. And then, a given image is represented as a binned histogram of visual words and the corresponding feature vector is formed by the bins. Finally, we employ an SVM classifier for image classification. Our experimental results demonstrate the efficient discrimination of the proposed features.

Keywords: image forensics, image classification, image authenticity, visual vocabulary.

1 Introduction

As rendering technology of computer graphics images leads to photorealism (a visual fidelity close to that of real-world photographic images), highly photorealistic *computer graphics* (PRCG) is regarded as a form of image forgery [1], [2]. Without knowing any information of image source, known as passive-blind image forensics, distinguishing photorealistic computer graphics from *photographic images* (PIM) has become an important problem in a wide range of social areas such as criminal inquisition as well as media credit system construction of journalism and publication, and has been regarded as one of the primary tasks in digital image authenticity forensics.

Photographic images are generated from natural scene by digital imaging tools and generally called as natural images. The acquisition process of a natural image taken by a digital camera is a complex transform from physical light to digital signal. The general signal processing process is outlined in both [3] and [4]: The light from a real-world

Y.Q. Shi, H.J. Kim, and F. Perez-Gonzalez (Eds.): IWDW 2011, LNCS 7128, pp. 292–305, 2012.

scene passes through the lens and the anti-aliasing blurring filter, and get to the color filter array (CFA). Then, the imaging sensor collects photons and converts them into voltages. The charge is amplified and quantized into digital signal. CFA interpolation (or demosaicking) algorithms are employed to estimate the absent color information of each pixel due to the use of CFA. Further post-processing manipulations in camera are performed to improve or enhance image visual quality, including color correction, white balance, gamma correction, low-pass filtering for denoising or sharpening, and so on. Finally, the raw image is stored in the camera storage in TIFF, JPEG or other formats. The imaging pipeline is shown in Fig. 1.

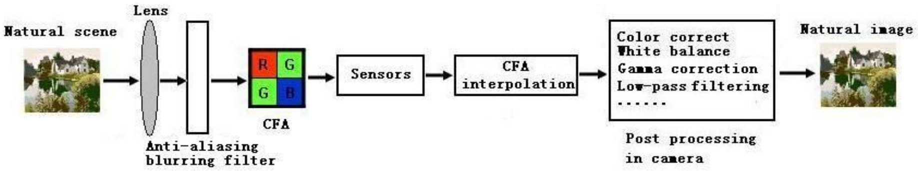


Fig. 1. Physical generation pipeline of a natural image captured by a digital camera

However, computer graphics (or computer generated images), in contrast, are of entirely virtual scene and generated via computer graphic tools, in which some artificial models are employed, including object description models, geometry models, illumination models as well as color and texture models. Finally, a virtual camera is applied to transform the designed virtual scene into the final image. A skilled artist or professional programmer is able to design out highly photorealistic computer graphics with graphic tools such as 3d MAX, Softimage-XSI and Maya. Although photorealistic computer graphics can deceive our eyes and make us doubtful of what we see, it is pointed out in [1] that they are not really parallel to natural images. The direct reason for this argument is that artificial models fail to entirely simulate complicated surface texture and geometrical shape of natural objects as well as illumination in natural scenes.

Factually, in view of time-cost and computation complexity, the generation process of computer graphics often has to be simplified, which inevitably causes some hypostatic differences between natural images and computer graphics, for example, in color number and histogram continuity. Lyu and Farid[5] proposed an approach to differentiating between computer graphics and natural images by building statistical models based on first- and higher-order statistics in wavelet domain and employing SVM (Support Vector Machine) for image classification. In [6], a similar method is used. Images in HSV color space, instead of in RGB as [5], are transformed into wavelet domain, and the statistical moments of characteristic function of wavelet coefficients in different subbands are used as the distinguishing features. Although statistical model-based methods similar to [5] and [6] can achieve some promising results, due to the lack of physical models for PIM and PRCG, these methods have insufficient ability to capture the essential differences between PIM and PRCG.

In general, imaging sensor is the heart of digital cameras. In [7], it is believed that the geometrical structure of high contrast local image patches can characterize the sensor properties of camera devices. Motivated by [7], in [8], Ng applied the local high-contrast patch features to cope with distinguishing PIM and PRCG. Ng studied various patterns of local image patches and calculated distribution histograms of these patterns respectively. In Ng's approach, each histogram contains 17,520 bins. The Kullback-Leibler (KL) distance between an image's histogram and two image model histograms was calculated. Consequently, multiple features for different patterns were gained. He employed SVM to distinguish PIM and PRCG and obtained 83% classification accuracy. Additionally, in [1], in order to identify images of different sources from physical insight, Ng proposed the combination of the similar features with fractal dimension, surface gradient and other differential geometrical features.

In this paper, we attempt to identify natural images and photorealistic computer graphics on the basis of statistically analyzing local edge patches. We propose a novel approach to constructing visual vocabulary with local image edge patches, which avoids some fatal problems of the previous bag-of-words model, and apply it to the image classification of natural images and photorealistic computer graphics. The organization of the paper is as follows: In Section 2, we describe the two data sets used in our experiments. In Section 3, we present the approach of differentiating photorealistic computer graphics from natural images based on an efficient image representation and visual vocabulary construction. Our experimental results are shown in Section 4 and the conclusions are given in Section 5.

2 Data Sets

We collected two data sets, i.e. PIM data set and PRCG data set, respectively consisting of 1000 camera images and 900 photorealistic computer graphics.

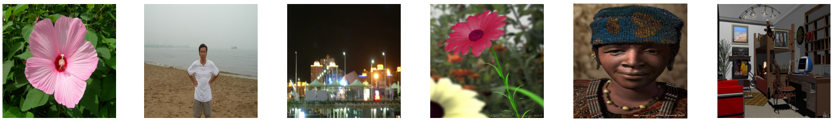


Fig. 2. Samples from the two data sets. Left three: samples of our PIM data. Right three: PRCG samples from Columbia dataset

We recognize that images from the Internet, like the Columbia PIM and PRCG datasets, may undergo various unknown post-processing and compression at various quality factors. To explore the essential properties of natural images, we collect all images in PIM set with high quality JPEG format from 8 consumer-end cameras and without any experience of post-process outside the cameras. The images are of different size, for example, 2048×1536, 2592×1944, and so on. The camera brands cover Canon, Sony, Nikon, Samsung and Olympus. The image content involved is of high diversity in illumination intensity (outdoor/indoor, daylight/night/rain) and object type (architecture/ persons/animal/plant/natural scene). In PRCG set, 800 images are from Columbia PRCG data set [9] with a wide range of styles including architecture, person, animal and natural scene. Additionally, we collect 100 CG images with high visual realism from website www.raph.com. The rendering software of these CG contains 3D Studio MAX, Cinema 4D, Lightwave, Maya and so on. See Fig. 2 for some samples of the two data sets.

3 Image Classification Based on Image Edge Vocabulary

The bag-of-words model [10, 11] originates from text categorization. When applied to visual categorization, it is called as bag-of-visual-words model, in which an image or a visual object is represented as a histogram of the number of occurrences of visual words. A visual word corresponds to a cluster center and the vocabulary is constructed by a set of cluster centers. In [10, 11], cluster centers are determined by k-means clustering algorithm. However, the selection of parameter k and initial cluster centers are difficult and greatly affect the final result.

The basic idea of the bag-of-visual-words helps us efficiently capture the significant difference in statistical distribution of geometrical structure of local edge patches between natural images and computer graphics. In this section, a new method for visual vocabulary construction based on key sampling points is proposed. The key sampling points are predetermined depending on the sparseness property of the geometrical structure of 3×3 local image edge patches, which is the reliable result of statistical analysis. A visual vocabulary is constructed by the key sampling points used as cluster centers. The data points for the local edge patches from a given image are assigned to the specific clusters. In this way, we carry out a supervised partitioning procedure, which successfully overcomes the instabilities of unsupervised learning algorithms. In our approach for image classification, the bags of visual words are treated as feature vector.

3.1 Presentation of Local Image Edge Patches

Unlike traditional image analysis means using marginal distribution to characterize the non-Gaussian traits of coefficients in transform domains, Lee [7] systematically studied the full probability distribution of 3×3 high-contrast local patches of natural images at pixel level. In this section, we preprocess the extracted edge patches according to the mathematical framework proposed in [7].

First, we transform color images from RGB color space into grey scale images and obtain the log-values of intensities. Then, edges are located and local edge patches are extracted. We apply Sobel edge detector to locate the edge points in an image and extract the 3×3 local patches corresponding to the edge points. For the purpose of obtaining local patches rich enough to carry diverse structure information, we divide an image into 9×9 non-overlapping blocks and then randomly select 1000 blocks from them. Finally, we extract 1000 disjoint 3×3 patches from the 1000 blocks. As for the computer generated images from the above dataset, it is possible that less than 1000 patches can be obtained for an PRCG image because of small image size or simple texture details of the image. Consequently, we collect 1M edge patches from the PIM data set and less than 900K from the PRCG data set.

Each 3×3 patch is regarded as a 9-tuple of real numbers (log of gray values), i.e. a vector in $\mathbb{R}^9$

$$\mathbf{x} \bullet\bullet [x_1 \bullet\bullet\bullet\bullet\bullet\bullet\bullet \bullet\bullet\bullet x_9]^T \bullet [I_{11} I_{21} I_{31} \tilde{I}_{12} I_{33}]^T \in \mathbb{R}^9 . \qquad (1)$$

In the following steps, patch data are preprocessed with centering, contrast normalizing and whitening according to [7], which gave us an efficient image local patch representation in high dimension state space.

- Data centering

Each patch vector $\mathbf{x}$ subtracts the patch mean

$$\tilde{\mathrm{x}} = \mathrm{x} - \frac{1}{9} \sum\nolimits_{i=1}^{9} x_i \ . \qquad (2)$$

New vector $\tilde{\mathbf{x}}$ with the mean zero is obtained and all the data points are projected into an 8-dimensional subspace within $\mathbb{R}^9$.

- Contrast normalizing

Lee [7] defined the D-norm to calculate the contrast of a 3×3 image patch $\mathbf{x}$

$$\| \mathbf{x} \|_D = \sqrt{\sum_{i \sim j} (x_i - x_j)^2} \ , \qquad (3)$$

where i~j denotes the 4-connected neighbors of a pixel in a 3×3 patch. Eq. (2) can be denoted in matrix form

$$\| \mathbf{x} \|_D = \sqrt{\mathbf{x}^T D \mathbf{x}} \ , \qquad (4)$$

where

$$D = \begin{vmatrix} 2 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 3 & -1 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & -1 & 4 & -1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & -1 & 3 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 & 3 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 2 \end{vmatrix} . \qquad (5)$$

According to the D-norm, all edge patch vectors $\tilde{\mathbf{x}}$ divided by $| \tilde{\mathbf{x}} \|_D$ are contrast-normalized, i.e.

$$\mathbf{y} = \frac{\tilde{\mathbf{x}}}{\| \tilde{\mathbf{x}} \|_D} . \tag{6}$$

After data centering and contrast normalizing, we obtain patches with the contrast equal to 1. This means that if one patch is obtained from another patch by translation or scaling of gray the two patches will be regarded as the same. It is noted that contrast normalization put all data points onto a 7-dimensional ellipsoid $\tilde{S}^7$, $\tilde{S}^7 \subset \mathbb{R}^9$, where

$$\tilde{S}^7 = \left\{ y \in \mathbb{R}^9 : \sum_{i=1}^{9} y_i = 0, \mathrm{y}^T D \mathrm{y} = 1 \right\} . \tag{7}$$

- **Data Whitening**

In this step, we make a change of basis. The 2-dimensional *Discrete Cosine Transform* (DCT) basis corresponding to 3×3 image patches are selected. The contrast normalized DCT basis in vector form are as follows

$$\begin{cases} \tilde{e}_1 = \frac{1}{\sqrt{6}}[1,\ 0,-1,\ 1,\ 0,-1,\ 1,\ 0,-1]^T \\ \tilde{e}_2 = \frac{1}{\sqrt{6}}[1,\ \mathbf{1},\ 1,\ 0,\ 0,\ 0,-1,-1,-1]^T \\ \tilde{e}_3 = \frac{1}{\sqrt{54}}[1,-2,\ 1,\ 1,-2,\ 1,\ 1,-2,\ 1]^T \\ \tilde{e}_4 = \frac{1}{\sqrt{54}}[1,\ 1,\ 1,-2,-2,-2,\ 1,\ 1,\ 1]^T \\ \tilde{e}_5 = \frac{1}{\sqrt{8}}[1,\ 0,-1,\ 0,\ 0,\ 0,-1,\ 0,-1]^T \\ \tilde{e}_6 = \frac{1}{\sqrt{48}}[1,\ 0,-1,-2,\ \mathbf{0},\ \mathbf{2},\ 1,\ \mathbf{0},-1]^T \\ \tilde{e}_7 = \frac{1}{\sqrt{48}}[1,-2,\ 1,\ \mathbf{0},\ \mathbf{0},\ \mathbf{0},-1,-2,-1]^T \\ \tilde{e}_8 = \frac{1}{\sqrt{216}}[1,-2,\ 1,-2,\ 4,-2,\ 1,-2,\ 1]^T \end{cases} , \tag{8}$$

where $| \tilde{e}_1 \|_D = \dots = \| \tilde{e}_8 \|_D = 1$ and the constant DCT basis vector is omitted. Then, construct a 9×8 matrix $B = [\tilde{e}_1, \tilde{e}_2 \dots, \tilde{e}_8]$.

Let

$$\Lambda = diag(1/\| e_1 \|^2, 1/\| e_2 \|^2, \dots, 1/\| e_8 \|^2) , \tag{9}$$

and finally, the basis change is made according to Eq. (10)

$$\mathbf{v} = A\mathbf{y} , \tag{10}$$

where $A = \Lambda B^T$. Since DCT is orthogonal, this is a whitening transform. At the same time, the change of basis transform the ellipsoid $\tilde{S}^7$ in Eq. (7) to a 7-sphere

$$S^7 = \left\{ \mathrm{v} \in \mathbb{R}^8 : \sum_{i=1}^{8} v_i = 0, \| \mathrm{v} \| = 1 \right\}, \tag{11}$$

where $S^7 \subset \mathbb{R}^8$, which is the state space of the data points.

Since all the data points lie on the 7-sphere, the distance between two data points $\mathbf{x}_1, \mathbf{x}_2$ ($\mathbf{x}_1, \mathbf{x}_2 \in \mathbf{S}^7$) can be calculated according to the angular distance

$$d(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{1\text{-}cos{<}x_1, x_2{>}} = \mathbf{1\text{-}} \frac{\mathbf{x}_1 \bullet \mathbf{x}_2}{\| \mathbf{x1} \| \| \mathbf{x2} \|}, \tag{12}$$

where $\bullet$ is the inner product of two vectors.

Upon that, all 3×3 image patches are represented as 8-dimensional vectors, which compose the data point set $\mathbf{v}$. Note that all data points v_i lie on the 7-dimensional sphere, i.e. $v_i \in S^7$ and $S^7 \subset \mathbb{R}^8$.

3.2 Construction of Visual Vocabulary

In [7], 17,520 Voronoi cells with roughly the same size and efficiently covering the 7–sphere are selected as the sampling point set. Therefore, the problem to observe the distribution of data points on the 7-sphere is converted to the one, which we should calculate the probabilities that data points fall into the corresponding Voronoi tessellations.

Each Voronoi cell is a convex hull which consists of a set of points that are closer to the particular lattice point than to any other lattice points (Note that the distance here is calculated according to Eq. (12)). The particular lattice point in a Voronoi tessellation is called as sampling point. That is,

$$O = \{o_1, o_2, ..., o_N\}, N = 17520, \tag{13}$$

where o_i is the sampling point in the ith lattice and is a 8-dimensional vector. The sampling point set is a dense set on the sphere $\mathbf{S^7}$ (a normed unit sphere as described above). The literature [12] further demonstrates the mathematical sense of the presented approach in [7] in topology.

As a result, we can use the histograms with 17,520 bins to respectively describe the probability distribution of geometrical structure of edge patches for natural images and computer graphics.

If a data point $\mathbf{x}$ ($\mathbf{x} \in \mathbf{S}^7$) falls into a Voronoi Cell Ω_i, it is at least equally close to the sampling point o_i than to any other sampling points. That is formulated as

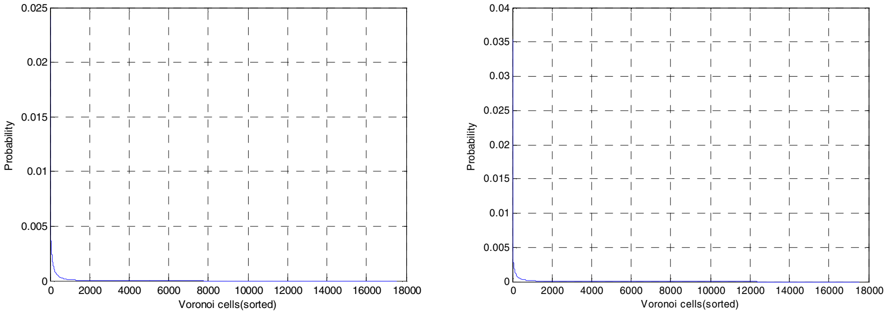
$$\Omega_i = \{\mathbf{x} \in \mathbf{S}^7 \mid d(x,o_i) \le d(x,o_j) \text{ , } \forall o_j \in O\} . \tag{14}$$

The probability density functions are estimated by

$$p(\Omega_i) = \frac{N(\Omega_i)}{\sum_i N(\Omega_i)} \qquad \boldsymbol{i\text{=}1,2,...,17520} , \tag{15}$$

where $N(\Omega_i)$ is the number of data points that fall into the corresponding Ω_i .

We randomly choose 200 PIM images from our PIM dataset (involving only 3 camera makes, i.e. Nikon, Sony and Sony) and get 200,000 PIM edge patches. We use the 100 CG images collected by ourselves from Internet as described in Section 2 and get 98,599 PRCG edge patches. All these edge patches are obtained according to the description in Subsection 3.1. Fig. 3 (a) and (b) respectively show the bins sorted according to decreasing probability for the data points with respect to the two subsets. It is fully demonstrated that when the edge patches are projected onto the 7-sphere, for both PIM and PRCG, the sparseness is evident. That is the majority of the data points are densely clustered in the sphere, and in a large number of Voronoi cells, there are very few or even no data points falling into them. In summary, the data points for both the image categories are scattered highly unequally in the 7-sphere.



(a) (b)

Fig. 3. Probabilities of edge patches in Voronoi cells that are sorted according to decreasing probability: (a) edge patches from the PIM data set; (b) edge patches from the PRCG data set

However, when choosing those bins with larger probability, for example, larger than 0.0015, Fig. 4 shows there are more such bins for PIM patches. We suppose that it is because the margin and texture of real-world objects more likely exhibit certain typical patterns whereas artificial images more frequently present new patterns with artifacts.

At the same time, we find a smaller set of Voronoi cells can pick up the majority of the patches for both the natural images and the computer generated images. For example, if observe those Voronoi cells with more than 100 data points, there are only 1,165 Voronoi cells but they collect over 83% of patch data. Moreover, when new image patches come, they fall into these Voronoi lattices more frequently. We look upon the sampling points corresponding to these Voronoi cells with larger possibility as key sampling points. Furthermore, we construct image edge vocabulary based on these key sampling points.

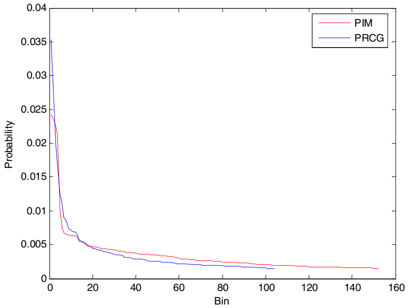


Fig. 4. Bins with possibility larger than 0.0015 (in descending order). 41.29 percents of PRCG patches accumulate in 104 bins while 53.03 percents of PIM patches accumulate in 152 bins.

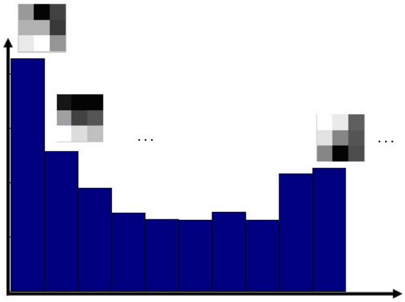


Fig. 5. Bags of visual words. Each bin corresponds to a key sampling point intuitively displayed as a local patch here

To be precise, we choose those sampling points with larger probability in both the histograms of PIM and PRCG (shown in Fig. 3) as the key sampling points. Each key sampling point corresponds to a visual word and all the visual words construct the vocabulary. Fig. 5 illustrates the idea based on bag-of-visual-words. The number of visual words is viewed as the size of the vocabulary denoted as $\|\mathcal{V}\|$. It is likely that an image patch falls outside the Voronoi cells corresponding to the key sampling points. In such a case, the image patch will be clustered into the corresponding visual word in terms of the nearest neighbor rule. As a result of the clustering, the 7-sphere $\mathbf{S}^7$ is covered non-equally by $\|\mathcal{V}\|$ Voronoi cells.

3.3 Image Classification

With the knowledge of the sparse distribution of geometrical structure of image edge patches, we construct the visual edge vocabulary. Herein we focus on the image classification.

We apply a non-linear SVM classifier. The classifier serves binary classification. In this case, an image is represented as a $\|\mathcal{V}\|$-dimensional feature vector. One extreme of this approach is to construct 17520-dimensional vectors. But the statistical analysis above motivates us to construct lower-dimensional vectors relying on the visual edge vocabulary. We expect that it not only dramatically reduce computation cost but also affect little on the classifier performance.

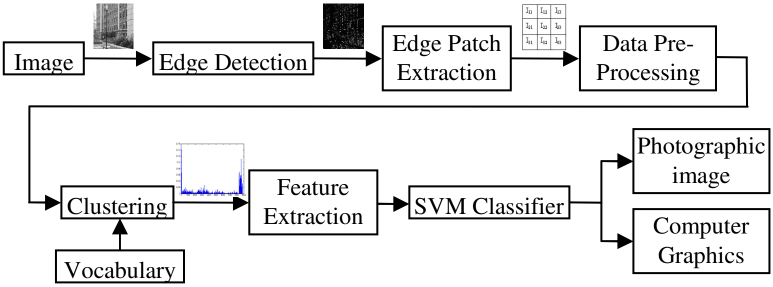


Fig. 6. The proposed image classification algorithm pipeline

In summary, the algorithm of distinguishing between photographic image and photorealistic computer graphics is illustrated as Fig. 6.

4 Experimental Results

We use the remaining 800 PIM and 800 PRCG images from our data sets to evaluate the proposed method on our data sets (see Section 2 for details). The 1,600 images are used for evaluating our method through 10-fold cross-validation. PIM images are labeled with 1 as positive and PRCG images with -1 as negative. We use LIBSVM implementation package [13].

4.1 Image Classification

First, we verify the classification capability on various vocabulary sizes. All samples are randomly divided into 2 groups. One contains 70% of the samples used for cross validation and training the SVM classifier, and the other contains the rest 30% used as test data. Each experiment below is conducted repeatedly on 10 random splits and the average detection rate calculated.

We determine vocabulary size $\|\mathcal{V}\|$ according to different possibility thresholds. The experimental results are shown in Table 1. Wherein TP (true positive) denotes the rate of correctly detected PIM, TN (true negative) denotes the rate of correctly detected PRCG, and the accuracy is the arithmetic average of TP and TN, as determined through 10-fold cross-validation. At the same time, the classification accuracy for training samples is given here.

As expected, it is shown in Table 1 that as $\|\mathcal{V}\|$ becomes larger, the classifier performance gets better. This result is consistent to the findings in large-scale image retrieval [14, 15] where larger vocabulary set leads to better retrieval accuracy. However, larger vocabulary size will bring computational intensity when clustering and training the classifier. In this work, we can select a better tradeoff in practical applications. In future work, we will explore the possibility of enlarging the vocabulary size to the order

Table 1. Classification accuracy of different vocabulary sizes

$\|\mathcal{V}\|$	TP	TN	Cross-validation Accuracy	Train-accuracy
14	88.2	87.7	88	91.3
172	96	94	95	97.9
256	96	95.4	95.7	99.1
1165	96.6	95.4	96	99.5
2782	96.6	95.5	96.1	99.6

of million on a larger data set by either adopting a more efficient representation such as the hierarchical vocabulary tree structure [14] or various efficient clustering methods such as an approximate k-mean [15].

We select 256-sized vocabulary for our approach here. Table 2 shows the 256 features based on visual vocabulary on local image edge can achieve classification accuracy comparable to the 216 wavelet based features [5].

Table 2. Comparison of classification performance

	TP	TN	Accuracy
Proposed (256f)	**96**	95.4	95.7
[5] (216f)	95.5	**98.1**	**97.2**

4.2 Generalization Capability

In our PIM set, images are taken by 8 cameras including 6 camera manufacturers. To verify the generalization capability of the two methods, we remove *all* (50) images taken by the Samsung camera from the PIM samples and train the classifier with the remaining 750 PIM images and 800 Columbia PRCG images. In this experiment, only one image is incorrectly classified (classification accuracy 98%) with the proposed method while all images fail to be correctly detected with that of [5]. Note that, we consider this experiment a test for the generalization capability of our method which, in this case, outperforms the method in [5]. This result may indicate that the visual vocabulary based on local edge patch can characterize the general property of a special image source better. In future work, we will perform a more comprehensive experiment on evaluating the generalizability of various competing methods with more sets of different holdout data.

4.3 Compression Attack

In the following experiments, we are interested to see how compression attack affects the classifier performance since compression can cause the loss of image edge information.

In fact, all of the above images in the data sets are in JPEG format. We compressed the 1600 images respectively with quality factor 90, 70, and 40. The 1600 original images from the data sets are used as training samples and the compressed images are used as unseen testing samples. Table 3 shows that, the proposed approach has rather stable test performance for PRCG for all three compression settings, as compared to the method in [5]. For PIM, the proposed approach outperforms [5] except for JPEG 40. We wish to understand how to ensure that a classification method is stable under almost all compression settings.

Table 3. Comparative experimental results of the proposed approach and [5] on datasets with different JPEG compression factors

Dataset	proposed	[5]
PIM, JPEG 90	**100**	61
PRCG, JPEG 90	96.1	**100**
PIM, JPEG 70	**82.4**	48.1
PRCG, JPEG 70	**95.9**	81.4
PIM, JPEG 40	50.4	**74.5**
PRCG, JPEG 40	**96.3**	91

5 Conclusions and Future Work

In this paper, we have proposed a new approach of PIM and PRCG classification through the combination of the mathematic framework in [7] and the idea of the bag-of-visual-words in [10]. First of all, we pay attention to local edge patches in images because those patches carry most of image information. By projecting the image patch data onto a 7-dimensional sphere with a series of transforms, we observe the distribution of data points in individual Voronoi lattice. The visual vocabulary is constructed through determining the key sampling points corresponding to specific Voronoi cells rather than finding cluster center by partitioning as [10]. The feature vectors are constructed by histogram bins of visual words. The classification is implemented via a binary-value SVM classifier. Our experimental results demonstrate that the proposed features not only have efficient image discrimination but also present certain generalization and resistance to JPEG compression attack.

Our statistical analysis and classification experimental results reveal the fact that the intrinsic difference between photographic images and photorealistic computer graphics may be captured by the geometry structure of local edge patches. The conclusion is of great significance for digital image forensic as well as photorealism evaluation for computer graphics.

Given the initial but promising results for the visual vocabulary approach in distinguishing photorealistic computer graphic images, as future work, we wish explore this approach further. We will consider a closer analogy to document retrieval by evaluating the effectiveness of the TF-IDF weighting (Term frequency–inverse document frequency) [11] and using stop words in our context for our problem of distinguishing photorealistic computer graphics. Ultimately, we wish to make sense of the visual vocabulary set as it holds the key to the understanding of photorealism, an elusive but important attributes in computer graphics research [16-19].

Acknowledgment. This work is supported by National Natural Science Foundation of China (NSFC:60873220, 61170137), Doctoral Fund of Ministry of Education of China (20103305110002), Zhejiang natural science foundation of China (ZJNSF:Y108022, Z1090622,Y1090285), Zhejiang science & technology preferred projects of China (2010C11025), Zhejiang province education department key project of China (ZD2 009012), Scientific Research Fund of Zhejiang Provincial Education Department of China (Y201018185), Ningbo science & technology preferred projects of China (2009 B10003), Ningbo Municipal Natural Science Foundation of China (2011A610191), Ningbo key service professional education project of China (2010A610115). In addition, our programs are supported by High School Special Fields Construction of Computer Science and Technology (TS10860), Zhejiang Province Excellent Course Project (2007), Zhejiang Province Key teaching material Construction project (ZJB2009074).

Authors would like to thank all who generously contributed their personal digital images for our research work.

References

1. Ng, T.-T., Chang, S.-F., Tsui, M.-P.: Physics-Motivated Features for Distinguishing Photographic Images and Computer Graphics. In: ACM Multimedia, Singapore, pp. 39–248 (2005)
2. Gallagher, A.C., Chen, T.: Image Authentication by Detecting Traces of Demosaicing. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops, Anchorage, pp. 51–62 (2008)
3. Swaminathan, A., Wu, M., Liu, K.J.R.: Digital Image Forensics via Intrinsic Fingerprints. IEEE Trans. Information Forensics and Security 3(1), 101–117 (2008)
4. Fridrich, J.: Digital Image Forensic Using Sensor Noise. IEEE Trans. Signal Processing 26, 26–37 (2009)
5. Lyu, S., Farid, H.: How Realistic is Photorealistic? IEEE Trans. Signal Processing 53(2), 845–850 (2005)
6. Chen, W., Shi, Y.-Q., Xuan, G.: Identifying Computer Graphics Using HSV Color Model and Statistical Moments of Characteristic Functions. In: IEEE International Conference on Multimedia and Expo., Beijing, pp. 1123–1126 (2007)
7. Lee, A.B., Pedersen, K.S., Mumford, D.: The Nonlinear Statistics of High-Contrast Patches in Natural Images. International Journal of Computer Vision 54(1-3), 83–103 (2003)
8. Ng, T.-T., Chang, S.-F.: Classifying Photographic and Photorealistic Computer Graphic Images Using Natural Image Statistics. ADVENT Technical Report #220-2006-6, Columbia University (October 2004)
9. Ng, T.-T., Chang, S.-F., Tsui, M.-P., et al.: Columbia photographic images and photorealistic computer graphics dataset. ADVENT Technical Report #205-2004-5, Columbia University (February 2005)
10. Csurka, G., Dance, C., Fan, L., Williamowski, J.: Visual Categorization with Bags of Keypoints. In: ECCV 2004 Workshop on Statistical Learning in CV, Prague, pp. 59–74 (2004)
11. Sivic, J., Zisserman, A.: Video Google: A Text Retrieval Approach to Object Matching in Videos. In: IEEE International Conference on Computer Vision (2003)
12. Carlsson, G.: Topology and Data. Bull. Amer. Math. Soc. 46, 255–308 (2009)

13. Hsu, C.-W., Chang, C.-C., Lin, C.-J.: A Practical Guide to Support Vector Classification, April 15 (2010), http://www.csie.ntu.edu.tw/~cjlin
14. Nister, D., Stewenius, H.: Scalable Recognition with a Vocabulary Tree. In: IEEE Conference on Computer Vision and Pattern Recognition (2006)
15. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object Retrieval with Large Vocabularies and Fast Spatial Matching. In: IEEE Conference on Computer Vision and Pattern Recognition (2007)
16. Ferwerda, J.A.: Three Varieties of Realism in Computer Graphics. In: SPIE Human Vision and Electronic Imaging, vol. 3 (2003)
17. Mayer, G.W., Rushmeier, H.E., Cohen, M.F., Greenberg, D.P., Torrance, K.E.: An Experimental Evaluation of Computer Graphics Imagery. In: ACM SIGGRAPH, pp. 30–50 (1986)
18. McNamara, A.: Exploring Perceptual Equivalence between Real and Simulated Imagery. In: ACM Symposium on Applied Perception in Graphics and Visualization, p. 128 (2005)
19. Rademacher, P., Lengyel, J., Cutrell, E., Whitted, T.: Measuring the perception of visual realism in images. In: Proceedings of the Eurographics Workshop on Rendering Techniques, pp. 235–248 (2001)

Exposing Original and Duplicated Regions Using SIFT Features and Resampling Traces*

David Vázquez-Padín[1] and Fernando Pérez-González[1,2,3]

[1] University of Vigo, Signal Theory and Communications Dept., Vigo, Spain
[2] GRADIANT, Vigo, Spain
[3] University of New Mexico, Dept. of Electrical and Computer Engineering, USA
{dvazquez,fperez}@gts.uvigo.es

Abstract. A common type of digital image forgery is the duplication of a region in the same image to conceal something in a captured scene. The detection of region duplication forgeries has been recently addressed using methods based on SIFT features that provide points of the regions involved in the tampering and also the parameters of the geometric transformation between both regions. However, considering this output, there is not yet any information about which of the regions are originals and which are the duplicated ones. A reliable image forensic analysis must provide this information. In this paper, we propose to use a resampling-based method to provide an accurate way to distinguish the original and the tampered regions by analizing the resampling factor of each area. Comparative results are presented to evaluate the performance of the combination of both methods.

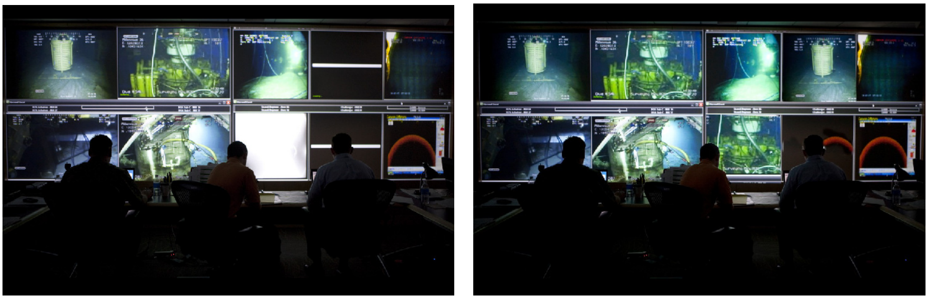
Keywords: Image forensics, region duplication, resampling estimation, SIFT

1 Introduction

Today, digital images are widely used to inform, communicate and interact with people, above all, through the Internet. Due to the huge proliferation of visual information, a lot of image editing tools were designed initially to enhance the quality of digital images, but in the meantime these tools also allow their manipulation, alteration and even the creation of realistic synthetic images. So, nowadays, we often have to deal with cases where an image cannot be considered as an undeniable proof of occurrence of an event. For instance, very recently, we

* Research supported by the European Union under project REWIND (Grant Agreement Number 268478), the European Regional Development Fund (ERDF) and the Spanish Government under projects DYNACS (TEC2010-21245-C02-02/TCM) and COMONSENS (CONSOLIDER-INGENIO 2010 CSD2008-00010), and the Galician Regional Government under projects "Consolidation of Research Units" 2009/62, 2010/85 and SCALLOPS (10PXIB322231PR), and by the Iberdrola Foundation through the Prince of Asturias Endowed Chair in Information Science and Related Technologies.

Y.Q. Shi, H.J. Kim, and F. Perez-Gonzalez (Eds.): IWDW 2011, LNCS 7128, pp. 306–320, 2012.



(a) Original image

(b) Tampered image

Fig. 1. Real example of a tampered image (*on the right*) shown in the BP Web site by copying and moving parts of the original image (*on the left*). Courtesy of The Washington Post.

have seen during the BP oil crisis, how the image shown in Fig. 1(a) was doctored on the BP Web site by filling the blank screens with other parts of the original photo (see the result in Fig. 1(b)).

As a mean to prove the authenticity or verify the integrity of an image and cope with these manipulations, a lot of techniques have arisen in the past few years that can be classified as active or passive. Active approaches require a known signal that is embedded in the image to detect forgeries, while passive approaches, also known as blind, work in the absence of any prior information of the original image. Currently, in the context of passive techniques there are several methods that exploit the intrinsic properties of an image [1], allowing for instance: the identification of the source or the origin of an image; the detection of lighting inconsistencies, double compressed images or region duplications; and also the detection and estimation of inconsistencies in the resampling factor of an image. In this paper, we will focus on the detection of duplicated regions and the estimation of the resampling factor on such regions.

Specifically, in this work we combine these two different but complementary forensic tools to get better results and to provide a more accurate forensic analysis of tampered images. The main idea is to mitigate the drawbacks of each technique by using the characteristics of the other. For example, by detecting a cloned region with one of the existing algorithms (e.g. [2] or [3]), it is viable to estimate the geometric relation between the original area and the cloned one, but it is not possible to know which of the two regions is the original and which is the clone. However, by estimating the resampling factor of each zone[1] using any of the methods in [4], [5] or [6], we can differentiate both regions as the original and the duplicated one, since their resampling factors will be different. In the other hand, if the cloned area has not been resized, the resampling estimator cannot help to infer such manipulation (since the resampling factors are equal), but using the region duplication detector this problem is solved.

[1] We are supposing that the copied region has been spatially transformed.

The pros and cons of each technique are discussed in more detail in the next section. In Section 3, the model used is described focusing on how we propose to combine both techniques to improve performance. Experimental results carried out with this image forensic scheme are summarized in Section 4. Finally, Section 5, provides the conclusions and further work.

2 Motivation

In the context of passive image forensics techniques, there does not exist a common framework to analyze images and detect forgeries, i.e. there is not a universal tool that can explicitly determine all the modifications or transformations applied to an image. Instead of that, there is a bunch of tools that exploit some of the inherent characteristics of an image, and in doing so, try to detect the alterations such image has been subject to.

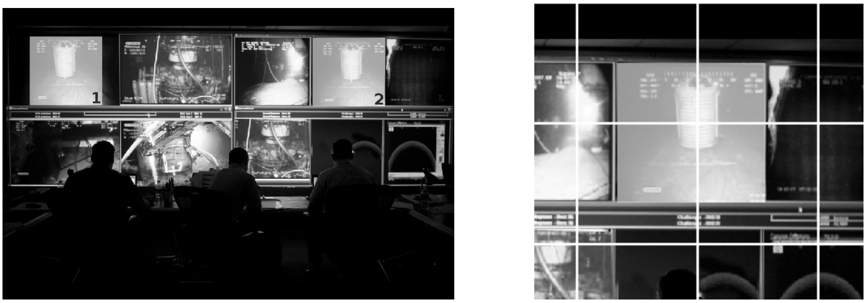
The main objective of this paper is to provide a novel image forensic tool to reach better results in terms of estimation accuracy of digital forgeries, by combining two different techniques that complement the needs of each other. As it was previously mentioned, one of the techniques allows the detection of region duplication forgeries where a part of an image itself is copied, probably geometrically transformed and pasted into another part of the same image to conceal something. The second technique, provides a way to statistically detect and estimate the resampling factor of an image block which gives information about the type of spatial transformation locally applied.

The complementary behavior of both techniques can be established from the analysis of advantages and drawbacks of each one, as it is summarized below.

2.1 Advantages/Drawbacks of Region Duplication Detectors

Starting from the first approach for detection of region duplication based on an exhaustive search and analysis of correlation properties of the image [1], until the most recent methods proposed in [2] or [3] capable of estimating the geometric transformation applied between the duplicated regions; the main shortcoming of all these techniques, supposing that they are able to find the duplicated regions, is the impossibility to identify which are the original regions and which are the duplicated ones.

For example, Fig. 2(a) represents the possible output of any of these methods, highlighting two duplicated regions (tagged with **1** and **2**). Taking only into account the provided output, can we assert that the region labeled as **1** is the source and the region labeled as **2** is the duplicate, from a mathematical point of view? The answer is negative, as these methods only provide a match between different pixel areas. Even being able to estimate the geometric relation between both regions (with the method proposed in [2] or [3]), it is not possible to distinguish, in a mathematical sense, the original region from the cloned patch. The more suitable solution to provide this information is to compute the resampling



(a) Region duplication detection

(b) Resampling analysis

Fig. 2. Examples of drawbacks of each technique. *On the left*, the detected regions are highlighted and tagged with **1** and **2**. *On the right*, the tampered region is highlighted and each analyzed block is denoted by a white border box.

factor of each region and also of the neighborhood and check if both are consistent. By analyzing this relation between the resampling factors, we can identify the tampered regions of the image.

Taking into account the advantages of the region duplication detectors, these methods are able to detect copy-move forgeries[2], while resampling detectors fail looking for inconsistencies in the resampling factor. Besides, the most recent proposed methods based on SIFT ([2] and [3]), allow a very fast analysis of an entire image, in terms of computation time. As a counterpart, they have also an important limitation in terms of detection performance since it is only possible to extract reliable keypoints from peculiar points of the image.

2.2 Advantages/Drawbacks of Resampling Detectors

The detection of resampling traces and the estimation of the resampling factor (or equivalently, the spatial transformation applied to an image block) are closely related and have been studied in several works [4,5,6]. Although these methods provide good results in controlled scenarios, when they are evaluated in more realistic situations, their performance get worse. For instance, looking for a more efficient forensic analysis in terms of computation time, these methods usually process an image using non-overlapped blocks of a fixed size (e.g. 128 × 128 pixels). However, with high probability, the location of a tampered region will not be aligned with the grid defined by these blocks, as it is shown in Fig. 2(b). Thus, in such cases, the detection of the tampered region will fail, since the number of resampled samples included in each block is small with respect to the number of original samples.

An important handicap of these methods is the impossibility to detect copy-move forgeries, since the resampling factor of the whole image remains constant.

[2] A copy-move forgery is considered when the duplicated region is not spatially transformed, just translated.

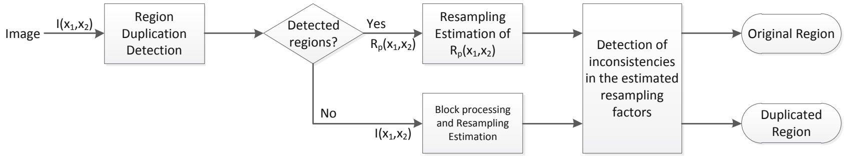


Image
I(x1,x2)
Region Duplication Detection
Detected regions?
Yes
Rp(x1,x2)
No
I(x1,x2)
Resampling Estimation of Rp(x1,x2)
Block processing and Resampling Estimation
Detection of inconsistencies in the estimated resampling factors
Original Region
Duplicated Region

Fig. 3. Block Diagram of the proposed image forensic analysis tool

We have just seen before that this problem can be easily solved by using a region duplication detector. Additionally, the processing of each block of the image, looking for inconsistencies in the resampling factor, is highly time-consuming.

Hence, once we have seen the positive characteristics and also the negative ones of each technique, it can be expected that the combination of both ideas will provide better performance and also a more complete and accurate forensic analysis of tampered images.

3 Model Description

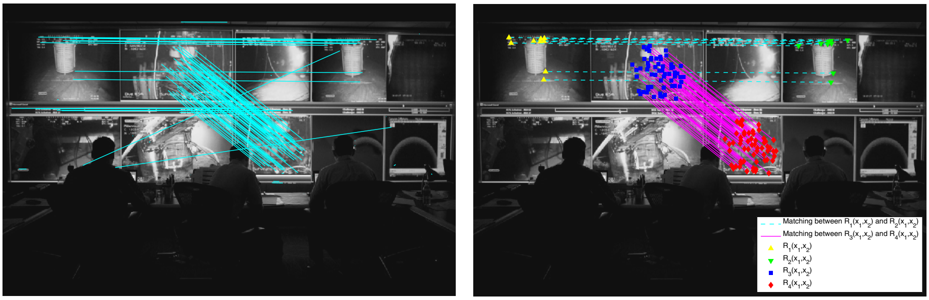
In order to overcome the problem related to the identification and differentiation of the original regions and the tampered ones using a region duplication detector and to avoid the previously mentioned misdetections of the resampling detectors, the proposed approach uses a combination of both techniques.

In Fig. 3 we represent in block diagram form the steps involved in the proposed forensic analysis of an image. As a first step we use a region duplication detector to extract the original and the cloned regions, but if the method is not able to find any tampered region, it is necessary to analyze the entire image by processing blocks and looking for inconsistencies in the resampling factor of each block. Nevertheless, if the region duplication detector is capable of finding the duplicated regions, then the resampling-based method is just applied to estimate the resampling factor of each area. Finally, according to the results obtained in the previous stages, the system determines and differentiates the original regions from the tampered ones.

Next, we describe the specific methods considered for each technique to provide a possible practical implementation of the proposed forensic analysis tool.

3.1 A SIFT-Based Method for Region Duplication Detection

As it was previously introduced, there are several recently published methods based on the matching of image features and keypoints (e.g. [2] and [3]), that provide good results for the detection of duplicated regions. In this paper, we consider the method proposed by Amerini et al.


Matching between R1(x1,x2) and R2(x1,x2)
Matching between R3(x1,x2) and R4(x1,x2)
R1(x1,x2)
R2(x1,x2)
R3(x1,x2)
R4(x1,x2)

(a) Matched keypoints

(b) Clustered and detected regions

Fig. 4. Steps followed for the detection of cloned areas. *On the left side*, solid lines represent the matching between keypoints and, *on the right side*, different markers are used to identify the clustered data and solid/dashed lines link the related regions.

Following the steps described in [2] we start with one of the color space component of a sampled image $I(\boldsymbol{x}) = I(x_1, x_2)$ with size $N_1 \times N_2$ pixels, where $0 \leq x_1 \leq N_1 - 1$ and $0 \leq x_2 \leq N_2 - 1$. We apply the algorithm proposed by Lowe in [7] to reach a set $\mathcal{X}$ of n keypoints: $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \mid \boldsymbol{x}_k = (x_1, x_2)\}$; with their respective SIFT descriptors: $\mathcal{D} = \{\boldsymbol{d}_1, \ldots, \boldsymbol{d}_n\}$, where each $\boldsymbol{d}_k$ is a 128-dimensional vector. Since the descriptors of a duplicated region will look like those of the original area, we want to identify the nearest neighbor of each descriptor to find a possible match of similar keypoints. Thus, a vector that contains the Euclidean distance between each pair of descriptors is computed for each descriptor $\boldsymbol{d}_k$, obtaining a set

$$\mathcal{S} = \{s_l = \|\boldsymbol{d}_i - \boldsymbol{d}_j\|_2 \mid j \in \{1, \ldots, n\}, j \neq i\}$$

that will be sorted in ascending order, for convenience. The matching between keypoints is satisfied if the ratio between the distance of the closest neighbor s_1 and that of the second-closest one s_2 is less than a threshold Υ, i.e.

$$\frac{s_1}{s_2} < \Upsilon.$$

For instance, considering a threshold $\Upsilon = 0.6$ and applying this procedure to the BP tampered image shown in Fig. 1(b), we get the result depicted in Fig. 4(a). Once the set of matched keypoints $\mathcal{X}_m$ is obtained, it is necessary to cluster these data in such a way as to be able to distinguish the different matched regions.

For clustering on the spatial location of the matched points, an agglomerative hierarchical clustering is used as it is proposed in [2]. Considering that we have at least two matched areas, the result of this process provides $P \geq 2$ different sets of matched points $\mathcal{M}_p$, so $\mathcal{X}_m = \mathcal{M}_1 \cup \cdots \cup \mathcal{M}_P$, and this allows the definition of the different duplicated regions.

Table 1. Followed steps by the method proposed in [4]

For each frequency pair (α_1, α_2) of the $N \times N$ 2-D FFT grid:
1. From the data $R_i(x_1, x_2)$, build up a cyclostationary vector $\hat{\boldsymbol{c}}_{xx}$, for a set of lags $\boldsymbol{\tau}$.
2. Estimate the covariance matrix $\hat{\boldsymbol{\Sigma}}_{xx}$.
3. Compute the test statistic $\mathcal{T}_{xx} = N^2 \hat{\boldsymbol{c}}_{xx}^H \hat{\boldsymbol{\Sigma}}_{xx} \hat{\boldsymbol{c}}_{xx}$.
4. Set $\varGamma$ based on the probability of false alarm P_F.
5. Consider a cyclic frequency pair if $\mathcal{T}_{xx} > \varGamma$.
From the set of cyclic frequencies estimate the transformation.

Continuing with the BP doctored image, we illustrate in Fig. 4(b), the four set of points that determine the two different tampered regions matched with the solid and dashed lines. Note that some outliers have been removed after the clustering process.

From the points in a region $\boldsymbol{x}_q \in \mathcal{M}_q$ and the corresponding matched points $\boldsymbol{x}_r \in \mathcal{M}_r$, we can estimate the geometric transformation applied between the two matched areas:

$$\begin{bmatrix} \boldsymbol{x}_q^T \\ 1 \end{bmatrix} = \boldsymbol{H}_{qr} \begin{bmatrix} \boldsymbol{x}_r^T \\ 1 \end{bmatrix},$$

where $\boldsymbol{H}_{qr}$ represents an affine homography. By using the Random Sample Consensus (RANSAC) algorithm, a Maximum Likelihood estimation of the affine homography $\boldsymbol{H}_{qr}$ can be carried out.

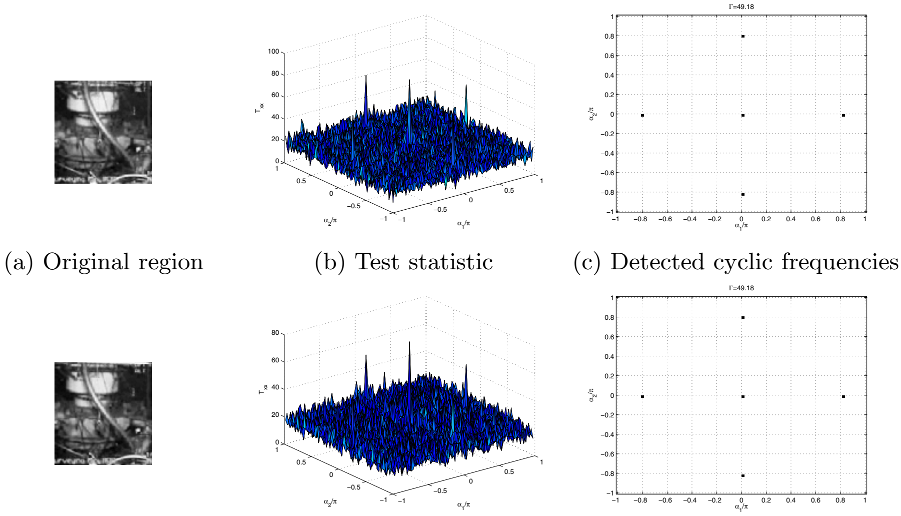
Now, suppose that from the SIFT-based method we obtain $P = 2$ identified regions $R_1(x_1, x_2)$ and $R_2(x_1, x_2)$ and also the estimation of the relation between both $\hat{\boldsymbol{H}}_{12}$, then, using this information, can we demonstrate that $R_1(x_1, x_2)$ is the original area and $R_2(x_1, x_2)$ is the duplicated one, or vice-versa? The method explained below will help to answer this question.

3.2 A Resampling-Based Method to Reveal Tampered Regions

An appropriate way to determine if a matched region is the source or the duplicated one, is to use a resampling estimator that gives a measure of the applied spatial transformation, based on the intrinsic properties of the image pixel region. If the SIFT-based method is not able to find any duplicated region, then we can use any of the proposed methods [4],[5] or [6] to make an exhaustive analysis, processing all the blocks of the image and looking for inconsistencies in the resampling factor.

However, we are more interested in the case when the SIFT-based method does provide the detected cloned regions. So, considering that we get two regions $R_1(x_1, x_2)$ and $R_2(x_1, x_2)$ and taking into account that these regions are generally non-square, for the identification of the original and the duplicated one, we will use the method proposed in [4], which takes a block of the image and applies a statistical test for the evaluation of the presence of almost cyclostationarity in the analyzed block. The steps followed by the method are summarized in Table 1.

Since this method works with square blocks, we have to adapt the detected regions to get a square form. A simple way to do that is to take a square region



(a) Original region (b) Test statistic (c) Detected cyclic frequencies

(d) Duplicated region (e) Test statistic (f) Detected cyclic frequencies

Fig. 5. Application of the two-dimensional statistical test to one of the pair of matched regions in the BP image: $R_3(x_1, x_2)$ (*top row*) and $R_4(x_1, x_2)$ (*bottom row*)

that includes the area under analysis and pad with zeros all the pixels that are not contained in the region $R_i(x_1, x_2)$. The zero-padding approach is probably a suboptimal solution, but doing this we can estimate the resampling factor for each region $R_i(x_1, x_2)$. One of the objectives of this paper is also to study the performance of this method in such scenario.

As we have stated before, a resampling detector cannot differentiate the original source from the duplicated versions if a copy-move forgery is performed. That is exactly what happens with the tampered regions, labeled as $R_2(x_1, x_2)$ and $R_4(x_1, x_2)$ in Fig. 4(b). In fact, applying the statistical test to the matched regions $R_3(x_1, x_2)$ and $R_4(x_1, x_2)$, we obtain the same resampling factor ($\hat{\rho} \approx 5/3$) in both cases, as we can see in Fig. 5. Thus, in this particular scenario, the resampling-based method only identifies the scaling factor applied to the image, but it is not able to distinguish the source region from the clone (since the resampling factor is the same).

However, considering that the pasted regions are geometrically adapted to the scene, then to determine which parts of the image are the copies and which one of those is the source, it is enough to analyze the neighborhood of each region. So, taking a square block that only includes the adjacent neighbor pixels of each region $R_i(x_1, x_2)$ (removing the pixels that belong to the area under analysis), the resampling factor of the neighborhood can be estimated. Finally, for the classification of the regions, we know that an original region will have the same resampling factor in the neighborhood and inside the region, but the tampered regions will have different values in both cases.

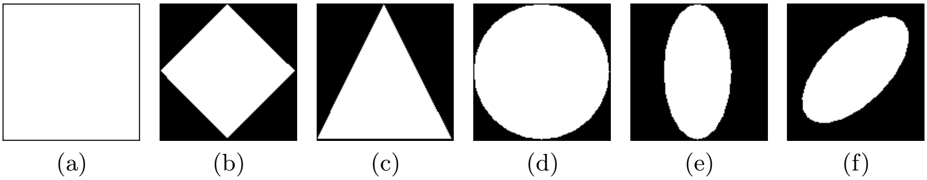


Fig. 6. Different masks used to test the performance of the proposed forensic tool

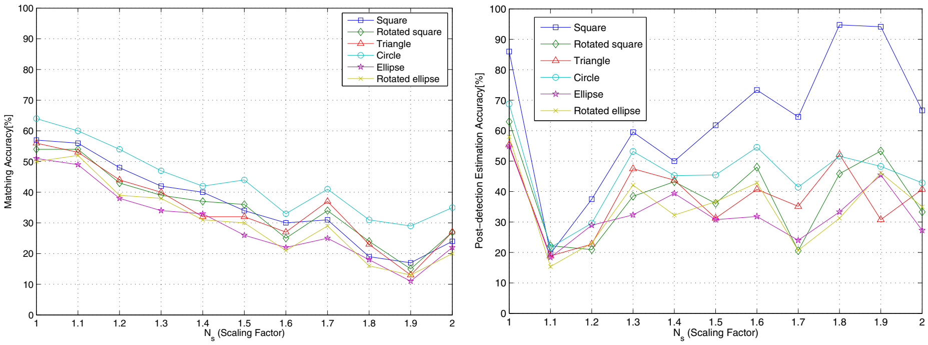
4 Experimental Results

For the evaluation of this image forensic scheme, we use 100 images from a personal image database composed by several realistic scenarios with different indoor and outdoor scenes. All the images in this collection have been captured in a RAW format by a Nikon D60 digital camera and have been converted into uncompressed TIFF images in the RGB color space. The original resolution of each image was 3872 × 2592, but due to the increase of computational complexity, all the images were cropped to 1024 × 1024 pixels. The resampling factor of each color channel is equal to 2, due to the color filter array (CFA) interpolation performed inside the camera. This fact will be taken into consideration along the application of the resampling-based method and for simplicity we will only process the green component of the RGB color space.

To test the performance of the proposed scheme (Fig. 3), as a first step we evaluate the SIFT-based method and the resampling-based method separately, and then we combine both to see how the results of the forensic analysis improve. In order to get more realistic forgeries in our experiments, we use six different patterns for the duplicated areas, that are depicted in Fig. 6. We use these masks to copy a region located at a random position of an image, then we scale this region by one of the scaling factors N_s in the set $\{1, 1.1, \ldots, 2\}$ and, finally, we paste the duplicated region in a new random location on the same image. The random position of both regions is the same for all masks in order to make a fear comparison, but this one changes for different scaling factors and for each image. Since the tampered regions tend to be relatively small, we have made the experiments in such a way that the resampled region fits always in a 128 × 128 block.

For the SIFT-based method we use a threshold $\Upsilon = 0.6$, we remove false positive matching points if their distance is less than 10 (i.e. $\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2 < 10$) and once we get the hierarchical clustering we remove the outliers of each region if their distance to the mean point of their corresponding region is higher than 3 times the variance of the points in the considered region. The implementation of the SIFT algorithm used in the following experiments has been taken from [8] and for the RANSAC homography estimation we have used the functions available from [9].

The configuration of the resampling-based method is almost the same as the one used in [4] (i.e. we use a spectral window to smooth the periodogram of size 11 × 11 and a set of $K = 9$ lags), but we do not use the threshold Γ to detect



(a) SIFT-based method

(b) SIFT-based method

Fig. 7. Matching and post-detection estimation accuracy (in terms of percentage), obtained with the SIFT-based method for different masks and scaling factors.

the cyclic frequencies, since we just estimate the applied transformation (i.e. a scaling factor) from the cyclic frequency with highest magnitude, excluding the frequency at zero (DC).

4.1 Detection Results Using the SIFT-Based Method

Taking into account the described set of tampered images, we consider that the SIFT-based method matches correctly a tampered area if it is able to find at least four common points between the original and the duplicated region. Fig. 7(a) depicts the matching accuracy of this method in terms of percentage, showing the different results for each used mask and for the different values of the scaling factor N_s.

Next to this graph, Fig. 7(b) shows the (post-detection) estimation accuracy of the affine transformation applied between the previously matched regions, using the RANSAC method. Note that we are drawing the post-detection estimation accuracy, i.e. the estimation accuracy of the scaling factor applied between the correctly matched regions in the previous step (thus, it is clear that the represented percentage of accurate estimation is not relative to the 100 images of the database). In this case, since we cannot know which region is the original we get two possible estimations: $\hat{\boldsymbol{G}}_{12} \approx \boldsymbol{H}_{12}$ or $\hat{\boldsymbol{G}}_{12} \approx \boldsymbol{H}_{12}^{-1}$. We consider that the estimation is correct if any of both estimated affine transform satisfies $|\hat{\boldsymbol{G}}_{12}(1,1) - N_s| \leq 0.05$ or $|\hat{\boldsymbol{G}}_{12}(2,2) - N_s| \leq 0.05$, where $\hat{\boldsymbol{G}}_{12}(i,j)$ represents the element of the matrix $\hat{\boldsymbol{G}}_{12}$ located at the ith row and at the jth column. In this case, $\hat{\boldsymbol{G}}_{12}(1,1)$ and $\hat{\boldsymbol{G}}_{12}(2,2)$ represent the estimation of the scaling factor applied in each axis in the affine transformation.

As we can observe from the two graphs of Fig. 7, with the SIFT-based method it is easier to match and estimate copy-move forgeries than duplicated regions that have been geometrically transformed. However, from the estimation point of view, it is more difficult to estimate the homography for scaling factors near one like $N_s = 1.1$ or $N_s = 1.2$, than for higher values. The matching accuracy is not very high, due to the lack of reliable keypoints in several images of the dataset (the number of keypoints per image was in the range $[250, 17500]$), but, as it was mentioned earlier, this is an intrinsic limitation of any SIFT-based method. With respect to the used masks, the intuitive idea that small areas are more challenging for detection and estimation purposes, comes up in both plots.

At this point we are just able to find matches between regions and estimate the relation between both, but we cannot indicate which is the source and which is the forged region.

4.2 Detection Results Using the Resampling-Based Method

Before considering the union of the two methods, we evaluate the resampling-based method when it is applied to the whole image, processing block by block to find inconsistencies in the resampling factor ρ. As it was previously noticed, due to the CFA interpolation applied by the camera, we know that the resampling factor of each non-tampered block is $\rho = \rho_{\text{orig}} = 2$, and then the corresponding value to a scaled version by N_s will be $\rho = \rho_{\text{orig}} \times N_s$. Therefore, once we attain a different value from ρ_{orig} we tag the block under analysis as a digitally forged region. In this case, because the tampered regions have a similar size, we use a 128×128 block of analysis.

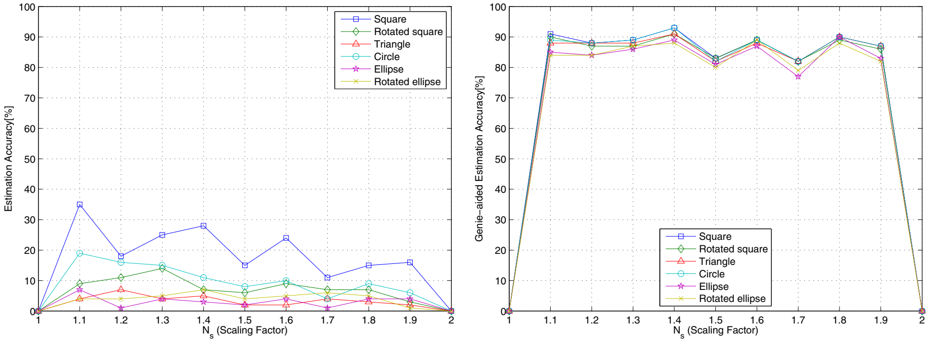
The classification of every single block is performed by analyzing the test statistic $\mathcal{T}_{xx}$ computed in each case. As we have said at the beginning of Section 4, the resampling factor is estimated from the cyclic frequency (α_1, α_2) with highest magnitude (excluding DC), and using the following relation:

$$\hat{\rho} = \max_{i \in \{1,2\}} \hat{\rho}_i = \max_{i \in \{1,2\}} \frac{2\pi}{|\alpha_i|}$$

where we have exploited the fact that, in this case, $\rho \geq 2$ since $1 \leq N_s \leq 2$. We consider that the detection of the tampered region is correct if we discover any inconsistency in the resampling factor (i.e. $\hat{\rho} \neq \rho_{\text{orig}}$) and the corresponding estimated resampling factor $\hat{\rho}$ satisfies $|\hat{\rho} - 2N_s| < 0.05$ and since we have the interference created by the CFA pattern we will also check if $|\hat{\rho}/(\hat{\rho} - 1) - N_s| < 0.05$ is satisfied.

Applying this approach to the tampered images of the database, we obtain the results shown in Fig. 8(a). As we have stated before, this method cannot detect copy-move forgeries, since there are not inconsistencies in the resampling factor along the whole image and that is the reason why the estimation accuracy is equal to zero at $N_s = 1$. Given the ambiguity inserted in the estimation, caused by the CFA pattern, we are not able to distinguish between a scaling factor $N_s = 1$ or $N_s = 2$, and that is why the estimation accuracy is also zero for $N_s = 2$. The rate of accurate estimation of the tampered region is not very high for all the masks used (in the best case we barely reach a 35%), so this method presents very bad performance when identifying forgeries.

Nevertheless, to demonstrate the generally good performance of the resampling estimator, we analyze the estimation accuracy in an ideal case where we



(a) Processing block by block (b) Using exact matching (genie-aided)

Fig. 8. Estimation accuracy (in terms of percentage), obtained through the application of the resampling-based method in two different scenarios for several masks and scaling factors

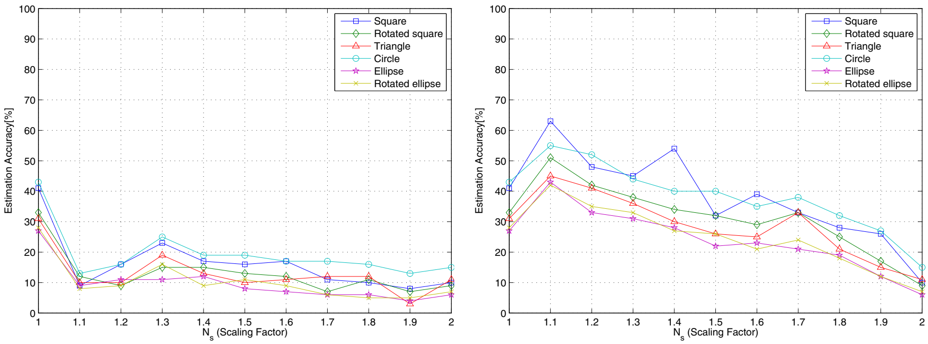
use the information supplied by a *genie* that tell us exactly the location of the original region and that of the tampered region (the application of a "genie-aided" detection is commonly used in communications to determine performance bounds). Thus, knowing exactly the location of both regions in the pixel domain and using the same criteria for the estimation of ρ, as in the previous scenario, we show in Fig. 8(b) the results of accurately estimate which region is the original and which is the duplicated. As it was said before, the correct distinction of the two regions when a spatial transformation has not been applied is not possible with the resampling-based method. However, the detection performance is very high (around a 90% for all the masks) if we compare it with the obtained when the image is processed block by block.

So, according to the results collected in this ideal case, the problem does not lie in the resampling estimator itself, but in the correct matching of the tampered area, and that is the reason why a SIFT-based method is needed.

4.3 Detection Results Combining Both Methods

As it was discussed along the paper, the combination of both methods provides a deeper and enhanced forensic analysis of the tampered regions (since we are able to identify which region is the source and which is the duplicated one) and it also brings a way to compensate the drawbacks of each method with the advantages of the other.

Certainly, since the SIFT-based method is not capable to find all the duplicated regions, mostly due to the unavoidable lack of reliable keypoints, combining both approaches we will get worse results than those depicted in Fig. 8(b) (i.e. the ideal "genie-aided" case where we perfectly match all the regions). However, with the use of the SIFT-based method, the detection of the tampered regions



(a) SIFT-based method

(b) Proposed forensic tool

Fig. 9. Comparative results of the estimation accuracy for the SIFT-based method and the proposed forensic tool

is more accurate than processing the image block by block, so we will get better results than those included in Fig. 8(a). Finally, since the estimation of the resampling factor is not so dependent on outliers as it is the case for the estimate of the homography, we will also get better results than those comprised in Fig. 9(a), where we represent the estimation accuracy of the SIFT-based method when it is able to correctly match the two regions and also estimate their geometric relation. Explicitly, the estimation accuracy ploted in Fig. 9(a), corresponds to the product of the accuracy rates achieved in the matching step (Fig. 7(a)) and in the post-detection estimation step (Fig. 7(b)).

In Fig. 9(b) we can see the inferred estimation accuracy of the proposed forensic tool for different masks and scaling factors. If we compare this plot with the corresponding estimation accuracy obtained with the SIFT-based method alone (depicted in Fig. 9(a)), we can observe that with the scheme described in Fig. 3, performance is improved for almost all the scaling factors and masks considered. It is important to note that the resampling estimator takes as input the exact matching of the detected regions by the SIFT-based method, so the results provided can be considered as an upper performance bound of the estimation accuracy that we can attain with this approach.

Note also that, even with the combination of both methods, we are still not able to distinguish the original region from the tampered one when a copy-move forgery is carried out. Besides, in this particular case, occasioned by the CFA interpolation of the camera, we are neither able to identify the duplicated regions by a factor $N_s = 2$. Hence, the estimation accuracy should be strictly zero for the scaling factors $N_s = 1$ and $N_s = 2$ in Fig. 9(b). However, since with the SIFT-based method we are able to match the involved regions in the tampering and also their relation, then we add the estimation accuracy of this method in both cases, and that is the reason why we have the same values of estimation accuracy for the scaling factors $N_s = 1$ and $N_s = 2$ in both graphs of Fig. 9.

By comparing the estimation accuracy of the resampling-based method (processing block by block) with that obtained with the concatenation of both methods, we achieve an improvement of the exact classification of each region for all the scaling factors and masks considered. In addition, as it was expected, the best results are reached with those masks that have the largest area.

According to the results shown in this section, we can conclude that the forensic analysis scheme proposed in this paper provides a more accurate analysis since we can identify in an image where are located and which are the original regions and the tampered ones when a region duplication forgery is performed. Moreover, the performance in terms of estimation accuracy is increased with respect to the use, in an independent way, of the SIFT-based and the resampling-based methods.

5 Conclusions and Further Lines

In this paper we have introduced a new scheme for image forensic analysis, by combining two complementary methods. The former, based on SIFT, is capable of finding duplicated regions and the latter, based on a resampling estimator, allows one to identify which region is the source and which is the tampered one. The proposed forensic analysis scheme provides better estimation results than considering each method separately.

As future research lines we will focus on the application of this method using JPEG compressed images trying to get similar results as in the case of uncompressed images. Another interesting question is the estimation of the resampling factor on a non-square area, since the zero-padding technique is not the optimal one.

References

1. Sencar, H.T., Memon, N.: Overview of state-of-the-art in digital image forensics. Part of Indian Statistical Institute Platinum Jubilee Monograph series titled Statistical Science and Interdisciplinary Research. World Scientific Press (2008)
2. Amerini, I., Ballan, L., Caldelli, R., Del Bimbo, A., Serra, G.: Geometric Tampering Estimation by Means of a SIFT-based Forensic Analysis. In: Proceedings of 2010 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP), pp. 1702–1705 (2010)
3. Pan, X., Lyu, S.: Region duplication detection using image feature matching. IEEE Transactions on Information Forensics and Security 5(4), 857–867 (2010)
4. Vázquez-Padín, D., Mosquera, C., Pérez-González, F.: Two-dimensional statistical test for the presence of almost cyclostationarity on images. In: Proceedings of 2010 17th IEEE International Conference on Image Processing (ICIP), pp. 1745–1748 (2010)
5. Mahdian, B., Saic, S.: Blind authentication using periodic properties of interpolation. IEEE Transactions on Information Forensics and Security 3(3), 529–538 (2008)
6. Kirchner, M., Gloe, T.: On resampling detection in re-compressed images. In: Proceedings of 2009 First IEEE International Workshop on Information Forensics and Security (WIFS), pp. 21–25 (2009)

7. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision 60, 91–110 (2004)
8. SIFT Keypoint Detector, `http://www.cs.ubc.ca/~lowe/keypoints`
9. RANSAC algorithm, `http://www.csse.uwa.edu.au/~pk/research/matlabfns`

Fingerprint Forensics Application Protocol: Semi-automated Modeling and Verification of Watermark-Based Communication Using CASPER and FDR

Ronny Merkel, Christian Kraetzer, Robert Altschaffel, Eric Clausing,
Maik Schott, and Jana Dittmann

Department of Computer Science, Research Group Multimedia and Security
Otto-von-Guericke-University of Magdeburg, Germany
{kraetzer,merkel,schott,dittmann}@iti.cs.uni-magdeburg.de
{robert.altschaffel,eric.clausing}@student.uni-magdeburg.de

Abstract. Recently, the technique of semi-automated protocol verification using model-checkers was transferred from cryptography to the domain of watermark-based communication protocols. This technique offers cost-effective security verification of such watermark-based protocols and an increased flexibility in comparison to traditionally applied, manual mathematical proofs. In this paper, we want to evaluate the feasibility of this approach, using the modeling language CASPER and the model-checker FDR. We extract the prospects and limitations of the approach by modeling and verifying a practical application scenario for forensic investigations using high-resolution biometric fingerprint data. We evaluate the security aspects, which can be verified by the scheme, as well as the syntactical limitations, complexity limitations and the methodological limitations and indentify necessary improvements for a practical usage of such scheme.

Keywords: watermark-based communication protocols, semi-automated security verification, model-checking, CASPER, FDR.

1 Introduction

Model-checking is a common technique in cryptography to evaluate the security of communication protocols. Using such an approach, the underlying cryptographic primitives are assumed to be 'secure' ([1]), while the protocol run itself is examined for flaws. Although the possibility of a manual mathematical proof exists as an alternative to the modeling and automated verification of such protocols, it is rather difficult and time consuming in comparison to an automated verification.

In prior work, we have transferred such model-checking techniques from cryptography to the domain of digital watermarking in [2] for the semi-automated modeling and verification of watermark-based communication protocols. We suggest in [2] a six step

Y.Q. Shi, H.J. Kim, and F. Perez-Gonzalez (Eds.): IWDW 2011, LNCS 7128, pp. 321–335, 2012.

procedure to model and verify a watermark-based communication protocol using XML, the formalization language CASPER ([3]) and the model-checker FDR ([4]). The approach sounds very promising, enabling a comparatively fast and reliable verification of the security of watermark-based communication protocols, which might be used in different application scenarios such as content protection applications, hidden transfer of secret payload or the chain of custody preservation in a forensic investigation. However, the scheme faces severe limitations at this point in time.

In this paper, we want to investigate the feasibility of our approach. We design a watermark-based communication protocol which might be used in a forensic investigation to protect high-resolution biometric fingerprint traces captured from a crime scene as well as to preserve the chain of custody in a forensic investigation. By modeling and verifying such exemplary watermark communication protocol with the scheme, we show its prospects and limitations within a practical application. We analyze in particular:

- Which *security aspects* can be evaluated using our scheme?
- What are the *syntactical limitations* of the CASPER modeling language?
- Which *complexity limitations* exist for the verification using CASPER and FDR?
- Which *methodology limitations* arise from the practical realization of the scheme?

The paper is structured as follows: in the next section, we want to briefly introduce our methodology from [2]. We then propose a practical application scenario in section three, using a watermarking scheme for enabling the privacy- and chain of custody preservation in a fingerprint forensics investigation scenario. In section four, we model and verify a communication protocol for that scenario using our methodology from [2]. We evaluate the encountered prospects and limitations in section five, proposing also how to improve the scheme. Section six concludes the paper and discusses future work.

2 State of the Art

To the best of our knowledge, we are the first to propose using a modeling and verification approach from the domain of cryptography for the evaluation of watermark-based communication protocols, as reported in [2]. We suggest a procedure comprised of six steps using XML, the modeling language CASPER ([3], [5]) and the model-checker FDR (Failures-Divergences Refinement; [4]). The scheme is depicted in figure 1.

In the first step of our methodology, a real world application scenario is modeled using the XML language. Here, the network, tasks, present watermarking schemes and message layout of the given application scenario need to be specified. For the description of the network, each network-node is modeled, including the present links to neighboring nodes, the processing capacities of the nodes and links as well as the known watermarking algorithms. The tasks are modeled in form of a given start- and destination node as well as transmission constraints, such as the watermark algorithm to be used, required capacities or infrastructures (such as PKI) and nodes which are required to be passed on the way. The specific watermarking algorithms known by each node of the network are also modeled in detail, specifying the used cover, capacity, hierarchy levels and the message layout of the payload, representing the main structure of the watermark.

In the second step of the methodology, a path search is performed, identifying all routes through the network, which fulfill the specifications given by a task. If no route is found, the scheme is exited with a failure statement, not being able to establish a logical link fulfilling the specified requirements. In the following third step, all found routes are parsed and the optimal route according to a cost function is selected.

From a final XML-structure, the results are automatically translated into the CASPER modeling language in the fourth step of our methodology. Here, a manual adjustment of the model has to be performed, adding possible attacker scenarios (specified by the attackers knowledge) and security aspects to be verified (at this point in time, only confidentiality and entity-authenticity can be verified with the scheme, see also section 5).

In the fifth step, the CASPER code is compiled into the Communicating Sequential Processes (CSP; see [6]) and is checked in the sixth step using the model-checker FDR. The model-checker assigns a security statement to a given protocol. In case this statement includes possible attacks found on the protocol, it needs to be hardened accordingly or alternative solutions of the path search in step two of the methodology have to be considered.

In the scope of this paper, we want to apply the introduced methodology to a proposed watermark-based communication protocol for forensic applications, to evaluate its feasibility for a practical usage.

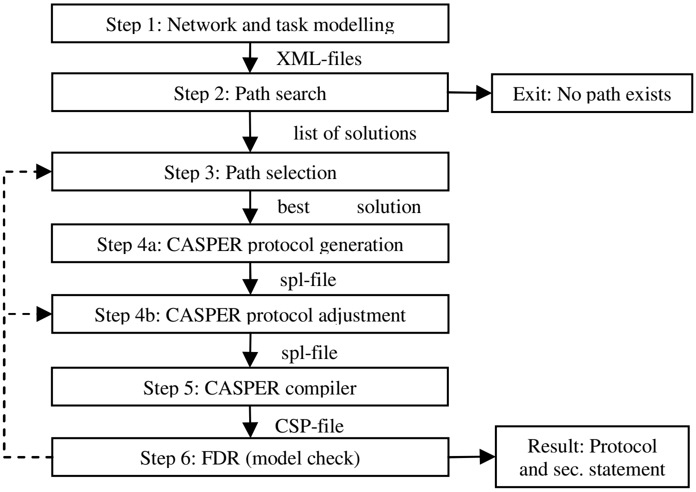

Step 1: Network and task modelling
XML-files
Step 2: Path search
Exit: No path exists
list of solutions
Step 3: Path selection
best solution
Step 4a: CASPER protocol generation
spl-file
Step 4b: CASPER protocol adjustment
spl-file
Step 5: CASPER compiler
CSP-file
Step 6: FDR (model check)
Result: Protocol and sec. statement

Fig. 1. Procedure for watermark protocol generation and verification according to our developed scheme, taken from [2]

A few other modeling and verification approaches also exist, such as AVISPA ([7]), REBECA ([8]) or Athena ([9]) which might also be adapted for the modeling, generation and/or verification of watermark-based communication protocols, but face similar severe restrictions than the approach using CASPER and FDR.

3 Design of Our Proposed Watermark-Based Communication Protocol for Forensic Investigations

To determine the feasibility of the semi-automated modeling and verification approach using CASPER and FDR, we model and verify an exemplary proposed digital watermarking scheme for forensic investigations. The scheme is designed to assure the privacy of high resolution biometric data captured optical and contactless by the FRT MicroProf 200 CWL 600 (Chromatic White Light) sensor [10]. This sensor is transferred from the domain of surface quality measurement to the non-invasive lifting of latent fingerprint traces. Adapting surface scanners from different areas to the field of fingerprint acquisition is a common trend in recent years. For such kind of data to be accepted as evidence in a court hearing, specific measures to assure the privacy of the high-resolution person related data as well as the generation and preservation of the chain-of-custody need to be assured (meaning that all data needs to be protected and documented for the complete path from the generation until the final disposition of the data, see also [11]).

Creating a watermark-based communication protocol and successfully verifying its security aspects would enable forensic experts to also use watermark-based communication protocols for the processing of forensic evidence, without losing the acceptability of the evidence in a law suit. Using watermark-based communication protocols might have advantages over common cryptographic approaches in some cases. For example, since the watermark payload is tightly coupled with the cover object, watermarking allows for a forensic expert to add meta-data without creating additional files or attachments. This is especially important in the case when unauthorized people should not be aware of the presence or amount of meta-data (such as fingerprint verification results) within the image. Furthermore, the embedding of a watermark leads to an increase in the entropy of the image, but not in its overall size. Also, watermarks can often not be removed easily without destroying the cover, whereas a signature might be cropped from an object. In an advanced application scenario, also hierarchical access can be implemented using hierarchical watermarking, e.g. by assigning different areas of a data object to different hierarchy levels (‘segmented watermarking’, see [12]).

Our proposed watermarking scheme for forensic investigations is based on the media authentication scheme of Dittmann et al. ([13]), presenting a reversible watermarking scheme using cryptographic signatures and hashes. The scheme substitutes an area *B* of an image *O* in the spatial domain by a watermark composed of the compressed original pixel values *Comp(B)* and the payload *w*, leaving the area *A* of the image unchanged (see figure 2). The integrity and data-origin-authenticity of the scheme are mathematically verified in [13].

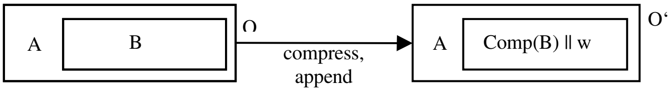


Fig. 2. The media authentication scheme of Dittmann et al., taken from [13]

We extend this scheme to assure the privacy of high-resolution biometric fingerprint images by substituting the fingerprint area with a compressed and encrypted watermark comprised of the original fingerprint values and a payload, which is only recoverable with the help of a symmetric key *sym_key*, only known to trusted personnel in a forensic investigation (see figure 3).

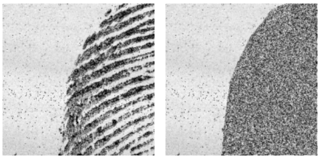


Fig. 3. A partial biometric fingerprint image before and after being watermarked with our proposed watermarking scheme for the privacy protection of the biometric data

We furthermore divide the watermark into a private and a public area, where the private area contains the original fingerprint pixels and additional confidential data, whereas the public area contains public meta-data and signatures.

Formally, our proposed scheme first divides an image O captured at a crime scene into an unchanged area A (background) and the fingerprint area B, creating a location map LM storing the location information. A hash value h is computed over the original picture O to ensure its integrity: $h = H(O)$. The original pixel values B and the private payload data D_{priv} are concatenated and compressed: $c = Comp(B||D_{priv})$. The result c is further concatenated with the computed hash h and encrypted using a symmetric cipher and a secret key *sym_key*: $e = Encrypt_{sym\_key}(c||h)$.

In the next step, the cryptographic signature s is computed over the unchanged background pixels A and the encrypted data e using the private key *asym_priv* of the signing entity: $s = S_{asym\_priv}(A||e)$. The final watermark w is generated by concatenating the encrypted data e, signature s and public data D_{pub}, as well as padding *padd* (up to the full capacity): $w = e \ || \ s \ || \ D_{pub} \ || \ padd$. The watermark w is then embedded by substituting the fingerprint area B specified by the location map LM, creating the watermarked image O': $O' = Substitute(LM,w)$. The location map LM is afterwards reversibly embedded into the watermarked image O' using the difference expansion technique from [14] for minimal visual distortion: $O'' = Emb(O',LM)$.

The watermark extraction process is done in an analogue way to the embedding process. In a fingerprint forensics application scenario, several entities might have to perform different operations on the watermark according to their respective tasks and known keys such as verifying signatures, reading or writing public and/or private data, signing the image or reconstructing the original cover (see figure 4). Whenever a signature or hash is extracted by the scheme, it is checked and in case of incorrectness the processing of the scheme is terminated with an error statement.

Action	**Operation**
Accessing/writing public data	D_{pub}
Accessing/Writing private data	$c=Comp(B \Vert D_{priv}), e=Encrypt_{sym\_key}(c \Vert h)$ $B \Vert D_{priv}=Decomp(c), c \Vert h=Decrpyt_{sym\_key}(e)$
Adding/verifying hash	$h=H(O), h^*=H(O^*)$
Adding/verifying signature/s	$s=S_{asym\_priv}(A \Vert e), s^*=S_{asym\_publ}(A^* \Vert e^*)$
Selection of new embedding area	$O \rightarrow A,B$
Reconstruction of original image	$O=A \Vert B$

Fig. 4. Possible actions offered to a user by our proposed watermarking scheme

In our exemplary forensic application scenario, a sensor *SEN* captures a high-resolution fingerprint trace (e.g. from a crime scene), watermarks it and forwards it to a forensic expert *FE1*. This expert might enhance the quality of the image for a better verification and adds additional meta-information. The image is then forwarded to a second forensic expert *FE2*, who might perform verification and stores the verification results in the payload of the watermark. The image is then sent to a court *COU*, where the image might be used as evidence. The procedure is depicted in figure 5.

4 Modeling and Verification of Our Proposed Watermark Communication Protocol Using CASPER and FDR

In this section, we want to model and verify our proposed watermarking scheme using the modeling language CASPER and the model-checker FDR, to evaluate the feasibility of such methodology for its usage in practical application scenarios. We therefore have to run through the six steps of the methodology, which are described in section 2. Due to the page limit of this paper, we cannot discuss each aspect of the modeling in detail, nor give an introduction into the CASPER syntax. We therefore refer to [2], [3] and [5] for further reading.

4.1 Network and Task Modeling, Path Search and Path Selection (Steps 1-3)

Within the first step of our methodology, we have to model the network, tasks, watermarking scheme and message layout of our communication protocol. According to the communication protocol description from section 3, we define four nodes for our forensic application scenario, namely a sensor *SEN*, two forensic experts *FE1* and *FE2* as well as a court *COU*. We also define a communication link between *SEN* and *FE1*, *FE1* and *FE2* as well as *FE2* and *COU* (see figure 5).

For our scenario, we assume that all nodes and links between the nodes fulfill the necessary requirements, such as sufficient capacity and knowledge of the used watermarking scheme described in section 3. Excerpts of the XML descriptions of the network, tasks, watermarking algorithm and the message layout (representing the structure of the watermark payload) are depicted figure 6.

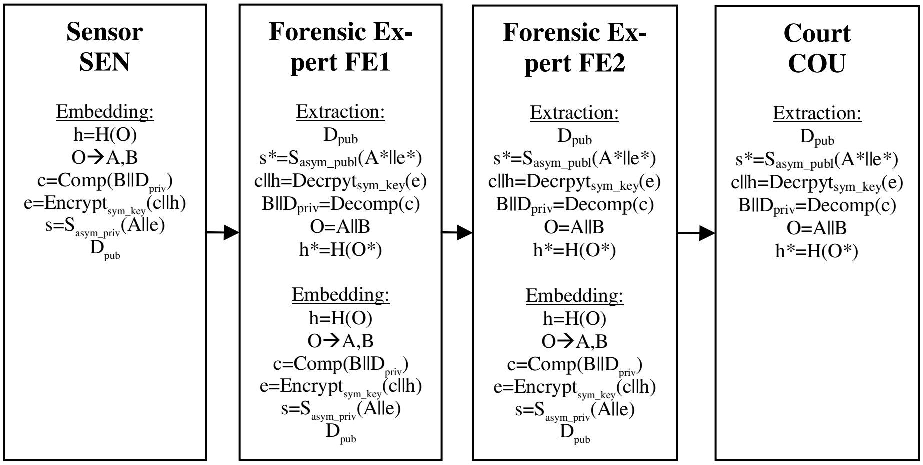


Fig. 5. The network nodes and communication links of our proposed watermark-based application scenario. For each node, the embedding and extraction operations are also depicted

Our modeled <*network*> consists of the <*nodes*> *SEN*, *FE1*, *FE2* and *COU*, each <*node*> having several characteristics, such as a <*cover*>, digital watermarking algorithms <*dwm*> known by the node, a capacity <*nodecap*> (exemplary set to 3000 in our example) and specifies certain requirements, such as a PKI <*pki*> or time-stamping <*ts*>. Each cover furthermore includes a cover channel <*cc*>, which has a <type> and a capacity <*chancap*> (exemplary set to 3000 in our example). Each network furthermore has one or more <*connections*>, linking a source node <*src*> to a destination node <*dst*>.

For the task definition, a global <*network-task*> exists, which is comprised of several tasks <*task*>, transporting data from a source node <*src*> to a destination node <*dst*>. For our example, each transfer of the data from one forensic entity to the next one is modeled as such a <*task*>. Each task furthermore comprises <*meta*> information, such as the used <*cover*>, used <*messages*> describing the structure of the watermark payload and additional requirements, such as the use of a public key infrastructure <*pki*>, time-stamping <*ts*>, a required capacity <*cap*> (exemplary set to 1500 in our example) and a hierarchy-level <*hierarchy*> (set to two for our example).

A specified watermarking algorithm <*dwm*> comprises an <*id*>, a <*cover*>, a required capacity <*dwmcap*> (exemplary set to 1500 in our example), a <*robustness*> level (exemplary set to ‘low’ in our example) and a <*hierarchy*> level (set to two for our example) as well as a general embedding key <*key*> (which is independent of the specific keys used for the hierarchical embedding).

The main structure of the watermark payload is described in the <*message*> tag, in our example the messages *msg0* and *msg1*, which represent the two hierarchy levels of our scheme (*msg0* represents the private area of the scheme whereas *msg1* represents the public area). The message *msg0* describes the first <*level*> of the watermark and contains a <*signaturechain*>, meaning that all content of the message has to be signed by each node processing or forwarding the message. Furthermore, the content of *msg0* is <*encrypted*> with the symmetric <*key*> *sym_key*, which is only known by people authorized for that hierarchy <*level*>. The main content of the message is a <*hash*>

over the original image (in our application scenario a <*hash*> over the unchanged area *A* of the cover image and a <*hash*> over the original embedding area *B*), the original pixel values of the embedding area *B* and the private payload D_{priv}. The unchanged area *A* of the cover image is also part of the message, but it is not encrypted, since it is an unchanged part of the original cover in which nothing is embedded.

```
<network>
 <nodes>
   <node>
     <id>SEN</id>
     <cover>
       <cc>
         <type>image</type>
         <chancap>3000</chancap>
       </cc>
     </cover>
     <dwm>Alg01</dwm>
     <nodecap>3000</nodecap>
     <pki>y</pki>
     <ts>n</ts>
   </node>
   <node>
     <id>FE1</id>
     <cover>
       <cc>
         <type>image</type>
         <chancap>3000</chancap>
       </cc>
     </cover>
     <dwm>Alg01</dwm>
     <nodecap>3000</nodecap>
     <pki>y</pki>
     <ts>n</ts>
   </node>
   ...
  </nodes>
  <lc>
   <connection>
     <src>SEN</src>
     <dst>FE1</dst>
   </connection>
   <connection>
     <src>FE1</src>
     <dst>FE2</dst>
   </connection>
   <connection>
     <src>FE2</src>
     <dst>COU</dst>
   </connection>
  </lc>
 </network>
```

```
 <dwms>
   <dwm>
     <id>Alg01</id>
     <cover>image</cover>
     <dwmcap>1500</dwmcap>
     <robustness>low</robustness>
     <hierarchy>2</hierarchy>
     <key>symmetric</key>
   </dwm>
 </dwms>
```

```
<network_task>
  <task>
    <id>Task1</id>
      <src>SEN</src>
      <dst>FE1</dst>
      <meta>
        <cover>
          <id>0</id>
          <type>image</type>
        </cover>
        <message>
          msg0
        </message>
        <message>
          msg1
        </message>
      </meta>
      <required>
        <pki>y</pki>
        <ts>n</ts>
        <cap>1500</cap>
        <hierarchy>
          2
        </hierarchy>
      </required>
  </task>
  ...
</network_task>
```

```
<message>
  <id>msg0</id>
  <level>1</level>
   <signaturechain>
     <content>
       <signed>
         <encrypted>
           <key>sym_key</key>
           <hashed>
             <hash0>A</hash0>
             <hash1>B</hash1>
           </hashed>
           <data0>B</data0>
           <data1>$D_{priv}$</data1>
         </encrypted>
         <data2>A</data2>
      </signed>
     </content>
   </signaturechain>
</message>
<message>
  <id>msg1</id>
  <level>2</level>
   <content>
     <data0>$D_{pub}$</data0>
   </content>
</message>
```

Fig. 6. Excerpts of the XML description of the network, tasks, watermarking algorithm and the message layout of our proposed watermark-based application scenario

The second message *msg1* represents the second hierarchy <*level*> of the scheme, which is the public area consisting of the public, unencrypted, unsigned and unhashed data $D_{pub.}$

Having successfully modeled our forensic application scenario (*network*, *task*, *watermarking algorithm* and *messages*), we can proceed to the second step of the methodology, which is the path search. Since there is only one path in our network, which is the route between the sensor *SEN* and the court *COU* passing by both forensic experts *FE1* and *FE2*, this task is straightforward. In step 3 of our methodology, the optimal path is to be selected using a cost function, which is also trivial since only one path is present.

4.2 CASPER Translation and Compilation (Steps 4-5)

In step 4 of our methodology, first the xml-representations are automatically translated into a description of the formalization language CASPER ([3], [5]) by means of an automated translation tool. However, it is to note here that the design of the message *msg0* cannot be automatically translated into CASPER-notation by the developed tool, since it is too complex for the current abilities of our translator. It therefore needs to be manually specified in the second part of step four of the methodology.

In this second part of the fourth step, apart from the manual adjustment of the message layout, the attacker scenario as well as the security aspects to be evaluated have to be manually specified. As we show in section 5, the CASPER/FDR-approach is currently only able to verify the security aspects of entity-authenticity and confidentiality, however, might be extended to also verify other aspects in future work.

Due to page limitations, we do not go into the details of the CASPER protocol adjustment here, we just want to mention that a generic attacker scenario according to Kerckhoffs' law is used (where a possible intruder has knowledge of all information which is not explicitly stated as being secret) and that the security aspects of entity-authenticity and confidentiality are verified (since these are the only two aspects which can be verified with the help of FDR at this point in time). The final CASPER-code is exemplary depicted for the first <*task*> in figure 7 (transferring the watermarked image from the sensor *SEN* to the first forensic expert *FE1*). For a detailed description of the CASPER-syntax please refer to [3] and [5].

The main functionality of the watermarking scheme can be seen in the *#Protocol description*, where the original fingerprint data *b_sen*, the private payload *dpriv_sen* and a hash over the original image, represented as *H(a_sen)* (hash over the unchanged background part *a_sen*) and *H(b_sen)* (hash over the fingerprint area *b_sen*) are encrypted with the shared symmetric key *sym_key* for the private area of the watermark. Such payload is then signed together with the unchanged background area *a_sen*, which is modeled by asymmetric encryption using the private key *asym_priv_sen* of the signing node *SEN* (in our case the sensor lifting the fingerprint data). The resulting data item together with the public payload *dpub_sen* represents the watermarked cover and is transmitted to the next node, which is the first forensic expert *FE1* in our example.

```
#Free Variables
sen,fe1 : Agent
a_sen : visiblepart
b_sen : fingerprint
dpub_sen : publicinfo
dpriv_sen : privateinfo
H : HashFunction
sym_key : SharedKey
asym_priv_sen, asym_priv_fe1, asym_priv_mal : PrivateKey
asym_publ_sen, asym_publ_fe1, asym_publ_mal : PublicKey
InverseKeys = (sym_key, sym_key),(asym_priv_sen, asym_publ_sen),
(asym_priv_fe1, asym_publ_fe1), (asym_priv_mal, asym_publ_mal)

#Processes
TASK1_SENDER(sen, a_sen, b_sen, dpub_sen, dpriv_sen, sym_key, asym_priv_sen,
asym_publ_sen)
TASK1_RECEIVER(fe1, sym_key, asym_priv_fe1, asym_publ_fe1, asym_publ_sen)

#Protocol Description
0. -> sen : fe1
1. sen -> fe1 : {a_sen,{b_sen, dpriv_sen, H(a_sen),H(b_sen)}{sym_key}}
{asym_priv_sen}, dpub_sen

#Specification
Secret (sen, b_sen, [fe1])
Secret (sen, dprivs_sen, [fe1])
Agreement(sen, fe1, [a_sen, b_sen, dpriv_sen])

#Actual variables
SEN, FE1, MAL : Agent
A_SEN: visiblepart
B_SEN: fingerprint
DPUB_SEN: publicinfo
DPRIV_SEN: privateinfo
SYM_KEY : SharedKey
InverseKeys = (SYM_KEY, SYM_KEY), (ASYM_PRIV_SEN, ASYM_PUBL_SEN),
(ASYM_PRIV_FE1, ASYM_PUBL_FE1), (ASYM_PRIV_MAL, ASYM_PUBL_MAL)
ASYM_PRIV_SEN, ASYM_PRIV_FE1, ASYM_PRIV_MAL : PrivateKey
ASYM_PUBL_SEN, ASYM_PUBL_FE1, ASYM_PUBL_MAL : PublicKey

#Functions

#System
TASK1_SENDER (SEN, A_SEN, B_SEN, DPUB_SEN, DPRIV_SEN, SYM_KEY, ASYM_PRIV_SEN,
ASYM_PUBL_SEN)
TASK1_RECEIVER (FE1, SYM_KEY, ASYM_PRIV_FE1, ASYM_PUBL_FE1, ASYM_PRIV_SEN)

#Intruder Information
Intruder = MAL
IntruderKnowledge = {SEN, FE1, A_SEN, DPUB_SEN, ASYM_PRIV_MAL, ASYM_PUBL_MAL,
ASYM_PUBL_SEN, ASYM_PUBL_FE1}
```

Fig. 7. CASPER syntax of the first <*task*> of our modeled application scenario (*SEN→FE1*)

For specifying the security aspects to be checked, we use the *#Specification* section of the CASPER protocol. Here, we can specify a *Secret* statement for the verification of the confidentiality: *Secret (sen, dprivs_sen, [fe1])*, meaning that the content of the object *dpriv_sen* is a secret only known to the nodes *sen* and *fe1*. The *Agreement* statement can furthermore be used to verify the entity-authenticity: *Agreement(sen, fe1, [a_sen, b_sen, dpriv_sen])*, specifying that after the protocol run, the nodes *sen* and *fe1* are correctly authenticated to each other using the objects *a_sen*, *b_sen* and *dpriv_sen* (see also [5]).

In the fifth step of our methodology, the CASPER-code is automatically compiled into CSP, which can then be verified using the model-checker FDR.

4.3 Model-Checking Using FDR (Step 6)

In the sixth step of our procedure, we verify our watermark-based communication protocol using the model-checker FDR. However, the protocol verification for the complete network (transfer of the data from *SEN* to *COU* over *FE1* and *FE2*) leads to a termination of the model-checker due to insufficient memory. Our test machine with 3GB of memory therefore is not able to cope with the state space of this rather middle-sized protocol. Since every node in our exemplary scenario is modifying and re-embedding the payload and maybe even changing the cover object, each communication step of the protocol can be seen as a transaction of independent data items, therefore allowing to simplify our protocol by only checking the first <*task*> (the transmission of the data from the sensor *SEN* to the first forensic expert *FE1*) with FDR, for which the computational requirements are sufficient. The security statement of FDR for this <*task*>, which is the final result of our methodology, is depicted in figure 8.

```
Checking assertion SECRET_M::SECRET_SPEC [T= SECRET_M::SYSTEM_S
No attack found

Checking assertion SECRET_M::SEQ_SECRET_SPEC [T=SECRET_M::SYSTEM_S_SEQ
No attack found

Checking assertion
AUTH1_M::AuthenticateTASK1_SENDERToTASK1_RECEIVERAgreement_a_sen_b_sen_dpriv_sen
[T=AUTH1_M::SYSTEM_1
Attack found:

Top level trace:
FE1  believes  (s)he  has  completed  a  run  of  the  protocol,  taking  role
TASK1_RECEIVER, with SEN, using data items A_SEN, B_SEN, DPRIV_SEN

System level:
Casper> 0. ->SEN : MAL
1.      SEN    ->    I_MAL    :    {A_SEN,    {B_SEN,    DPRIV_SEN,    H(A_SEN),
H(B_SEN)}{SYM_KEY}}{ASYM_PRIV_SEN}, DPUB_SEN
1.      I_SEN    ->    FE1    :    {A_SEN,    {B_SEN,    DPRIV_SEN,    H(A_SEN),
H(B_SEN)}{SYM_KEY}}{ASYM_PRIV_SEN}, DPUB_SEN
```

Fig. 8. FDR security statement checking the modeled protocol for the first <*task*> (*SEN→FE1*)

The results of the model-checking show that for the first two security specifications, which are representing the confidentiality requirements, no attack is found and therefore the confidentiality is verified. For the third security specification, which is representing the entity-authenticity, a classical man in the middle attack is found, with an intruder intercepting the data sent by the sensor *SEN* and forwarding it to the first forensic expert *FE1* at an arbitrary point in time. This information can be used to harden the protocol, e.g. by including a timestamp into the payload of the watermark or by applying additional authentication mechanisms.

Especially in watermark-based protocol contexts it might be of great interest to also verify the data-origin-authenticity and the integrity of the data, since watermarked objects are often transferred over the internet or other insecure networks with a huge number of non-trusted nodes, where the entity-authenticity might not be assurable. This is not possible with CASPER/FDR at this point in time, as shown in the next section. However, with some adjustment of the CASPER-notation, the verification of these security aspects might be possible.

5 Prospects and Limitations of the CASPER/FDR Based Semi-automated Modeling and Verification Approach

Concerning the *security aspects* which can be verified, the CASPER/FDR approach is so far limited to the verification of the confidentiality and the entity-authenticity of a protocol. Although the CASPER syntax provides a general framework for checking the integrity (such as hashes) as well as constructs for a possible verification of the data-origin authenticity (such as hashes and asymmetric encryption, which might be used for a digital signature scheme), the current realization of the CASPER language does not allow for a verification of such security aspects. An example is given here, showing that integrity cannot be verified using the CASPER/FDR approach. In an exemplary scenario, a node *X* sends an unencrypted message *m* to a node *Y*, which forwards it to a node *Z* (see figure 9). To model the integrity of such scenario, the nodes *X* and *Z* need to be given predefined knowledge of each other and an agreement between *X* and *Z* on the message *m* needs to be defined. In this case, trivially an attack on the integrity should be found by the model-checker FDR, representing a man-in-the-middle attack with an intruder claiming to be node *Y*, capturing the message *m* from node *X* and forwarding a changed message *m'* to the node *Z*. However, the results show that no attack is found, counter-proving such integrity-verification approach.

#Free Variables
`x,y,z : Agent`
`m : Message`
	#Processes
	`SENDER(x,z,m)`
	`FORWARDER(y)`
	`RECEIVER(z,x)`

#Protocol description
`0. -> x : y`
`1. -> y : z`
`2. x -> y : m`
`3. y -> z : m`
	#Specification
	`Agreement(x,z,[m])`

#Actual variables
`X,Y,Z,MAL : Agent`
`M : Message`
	#Functions

#System
`SENDER(X,Z,M)`
`FORWARDER(Y)`
`RECEIVER(Z,X)`
	#Intruder Information
	`Intruder = MAL`
	`IntruderKnowledge = {X,Y,Z}`

`Checking assertion AUTH1_M::AuthenticateSENDERToRECEIVERAgreement_m`
`T=AUTH1_M::SYSTEM_1`
`No attack found`	

Fig. 9. Casper syntax and FDR security statement of our counter-example, showing that integrity cannot be verified by CASPER and FDR at this point in time

The reason for this behavior seems to lie in the CASPER realization, which is designed to (apart from confidentiality) check if two nodes exchanging certain messages are correctly authenticated to each other (see [5]). Once this authentication is given, fulfillment of the agreement-statement (which is in our example used for the integrity verification) is declared, which is the case from the beginning of the protocol run on because the source and destination node know each other already by predefined knowledge. The agreement-statement can therefore in its current form not be used for the integrity verification. The data-origin-authenticity, being modeled as verifying the integrity of the authenticity information, is a subset of the integrity. As such, it can therefore also not be checked by CASPER, which is logically taking the fact into account that the data-origin-authenticity cannot be attacked by a possible man-in-the-middle if the two communicating nodes are already authenticated to each other. Therefore, while entity-authenticity and confidentiality can be obviously verified by the CASPER/FDR approach, integrity and data-origin-authenticity cannot due to the realization of the CASPER language. However, the general structure exists, which might allow the verification of these security aspects if the CASPER language would be extended.

Availability cannot be checked with the scheme at the moment. Given the syntactical framework of CASPER and FDR, it might be possible with an extended syntax to check if a certain message ultimately reaches its destination node, the availability of the nodes themselves, however, cannot be verified due to the strict sequential design of the scheme, which does not allow for dynamic changes of network nodes. The non-repudiation might be verifiable, if the number of possible sources of modification, which might be extractible in the scope of a possible integrity-verification, is limited to exactly one entity.

The syntactical limitations of the CASPER language represent significant limitations to the methodology. Here, the adoption of the approach from the domain of cryptography can clearly be seen. Many watermarking primitives cannot easily be modeled. While an invertible blind watermark might be modeled quite easily by simply encrypting the payload data and sending it (together with the cover) to the receiver who is decrypting it, non-blind or non-invertible watermarks are much harder to model and would require additional basic functionality in CASPER. Furthermore, CASPER offers no structures to store data at a network node for later use, such as a hash value or timestamp, which might have to be verified at a later stage of the protocol run. At the same time, it is not possible to forward data without interpretation for more than one step. CASPER allows for a node to store data in a variable without interpreting it, but the next node the data is sent to needs to interpret it, which is not desired in a scenario, where data is to be forwarded through many nodes without processing.

The *complexity limitations* pose another major problem to the methodology. Even the earlier proposed watermark-based communication protocol, which is rather small in comparison to other practical application scenarios (possibly being much more complex), cannot be verified with FDR using a RAM size of 3GB, causing the model-checker to exit with not having enough memory. The limitations of the state-space size of the model-checker are therefore another challenge. However, protocol simplifying transformations might be used in advance, as proposed in [15]. Furthermore, the strict sequential processing of the protocols required by FDR does not allow for the analysis of dynamic or fuzzy networks, as well as erroneous channels, which are very common in practical application scenarios.

Our methodology itself introduces some further limitations, which are referred to as methodology limitations in this paper. The methodology is limited by design to the evaluation of the security of a protocol, not of its underlying primitives, which have to be separately evaluated. A significant challenge for the presented framework is furthermore to define automated primitives and translations rules, which allow for a generalized protocol modeling and verification. While the definition of the network and tasks is comparatively easy, the definition of the watermarking scheme, especially the structure of the message transferred (see section 4.1), is very hard to be specified in a general manner, since these messages are often very algorithm-specific with different granularity or embedding strategies and are allowing for different modifications of the used embedding scheme and/or –area. Even if such a message can successfully be modeled, a general automated translation into the CASPER language seems to be very challenging, due to the complexity and different requirements of the different messages. For our introduced communication protocol, the defined message msg0 (see section 4.1) could not completely be translated by the translation-tool, leaving the core-functionality of the message to be adapted manually. Also, the security aspects to be verified as well as the definition of different attacker-scenarios (represented by different knowledge of a possible intruder) need to be specified manually, leading to the need of a manual adaption of the scheme, which requires knowledge of the CASPER-language.

The limitations summarized within this section can partially be overcome by extending the scheme. The *security aspects* of data-origin-authenticity as well as integrity verification and even non-repudiation might be included by adapting the CASPER-language. Also the *syntactical limitations* can be overcome by adding new, watermark-related functionality to CASPER. The *complexity limitations* are much more challenging to improve. However, they can be partially overcome by simplifying the protocol in advance (see [15]) or by using higher computational power. The *methodology limitations* so far lead to the need of an increased manual adjustment but do not limit the scheme itself. It could be further improved in future research and by systematically collecting common used primitives to cover at least the common watermarking schemes and protocol functions.

The practical prospects of our approach using semi-automated protocol modeling, generation and security verification can be summarized to its cost-effectiveness. In contrast to its alternative, the manual security verification of protocols, it is assumed to be faster and easier adaptable to specific application scenario requirements. However, the mentioned limitations need to be addressed first, for it to be feasible for the practical usage.

6 Conclusions and Future Work

In this paper, we evaluated the feasibility of our previously introduced semi-automated verification approach for watermark-based communication protocols, using XML, the modeling language CASPER and the model-checker FDR. For our evaluation, we proposed an exemplary application scenario where digital watermarking is applied to high-resolution forensic fingerprint data for the privacy- and chain of custody preservation in a forensic investigation. We showed that although the methodology offers great advantages, it has also severe limitations, preventing a practical application to many scenarios at the moment. However, we also showed that there are comparatively easy ways to improve the scheme to be suitable for a practical application, such as the enhancement of the CASPER-syntax.

In future work, the suggested improvements, especially the enhancement of the CASPER syntax and further improvement of the translation rules and modeling framework should be conducted, to make the methodology feasible for a practical application.

Acknowledgements. The fingerprint related work in this paper has been funded in part by the German Federal Ministry of Education and Science (BMBF) through the Research Programme under Contract No. FKZ: 13N10818.

The effort for watermarking algorithm aspects related work was in part sponsored by the Air Force Office of Scientific Research, Air Force Material Command, USAF, under grant number FA8655-09-M3061. The U.S. Government is authorized to reproduce and distribute reprints for Government purpose notwithstanding any copy-right notation thereon. The views and conclusions contained herein are those of the author and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Office of Scientific Research or the U.S. Government.

References

[1] Pimentel, J.C.L., Monroy, R.: Formal Support to Security Protocol Devel-opment: A Survey. Computación y Sistemas 12(1), 89–108 (2008) ISSN 1405-5546

[2] Kraetzer, C., Merkel, R., Altschaffel, R., Clausing, E., Schott, M., Dittmann, J.: Modelling Watermark Communication Protocols using the CASPER Modelling Language. In: Proc. ACM Multimedia and Security Workshop 2010 (MMSec 2010), Rome, Italy, pp. S.67–S.72. ACM Press, New York (2010) ISBN 978-1-450-30286-9

[3] Lowe, G.: CASPER: A Compiler for the Analysis of Security Protocols. Journal of Computer Security (1998)

[4] FDR user manual. Formal Systems (Europe) Ltd., http://www.fsel.com/documen-tation/fdr2/html/ (last accessed August 30, 2011)

[5] Lowe, G., Broadfoot, P., Dilloway, C., Hui, M.L.: CASPER - A Compiler for the Analysis of Security Protocols. User Manual and Tutorial, Version 1.12, Oxford University Computing Lab (2009)

[6] Roscoe, A.W.: Model-checking CSP. In: A Classical Mind: Essays in Honour of C. A. R. Hoare. Prentice Hall International (UK) Ltd. (1994)

[7] Viganò, L.: Automated Security Protocol Analysis with the AVISPA Tool. In: Proc. XXI Mathematical Foundations of Programming Semantics, MFPS 2005 (2005)

[8] Reactive Objects Language, http://ece.ut.ac.ir/FML/rebeca.htm (last accessed August 30, 2011)

[9] Song, D.X.: Athena: a new efficient automatic checker for security protocol analysis. In: Proceedings of the Twelfth IEEE Computer Security Foundations Workshop, pp. 192–202. IEEE Computer Society Press (1999)

[10] Fries Research Technology, http://www.frt-gmbh.com/en/ (last accessed August 30, 2011)

[11] Newman, R.: Computer forensics: evidence, collection, and management. Auerbach (2007)

[12] Sheppard, N.P., Naini, R.S., Ogunbona, P.: On multiple watermarking. In: Proc. ACM Multimedia and Security Workshop 2001 Ottawa, Ontario, Canada, October 5, pp. 3–6. ACM (2001)

[13] Dittmann, J., Katzenbeisser, S., Schallhart, C., Veith, H.: Provably secure authentication of digital media through invertible watermarks. Cryptology ePrint Archive, Report 2004/293 (2004), http://eprint.iacr.org/

[14] Coltuc, D., Chassery, J.M.: Very fast watermarking by reversible contrast mapping. IEEE Signal Process. Lett. 14(4), 255–258 (2007)

[15] Hui, M.L., Lowe, G.: Safe simplifying transformations for security protocols. In: 12th Computer Security Foundations Workshop Proceedings, pp. 32–43. IEEE Computer Society Press (1999)

Image Forensics of High Dynamic Range Imaging

Philip J. Bateman*, Anthony T. S. Ho, and Johann A. Briffa

Department of Computing, University of Surrey,
Guildford, Surrey, GU2 7XH, UK
{P.Bateman,A.Ho,J.Briffa}@surrey.ac.uk
http://www.surrey.ac.uk/computing/

Abstract. This paper introduces a novel area of research to the Image Forensic field; identifying High Dynamic Range (HDR) digital images. We create a test set of images that are a combination of HDR and standard images of similar scenes. We also propose a scheme to isolate fingerprints of the HDR-induced haloing artifact at "strong" edge positions, and present experimental results in extracting suitable features for a successful SVM-driven classification of edges from HDR and standard images. A majority vote of this output is then utilised to complete a highly accurate classification system.

Keywords: Image Forensics, High Dynamic Range Imaging.

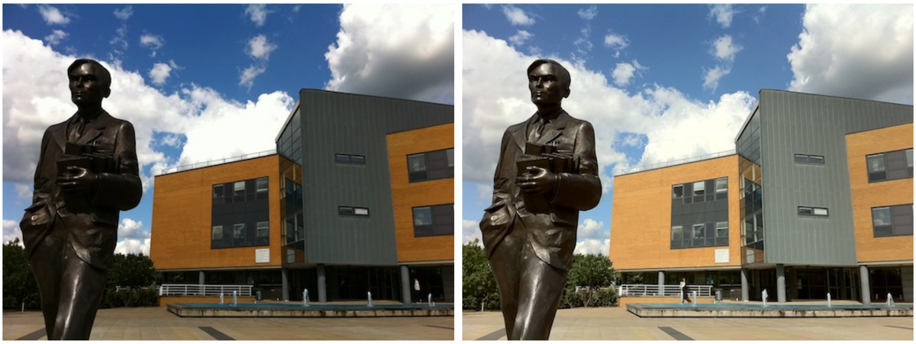
1 Introduction

In standard 24-bit imaging (one byte per channel), a single pixel can be represented by one of over 1.6 million colours. In real-world environments, however, the depth of colour is significantly larger, meaning that digital images often misrepresent them. Consequently, the images would likely contain under or over-exposed regions that degrade texture, detail, and colour. Throughout this paper, we refer to such images as *Low Dynamic Range (LDR)* images. In contrast to LDR images, in *High Dynamic Range (HDR)* imaging, a set of differently exposed images are combined and processed to create an image with a greater range of luminance between the light and dark areas of an image. This creates a more balanced version of the original image that matches the real-world environment more closely. Figure 1 illustrates one example of the difference between LDR and HDR versions of the same scene.

The quality of HDR processed images has led many manufacturers to use it for enhancing images automatically, directly after they are captured. Popular digital cameras with onboard HDR processing include camera phones such as the Apple iPhone 4 and a wide range of DSLR cameras manufactured by Canon, Sony, Nikon, Casio, Pentax, and many more. Furthermore, since HDR imaging

* The author's research is funded by an EPSRC CASE award in collaboration with Charteris Plc.

Y.Q. Shi, H.J. Kim, and F. Perez-Gonzalez (Eds.): IWDW 2011, LNCS 7128, pp. 336–348, 2012.



(a) A standard LDR image

(b) An HDR processed version

Fig. 1. LDR *vs.* HDR imaging. Note that in the LDR image (a) the dark areas of the statue are under-exposed and therefore lack the depth and detail that is present in the HDR image (b). Similarly, the texture of the clouds are mostly washed out in the LDR image due to over-exposure, where as the HDR version still possesses detail in these regions.

is purely software-driven, the set of camera phone devices increases when we consider downloadable apps available for Android users.

Previously, HDR imaging was considered an image editing tool that was entirely separate from the image acquisition pipeline. Most HDR creations therefore required the use of image editing software such as Adobe Photoshop and GIMP, as well as a bespoke collection of applications such as *Photomatix Pro*, *easyHDR*, and *DynamicPhoto HDR*. However, as an obvious consequence of the increasing availability to HDR imaging, more and more images that appear online are a result of HDR processing. Developing a strategy for the accurate forensic detection of digital images produced from the HDR imaging pipeline is therefore of great importance to ensure we understand the history of a digital image. In this paper, we direct our attention to images captured from HDR devices, with the aim of distinguishing between standard and HDR images taken from the same device. Specifically, this paper discusses and works from the creation of a test set of images captured from the Apple iPhone 4. This device is consistently reported by imaging-hosting website, *Flickr.com* as the most popular device amongst their community of over 60 million users. When we therefore consider the potential frequency of digital images currently in circulation that originate from this device, it is important to consider it for forensic experiment. Furthermore, the Apple iPhone 4 allows users to capture and process images as HDR natively, meaning it is highly probable that many images in circulation are of this type. We also explore anomalies associated with the HDR imaging pipeline, and combine the output of a trained Support Vector Machine (SVM) with majority voting to classify the images accurately.

Much of the image forensic research is currently geared towards camera identification. Moreover, the identification of anomalies in the image acquisition pipeline can be used to intrinsically link digital images to their source device.

Some of the more acclaimed approaches for achieving this include the analysis of sensor noise patterns [1], radial distortions caused by lens misalignments [2], the affect of the Colour Filter Array (CFA) on pixel colours [3], [4], [5], and the creation of a 34 featured SVM classifier extracted from a range of image properties [6]. Since some manufacturers' allow the direct processing of HDR imaging, it can be considered a possible extension to the image acquisition pipeline, providing a wider range of potential features to aid camera identification. The HDR imaging pipeline is complicated and varies with the implementation. It is therefore likely that fingerprints of individual manufacturer implementations of HDR imaging can directly aid camera identification. Currently, we believe that there is no literature related to Image Forensics for solving the HDR *vs.* standard problem. It seems logical to suggest that this area will grow in popularity based on the growing trend of applications created for producing HDR images.

This paper introduces the key steps in creating an HDR image, before defining the usage of Homomorphic Filtering for compressing the dynamic range to one that is compatible with output devices. A resulting anomaly from this process is then highlighted, before presenting our strategy for classifying HDR and LDR images. Our experimental results are followed by a concluding summary of our work and potential paths for future work.

2 High Dynamic Range Imaging Pipeline

The High Dynamic Range imaging pipeline is a complicated mixture of many processes, sub-processes, and bespoke strategies. Each implementation of HDR imaging can create output images that are significantly different from each other. For example, some implementations seek to produce artistic images that appear as though they have been heavily post-processed, whereas other implementations aim to produce more natural-looking images that closely match real-world scenes. Even when multiple separate implementations share the same desired output, there will still exist some subtle differences that can be useful for forensic analysis.

In Figure 2 we present a simplified model of the HDR pipeline that encompasses the main activities. The figure shows that the dynamic range of a real-world scene is much larger than that of a digital camera. Consequently, when a digital camera captures the scene, a lot of detail in the bright and dark regions is lost in over and under-exposed regions. In contrast, HDR imaging solves this problem by capturing a minimum of two differently exposed images that capture the detail in the highlights and shadows of the scene, as well as the main focal point. These images are then combined to produce a single HDR image with a dynamic range much closer to the real-world scene. However, since display devices and printers have a much smaller dynamic range, the image must be compressed to remove insignificant luminance data so that it can be viewed or printed. This process is known as *Tone Mapping*, and has been the focus of much HDR research in the past few years [7], [8], [9], [10]. The main challenge lies in the fact that there is no universal solution to the tone-mapping problem, since

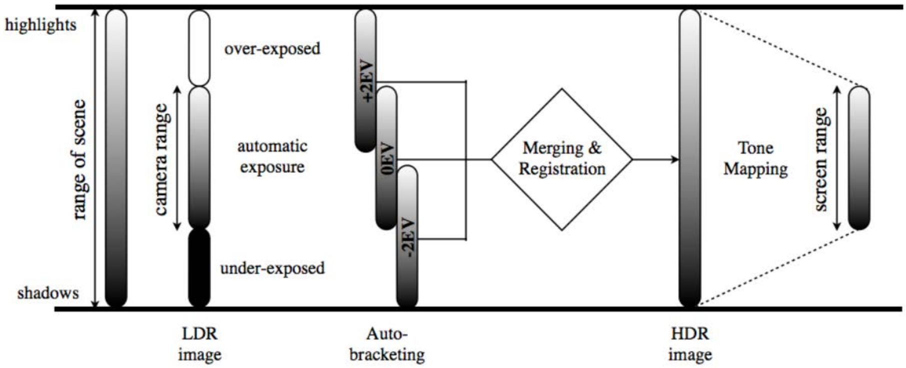


Fig. 2. The HDR imaging pipeline (adapted from [12])

it depends on the scene captured as to how the dynamic range can be reduced without damaging the natural feel of the image.

Oppenheim *et al.* were amongst the first authors to formally introduce this dilemma, and presented several ideas for tone mapping that are present in many of today's tone mapping operators [11]. Arguably the most dominant of these ideas was the use of *Homomorphic Filtering* for frequency-dependant compression of luminance components. This concept is discussed in the following section.

3 Homomorphic Filtering

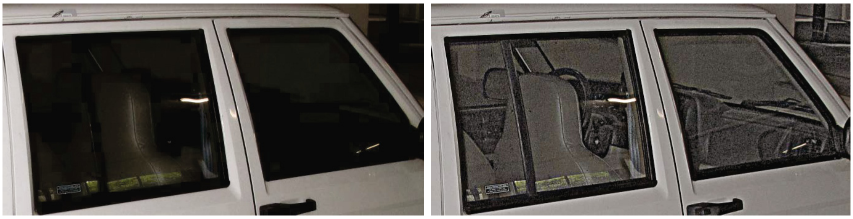
It is well documented in image processing literature that a given image is comprised of two respective brightness and contrast components: *illuminance* and *reflectance*, and that the human brain uses these components to complete the scenes that our eyes capture [13]. The illuminance component refers to the available light that radiates the scene, and reflectance refers to the light that is reflected off of the various objects that the scene contains. For example, a lake will reflect more light than a park bench (reflectance), but both could be lit by the sun (illuminance). A digital image can therefore be expressed in simplified luminance terms as the product of illuminance and reflectance components:

$$L_{(x,y)} = i_{(x,y)} \cdot r_{(x,y)}. \tag{1}$$

where $L_{(x,y)}$ refers to a digital image expressed in luminance terms, and $i_{(x,y)}$, $r_{(x,y)}$ refer to the illuminance and reflectance components respectively. In [11], the authors observe that the illuminance component contains significant redundancy, and can be compressed with a minimal impact to the detail and contrast of the image. Since luminance is a multiplicative product of i and r, the components must be separated such that dynamic range compression may be performed on i only. This is achieved by firstly computing the luminance data in terms of the logarithmic space such that:

$$\log(L_{(x,y)}) = \log(i_{(x,y)}) + \log(r_{(x,y)}). \tag{2}$$

Note how the luminance data is now expressed as an additive mix of i and r. The authors in [11] state that if the data are further expressed according to the Discrete Fourier Transform (DFT), then low frequencies can be associated with the illuminance component, and high frequencies with the reflectance component. The final task is to apply Homomorphic Filtering to attenuate the low frequencies whilst preserving the high frequencies in order to reduce the negligible illuminance data whilst simultaneously increasing the contrast of the image.



(a) Before Homomorphic Filtering (b) After Homomorphic Filtering

Fig. 3. The effect of Homomorphic Filtering [14]. In the original image image (a) the contrast range is so low that the interior of the vehicle appears dark and lacks detail. However, in the homomorphic filtered image (b) the contrast associated with the detailed regions is emphasised.

Dynamic range compression by Homomorphic Filtering is a popular methodology amongst recent HDR imaging implementations since it is very effective at producing dynamically compressed images with natural-looking end results [15]. However, depending on the contrast difference of two objects, and indeed the attenuation function used, it is common for the filtering process to produce a *halo* artifact, where objects appear to glow against the background. This artifact is most apparent in edge pixels, since they are a combination of both high and low frequencies, and only the low frequencies are attenuated. The degree to which this artifact is apparent depends on the specific implementation, but in forensic terms this makes it a useful source for fingerprinting. It is this homomorphic filtering artifact that this paper uses for classifying whether an image is HDR or LDR.

4 Evaluation Methodology

In this paper, we hypothesise that digital images will exhibit signs of the halo artifact if they are composed from the HDR imaging pipeline. Similarly, the artifact will not exist to the same magnitude for standard images produced by conventional photography. The identification of haloed regions should therefore help to distinguish HDR and LDR images accurately.

4.1 Test Strategy

In our initial work, we establish a library of 100 ‘landscape’ images that are an equal combination of HDR and LDR images taken with the same device (Apple iPhone 4 running iOS 4.3.3). The images are captured with the native camera application, and the device is mounted to a tripod to ensure that the alignment between the HDR and LDR versions of the same scene is consistent. For each image, ‘strong’ vertical edge points – such as where bright light from the sky merges with a dark object – are extracted and data from these regions are tested to isolate the halo artifact. As discussed in Section 3, the halo artifact is most evident at these such edge regions, as they are comprised of both high and low frequency data.

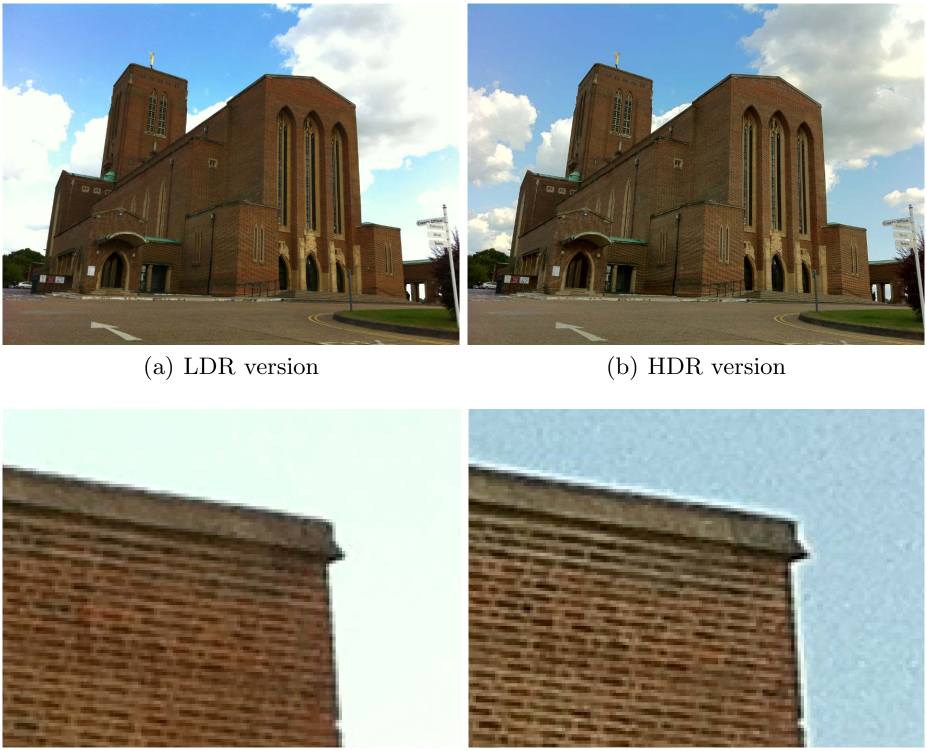
Following on from a system we have designed to isolate strong edges, we propose the extraction of a frame of pixel data (obtained from the left and right of these edge points) for 100 edges from each image. We then use this data in conjunction with the LibSVM classifier to predict the likelihood that a given edge is an HDR edge. Based on the results of this classification, we then employ majority voting to establish the generalised accuracy for detecting an HDR image.

Figure 4 illustrates a typical example of the degree of haloing associated with LDR and HDR images captured from the Apple iPhone 4. The HDR image shows clear edge anomalies around the strong edge points compared to the equivalent LDR image. What we see is a single bright pixel that separates the boundary of the object with the bright sky background. This is a strong indication that homomorphic filtering has taken place at these two contrasting points, and subsequently a useful fingerprint for detecting HDR imaging.

4.2 Edge Selection

The successful extraction of strong edges is key to identifying the halo artifact that forms the basis of our classification process. In essence, we need to establish some automated scheme for extracting the most suitable edges to be considered for evaluation. Figure 5 presents a block diagram of our initial schema for ‘Strong Edge’ extraction.

Canny Edge Detection. We start by extracting the luminance component of the image, and applying Canny edge detection to find as many edges as possible. Canny edge detection is the preferred choice as our preliminary testing showed that this operator extracted a higher frequency of the most relevant edges. Figure 6 illustrates how the Sobel, Roberts, and Prewitt operators misses important edges when the brightness of an object closely matches that of the background. In particular, note how the Canny edge operator provides more consistent edge information for the “Refectory” and“Shop” signposts, with respect to other operators. However, since the output of the canny edge detection over-detects many edges from detailed regions, the information must be reduced in order to locate a single edge of interest. To achieve this, the image itself must first undergo several transformations, as discussed in the next section.



(a) LDR version

(b) HDR version

(c) LDR ‘strong’ edges (magnified)

(d) HDR ‘strong’ edges (magnified)

Fig. 4. HDR and LDR images captured from Apple iPhone 4, and their respective ‘strong’ edges. Note that whilst the edges of the LDR image (c) do not show visible signs of haloing, the haloing is abundantly obvious for the corresponding edges in the HDR image (d).

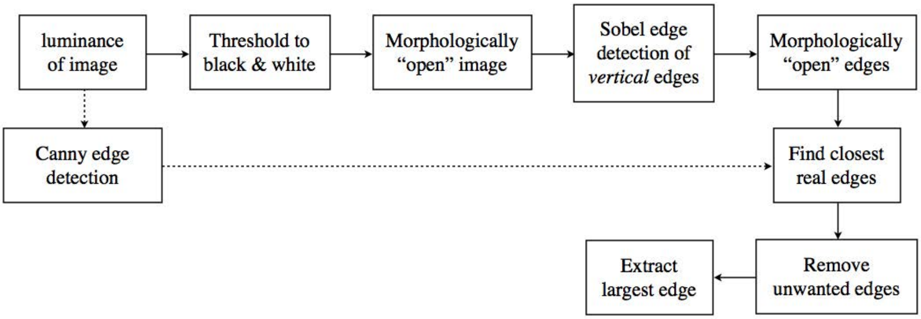
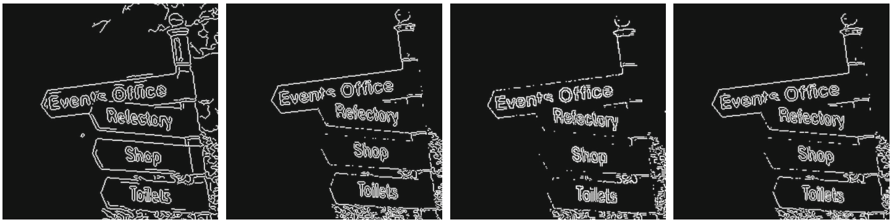

luminance of image
Threshold to black & white
Morphologically “open” image
Sobel edge detection of *vertical* edges
Morphologically “open” edges
Canny edge detection
Find closest real edges
Remove unwanted edges
Extract largest edge

Fig. 5. The ‘Strong Edge’ extraction process



(a) Canny (b) Sobel (c) Roberts (d) Prewitt

Fig. 6. Edge detection using different operators, each with perceptually optimum threshold values

Threshold to Binary Form. In order to reduce the amount of edges that are detected, the detailed regions of the image must be reduced such that all that remains is the edge associated with objects that neighbour significantly contrasting backgrounds. In this model, we firstly threshold the luminance data to produce a binary, black and white pixel representation. The "open" morphological operator is then used to merge the detailed regions in the black and white regions.

Edge Reduction. Since this initial work focuses only on vertical edges, we can apply Sobel edge detection on the opened image to extract only the edges that are likely to be of interest. Now that the detail has been removed from the image, we obtain a much smaller number of edges than we obtained from the Canny edge detection phase. However, due to the fact that morphological opening is a combination of erosion and dilation, the Sobel edges we extract may be slightly misaligned with the original image. We therefore consider the Sobel edges an estimate of interesting edges, and obtain the closest neighbouring edges from the Canny edge image. The remaining connected edges are then iteratively processed to determine whether or not they satisfy certain conditions. Assuming for this example that the object is expressed as pixel value 0, and the background is expressed as 1, then the conditions are:

- mode of 5 pixels left of edge = 0.
- mode of 5 pixels right of edge = 1.
- $|$ angle of connected edge $| \leq 30\,^\circ$ of vertical axis.

The first rule checks that the object is on the left, while the second rule also checks that the background is on the right of the connected edges. The third rule checks that the edge is close to vertical. The haloing artifact exists as a normal vector to the edge. To simplify the scheme, we consider the haloing artifact that is a normal vector of the vertical axis such that a horizontal extraction of pixel data would encapsulate the full artifact. If the connected edge were at $45\,^\circ$, for example, then we would need to consider and evaluate diagonal pixels. When a connected edge does not satisfy all of the conditions, they will be dropped, and only the edges that are likely to yield the most useful results remain. Figure

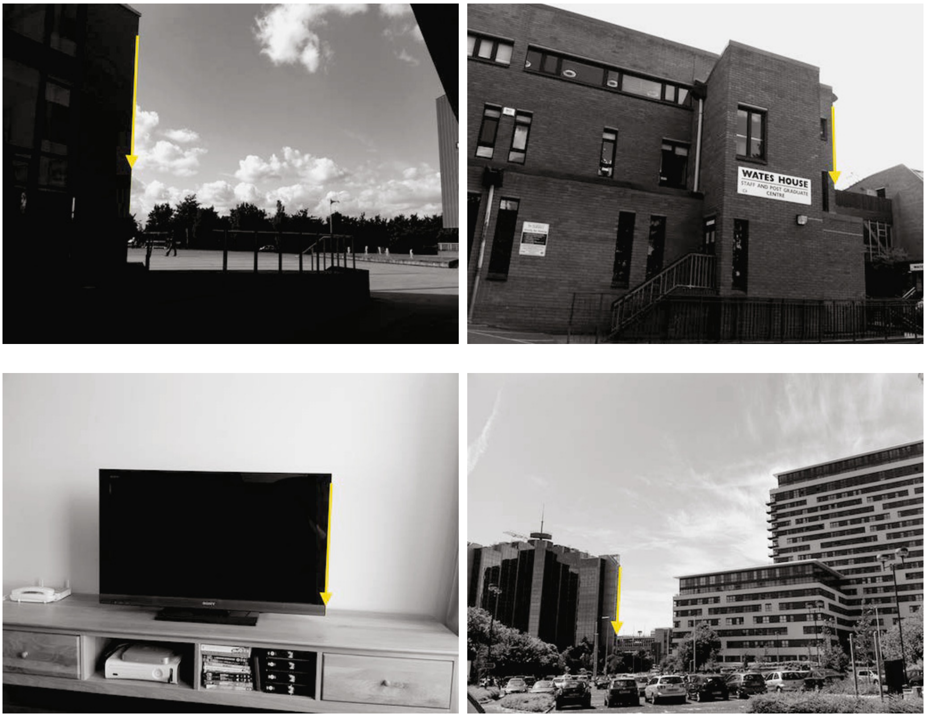


Fig. 7. Examples of edges extracted from the Edge Selection process

7 illustrates several examples of the selected edges that will be used to isolate signs of the haloing artifact.

5 Experimental Results

5.1 Pixel Distribution

Using the edge selection scheme discussed in Section 4.2, we obtain the selected edges for each individual image. We then extract a horizontal frame of pixel data centred on each individual edge position, that describes the distribution of pixel data for the object (left) and background (right) that neighbour the edge. Figure 8 illustrates the typical pixel distribution extracted from LDR and HDR images, where a frame radius of 128 pixels is selected.

The vertical line at zero represents the location of the edge pixel. Of particular interest is the distribution of the pixel data that immediately follows the edge. For the LDR image, there is a small peak where the pixels are seemingly slightly brighter than the remainder of the data. In comparison, this peak is much larger for HDR images where it is noted that the pixels immediately right of the edge are more contrasting to the remainder of the data. This is the result of the HDR version containing more colour and texture data than its LDR counterpart, and a clear graphical representation of what was noted from Figure 4.

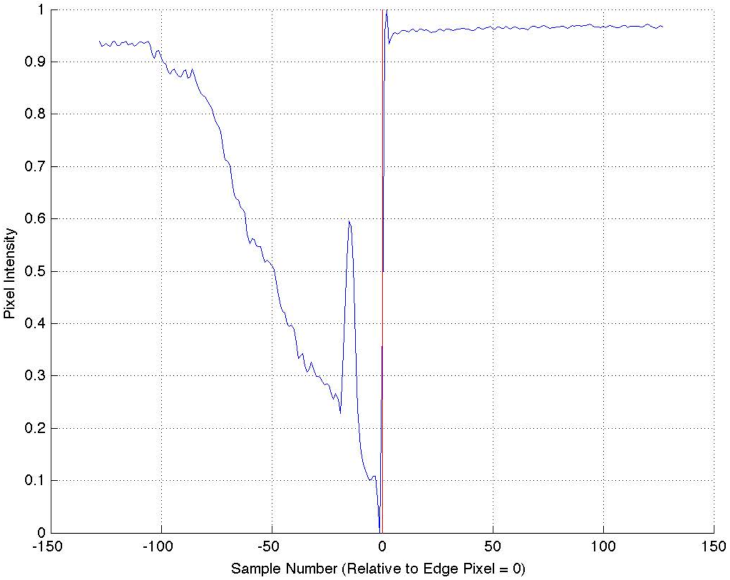
5.2 Classification of HDR *vs.* LDR Edges

The plots illustrated in Figure 8 are actually a fair representation of the plots obtained for each of the images. In each case, the LDR version shows a smaller peak when compared to the equivalent HDR image. This provides further confidence that extracting the correct features from this data could help to distinguish between both image types. Of course, the magnitude to which this peak extends is proportionate to the image pixels. If the background is relatively white (for example, in a cloudy scene) then the peak will be smaller than if the sky is a deep blue. However, since it is unlikely that the distribution of pixel data in white clouds will exactly equal the intensity of the halo, a peak always exists. For this reason, it is arguably inappropriate to base feature extraction in the spatial domain. However, accepting that the peak following the edge pixel exists as a common characteristic between the images, it is likely that a DFT transformation will emphasise this trait. We therefore convert each extracted frame of pixels to the DFT domain. By randomly selecting 100 edges from the edge selection process, we sample the pixel data and obtain the magnitude of DFT for each. We then reduce the DFT output to half the original length by removing the symmetric data. Using simply this output, we create a training data set for LibSVM classification. The training set is comprised of 100 edges from 90% of the images. More precisely, the training set will be a 9000x128 data matrix, since 90 of the 100 images are used for training, and a 1x128 DFT vector is computed for 100 edges for each image. The remaining 10 images are processed in exactly the same way and are used as test data for the LibSVM classifier.

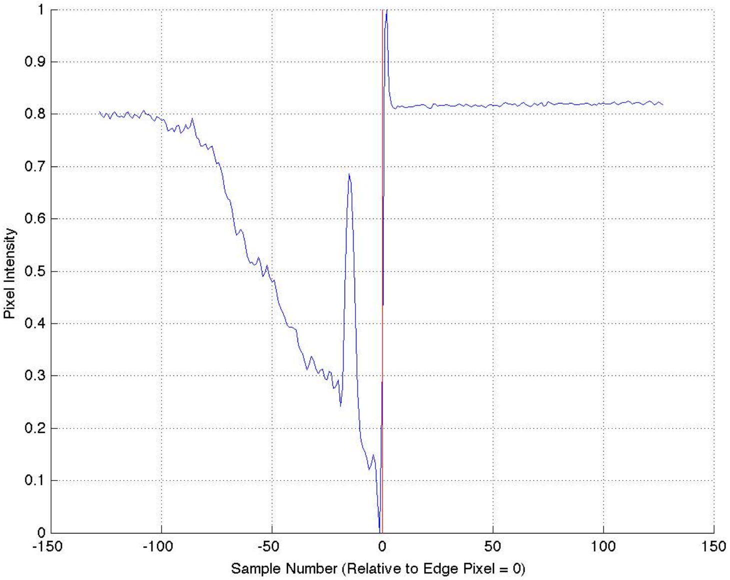
5.3 Results

The LibSVM classifier was able to classify each individual edge with an overall accuracy of 85.1%. Considering that we use 100 edges from every image, this preliminary result provides further proof that the proposed use of the halo artifact is consistent for both image types. However, this classification alone does not demonstrate the success as to whether an entire image is classified as either LDR or HDR. To achieve this, the output of the data classification must be aggregated to groups of 100, where each group represents all the edges for the same image. Using a majority voting strategy, the individual images can be classified with respective levels of confidence. Table 1 shows the ‘actual’ class type (LDR or HDR) for the 10 images tested. The majority voting process correctly predicts the class type in each instance leading to a generalised classification accuracy of 100%. For each image, the percentage of correctly classified edges is computed to provide a confidence level to which each classification is made.

Since the majority voting process implies an accuracy of 100%, we have proved that our scheme can accurately use a single property of HDR imaging to distinguish between LDR and HDR images. The individual levels of confidence from each of these classifications is as high as 100%, but also decreased to 55%. After a brief evaluation of the image this data corresponds to, it is clear that the image is highly textured, and this negatively affects our scheme for extracting strong



(a) Pixel distribution from HDR Image



(b) Pixel distribution from LDR Image

Fig. 8. The pixel distribution of LDR (a) and HDR (b) edges. The data has been normalised to intensity values for clearer interpretation.

Table 1. Final classification of LDR *vs.* HDR images, and the respective confidence levels

Test Image	Actual	Predicted	Accuracy (%)
1	HDR	HDR ✔	88
2	HDR	HDR ✔	99
3	HDR	HDR ✔	80
4	HDR	HDR ✔	69
5	HDR	HDR ✔	55
6	LDR	LDR ✔	87
7	LDR	LDR ✔	92
8	LDR	LDR ✔	100
9	LDR	LDR ✔	91
10	LDR	LDR ✔	90

edges. In this example, a weaker edge was selected, and the data corresponding to the halo was not captured fully. The requirement of developing a more precise scheme is therefore of interest to future work.

To check for manufacturer-specific consistencies in the halo artifact, a small number of random HDR and LDR images have been collected from 2 other Apple iPhone 4 devices. For each of the images tested, the halo artifact exists to the same degree, and similar classification accuracies were obtained. Furthermore, the latest iOS software beta 5.0 was installed to a 4th Apple iPhone 4 device to ensure that the halo artifact has not been addressed by a software update distributed by the manufacturer. Again, the halo artifact was still notably present in HDR images.

It is expected that each manufacturer-driven implementation of the HDR imaging pipeline will inherit anomalies that can be traced to the device in much the same way that camera identification research functions. Furthermore, HDR-induced anomalies such as the haloing artifact are likely to be handled differently depending on the implementation. Since there are many applications available for mobile devices to produce HDR images, it is therefore feasible that the specific application used to create them can be traced.

6 Conclusion

In this paper, we have introduced a novel area for image forensic research, and proposed a forensic scheme for identifying the halo artifact induced by HDR imaging, in images collected from the Apple iPhone 4. The scheme is capable of extracting the most suitable edges for analysis, converting them to an appropriate feature representation, and classifying the image successfully. We have presented a proof of concept by presenting our initial experiments and methodology for correctly classifying LDR and HDR images with a 100% success rate on a small test set. Of interest to future work is the obvious requirement of improving the confidence of individual classifications; most logically by refining the edge selection scheme. Our current strategy for identifying the halo artifact functions only on vertical edges, but can be expanded for horizontal edges in order to strengthen the likelihood that the extracted edge data captures haloed regions. Beyond this, a system for processing any edge could be engineered such that the frame data is extracted as a normal vector to the edge direction. These modifications would enable the scheme to be more compatible with a wider range of scenes.

Acknowledgements. The authors would like to extend thanks and gratitude to Charteris Plc. for supporting this research.

References

1. Lukáš, J., Fridrich, J., Goljan, M.: Digital Camera Identification From Sensor Pattern Noise. IEEE Transactions on Information Security and Forensics 1(2), 205–214 (2006)
2. Choi, K.S., Lam, E.Y., Wong, K.K.Y.: Source Camera Identification Using Footprints From Lens Aberration. In: Proceedings of the SPIE, vol. 6069, pp. 172–179 (2006)
3. Bayram, S., Sencar, H.T., Memon, N., Avcibas, I.: Source Camera Identification Based on CFA Interpolation. In: Proceedings of IEEE ICIP, vol. 3, pp. 69–72 (2005)
4. Celiktutan, O., Avcibas, I., Sankur, B., Memon, N.: Source Cell-phone Identification. In: IEEE Signal Processing and Communications Applications, pp. 1–3 (2005)
5. Long, Y., Huang, Y.: Image Based Source Camera Identification using Demosaicking. In: IEEE 8th Workshop on Multimedia Signal Processing, pp. 419–424 (2006)
6. Kharrazi, M., Sencar, H.T., Memon, N.: Blind Source Camera Identification. In: International Conference on Image Processing, vol. 1, pp. 709–712 (2004)
7. Fattal, R., Lischinski, D., Werman, M.: Gradient domain high dynamic range compression. In: Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques, vol. 21, pp. 249–256 (2002)
8. Mantiuk, R., Myszkowski, K., Seidel, H.P.: A Perceptual Framework for Contrast Processing of High Dynamic Range Images. ACM Transactions on Applied Perception 3, 286–308 (2006)
9. Krawczyk, G., Myszkowski, K., Seidel, H.P.: Computational Model of Lightness Perception in High Dynamic Range Imaging. In: Human Vision and Electronic Imaging XI, IS&T/SPIE's 18th Annual Symposium on Electronic Imaging (2006)
10. Qiu, G., Guan, J., Duan, J., Chen, M.: Tone Mapping for HDR Image using Optimization A New Closed Form Solution. In: 18th International Conference on Pattern Recognition, pp. 996–999 (2006)
11. Oppenheim, A.V., Schafer, R., Stockham, T.: Nonlinear Filtering of Multiplied and Convolved Signals. Proceedings of the IEEE 56(8), 1264–1291 (1968)
12. Debevec, P., Yu, Y., Borshukov, G.D.: Efficient View-Dependent Image-Based Rendering with Projective Texture-Mapping. In: Eurographics Rendering Workshop, pp. 105–116 (1998)
13. Gonzalez, R.C., Woods, R.E.: Digital Image Processing, 3rd edn. Pearson Prentice Hall (2007) ISBN: 978-0-13-168728-8
14. Russ, J.C.: Forensic Uses of Digital Imaging. CRC Press (2001) ISBN: 978-0-84-930903-8
15. Reinhard, E., Ward, G., Pattanaik, S., Debevec, P.: High Dynamic Range Imaging: Acquisition, Display, and Image-Based Lighting. Morgan Kauffman (2005) ISBN: 978-0-12-585263-0

Improved Run Length Based Detection of Digital Image Splicing

Zhongwei He[1], Wei Lu[2], and Wei Sun[1]

[1] School of Software, Sun Yat-sen University, Guangzhou 510006, China
zhuge_2003@hotmail.com, sunwei@mail.sysu.edu.cn
[2] School of Information Science and Technology, Guangdong Key Laboratory of Information Security Technology, Sun Yat-sen University, Guangzhou 510006, China
luwei3@mail.sysu.edu.cn

Abstract. Image splicing is very common and fundamental in image tampering, which severely threatens the integrity and authenticity of images. As a result, there is a great need for the detection of image splicing. In this paper, an improved run length based scheme is proposed to detect this specific artifact. Firstly, the edge gradient matrix of an image is computed. Secondly, approximate run length is defined and calculated along the edge gradient direction. Thirdly, features are constructed from the related histograms of the approximate run length. Finally, support vector machine (SVM) is exploited to classify the authentic and spliced images using the constructed features. The experiment results demonstrate that the proposed approach can achieve a moderate accuracy with far less computational cost and much fewer features when compared with a similar method.

Keywords: Image splicing detection, Digital image forensics, Approximate run length, Edge detection, Characteristic function.

1 Introduction

Nowadays, digital images have been widely used in our daily life. However, with the development of computer software technology and the rise of Internet, it's so easy to tamper an image and distribute it that the saying "seeing is believing" is no longer always true. As a result, digital image forensics that aims at authenticating digital images and detecting possible image forgeries becomes a hot research area.

There are lots of different kinds of image forgeries, such as image splicing, copy-move, blurring, re-sampling. Among all these tampering operations, image splicing is the most common and fundamental one that creates composite image by cutting and joining two or more photographs. Spliced images could be deceivably authentic to human eyes even without any post-processing such as matting and blending, see Fig. 1 for some examples [11]. What's more, spliced images could be used for malicious purposes, because they could lead to people's misunderstanding to the contents, or convince people to something that never exists. Therefore, the detection of image splicing is of great importance.

Y.Q. Shi, H.J. Kim, and F. Perez-Gonzalez (Eds.): IWDW 2011, LNCS 7128, pp. 349–360, 2012.

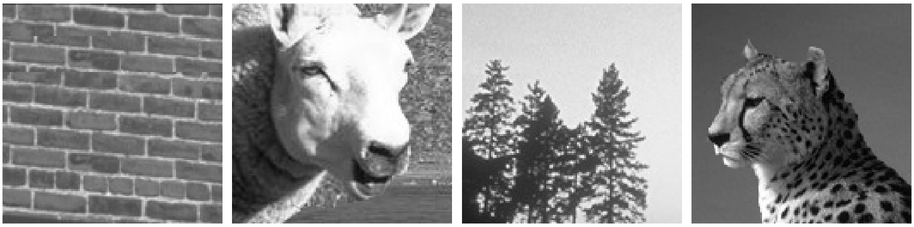


Fig. 1. Examples of spliced images

Generally speaking, there are two different kinds of techniques in digital image forensics, referred to as active [9] and passive (blind) [8], respectively. Though the former one could achieve a better performance and robustness in image tampering detection, it needs some information to be inserted into digital images before the images are distributed, which restricts its application area. The latter one, on the contrary, can do the detection in the absence of such information, and consequently attracts more attentions.

The logic behind the passive (blind) detection is that, though visual clues are erasable, image tampering would inevitably alter the underlying statistical characteristic of an image. Based on this idea, lots of researches focusing on different kinds of image forgeries have been proposed. In the field of image splicing, some blind detection approaches have been developed in the past. In a series of papers [10,12,13], Ng et al. proposed to use a higher order moment spectra, bicoherence, as features to identify spliced images. It is claimed that bicoherence is sensitive to quadratic phase coupling (QPC) caused by splicing discontinuity. The bicoherence based approach was tested on a well designed image data set [11], and a detection accuracy was reported as high as 72%. In [4], Shi et al. utilized Hilbert-Huang Transform (HHT) to capture the high non-linear and non-stationary nature introduced by image splicing. Besides, a statistical natural image model based on moments of characteristic function using wavelet decomposition was proposed in this work. By combining features extracted from these two methods, a detection accuracy of 80.15% was reported. In [6], camera response function (CRF) was exploited by Hsu et al. for image splicing detection in a semi-automatic manner. And this work was extended to a fully automatic one by incorporating automatic image segmentation in [7]. 2D phase congruency as well as statistical moments of wavelet characteristic function were proposed by Chen et al. as features to detect spliced images in [2]. They achieved a detection accuracy of 82.32% on Columbia Image Splicing Detection Evaluation Dataset [11] using SVM as the classifier. In [14], Shi et al. proposed a natural image model consisting of two kinds of statistical features. The model could achieve a detection accuracy of 91.87%.

The approaches mentioned above are very promising in terms of detection accuracy. However, detection accuracy is not all, computational cost and feature dimensionality should be taken into consideration too. Since the detection accuracy achieved is more than 90%, the next step would be searching for some approaches with less computational complexity, while keeping the detection accuracy as high as possible. For this purpose, inspired by the run length concept mentioned in [3], an image splicing detection algorithm based on approximate run length is proposed in this paper.

The rest of this paper is organized as follows. In section 2, the approximate run length based approach is described in detail. The experiment results of our proposed approach as well as the comparison with the scheme proposed in [3] are shown in section 3. Finally, the conclusions are drawn in section 4.

2 The Proposed Approach

In this section, the approximate run length extended from the original one is presented firstly. Then, the proposed approach based on the approximate run length is described in detail.

2.1 The Original Method Based on Run Length

Inspired by the conclusion in [15] that approaches used in steganalysis can also be used in splicing detection if appropriately applied, Jing Dong et al. proposed to use Run-Length based statistic moments for image splicing detection in [3] (denoted as the original method based on run length here). A run is defined as a string of consecutive pixels having the same gray-scale value along a specific linear orientation θ (e.g. 0°, 45°, 90° and 135°). The length of the run is the number of repeating pixels in the run. The image run length histogram is then defined as a vector:

$$H_\theta(j) = \sum_{i=1}^{M} p_\theta(i,j), \quad 1 < j < N \tag{1}$$

where $p_\theta(i,j)$ is a run length matrix defined as the number of runs with pixels of gray-scale value i and run length j. M is the number of gray levels while N is the maximum run length. Using Eq. (2), Jing Dong et al. extracted the first three moments of characteristic function from image run length histograms in four directions as features for image splicing detection, and a detection accuracy of 69.75% on Columbia Image Splicing Detection Evaluation Dataset was reported in [3].

$$M_n = \frac{\sum\limits_{j=1}^{L/2} f_j^n |F(f_j)|}{\sum\limits_{j=1}^{L/2} |F(f_j)|} \tag{2}$$

Here, n is the order of moment, $F(\cdot)$ represents the Discrete Fourier Transform (DFT), and L is the total number of points in the horizontal axis of the histogram. More details about Eq. (2) and the reason for using moments of characteristic function instead of moments of histograms can be found in [14,16].

2.2 The Improved Method Based on Approximate Run Length

The original run length based method mentioned above is simple and fast, however, its detection accuracy is not high enough. Here, we propose an improved method based on approximate run length, which can achieve higher detection accuracy with fewer features. The approximate run length is developed by making some modifications to the original one. The modifications and the reasons making these modifications are discussed as follows:

Firstly, instead of computing all the run lengths of an image, we only compute run lengths on the edge pixels. The global discontinuity and incoherency on image structure and pixel correlation introduced by splicing are too limited to be detected by computing all the run lengths of an image, because most of the run lengths (the ones within the same spliced region in a spliced image) are totally not affected by the splicing operation. Computing run lengths on the edge pixels, on the contrary, is better for the reason that splicing normally introduces extra edges to the image. Extra edges introduced by splicing are sharper than the ones existing in the authentic images (non-spliced images) in general, and hence, can be taken advantage of distinguishing spliced images from natural ones.

Secondly, instead of computing run lengths along four specific linear orientations (i.e. 0°, 45°, 90° and 135°), we compute run length along the respective gradient direction of each edge pixel only. Computing run lengths along four specific linear orientations is not necessary, see Fig. 2 for a simple example of a spliced image. In this simple case, run lengths along the orientation 90° alone is enough to reveal the splicing forgery, and run lengths along the linear orientation 0° is completely not affected by the splicing, while run lengths along the orientation 45° and 135° bring no more helpful information than those along the orientation 90°. In fact, for this specific example, the orientation 90° is the gradient direction of all the edge pixels. Gradient direction of an edge pixel is the linear orientation spanning the edge, which could possibly be an edge between two spliced regions in a spliced image. As mentioned above, extra edges introduced by image splicing are different from the original ones, so they do help in authenticating an image. As the shape and direction of an edge can be varied in an image, we compute run lengths along the respective gradient direction on every single edge pixel rather than along a fixed orientation.

Thirdly, instead of computing the strict run length which requires all the pixels in a "run" have exactly the same gray-scale value, we compute "run length" in a

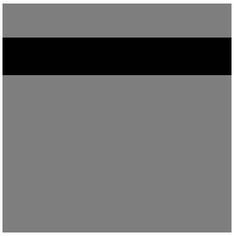


Fig. 2. A simple example of a spliced image

different sense. In our method, we introduce a threshold t, if the absolute value of the difference of two neighboring pixels' gray-scale value is not greater than the threshold t, we consider the two pixels are in an approximate run. When computing an approximate run from a specific edge pixel, the gray-scale value of the edge pixel is taken as a pivot, and then all the pixels in the approximate run derived from this edge pixel are supposed to abide by the rule mentioned above, as shown in Eq. (3).

$$|g(\boldsymbol{x_i}) - g(\boldsymbol{x_0})| \leq t \tag{3}$$

where $\boldsymbol{x}_0$ is the edge pixel coordinate, $\boldsymbol{x}_i$ is any pixel in the approximate run $\boldsymbol{\alpha}_0 = (\boldsymbol{x}_{-m}, \cdots, \boldsymbol{x_0}, \cdots, \boldsymbol{x_n})$ derived from $\boldsymbol{x}_0$ along its gradient direction, and $g(\cdot)$ represents the gray-scale value of the given pixel. When $i = -(m+1)$ or $i = n+1$, $\boldsymbol{x}_i$ doesn't satisfy Eq. (3). Therefore, the approximate run length on the edge pixel $\boldsymbol{x}_0$ is the length of $\boldsymbol{\alpha}_0$, i.e., $m+n+1$. The reason we are doing so is based on the fact that, even in an authentic (non-spliced) image, the gray-scale value of consecutive pixels fluctuate to some extent. The only difference between an authentic image and a spliced image is that, the fluctuation in a spliced image is more dramatic. Therefore, a reasonably and carefully chosen threshold t can be utilized to distinguish spliced images from natural ones.

Finally, since we have introduced a threshold t to the approximate run length, gray-scale values of pixels in a run are not always the same. Here, we introduce a new concept "Fluctuation Degree" to reflect the change of gray-scale values of pixels in an approximate run. Fluctuation Degree can be calculated using Eq. (4).

$$FD = \frac{\sum\limits_{i=-(m+1)}^{n+1} |g(\boldsymbol{x}_i) - g(\boldsymbol{x}_0)|}{L+2} \tag{4}$$

where $\boldsymbol{x}_0$ is an edge pixel, and its approximate run along gradient direction is $\boldsymbol{\alpha}_0 = (\boldsymbol{x}_{-m}, \cdots, \boldsymbol{x}_0, \cdots, \boldsymbol{x}_n)$, $L = m+n+1$ is the approximate run length on $\boldsymbol{x}_0$, and FD is the Fluctuation Degree of $\boldsymbol{\alpha}_0$. One may note that, the computation of FD involves $\boldsymbol{x}_{-(m+1)}$ and $\boldsymbol{x}_{n+1}$ which are not pixels in the approximate run but pixels directly adjoin to it, so FD can not only reflect the change of gray-scale values of pixels in an approximate run, but also reflect the change of gray-scale values of pixels at both ends of the run.

After making the modifications mentioned above, we develop an improved method based on approximate run length, which is shown in Fig. 3. And it detailed as follows.

Firstly, given a source image, we compute its edges and gradient direction 2-D matrix Θ using Sobel operator.

Secondly, the obtained gradient direction 2-D matrix Θ is quantified. The value scope of the above gradient direction θ is $[-90°, 90°]$. However, there are only four kinds of direction relationship between two consecutive pixels, which are $-90°$ and its opposite (vertical direction, denoted as θ_v), $-45°$ and its opposite (main diagonal direction, denoted as θ_d), $0°$ and its opposite (horizontal direction, denoted as

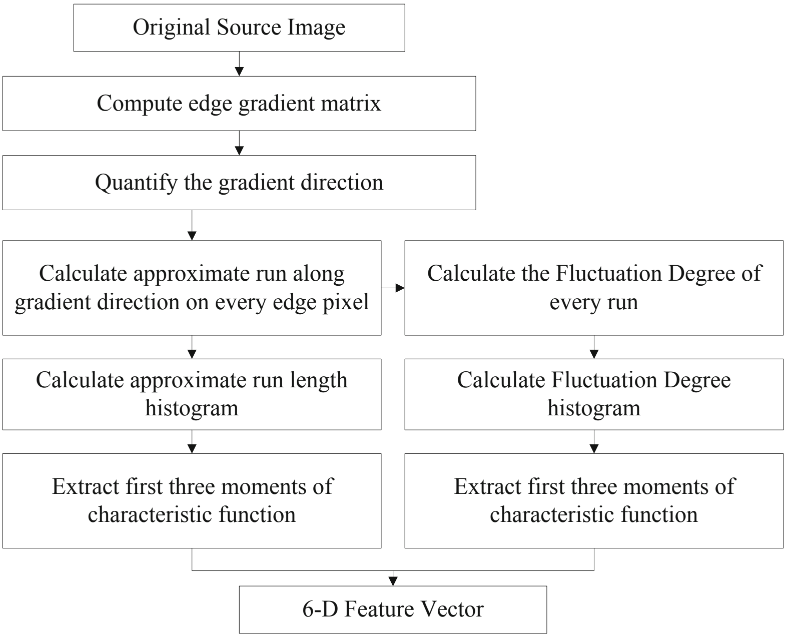

Original Source Image
Compute edge gradient matrix
Quantify the gradient direction
Calculate approximate run along gradient direction on every edge pixel
Calculate the Fluctuation Degree of every run
Calculate approximate run length histogram
Calculate Fluctuation Degree histogram
Extract first three moments of characteristic function
Extract first three moments of characteristic function
6-D Feature Vector

Fig. 3. The improved method based on approximate run length

θ_h), 45° and its opposite (minor diagonal direction, denoted as θ_m), respectively. Thus, we quantify the gradient direction θ using Eq. (5).

$$\theta_Q = \begin{cases} \theta_v, & -90^\circ \leq \theta < -67.5^\circ \text{ or } 67.5^\circ < \theta \leq 90^\circ \\ \theta_d, & -67.5^\circ \leq \theta < -22.5^\circ \\ \theta_h, & -22.5^\circ \leq \theta < 22.5^\circ \\ \theta_m, & 22.5^\circ \leq \theta \leq 67.5^\circ \end{cases} \tag{5}$$

Thirdly, we compute the approximate run length along gradient direction on every edge pixel, and extract the first three moments of characteristic function from the corresponding histogram of the approximate run length using Eq. (2).

Fourthly, we calculate the Fluctuation Degree of each "approximate run" obtained in the last step using Eq. (4), and extract the first three moments of characteristic function from the corresponding Fluctuation Degree histogram using Eq. (2).

Finally, we concatenate the six moments of characteristic function and obtain a 6-D feature vector for image splicing detection.

3 Experiments and Results

In this section, we first introduce the experiment conditions, and then present a set of experiments to demonstrate the high performance and effectiveness of the proposed approach. Finally, analysis of the images misjudged by the proposed approach is given.

3.1 Experiment Conditions

The public available and well recognized image dataset for splicing detection is the Columbia Image Splicing Detection Evaluation Dataset provided by DVMM, Columbia University. It consists of 933 authentic and 912 spliced images with size of 128 × 128 pixels. The dataset is carefully designed for benchmarking the blind passive image splicing detection algorithms, so all the experiments presented in this paper are conducted on this specific dataset. More details about the image dataset can be found in [11].

Support vector machine is utilized as the classifier in our experiments. Specifically, we choose the LIBSVM [1], and a RBF kernel is used. The optimal values for the parameter c and g are set through a "grid-search" method [5]. In the experiments, all the authentic images are labeled as $+1$ (positive), while all the spliced images are labeled as -1 (negative). Then the classification of authentic and spliced images can be viewed as a binary decision problem, and the classifier LIBSVM is trained to solve it.

To evaluate the performance of the proposed approach, all the experiments and comparisons are tested on the above mentioned dataset and the same classifier. The software platform is Matlab R2009b, and the hardware platform is a PC with a 2G duo core processor. In each experiment, the average rate of 50 repeating independent tests is recorded. In each of the 50 runs, we randomly select 5/6 of the authentic images and 5/6 of the spliced images from the dataset to train the SVM classifier. Then the remaining 1/6 of the authentic and spliced images are used to test the trained classifier.

3.2 The Comparison between the Original and the Approximate Run Length Based Method

A set of comparison experiments is conducted between the improved approximate run length based method and the original run length based method proposed by Jing Dong et al. in [3]. The detailed results are given in Tab. 1. Here, TP rate is the ratio of correct classification of authentic images. TN rate is the ratio of correct classification of spliced images. Accuracy is the weighted average value of TP rate and TN rate. The threshold t is set as a percentage, which means that the threshold t is a percentage of the respective dynamic scope of image gray-scale (i.e. the absolute value of the difference between the maximum and the minimum gray-scale value in an image).

As shown in Tab. 1, when the threshold t is set as 2%, the approximate run length based method can achieve a detection accuracy as high as 76.80%, and the corresponding ROC curve is shown in Fig. 4.

Compared with the original run length based method, the approximate run length based method can achieve 7% higher detection accuracy (76.80% vs. 69.75%) with half the feature dimensionality. It is obvious that the approximate

Table 1. Results on the comparison between the original and the approximate run length based method

Method	Dimensionality	Threshold t	TP(%)	TN(%)	Accuracy(%)
Original run length	12	0	69.74	65.81	69.75
Approximate run length	6	1	81.88	63.51	72.79
Approximate run length	6	2	78.12	71.49	74.83
Approximate run length	6	3	80.23	69.47	74.91
Approximate run length	6	4	84.67	65.80	75.33
Approximate run length	6	5	83.72	66.64	75.26
Approximate run length	6	6	78.68	69.76	74.27
Approximate run length	6	1%	77.69	72.82	75.28
Approximate run length	6	2%	86.45	67.46	76.80
Approximate run length	6	3%	78.67	71.05	74.90
Approximate run length	6	4%	77.97	68.41	73.24

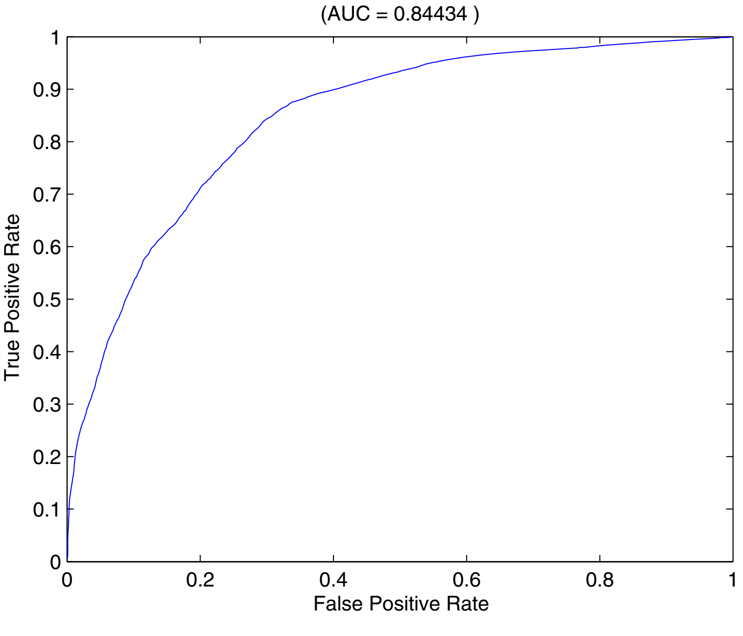


Fig. 4. The ROC curve of the approximate run length based method (t=2%)

run length based method is more effective in classifying authentic and spliced images than the original.

A percentage (e.g. 2%) of the respective dynamic scope of image gray-scale as threshold t is better than a fixed value for taking into account the diversity of image contents. It is more reasonable for every single image. Specifically, a threshold of 2% can tolerate the small inherent fluctuation of gray-scale value of consecutive pixels in an authentic image, and not conceal the large fluctuation or step change of gray-scale value introduced by image splicing.

Since the 6-D feature vector extracted using the approximate run length based method comprised of two different kinds of feature components, i.e. the first three moments of characteristic function calculated from the approximate run length histogram (denoted as $M1$) and the newly introduced first three moments of characteristic function calculated from the corresponding Fluctuation Degree histogram (denoted as $M2$), some experiments are further conducted by using $M1$ only to see whether the introduction of $M2$ is necessary. The results are given in Tab. 2 in detail.

Table 2. Results on the detection accuracy of the feature vector $M1$

Feature Vector	Dimensionality	Threshold t	TP(%)	TN(%)	Accuracy(%)
$M1$	3	1	64.71	66.99	65.84
$M1$	3	2	82.18	54.55	68.50
$M1$	3	3	73.86	66.67	70.30
$M1$	3	4	66.12	75.25	70.64
$M1$	3	5	67.99	69.38	68.68
$M1$	3	6	69.95	67.00	68.49
$M1$	3	1%	91.95	50.75	71.55
$M1$	3	2%	77.69	71.66	74.70
$M1$	3	3%	77.79	64.68	71.30
$M1$	3	4%	74.77	63.68	69.28

From Tab. 1 and Tab. 2 we can see that, in terms of Accuracy, the original 6-D feature vector ($M1 + M2$) is better than $M1$ no matter what threshold is chosen. The highest detection accuracy achieved by $M1 + M2$ is 2% higher than the one achieved by $M1$ (76.80% vs. 74.70%). Therefore, the introduction of $M2$ is worthwhile.

3.3 Further Experiments on Comparison

The feature vector proposed in [3] comprised of two different feature sets, of which one is based on original run length (denoted as RL) and the other one is based on edge statistic (denoted as SP). To evaluate the 6-D feature vector proposed in our approach (denoted as ARL, i.e. $ARL = M1+M2$) comprehensively, we make a comparison between our approach and the overall scheme proposed in [3], not only in terms of detection accuracy, but also the computational costs and feature dimensionality. The results of the experiments are shown in Tab. 3, and the corresponding ROC curves are shown in Fig. 5.

As shown in Tab. 3 and Fig. 5, the feature set ARL proposed by us can achieve a comparable detection accuracy with far less computational cost and much fewer features than $RL + SP$ proposed in [3], which makes our proposed scheme more suitable for real applications and large-scale analysis. What's more, the feature set SP is a supplement to the run length (either original run length or approximate run length) based feature sets. The feature set $ARL + SP$ can reach a detection accuracy as high as 80.02%, and its computational time is still quite manageable.

Table 3. Results on the comparison between our approach and the scheme proposed in [3]

Feature set	Dimensionality	Accuracy(%)	Extraction time(s)
SP	49	74.27	0.2005
RL	12	69.75	0.0125
$RL + SP$	61	76.52	0.2130
ARL	6	76.80	0.0341
$ARL + SP$	55	80.02	0.2346

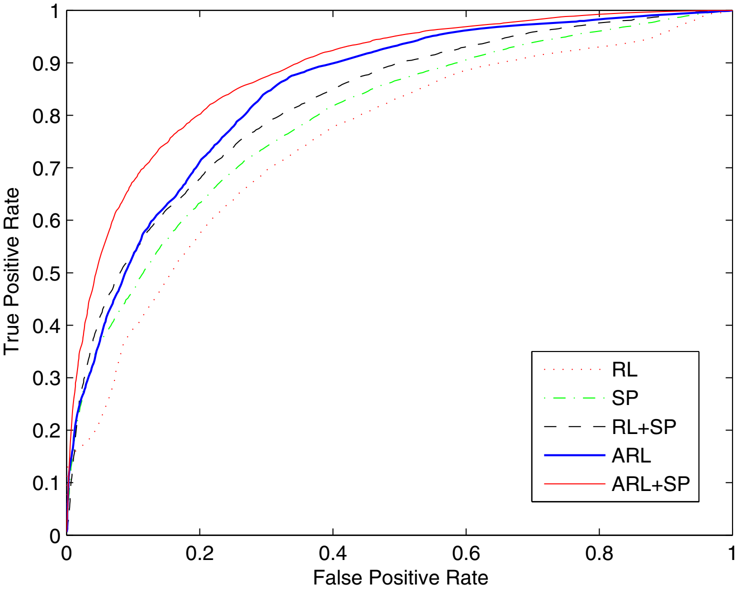


Fig. 5. The ROC curves of several different feature sets

3.4 Analysis of the Misjudged Images

Although the proposed approach is simple and efficient, its detection performance is still somewhat less than satisfactory. Some examples of the misjudged images list in Fig. 6, of which the first row are authentic images and the second row are spliced images.

After careful scrutiny, we find that images with complex texture are more likely to be misclassified. The reason is two-fold. One is, complicated texture will greatly threaten the accuracy of the edge detection of an image, which is vitally important to the proposed approach. The other one is, the fluctuation of gray-scale values of consecutive pixels will tend to be more dramatic in an image with complex texture, and consequently make the authentic images and the spliced ones less distinguishable. Though the introduction of threshold t in the proposed approach has lessened the influence of complex texture to some

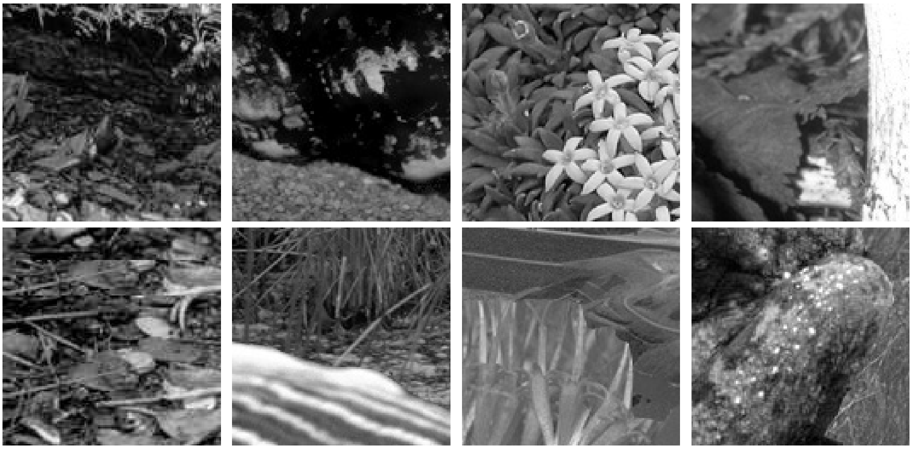


Fig. 6. Some examples of the misjudged images

extent, it's still far from enough. Further research work on this aspect should be done in the future.

4 Conclusions

In this paper, a novel approach based on approximate run length is proposed for image splicing detection. The approximate run length is first defined, which is developed by making some modifications to the original one. Then, an approximate run length based scheme is proposed to classify authentic and spliced images. Compared with a similar method, the proposed approach can achieve a moderate detection accuracy with far less computational complexity and much fewer features. Thus, we believe our approach is more suitable to be used in the areas of real applications and large-scale analysis.

Acknowledgements. This work is supported by National Natural Science Foundation of China (NSFC No. 60803136), the Fundamental Research Funds for the Central Universities, and Guangzhou Science and Technology Program (No. 2009J1-C541-2).

References

1. Chang, C.C., Lin, C.J.: LIBSVM – a library for support vector machines, `http://www.csie.ntu.edu.tw/~cjlin/libsvm`
2. Chen, W., Shi, Y.Q., Su, W.: Image splicing detection using 2-d phase congruency and statistical moments of characteristic function. In: Imaging: Security, Steganography, and Watermarking of Multimedia Contents, p. 65050R (2007)

3. Dong, J., Wang, W., Tan, T., Shi, Y.Q.: Run-Length and Edge Statistics Based Approach for Image Splicing Detection. In: Kim, H.J., Katzenbeisser, S., Ho, A.T.S. (eds.) IWDW 2008. LNCS, vol. 5450, pp. 76–87. Springer, Heidelberg (2009)
4. 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.) IWDW 2006. LNCS, vol. 4283, pp. 177–187. Springer, Heidelberg (2006)
5. Hsu, C.W., Chang, C.C., Lin, C.J.: A Practical Guide to Support Vector Classification. Department of Computer Science, National Taiwan University (April 2010)
6. Hsu, Y.F., Chang, S.F.: Detecting image splicing using geometry invariants and camera characteristics consistency. In: IEEE ICME 2006, pp. 549–552 (2006)
7. Hsu, Y.F., Chang, S.F.: Image splicing detection using camera response function consistency and automatic segmentation. In: IEEE ICME 2007, pp. 28–31 (2007)
8. Mahdian, B., Saic, S.: A bibliography on blind methods for identifying image forgery. Signal Processing: Image Communication 25, 389–399 (2010)
9. Potdar, V.M., Han, S., Chang, E.: A survey of digital image watermarking techniques. In: 3rd IEEE International Conference on Industrial Informatics (INDIN), pp. 709–716 (2005)
10. Ng, T.T., Chang, S.F.: Blind detection of digital photomontage using higher order statistics. Tech. Rep. 201-2004-1, Columbia University (2004)
11. Ng, T.T., Chang, S.F.: A data set of authentic and spliced image blocks. Tech. Rep. 203-2004-3, Columbia University (2004)
12. Ng, T.T., Chang, S.F.: A model for image splicing. In: IEEE International Conference on Image Processing, pp. 1169–1172 (2004)
13. Ng, T.T., Chang, S.F., Sun, Q.: Blind detection of photomontage using higher order statistics. In: IEEE ISCAS, pp. 688–691 (2004)
14. Shi, Y.Q., Chen, C., Chen, W.: A natural image model approach to splicing detection. In: MM&Sec 2007, pp. 51–62. ACM, Dallas (2007)
15. Shi, Y.Q., Chen, C., Xuan, G., Su, W.: Steganalysis Versus Splicing Detection. In: Shi, Y.Q., Kim, H.-J., Katzenbeisser, S. (eds.) IWDW 2007. LNCS, vol. 5041, pp. 158–172. Springer, Heidelberg (2008)
16. 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, pp. 269–272 (2005)

Median Filtering Detection Using Edge Based Prediction Matrix

Chenglong Chen and Jiangqun Ni[⋆]

School of Information Science and Technology, Sun Yat-Sen University
Guangzhou 510006, P.R. China
c.chenglong@gmail.com, issjqni@mail.sysu.edu.cn

Abstract. In digital image forensics, there is an increasing need for the development of techniques to identify general content-preserving operations, such as resampling, compression, contrast enhancement and median filtering (MF). As a contribution towards this goal, we present, in this paper, a new blind forensic scheme for MF detection in images. The proposed method is based on the observation that, compared with original and linear filtered images, median filtered images exhibit distinct intrinsic traces around edges, e.g. neighborhood correlation, noise suppression and good edge preservation. Such MF intrinsic fingerprints are characterized as the Edge Based Prediction Matrix (EBPM), which contains the estimated prediction coefficients of neighborhood prediction among different edge regions in images. By incorporating the support vector machine (SVM), the MF detector is developed based on EBPM. Extensive simulations are carried out, which demonstrates the superior performance of the proposed scheme in terms of effectiveness and robustness.

Keywords: Median Filtering, Digital Image Forensics, Neighbor Prediction.

1 Introduction

In recent years, digital images have been widely used in news media, law enforcement, and military applications. With such high popularity and widespread availability of digital image editing software, we can no longer take the originality and authenticity of digital images for granted. Traditional methods for image authentication including digital signature and digital watermarking belong to active image authentication techniques. However, some pre-processing of these approaches such as signature generation and watermark embedding must have been done before the distribution of images, which would limit these approaches to specially equipped digital cameras in practical applications [1]. As a result, there is an increasing need for developing techniques to assess the authenticity of images without relying on such pre-processing.

[⋆] Corresponding author.

Y.Q. Shi, H.J. Kim, and F. Perez-Gonzalez (Eds.): IWDW 2011, LNCS 7128, pp. 361–375, 2012.

Recently blind forensic methods, which work in the absence of any watermark or signature, have become the domain of extensive research. In contrast to the active techniques, these methods operate under the premise that the only information available is the image itself with undetermined authenticity [1]. Although digital forgeries may leave no visual clues that indicate modification, they may indeed alter the underlying statistics of an image [1]. By identifying the traceable statistical artifacts left behind by manipulations imposed on images, blind forensic methods can assess the authenticity of digital images and identify image alterations without access to the source images or source device.

Image manipulations can generally be classified into malicious tampering and content-preserving manipulations. Correspondingly, the works in image forensics fall into two main categories. In the first category, numerous forensic methods focus on the detection of malicious tampering of the image content, such as copy & move and image splicing. In the second one, methods have been proposed to detect content-preserving manipulations, such as resampling [1] [2], JPEG compression [3], contrast enhancement [4] and median filtering [5] [6]. Although the content-preserving manipulations do not change the visual appearance of an image, their blind detection is still of forensic interest. First, these manipulations are usually employed as post-processing operations, which destroy the actual state of an image after previous tampering operations, thus may affect the set of forensic tools we are using and decrease the reliability of forensic algorithms. Second, certain post-processing operations may interfere with or diminish visual traces of previous tampering operations, such as the resampling operator [7]. Hence evidence obtained from image forensic analysis of such content-preserving manipulations would provide useful information to assess the authenticity of digital images.

In this paper, we focus on the detection of median filtering (MF). As most of existing forensic methods, such as resampling and CFA interpolation detection schemes [1] [2], rely on some kind of linearity assumption, blind detection of non-linear MF becomes especially challenging. Due to its non-linearity, MF can serve as an anti-forensics technique to destroy the linear constraints with image forgery operations and affect the reliability of existing forensic methods. An example is the new resampling scheme reported in [7], which applies MF to hide interpolation traces and is undetectable by Popescu and Farids resampling detector [2]. Therefore, forensic detection of MF can provide useful forensic information to identify the possible resampling operation.

Blind MF detection has already been studied by Kirchner [5] and Cao [6]. In [5], streaking artifacts, which are measured using distribution of first-order differences, are employed to detect MF in bitmap images. While for MF detection in JPEG post-compressed images, instead of streaking artifacts, the subtractive pixel adjacency matrix (SPAM) features are incorporated to analyze the conditional joint distribution of first-order differences. In [6], the probability of zero values on the first-order difference image in textured regions is adopted as MF statistical fingerprint to detect MF. The works in [5] [6] depend on, to some extent, the measurements of streaking artifacts by means of first-order differences as fingerprints for MF detection. Since the distributions of first-order differences are vulnerable to image manipulations such as quantization and rounding, the streaking artifacts features used in [5] [6] cannot survive the JPEG compression except the high dimensional SPAM features. This motivates the development of robust features which well characterize the MF manipulation for MF detection. It has been observed that, compared with original and linear filtered images, median filtered images exhibit distinct intrinsic traces around edges, e.g. neighborhood correlation, noise suppression and good edge preservation. Based on this observation, we present a simple yet effective statistical model to characterize the MF operation. The image is first divided into blocks and then classified into different types of edge blocks based on the gradient features. For each edge block type, we fit the neighborhood prediction model to the corresponding edge blocks across the image and extract the prediction coefficients as MF intrinsic fingerprints, known as the Edge Based Prediction Matrix (EBPM). By incorporating the support vector machine (SVM), the MF detector is developed based on EBPM. Extensive simulation results have demonstrated the effectiveness and robustness of the proposed scheme.

The rest of this paper is organized as follows. In Section 2, a brief review of the median filter and its characterization in term of edge preservation are given. Section 3 elaborates the extraction of EBPM features and the proposed MF detection scheme. Experiment results and analysis with the proposed MF detection scheme, and comparison with previous MF detection schemes are reported in Section 4. Finally the conclusion is drawn in Section 5.

2 The Median Filter and Its Characterization

Nonlinear image processing techniques are widely used in a variety of image processing applications, such as digital image filtering, image enhancement and edge detection [8]. One of the most important families of nonlinear image filters for noise removal are order statistic filters. Among this family, median filter is the simplest one, which was first proposed by Tukey in 1971 [9]. Since then, it has become the best known filter of this family and been successfully applied to signal and image processing. In the following, a brief review of the median filter and its characterization are given.

2.1 The Median Filter

Let $\boldsymbol{X} = (X_1, X_2, \ldots, X_n)$ be a finite set of random variables. By arranging the elements of $\boldsymbol{X}$ in ascending order of magnitude such that $X_{(1)} \leq X_{(2)}, \ldots, \leq X_{(n)}$, the order statistics associated with $\boldsymbol{X}$ is defined as $\boldsymbol{X}_{(\cdot)} = (X_{(1)}, X_{(2)}, \ldots, X_{(n)})$. Thus the median (or the middle) value $med(\boldsymbol{X})$ of $\boldsymbol{X}$ is given by:

$$med(\boldsymbol{X}) = \begin{cases} X_{(v+1)}, & \text{for } n = 2v+1 \\ (X_{(v)} + X_{(v+1)})/2, & \text{for } n = 2v \ . \end{cases} \tag{1}$$

In the interest of simplicity in definition and computation, median filters with window of odd dimension are usually employed in practice. A one-dimensional median filter of length n, $n = 2v + 1$, is defined as:

$$\hat{X}_i = med\big\{X_{i+r}; r \in \{-v, -v+1, \ldots, v-1, v\}\big\}, \tag{2}$$

where $\hat{X}_i$ is the output of the median filter. In the community of statistics,the median has long been recognized as a robust estimator of the location (mean) parameter and played an important role in the robust analysis of data contaminated with outlying observations, called outliers [8]. The median filter has also found to be very effective for signal smoothing and denoising, especially for one- or two-dimensional signals contaminated with impulsive noise [10].

In digital image processing, the two-dimensional median filters are widely used. Given a grayscale image with intensity value $X_{i,j}$, a 2-D median filter is defined by the following equation:

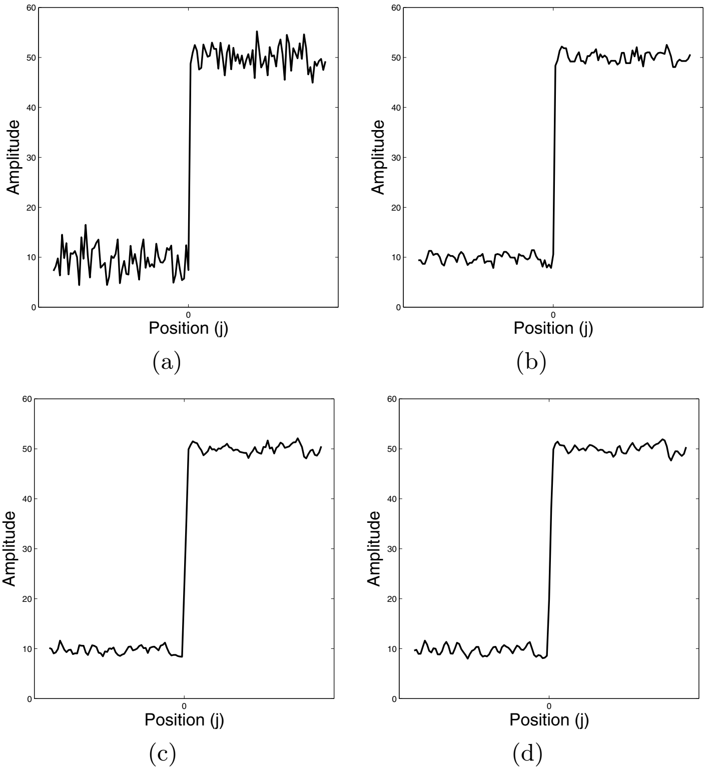
$$\hat{X}_{i,j} = med\{X_{i+r,j+s}; (r,s)\in W(i,j)\}, \tag{3}$$

where $\hat{X}_{i,j}$ is the output of the median filter and $W(i,j)$ is the 2-D filter window centered at image coordinates (i,j) with different shape such as square-, circle-, cross- and X-shaped of different sizes. The geometry and size of filter window may affect the spatial performance of the median filter, e.g. edge and image detail preservation [11]. The symmetric square window of odd dimension is assumed throughout the paper, which is the most widely used window in practical applications.

2.2 Edge Preservation with MF

In this section the characterization of MF is given. One of the main differences between the median filter and other linear filters, e.g. the moving average filter is that, for a median filter with window of odd dimension, the output image pixels are directly drawn from the set of input pixels with no rounding operation. Since the same pixels might be shared by the adjacent windows in median filtering, there exists a trend that the output pixels in a certain neighborhood would take the same value in median filtered image. Therefore, the median filter tends to produce regions of constant or nearly constant intensities. As the shape of these regions is usually either linear patches (streaks) or blotches, this effect is referred to as *streaking artifacts* [12]. Although the streaking artifacts can be explored to identify the MF operation [5] [6], the features, however, are proved to be vulnerable to some post processings such as JPEG compression. We then proceed to characterize and explore another intrinsic trace of MF, i.e., the edge preservation.

Human vision is very sensitive to high frequency information. Therefore, image edges and image details (e.g. corners and lines) usually carry important information for human visual perception. This makes edge preservation and edge enhancement a very important objective for image filter design [8]. Most of the



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

Fig. 1. (a) a section crossing the noisy image edge, (b) 3×3 median filter output, (c) 3×3 moving average filter output, (d) 3×3 gaussian filter output with delta=0.5

conventional linear filters, e.g. the moving average filter, cannot cope with edge and image detail preservation because of their low-pass characteristics. In fact, they tend to smooth and even destroy edges, and finally produce images which are unpleasant to the eyes. In contrast, the median filter performs much better in image edge preservation compared to linear filters [8] [11]. Bovik et al. in [11] found that median prefiltering can improve the performance of edge detector, as measured by the increased noise suppression away from edges and insubstantial loss of edge details. It is the good edge and image detail preservation properties that make the median filter quite applicable to digital image filtering.

To demonstrate the characteristic of good edge preservation with MF, we then consider the behavior of median filtered and linear filtered images in the vicinity of an idealized step edge with added noise. An idealized noisy vertical edge model $\{X^v_{i,j}\}$ of height $h > 0$, as shown in Fig. 1(a), is defined as follow:

$$X^v_{i,j} = \begin{cases} N_{i,j}, & j \leq 0 \\ h + N_{i,j}, & j \geq 1, \end{cases} \tag{4}$$

where $\{N_{i,j}\}$ denotes the white noise.

Fig. 1 shows that the median filter perform better edge preservation than both the moving average filter and gaussian filter in term of edge blurness. Similar results are also observed from images after JPEG compression. Based on this observation, we then proceed to develop a new forensic scheme by exploring the edge preservation as the intrinsic fingerprint for robust MF detection.

3 The Proposed MF Detection Scheme

The median filter employs different filter windows when performing filtering as other linear filters do. The overlapped filter window makes the output pixels of filtered images in a neighborhood be correlated to some extents, especially for the linear filtered images. Considering the non-linearity and good edge preservation of the median filter, we believe that the pixels of the median filtered images correlate to their neighbors in a quite different way compared to the pixels of original and linear filtered images. The observations motivate us to explore the statistics among edge regions of images as the intrinsic fingerprints for MF detection. We found that the neighborhood prediction model [13] provides a simple yet effective solution. With this framework, the edge based MF intrinsic fingerprints are characterized as EBPM, which contains the estimated prediction coefficients of neighborhood prediction among different edge regions in images. In the following, we will elaborate the proposed MF detection scheme.

3.1 Edge Block Classification

We first divide the image under investigation into N non-overlapping blocks of size $B \times B$ and obtain set $\mathbf{\Omega}$ (B=5 in our scheme). We then classify set $\mathbf{\Omega}$ into three types of edge blocks based on the gradient features of each block [14].

For the 5×5 block Ω_i shown in Fig. 2, the horizontal and vertical gradient features of Ω_i are defined as the linear combination of the absolute second-order gradient values:

$$\begin{aligned} G_H = {} & |I_{i,20} + I_{i,1} - 2I_{i,11}| + |I_{i,15} - I_{i,6}| \\ & + |I_{i,21} + I_{i,2} - 2I_{i,12}| + |I_{i,16} - I_{i,7}| \\ & + |I_{i,22} + I_{i,3} - 2I_{i}| + |I_{i,17} - I_{i,8}| \\ & + |I_{i,23} + I_{i,4} - 2I_{i,13}| + |I_{i,18} - I_{i,9}| \\ & + |I_{i,24} + I_{i,5} - 2I_{i,14}| + |I_{i,19} - I_{i,10}|, \end{aligned} \tag{5}$$

$$\begin{aligned} G_V = {} & |I_{i,5} + I_{i,1} - 2I_{i,3}| + |I_{i,4} - I_{i,2}| \\ & + |I_{i,10} + I_{i,6} - 2I_{i,8}| + |I_{i,9} - I_{i,7}| \\ & + |I_{i,14} + I_{i,11} - 2I_{i}| + |I_{i,13} - I_{i,12}| \\ & + |I_{i,19} + I_{i,15} - 2I_{i,17}| + |I_{i,18} - I_{i,16}| \\ & + |I_{i,24} + I_{i,20} - 2I_{i,22}| + |I_{i,23} - I_{i,21}| \ . \end{aligned} \tag{6}$$

Block Ω_i is then classified into one of the three edge blocks, i.e., H, V and O, based on its gradient features as follows (T is the predetermined threshold):

$I_{\mathrm{i},1}$	$I_{\mathrm{i},6}$	$I_{\mathrm{i},11}$	$I_{\mathrm{i},15}$	$I_{\mathrm{i},20}$
$I_{\mathrm{i},2}$	$I_{\mathrm{i},7}$	$I_{\mathrm{i},12}$	$I_{\mathrm{i},16}$	$I_{\mathrm{i},21}$
$I_{\mathrm{i},3}$	$I_{\mathrm{i},8}$	I_{i}	$I_{\mathrm{i},17}$	$I_{\mathrm{i},22}$
$I_{\mathrm{i},4}$	$I_{\mathrm{i},9}$	$I_{\mathrm{i},13}$	$I_{\mathrm{i},18}$	$I_{\mathrm{i},23}$
$I_{\mathrm{i},5}$	$I_{\mathrm{i},10}$	$I_{\mathrm{i},14}$	$I_{\mathrm{i},19}$	$I_{\mathrm{i},24}$

Fig. 2. Block Ω_i of size 5×5

a) H: blocks containing roughly horizontal edge with a significant vertical gradient, i.e., $G_{V,i} - G_{H,i} > T$.
b) V: blocks containing roughly vertical edge with a significant horizontal gradient, i.e., $G_{H,i} - G_{V,i} > T$.
c) O: Other blocks containing either horizontal and vertical edges or smooth regions.

For natural images, there may exist blocks with edges that are neither predominantly horizontal nor vertical. In addition, the image block may even have fine-detailed features which cannot be well characterized with a single edge orientation. Therefore, to well represent the edge blocks of natural images, a better edge block classification model could be adopted. Actually, we can easily extend our edge classification model into eight directions using the threshold-based variable number of gradients (VNG) algorithm in [15]. Although a slightly better performance is achieved, the edge block classification model with eight directions suffers from significantly high computational burden due to the significantly increased feature dimension. Consequently, we take the 3 block types classification model in our scheme.

3.2 Extraction of EBPM

In order to capture the statistics among different types of edge blocks, we make use of the neighborhood prediction model for H, V and O blocks in an image. We illustrate the procedure of feature extraction for H blocks, the similar approaches can be applied to V and O blocks. Let N_H be the number of H blocks in an image. For the Ω_i block of type H, a linear predictor for the central pixel value I_i using its $(B^2 - 1)$ neighbors is given by:

$$I_i = \sum_{k=1}^{B^2-1} \alpha_{H,k} I_{i,k}, \tag{7}$$

where $I_{i,k}$ is the pixel value of the (i, k) neighbor of block Ω_i as indicated in Fig. 2, and is the prediction coefficient associated with $I_{i,k}$. We assume that the same prediction model is applied for the all the blocks of same type in an image. We denote the N_H center pixel values and their corresponding prediction coefficients as $\boldsymbol{y} = [I_1, I_2, \ldots, I_N]^T$ and $\boldsymbol{\alpha}_H = [\alpha_{H,1}, \alpha_{H,2}, \ldots, \alpha_{H,B^2-1}]^T$, respectively. We then represent the $(B^2 - 1)$ neighbors of each center pixel as a row vector, and organize the neighbor pixels in all H blocks as a matrix Y of dimension $N_H \times (B^2 - 1)$. The neighborhood prediction model is then expressed compactly in matrix form as:

$$\boldsymbol{y} = Y \boldsymbol{\alpha}_H, \tag{8}$$

To find a solution for Eqn.(8), we have

$$\boldsymbol{e}(\boldsymbol{\alpha}_H) = \boldsymbol{y} - Y \boldsymbol{\alpha}_H \ . \tag{9}$$

The estimation of the prediction coefficients $\boldsymbol{\alpha}_H$ can then be formulated as a least-squares (LS) problem with respect to $\boldsymbol{e}(\boldsymbol{\alpha}_H)$ in Eqn.(9), which is minimized using a quadratic cost function:

$$J(\boldsymbol{\alpha}_H) = \boldsymbol{e}(\boldsymbol{\alpha}_H)^T Q \boldsymbol{e}(\boldsymbol{\alpha}_H), \tag{10}$$

where Q is the estimated covariance of $\boldsymbol{e}(\boldsymbol{\alpha}_H)$ (we assumed an identity matrix in this case). Setting the differentiation of $J(\boldsymbol{\alpha}_H)$ with respect to $\boldsymbol{\alpha}_H$ to zero, we obtain:

$$\boldsymbol{\alpha}_H = (Y^T Y)^{-1} Y^T \boldsymbol{y} \ . \tag{11}$$

By applying the same approach to other block types in image, we obtain the prediction coefficients $\boldsymbol{\alpha}_V$ and $\boldsymbol{\alpha}_O$ for V and O blocks. The obtained $3 \times (B^2 - 1)$ prediction coefficients are arranged as Edge Based Prediction Matrix (EBPM) defined in (12) and used as the intrinsic fingerprint for MF detection.

$$F = [\boldsymbol{\alpha}_H, \boldsymbol{\alpha}_V, \boldsymbol{\alpha}_O] \ . \tag{12}$$

It is obvious that the performance of the proposed blind MF detector is, to a great extent, dependent on the choice of the block size B. Although a larger block size leads to improved performance due to a larger feature vector, it may take risk to have an ill-posed matrix $Y^T Y$ when the image size under investigation is not large enough. Therefore B must be properly determined according to the image size and the expectation of allowed false alarm rate.

Fig. 3 gives the prediction coefficients $\boldsymbol{\alpha}_H$ obtained from H blocks of original and filtered Lena images, where $B = 5$ and $T = med\{|G_{V,i} - G_{H,i}| \, ; i \in N\}$. For H blocks in non-filtered image, it is observed that the prediction coefficients in the horizontal direction (0.419 and 0.415) are relatively greater than those in the vertical direction (0.308 and 0.379) as shown in Fig.3 (a). This indicates that, compared to the pixel values oriented along the vertical direction, the pixel values in the horizontal direction make relatively larger contribution to the prediction of the central pixel value. For H blocks in median filtered image, however, the prediction coefficients in the horizontal direction are significantly greater than those in vertical direction as shown in Fig. 3 (b), due to the effect of noise suppression and edge preservation with median filter. In contrast, for

0.020	-0.013	-0.058	-0.064	0.024	0.036	-0.065	0.044	-0.072	0.017
0.058	-0.104	0.308	0.030	-0.002	-0.020	0.121	-0.050	0.195	-0.032
-0.138	0.419	0	0.415	-0.104	-0.209	0.467	0	0.518	-0.189
0.049	-0.042	0.379	-0.115	0.060	0.076	0.156	0.014	0.089	0.018
0.021	-0.042	-0.122	0.015	0.005	-0.047	-0.041	-0.021	0.014	-0.023
		(a)					(b)		
0.033	-0.007	-0.031	-0.053	0.042	-0.016	0.065	-0.112	0.014	-0.000
-0.002	-0.277	0.559	-0.242	0.015	0.102	-0.272	0.482	-0.237	0.086
-0.043	0.525	0	0.551	-0.102	-0.212	0.573	0	0.616	-0.221
-0.019	-0.224	0.524	-0.281	0.044	0.127	-0.272	0.516	-0.305	0.133
0.038	-0.011	-0.075	0.034	0.003	-0.018	0.035	-0.124	0.062	-0.025
		(c)					(d)		

Fig. 3. Prediction coefficients obtained from H blocks of image Lena (a) original image, (b) 3×3 median filtered image, (c) 3×3 moving average filtered image, (d) 3×3 gaussian filtered image with delta=0.5

average and gaussian filtered images (see Fig. 3 (c) and (d)), the difference between coefficients in horizontal and vertical directions becomes much smaller, this is because the linear filters tend to blur edges and destroy image details. The similar results can also be observed from the prediction coefficients of V blocks in images.

3.3 MF Detector Based on EBPM

The EBPM features can well characterize the statistics around edge regions to distinguish among the original images, the median filtered images and the linear filtered images. By incorporating the support vector machine (SVM), the MF detector based on EBPM is developed. Fig.4 gives the sketch of the proposed MF detector. First, the image under investigation is divided into non-overlapping blocks of size $B \times B$, and each block is classified into one of the three block types according to its horizontal and vertical gradient features. Then, for each edge block type, we fit the neighborhood prediction model to the corresponding blocks in image, and estimate the prediction coefficients using LS algorithm. Finally, we take all the prediction coefficients as the EBPM features and feed them as inputs to SVM for MF detection.

4 Experimental Results and Analysis

In this section, experiment evidence and analysis on a large image database are presented to demonstrate the performance of the proposed MF detection scheme. The comparisons with previous MF detection schemes are also included.

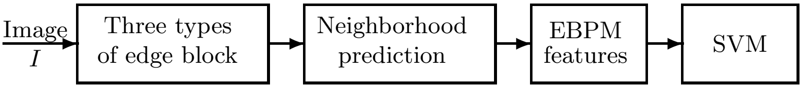


Fig. 4. Sketch of EBPM based MF detector

4.1 Experimental Setup

To evaluate the performance of the proposed MF detection scheme, we make use of the well-known image database UCID [16], which includes 1338 uncompressed TIFF images of 512 × 384 or 384 × 512. The database consists of images with different texture characteristics and taken under varying lighting conditions. All of these images are converted to grayscale ones prior to any further processing. In our experiments, the block size is $B = 5$ and the threshold for edge block type classification is set as $T = med\{|G_{V,i} - G_{H,i}|; i{\in}N\}$. In order to enhance the contribution of edges and fine image details, the image under investigation is convolved with a 3×3 Laplacian operator prior to extraction of EBPM features. For each input image, we extract a $3 \times (5^2 - 1) = 72$ -D EBPM features as discussed in Section 3.

We then linearly scaling the feature values associated with individual features to the range of $[-1, 1]$ and feed them as inputs to a support vector machine LIBSVM [17] for detection. Throughout all experiments, C-support vector classification with the non-linear RBF kernel $K(x_i, x_j) = \exp(-\gamma||x_i - x_j||^2)$ are performed with parameters chosen for ensuring no overfitting [18]. The parameters i.e., the error penalty parameter C and the RBF kernel parameter γ, are determined with an exhaustive grid search strategy suggested in [18] over all possible pairs on the following multiplicative grid:

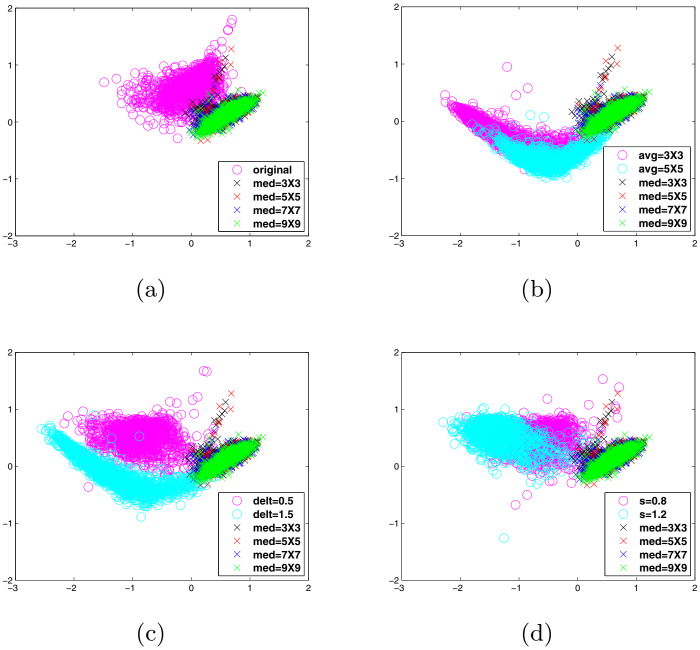
$$C \in \left\{2^c | c \in \{-5, -4, \ldots, 5\}\right\} \times \gamma \in \left\{2^g | g \in \{-5, -4, \ldots, 5\}\right\} \tag{13}$$

using five-fold cross-validation. The (C_0, γ_0) with the highest average cross-validation accuracy is adopted. To demonstrate the performance, the SVM classification results are depicted in the form of receiver operating characteristic (ROC) curve obtained from the five-fold cross-validation test with best (C_0, γ_0).

In the following, we will evaluate the performance of the proposed MF detection scheme from two aspects: 1) distinguish MF from original; 2) distinguish MF from other manipulations.

4.2 Distinguish MF from Original

To evaluate the feature separability, the 72-D EBPM features are extracted from original and manipulated images using image database UCID. Fig. 5 shows the results of projecting the original 72-D vectors onto the top two principal components (as computed from a principle component analysis(PCA) [19]), which capture about 67% of the total variance. It is noted that projection points corresponding to different types of sample images tend to clump together in clusters.



(a) (b)

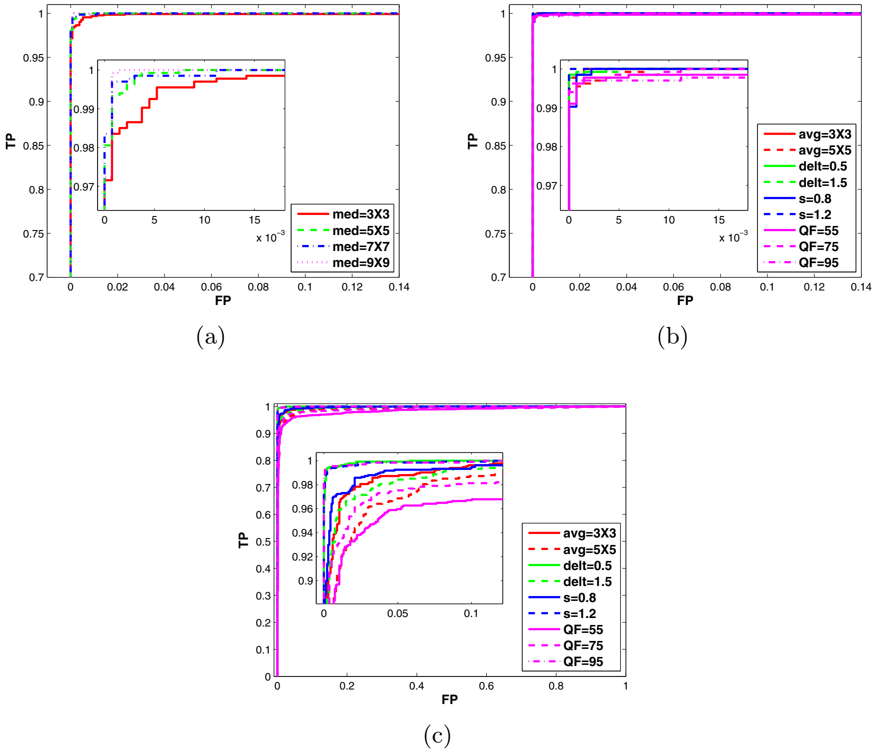
(c) (d)

Fig. 5. Projections of 72-D EBPM features extracted from different types of sample images using UCID database onto a 2-D subspace spaned with top two PCA components for median filtered (window size med=3×3, 5×5, 7×7, 9×9) and (a) original, (b) average filtered (window size avg=3×3, 5×5, (c) gaussian filtered (delt=0.5, 1.5), (d) bilinear scaled (scaling factor s=0.8, 1.2)

Even in the reduced space, there is a clear separation between original and median filtered images, suggesting that the proposed EBPM features are useful fingerprints to distinguish MF from original.

A. MF Detection without Other Manipulations

When no other manipulations are involved, the original images and their median filtered ones with different filter window sizes are taken as negative (N) and positive samples (P), respectively. Perfect ROC performances are achieved for all the filter window sizes as shown in Fig. 6(a). The ROC performance in low FP rates is zoomed-in and displayed in the center of each ROC curve. It is also observed that, even for the median filtered image with small window size (e.g., med=3×3), which usually has no visible distortion, the true positive (TP) rate is consistently kept above 0.95 for all false positive (FP) rates. The proposed MF detector has a comparable performance to that of Caos scheme [6] for small to medium window sizes, and slightly better performance for large window sizes (e.g. med=9×9). The zoomed-in part in Fig. 6(a) also reveals the facts that, the performance of our scheme is improved with the increase of filter window size. When using Caos scheme [6], however, the detection performance



(a)

(b)

(c)

Fig. 6. ROC curve for MF detection (a) without other manipulations, window size for median filter: med=3×3, 5×5, 7×7, 9×9, (b) with manipulations before MF, window size for median filter: med=5×5, (c) with manipulations after MF, window size for median filter: med=5×5

is degraded slightly for larger window size. This can be explained by the fact that the probability of zero values on the first-order difference image in texture regions, the MF fingerprint used in Caos scheme, tends to decrease when large filter window size is used.

B. MF Detection with Other Manipulations before MF

Fig. 6(b) shows the effects of other image manipulations before MF on the ROC performance of our MF detection scheme. Several common image manipulations, including the moving average filtering, the gaussian filtering, bilinear scaling and JPEG compression, are considered. It is observed from Fig. 6(b) that an almost perfect discrimination between different manipulated images (N) and their 5×5 median filtered ones (P) is achieved with the proposed MF detector. Compared to Caos scheme in [6], our EBPM based scheme achieves significant performance improvements for gaussian filtered and average filtered images, and comparable performance for other manipulated images. Perfect performance is also achieved for MF detection on resampled images (bilinear scaling as shown in Fig. 6(b)), suggesting the feasibility of the proposed MF detection scheme to tackle the effect of previously resampling operation. The result is encouraging as it shows the initial promise of the forensic detection on the new resampling algorithm in [7], which applies median filtering to hide the interpolation traces.

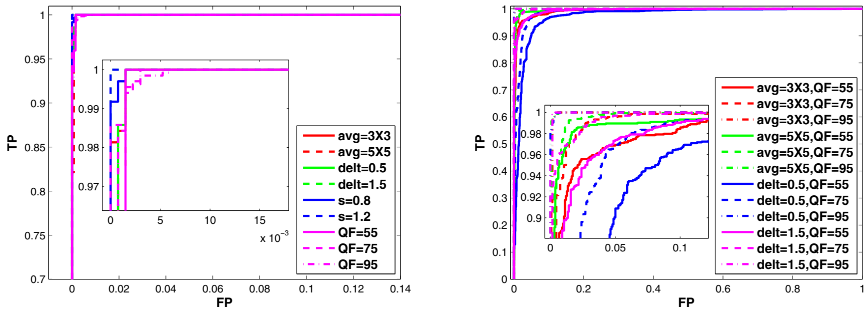
C. MF Detection with Other Manipulations after MF
Fig. 6(c) shows the ROC performance of classification between the median-filtered (5 × 5) sample images' different post-manipulated versions (P) and the manipulated original images (N). Common post-processing, such as the moving average filtering, the gaussian filtering, bilinear scaling and JPEG compression are applied. For all the manipulations tested, the true positive (TP) rate is kept above 0.9 for every false positive (FP) rate greater than 0.05, suggesting the good robustness performance of our MF detection scheme against post-processing. It is worthwhile to point out that, compared to the Kirchners scheme [5] and Caos scheme [6], the proposed MF detection scheme is robust against JPEG compression with different QF. While both the features of streaking artifacts in [5] and probability of zero values in first-order differences in [6] can not survive even the moderate JPEG compression. On the other hand, robustness against post linear filtering is also of forensic interest. Rabiner et al. in [20] proposed to use short linear filter to diminish streaking artifacts generated with median filter. Fig. 6(c) also shows that the proposed scheme can detect the median filtered images post-processed by the average filter and gaussian filter with high accuracy, and performs much better than Caos scheme [6].

4.3 Distinguish MF from Other Manipulations

For image forensics, it is beneficial to know as much as possible about the general processing history of an image, thus a wide variety of operations must be tested. Herein we shall evaluate the performance of the proposed scheme on distinguishing MF from other manipulations especially from linear smoothers. As is shown in Fig.5, the cluster of median filtered images has clear separation with those of other manipulated images, which indicates the effectiveness of the EBPM features in distinguishing MF from other manipulations.

As is depicted in Fig. 7(a), perfect performance for distinguishing 5×5 median filtered images (P) from other manipulated images (N) is achieved. Compared to Cao's scheme [6], our scheme performs better in classification between MF and the linear filters. This is because the better edge preservation property of MF comparing to other linear filters.

Considering that JPEG is the widely used format for image storage and transmission, it is interesting to evaluate the performance of the proposed MF detection scheme for classification between MF and the linear filters after JPEG compression. It is observed in [5] that the linear filters have similar effects on the SPAM features as the median filter after JPEG compression, which proliferates the ambiguities in determination the processing history of images using SPAM



(a) (b)

Fig. 7. ROC curve for the classification between (a) median-filtered (med=5×5) images and the images processed by other manipulations, (b) the images processed by median filter (med=5 × 5) and linear filters after JPEG compressed

features. Fig. 7(b) shows that an almost perfect discrimination between JPEG compressed median filtered images (P) and linear filtered images (N) is obtained for different QF in the range of 55, 75, 95.

5 Conclusion

In this paper, an effective forensic scheme for median filtering detection is proposed using Edge Based Prediction Matrix (EBPM). The statistics around edge regions, such as neighborhood correlation, noise suppression and edge preservation, are characterized and analyzed. Through neighborhood prediction model, the EBPM features are taken as the prediction coefficients for different edge regions. By incorporating the support vector machine (SVM), the MF detector is developed based on the EBPM features. Extensive simulations are carried out to demonstrate the effectiveness and robustness of the proposed MF detection scheme.

In our future work, we will extend the EBPM features as a general purpose filter detector to identify linear and other non-linear filters.

Acknowledgments. This work is supported by National Natural Science Foundation of China under contract 60970145.

References

1. Popescu, A.C., Farid, H.: Statistical Tools for Digital Forensics. In: Fridrich, J. (ed.) IH 2004. LNCS, vol. 3200, pp. 128–147. Springer, Heidelberg (2004)
2. Popescu, A.C., Farid, H.: Exposing digital forgeries by detecting traces of resampling. IEEE Transactions on Signal Processing 53(2), 758–767 (2005)

3. Neelamani, R., de Queiroz, R., Fan, Z., Dash, S., Baraniuk, R.G.: JPEG compression history estimation for color images. IEEE Transactions on Image Processing 15(6), 1365–1378 (2006)
4. Stamm, M., Liu, K.J.R.: Blind forensics of contrast enhancement in digital images. In: Proceedings of the 15th IEEE International Conference on Image Processing (ICIP), pp. 3112–3115 (2008)
5. Kirchner, M., Fridrich, J.: On detection of median filtering in digital images. In: SPIE Electronic Imaging: Security, Steganography, and Watermarking of Multimedia Contents, San Jose, CA, USA (2010)
6. Cao, G., Zhao, Y., Ni, R., Yu, L., Tian, H.: Forensic detection of median filtering in digital images. In: Proceedings of the 2010 IEEE International Conference on Multimedia and Expo. (ICME), pp. 89–94 (2010)
7. Kirchner, M., Böhme, R.: Hiding traces of resampling in digital images. IEEE Transactions on Information Forensics and Security 3(4), 582–592 (2008)
8. Pitas, I., Venetsanopoulos, A.N.: Order statistics in digital image processing. Proceedings of the IEEE 80(12), 1893–1921 (1992)
9. Tukey, J.W.: Nonlinear (nonsuperposable) methods for smoothing data. In: Conf. Rec., EASCON, p. 673 (1974)
10. Gallagher Jr., N.C., Wise, G.L.: A theoretical analysis of the properties of median filters. IEEE Transactions Acoust., Speech, Signal Processing ASSP-29, 1136–1141 (1981)
11. Bovik, A.C., Huang, T., Munson, D.: The effect of median filtering on edge estimation and detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 9(2), 181–194 (1987)
12. Bovik, A.C.: Streaking in median filtered images. IEEE Transactions on Acoustics, Speech and Signal Processing 35(4), 493–503 (1987)
13. Gou, H., Swaminathan, S., Wu, M.: Robust Scanner Identification based on Noise Features. In: Proc. SPIE, Security, Steganography, and Watermarking of Multimedia Contents IX (2007)
14. Swaminathan, A., Wu, M., Liu, K.J.R.: Nonintrusive Component Forensics of Visual Sensors Using Output Images. IEEE Transactions on Information Forensics and Security 2(1) (2007)
15. Chang, E., Cheung, S., Pan, D.Y.: Color filter array recovery using a threshold-based variable number of gradients. In: Proc. SPIE, Sensors, Cameras, and Applications for Digital Photography, vol. 3650, pp. 36–43 (1999)
16. Schaefer, G., Stich, M.: UCID - An uncompressed colour image database. In: Proc. SPIE, Storage and Retrieval Methods and Applications for Multimedia, San Jose (2004)
17. Chang C.C., Lin, C.J.: LIBSVM: a library for support vector machines. (2001), Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`
18. Hsu, C.W., Chang, C.C, Lin, C.J.: A practical guide to support vector classification, `http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf`
19. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. Wiley Interscience (2000)
20. Rabiner, L.R., Sambur, M.R., Schmidt, C.E.: Applications of a nonlinear smoothing algorithm to speech processing. IEEE Transactions Acoust., Speech, Signal Processing 23, 552–557 (1975)

New Feature Presentation of Transition Probability Matrix for Image Tampering Detection*

Luyi Chen[1], Shilin Wang[2], Shenghong Li[1], and Jianhua Li[1]

[1] Department of Electrical Engineering, Shanghai Jiaotong University, Shanghai, China
lychen1109@gmail.com, {shli,lijh888}@sjtu.edu.cn
[2] School of Information Security, Shanghai Jiaotong University, Shanghai, China
wsl@sjtu.edu.cn

Abstract. Extraction of discriminative feature is crucial to machine learning approach of image tampering detection. The state-of-the-art Markov transition probability feature is extended in this paper. We show that correlation between adjacent elements on the difference array of block DCT coefficients can be theoretically calculated and provides little information to the classification problem. We propose to decorrelate the variables and use the marginal distribution as feature in image tampering detection. The framework is applied to 1st and 2nd order Markov transition probability feature. Our experiment result shows the new presentation of the feature has competitive performance and greatly reduced dimensionality.

Keywords: Image Tampering Detection, Image Splicing Detection, Markov Transition Probability, Machine Learning, Pattern Recognition.

1 Introduction

Widely availability of powerful digital image processing software has made it easy to produce tampered digital images which are indistinguishable with human eyes. Thus effective technique is needed to detect those changes.

Generally there're active and passive methods. Active methods solve the problem by embedding watermarks or recording checksum to insure the integrity of the image. But it cannot deal with legacy content produced before the algorithm was designed. And it also changes the original content, which makes it unacceptable in some situations.

On the contrary, passive methods exploit knowledge of characteristics of natural images and their producing procedure. It collects information from cameras, compression algorithms, optics, and natural image models. For a good survey of this field, readers are referred to [1].

* This research work is funded by the National Science Foundation of China (61071152, 60702043), 973 Program (2010CB731403, 2010CB731406) of China, Shanghai Educational Development Foundation and Project of Beijing Key Laboratory of Communication and Information System of Beijing Jiaotong University.

Y.Q. Shi, H.J. Kim, and F. Perez-Gonzalez (Eds.): IWDW 2011, LNCS 7128, pp. 376–386, 2012.

One popular approach, in passive methods category, is to solve the problem under the framework of two-class pattern recognition. Effective features are extracted and used by classifier to predict if an image is an authentic image or a tampered one. Features with discriminative power have been discovered [2-3]. And benchmark dataset [4] has also been created.

Markov transition probability feature [5], proposed by Shi, et al., is currently the state-of-the-art in this field. The feature has been proved effective in both image tampering detection and steganalysis [6]. The problem of the feature is that the dimensionality is very large. Even with thresholding technique, the dimensionality of the feature is still $(2T+1)^2$. T is the threshold of the difference array of block DCT coefficient. Feature selection has been used for dimension reduction [7-8].

In this paper, we extend the Markov transition probability feature from another angle. We show that correlation between adjacent elements on difference array of block DCT coefficients can be theoretically calculated, using only property of DCT transform. The result holds regardless of the class of image processed, so it is not useful for classification. We propose to decorrelate adjacent elements on the difference array, and use the marginal distribution of the new variables as feature. Our experiment result shows this not only gives competitive performance, but also solves the dimensionality problem of the original Markov transition probability feature.

The same method has been used to approximate co-occurrence matrix [9]. It may not be a good approximation, since the independence of the new variables is not assured. But in our problem, it reduces the correlation which has no value to the classification, while keeps other aspects of the signals.

The rest of the paper is organized as follows. In section 2, Markov transition probability feature is introduced and the correlation between adjacent elements of the difference array is analyzed. In section 3, KL transform is used to decorrelate the random variables. Experimental result is reported in section 4. Section 5 draws the conclusion.

2 Markov Transition Probability Feature and the Motivation of Decorrelation

2.1 Markov Transition Probability Feature

An N×M digital image is processed with 8x8 block DCT transform. The absolute value of block DCT coefficient array is denoted as $\{x_{ij}\}$ (i=1…N, j=1…M). Then, the difference array of horizontal adjacent elements of $\{x_{ij}\}$ is denoted as

$$y_{ij} = x_{ij} - x_{i+1,j} \quad (i \in [1 \ldots N-1], j \in [1 \ldots M]) \tag{1}$$

Horizontal Markov transition probability on the difference array is denoted as $P(y_{i+1,j} \mid y_{ij})$. In [5] and [7], this Markov transition probability is used as feature in classification of authentic and spliced images.

2.2 Correlation between Adjacent Elements on the Difference Array

DCT transform approaches KL transform, which is optimal in the decorrelation sense, so we can approximate correlation between adjacent block DCT coefficients with zero.

$$\rho(x_{ij}, x_{i+k,j}) = \frac{E[(x_{ij} - \overline{x})(x_{i+k,j} - \overline{x})]}{\sigma_x^2} \approx 0 \ (k \in [1 \ldots N-1]) \tag{2}$$

$\overline{x}$ and σ_x are mean and standard deviation of $\{x_{ij}\}$ respectively. We can preprocess $\{x_{ij}\}$ to subtract its mean, which will not affect $\{y_{ij}\}$ according to (1). So we have

$$\overline{x} = E(x_{ij}) = 0 \tag{3}$$

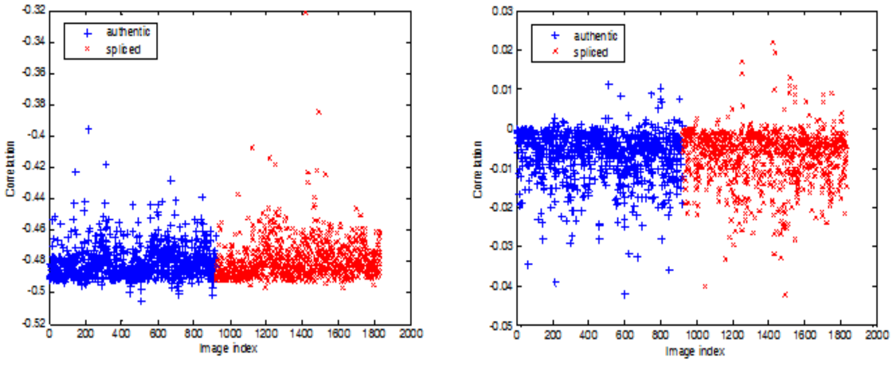
The following equation can be derived from (2) and (3).

$$E(x_{ij} x_{i+k,j}) \approx 0 \quad (k = 1, 2, \ldots N-1) \tag{4}$$

In reality, $\{x_{ij}\}$ is the absolute value of block DCT coefficients, but (4) is still a good approximation. Using (1) and (4), we can calculate correlation between adjacent elements on the difference array. The proof is left in the appendix.

$$\rho(y_{ij}, y_{i+k,j}) \approx \begin{cases} -0.5 & (k = 1) \\ 0 & (k > 1) \end{cases} \tag{5}$$

Since this correlation is dominated by the property of DCT transform, class information is overwhelmed. We calculated the correlation from Columbia Splicing Evaluation Dataset (section 4), and plotted it on Figure 1. The figure clearly shows that the correlation has similar distribution between two classes of images.



(1) (2)

Fig. 1. Correlation between adjacent elements on difference array of block DCT coefficients: (1) k=1; (2) k=2

We propose that this correlation in $\{y_{ij}\}$ can be removed. We use KL transform to decorrelate the variables, while any orthogonal transform can be used.

3 Decorrelation with KL Transform

3.1 Two Adjacent Elements

KL transform performs eigenvalue decomposition of the covariance matrix of random variables. Then the eigenvectors are used as basis in the transformed space. From the above analysis, the correlation of two adjacent elements in $\{y_{ij}\}$ is a symmetric matrix like

$$\begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix} \tag{6}$$

The ideal value of ρ is -0.5. In reality, since the sample image has limited size, ρ estimated from sample image is not exactly -0.5, but close enough. In the following analysis, we will show that no matter what the real ρ is, the KL transform matrix is not changed.

Eigenvalue of (6) is

$$\begin{cases} \lambda_1 = 1-\rho \\ \lambda_2 = 1+\rho \end{cases} \tag{7}$$

And the corresponding eigenvectors are

$$\beta_1 = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{pmatrix} \quad \beta_2 = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} \tag{8}$$

Be aware that β has nothing to do with ρ, so the real ρ only affects eigenvalue. Using (8), we can transform the adjacent elements in the difference array into new valuables

$$\begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = (\beta_1 \quad \beta_2)^T \begin{pmatrix} y_{ij} \\ y_{i+1,j} \end{pmatrix} \tag{9}$$

In this transform, we constructed two new variables, which are orthogonal to each other. Their physical meaning is a pair of linear filters operated on the difference array. Our feature is extracted from the marginal histogram of the new variables.

3.2 Three Adjacent Elements

Second order Markov transition probability has been used as feature in steganalysis [10]. The definition of the second order Markov transition probability is $P(y_{i+2,j} \mid y_{ij}, y_{i+1,j})$. Similar result as in section 3.1 can be produced. According to the analysis in section 2.2, the correlation of three adjacent elements in $\{y_{ij}\}$ is

$$\begin{pmatrix} 1 & \rho & 0 \\ \rho & 1 & \rho \\ 0 & \rho & 1 \end{pmatrix} \tag{10}$$

The eigenvalues are

$$\begin{cases} \lambda_1 = 1 \\ \lambda_2 = 1 - \sqrt{2}\rho \\ \lambda_3 = 1 + \sqrt{2}\rho \end{cases} \tag{11}$$

And the corresponding eigenvectors are

$$\beta_1 = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ -\frac{1}{\sqrt{2}} \end{pmatrix} \quad \beta_2 = \begin{pmatrix} -\frac{1}{2} \\ \frac{1}{\sqrt{2}} \\ -\frac{1}{2} \end{pmatrix} \quad \beta_3 = \begin{pmatrix} \frac{1}{2} \\ \frac{1}{\sqrt{2}} \\ \frac{1}{2} \end{pmatrix} \tag{12}$$

Again, β has nothing to do with ρ. Thus we can transform the adjacent elements into three new variables

$$\begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix} = (\beta_1 \quad \beta_2 \quad \beta_3)^T \begin{pmatrix} y_{ij} \\ y_{i+1,j} \\ y_{i+2,j} \end{pmatrix} \tag{13}$$

Marginal histograms of z_1, z_2 and z_3 are used to generate features for image tampering detection.

3.3 Alternative Form of Joint Statistics of Adjacent Elements

Similar to Markov transition probability, co-occurrence matrix has also been used as feature for splicing detection [11-12]. The difference between Markov transition probability and co-occurrence matrix is: the former is conditional probability, the latter is join probability. Their relationship is shown with the following equation

$$P(y_{ij}, y_{i+1,j}) = P(y_{i+1,j} \mid y_{ij}) P(y_{ij}) \tag{14}$$

If we know the joint probability, we will be able to calculate the marginal probability of each variable. Thus we can calculate the conditional probability from (14). But this is not the case the other way round. We will not be able to know joint probability when we only know the conditional probability, because we cannot calculate marginal probability from conditional probability.

This shows that joint probability contains more information than conditional probability. In real calculation of conditional probability, we first construct the joint probability matrix, and then we normalize every row of the matrix. By doing this, we lose n degree of freedom of the data, where n is the number of rows of the matrix.

But based on our test, they provide similar performance in classification. So we treat them as alternative form that describes joint statistics of adjacent elements. Without particular notice, we use conditional probability form as proposed by Shi, et al.

4 Implementation and Experiment Result

4.1 Concrete Implementation

We exclude DC component in construction of our marginal histogram. Based on our test, including or excluding DC component has little effect on the performance of the features.

In construction of Markov transition probability, the coefficient of block DCT transform is quantized to the nearest integer. In our new form, this quantization generates rounding error which affects the performance of the classification, so we omit this quantization step in our implementation.

In order to reduce the dimensionality of the Markov transition probability feature, the elements on the difference array is processed with threshold T. T is a selected positive integer, usually 3 or 4. In our new presentation, the threshold is still necessary. The performance decreases without the threshold. Our guess is that without the threshold, the dynamic range of the difference array causes instability in the tail of the new constructed marginal histogram.

The bins of the new marginal histogram are decided according to the dynamic range of the difference array. We take (9) as an example. The dynamic range of z_1 is decided in the following way.

$$\begin{aligned} |z_1| &= \frac{1}{\sqrt{2}} |y_{ij} - y_{i+1,j}| \\ &\le \frac{1}{\sqrt{2}} (|y_{ij}| + |y_{i+1,j}|) \\ &\le \frac{2}{\sqrt{2}} T \end{aligned} \tag{15}$$

The bins of the marginal histogram is selected as [-B … B], B is calculated with

$$B = floor(\frac{2}{\sqrt{2}}T) \tag{16}$$

Floor operation means rounding to the nearing integer towards minus infinity. Dimension of the feature extracted from this histogram is 2B+1. Dimension of other features can be decided accordingly.

For Simplicity, the constant $\frac{1}{\sqrt{2}}$ in (15) can be omitted. The range of z is roughly from -2T to 2T. So we can use 4T+1 bins to represent one histogram.

4.2 Experiment Result

We tested the performance of our features on Columbia splicing detection evaluation datasets [4]. Duplicate images in the dataset are removed. There are totally 1831 effective images left in the dataset, 921 authentic images and 910 spliced images.

We use roughly 2/3 of the data as training set, and 1/3 as test set. The dataset is randomly split into 20 sets of 1231 training samples and 600 test samples. Average test accuracy and their standard deviation are reported in Table 1. Here, we choose T=8. And the bins are selected with (16).

LibSVM [13] is used as classifier. We use Gauss RBF kernel. Parameters are selected by doing grid search with the first 5 sets. Then we take the median of the best parameters.

Our result shows that the performance of the new presentation is comparable with the original Markov transition probability feature. For first order Markov transition probability, the new presentation is significantly better than the original. P-value of the pairwise T-test of the testing accuracy on 20 sets is 0.002.

The new presentation of the 2nd order Markov transition probability is almost the same as the original, but not better. P-value of the pairwise T-test is 0.34. Feature dimensionality is 1/10 of the original.

Table 1. Classification accuracy of Markov transition probability matrix as feature and histogram of decorrelated variables as feature on Columbia splicing detection evaluation dataset

Type of Joint Statistics Feature		Dimension	Accuracy
2 elements	1st order Markov Transition Probability	81	87.09 (1.39)
	Our new form (Sec. 3.1)	46	87.97 (1.45)
3 elements	2nd order Markov Transition Probability	729	85.84 (0.92)
	Our new form (Sec. 3.2)	77	85.54 (1.34)

Table 1 also shows 2nd order Markov transition probability feature is worse than 1st order one on this particular dataset. This is not an intuitive result, because 2nd order transition probability includes one more state, so it includes more information as the following equation shows.

$$P(y_{i+2,j} \mid y_{i+1,j}) = \sum_{y_{i,j}} P(y_{i+2,j} \mid y_{i,j}, y_{i+1,j}) P(y_{i,j}) \tag{17}$$

One possible reason is that images contained in this dataset are too small, which are all 128x128. So they cannot provide an accurate estimation of higher order transition probability. More theoretical analysis is left to future research.

We also tested the performance of our new form when it is combined with moment features used in [5, 7]. The result is shown in Table 2.

From Table 2 we can see that the new form has almost the same performance as the original. While the mean is slightly lower, they do not pass T-test. We cannot simply say which is better with only 20 randomly selected sets. The advantage of the new form is that the dimension of the feature grows linearly with the threshold.

4.3 Computing Complexity

In case of two adjacent elements, our new form replaces one 2nd order histogram with two 1st order histograms. In order to compare their computing complexity, we recorded running time of our Matlab code on 30 randomly selected images in the same dataset. The Result is shown in Table 3.

Result in Table 3 shows that our new form is slightly faster than the original algorithm. Also 1st order histogram needs far less memory to store the data, so our new form is more efficient than the original.

Table 2. Test accuracy of our new form plus moment feature

Feature	T	Dimension	Accuracy
Moment+Transition Probability Matrix	3	266	89.86 (1.02)
Moment+New Form	3	220	89.62 (0.91)
	4	236	89.78 (1.03)
	5	252	89.78 (1.09)

Table 3. Average Computing time of feature extraction on 30 randomly selected images

Feature Type	Computing Time (seconds)
Transition Probability Matrix	0.0516 (0.0054)
Marginal Distribution of two new variables	0.0502 (0.0005)

5 Conclusion

In this paper, we evaluated the transition probability matrix feature proposed by Shi, et al. We showed that correlation between adjacent elements on the difference array of block DCT coefficients can be theoretically calculated using the property of DCT transform. This correlation provides little information to our classification problem. KL transform is used to construct new variables which are uncorrelated to each other. Marginal probability of the new variables is used as feature in image tampering detection.

Our new form of feature construction gives almost the same performance as the original transition probability matrix. The advantage is that the dimension is more manageable. It grows linearly with the threshold, while the original has square growth rate. In terms of computing complexity, it is also more efficient.

Under our new form, the dimension of 2nd order Markov transition probability feature has become manageable. But it still cannot give better performance. New ways of feature construction other than joint statistics of adjacent elements of difference array is needed to further improve the performance. This has motivated us to dig deeper into the statistical property of the feature instead of pursuing higher order transition probability.

Acknowledgements. The authors would like to thank Professor Yunqing Shi for reviewing the draft of the paper, and Doctor Candidate Xudong Zhao and Yujing Zhang for their useful discussions. Credits for the use of the Columbia Image Splicing Detection Evaluation Dataset are given to the DVMM Laboratory of Columbia University, CalPhotos Digital Library and the photographers listed in http://www.ee.columbia.edu/ln/dvmm/downloads/AuthSplicedDataSet/photographers.htm.

References

1. Farid, H.: A Survey of Image Forgery Detection. IEEE Signal Processing Magazine 26(2), 16–25 (2009)
2. Ng, T.-T., Chang, S.-F.: A model for image splicing. In: IEEE International Conference on Image Processing (ICIP), Singapore (2004)
3. Ng, T.-T., Chang, S.-F., Sun, Q.: Blind Detection of Photomontage Using Higher Order Statistics. In: IEEE International Symposium on Circuits and Systems (ISCAS), Vancouver, Canada (2004)
4. Ng, T.-T., Chang, S.-F.: A Dataset of Authentic and Spliced Image Blocks, ADVENT Technical Report, #203-2004-3, Columbia University (2004)
5. Shi, Y.Q., Chen, C., Chen, W.: A Natural Image Model Approach to Splicing Detection. In: The 9th Workshop on Multimedia & Security, pp. 51–62. ACM, New York (2007)
6. 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)
7. Sutthiwan, P., et al.: New developments in color image tampering detection. In: IEEE International Symposium on Circuits and Systems (ISCAS), Paris, pp. 3064–3067 (2010)
8. Lin, J.-Q., Wang, X.-D., Zhong, S.-P.: Reduction of Markov Extended Features in JPEG Image Steganalysis. In: 2nd International Congress on Image and Signal Processing (CISP), pp. 1–5. IEEE, Tianjin (2009)
9. Unser, M.: Sum and difference histograms for texture classification. IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-8(1), 118–125 (1986)
10. Pevny, T., Bas, P., Fridrich, J.: Steganalysis by Subtractive Pixel Adjacency Matrix. IEEE Transactions on Information Forensics And Security 5(2), 215–224 (2010)
11. Wang, W., Dong, J., Tan, T.: Effective image splicing detection based on image chroma. In: 16th IEEE International Conference on Image Processing (ICIP), pp. 1257–1260. IEEE, Cairo (2009)
12. Haralick, R.M., Shanmugam, K., Dinstein, I.: Textural Features for Image Classification. IEEE Transactions on Systems, Man, and Cybernetics SMC-3(6), 610–621 (1973)
13. Chang, C.-C., Lin, C.-J.: LIBSVM: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2(3) (2011)

Appendix: Proof of Equation (5)

Correlation between adjacent elements in $\{y_{ij}\}$ can be denoted as

$$\rho(y_{ij}, y_{i+1,j}) = \frac{E[(y_{ij} - \overline{y})(y_{i+1,j} - \overline{y})]}{\sigma_y^2} \tag{18}$$

According to (1)

$$\begin{aligned} \overline{y} &= E(x_{ij} - x_{i+1,j}) \\ &= E(x_{ij}) - E(x_{i+1,j}) = 0 \end{aligned} \tag{19}$$

Numerator of (18) can be calculated as follows

$$\begin{aligned} E(y_{ij} y_{i+1,j}) &= E[(x_{ij} - x_{i+1,j})(x_{i+1,j} - x_{i+2,j})] \\ &= E(x_{ij}x_{i+1,j} - x_{i+1,j}^2 - x_{ij}x_{i+2,j} + x_{i+1,j}x_{i+2,j}) \\ &= E(x_{ij}x_{i+1,j}) - E(x_{i+1,j}^2) - E(x_{ij}x_{i+2,j}) + E(x_{i+1,j}x_{i+2,j}) \end{aligned} \tag{20}$$

Using (4), (20) can be approximated with

$$E(y_{ij} y_{i+1,j}) \approx -E(x_{i+1,j}^2) \tag{21}$$

Denominator of (18) can be calculated with

$$\begin{aligned} \sigma_y^2 &= E(y_{ij} - \overline{y})^2 \\ &= E(y_{ij}^2) \\ &= E[(x_{ij} - x_{i+1,j})^2] \\ &= E(x_{ij}^2 + x_{i+1,j}^2 - 2x_{ij}x_{i+1,j}) \\ &= 2E(x_{ij}^2) - 2E(x_{ij}x_{i+1,j}) \\ &\approx 2E(x_{ij}^2) \end{aligned} \tag{22}$$

Substitute (21) and (22) back to (18), we get

$$\rho(y_{ij}, y_{i+1,j}) \approx \frac{-E(x_{i+1,j}^2)}{2E(x_{ij}^2)} = -0.5 \tag{23}$$

Similarly, covariance between y_{ij} and $y_{i+k,j}$ (k>1) is

$$\begin{aligned} E(y_{ij}y_{i+k,j}) &= E[(x_{ij}-x_{i+1,j})(x_{i+k,j}-x_{i+k+1,j})] \\ &= E(x_{ij}x_{i+k,j}-x_{ij}x_{i+k+1,j}-x_{i+1,j}x_{i+k,j}+x_{i+1,j}x_{i+k+1,j}) \\ &= E(x_{ij}x_{i+k,j})-E(x_{ij}x_{i+k+1,j})-E(x_{i+1,j}x_{i+k,j})+E(x_{i+1,j}x_{i+k+1,j}) \\ &\approx 0 \end{aligned} \tag{24}$$

According to (22), we know

$$\sigma_y^2 \neq 0 \tag{25}$$

So that we have

$$\rho(y_{ij}, y_{i+k,j}) \approx 0 \tag{26}$$

Thus (5) is derived.

Performance and Robustness Analysis for Some Re-sampling Detection Techniques in Digital Images

Hieu Cuong Nguyen and Stefan Katzenbeisser

Computer Science Department, Darmstadt University of Technology, Germany
cuong@seceng.informatik.tu-darmstadt.de

Abstract. In order to create convincing forged images, manipulated images are usually exposed to some geometric operations which require a re-sampling step. Therefore, detecting traces of re-sampling became an important approach in the field of image forensics. There are many re-sampling detection techniques described in the literature, but their performance has been often examined with a small dataset. Besides, performance and robustness have been tested under different conditions, so it is difficult to evaluate and compare them. In this paper, we analyze the performance of some selected re-sampling detection techniques by using a common testing framework and a large dataset. We also employ several kinds of image post-processing to defeat the detectors in order to evaluate their robustness and security. We show that the tested techniques obtain the best results when detecting up-sampled images. Unfortunately, most techniques are not secure and they can be defeated on different levels by applying post-processing operations to the forged images.

Keywords: Image forensics, re-sampling detection, robustness, security.

1 Introduction

Nowadays, digital images can easily be altered without leaving visual evidence due to the availability of powerful digital image processing tools. Therefore, developing techniques for deciding on image authenticity became an urgent need. There are many different types of image tampering, which can be detected by different forensic methods [1-3].

In order to create convincing forged images, manipulated images usually undergo geometric transformations, which involve a re-sampling step. Thus, detecting re-sampling traces is a popular approach in the field of image forensics. Techniques that detect re-sampling artifacts are often based on analyzing local linear predictors [4-6] or the variance of the second derivatives of images [7-9]. To evaluate the performance of the techniques, in [4] the authors used 200 uncompressed images as the original dataset, and selected 50 images to create re-sampled versions. The number of tested images in [5] and [6] is 200 and all of them have been used to produce re-sampled images. The techniques in [7] were tested with only one image, while in [8] used 114, and [9] used 40 images.

Y.Q. Shi, H.J. Kim, and F. Perez-Gonzalez (Eds.): IWDW 2011, LNCS 7128, pp. 387–397, 2012.

Robustness is another important characteristic of re-sampling detection techniques along with detection capacity. A forensic technique is called robust if it is able to detect forged images even after post-processing is applied to the forged images. To evaluate the robustness of a detection technique, the authors usually apply several post-processing operations, such as Gaussian noise addition or JPEG compression to re-sampled images. The purpose of the operations is to defeat the technique so that it declares a post-processed forged image as authentic. All authors in the aforementioned papers used simple image processing operations. However, more complex attacks can be defined which are tailored towards attacking a specific technique. Recently, Kirchner and Boehme [10] proposed some methods to hide traces of re-sampling, and Stamm et al. [11, 12] offered anti-forensic operations capable of disguising key evidence of JPEG compression.

It is obviously difficult to judge which detection technique is better since they were evaluated on different datasets under different testing conditions. To fill this gap, Uccheddu et al. [13] proposed an experimental methodology and applied it to evaluate and compare two re-sampling detection techniques of Kirchner and Gloe [6] and Mahdian and Saic [9]. In the paper, the authors used a dataset of 200 images in different categories; both analyzed techniques were tested following the same framework. However, in the performance test they considered only re-scaled images and in the robustness test, they limited their study to JPEG compression.

In this paper, we study and implement three widely used re-sampling detection techniques of Gallagher [8], Mahdian and Saic [9], and Popescu and Farid [4]. Next, we test the techniques with a common large dataset under the same conditions. Subsequently, we apply post-processing to the re-sampled images in order to evaluate the robustness of the techniques. We show that these techniques obtain the best results when detecting up-sampled images. The techniques in [4, 9] can also work with rotated images with high degrees, but the technique in [8] should not be applied to rotated images. We also show that all techniques are easily defeated by even simple post-processing operations that aim at disguising a forged image as authentic.

The structure of the paper is as follows. In the next section, we review the implementation of the considered re-sampling detection techniques. In Section 3, we analyze the performance and in Section 4 we evaluate the robustness of the selected techniques. Lastly, we conclude the paper.

2 Implementation of Detection Techniques

The first barrier for testing re-sampling detection techniques is that their implementations are often not available. Therefore, we implemented the algorithms according to the description in published papers. In this section, we review the techniques in [4, 8, 9] and details on their implementation.

Gallagher realized that low-order interpolated signals (bilinear and bicubic interpolations) introduce periodicity in variance function of their second derivatives with a period that is equal to the re-sampling factor [8]. This observation can be used to detect whether an image has been re-sampled. Specifically, the periodicity is examined by computing the discrete Fourier transform (DFT) of the second derivatives of the analyzed signals. In image forensics, the signals are rows (or columns) of the analyzed image.

Although Mahdian and Saic [9] proved more generally that the variance of the *n*-derivative of a re-sampled signal is also periodic, they used only the second derivative in their experiments. This detection algorithm consists of the following steps. Firstly, in a similar way as [8], the second derivative of the analyzed signal is calculated. Next, the Radon transformation is employed to compute projections of magnitudes of the second derivative along specified directions determined by angles (from 0° to 179° in 1° increments). The authors apply this algorithm to every row (or column) of the examined image. The implementation of the core part of the technique is available on the website of the authors [14].

The algorithm of Popescu and Farid [4] is probably the most widely used method. The authors noted that there are correlations between neighbouring pixels only in re-sampled images. In order to determine the correlations, they employed the expectation/maximization (EM) algorithm [15]. The EM algorithm estimates the linear correlation between each pixel and its neighbours, then computes the probability of each pixel being correlated to its neighbours. The main output of the algorithm is the correlation probability map (p-map) which contains periodic patterns in re-sampled images. The corresponding p-map will be more pronounced by transforming into frequency (DFT) domain. In our implementation, we used to the EM function of Nataraj et al. [16].

To quantify the performance and robustness, we implement automatic testing functions for all techniques. In these techniques, the images to be analyzed are transformed to frequency domain by applying DFT in order to uncover interpolation artefacts in the form of peaks. We use a strict threshold-based peak detector for searching local maximum (peaks) in the frequency domain. Since there is a trade-off between detection rate and false positive rate (FPR), the threshold has been chosen carefully through experiments. In this work, we use a simple method to support choosing thresholds. The method is as follows:

1. For each image in the analyzed dataset, apply the detection techniques to produce a representation of the image in frequency (DFT) domain.
2. In the frequency domain, find the highest peak, which is evidence of re-sampling.
3. Iterate the steps 1 and 2 for all images in the analyzed dataset then gather the highest peaks of all images in the dataset and draw their histogram.
4. Iterate the above steps for different re-sampled datasets and an original dataset. Subsequently, we have a set of histograms. These histograms are useful in choosing a threshold which satisfies the trade-off between detection rate and FPR.

Based on the above method, in our test we defined thresholds for each technique so that its detection rate is larger than 90% in the test with up-sampled images (with the scaling factor of 1.2), and its FPR is lower than 20% in the test with the original images. Specifically, by using the thresholds, the detections rates of all analyzed techniques are larger than 90%, and the FPR of the techniques of Gallagher [8], Mahdian and Saic [9], Popescu and Farid [4] are 18.5%, 19%, and 6.5% respectively. As an effort to reduce the FPR for the techniques in [8, 9] to below 10% by adjusting the thresholds which they used, we found that their detection rates were decreased significantly, so we missed many forgeries. The reason of higher FPR is that many false positives were caused by strong textures. Since the techniques in [8, 9] are based on detecting the second derivatives of images, strong textures produce clear periodicity, even in original images, so they can yield peaks in the frequency spectrum similar to re-sampled images.

Moreover, since all of these techniques are statistical methods, their test results are affected by the dataset in use. Therefore, we use a dataset of both 256 × 256 images and 128 × 128 images. In this paper, when we do not specify the size of images, we work with images of size 256 × 256 pixels.

3 Performance Analysis of Detection Techniques

To assess the performance and robustness of detection techniques correctly, we prepare a large image dataset. In this work, we use the same dataset in order to allow a fair comparison. First of all, to create a dataset of original images, we collected randomly 200 uncompressed images from [17], then converted them to gray-scale, and cropped each of them to the size of 256 × 256 pixels. Next, we created different datasets of up-sampled, down-sampled, and rotated images with different factors. Similarly, we created a dataset of 200 images with the size of 128 × 128 pixels.

We evaluate the performance of the techniques by testing them with different kinds of re-sampled images. Firstly, we test the techniques with up-sampled images. All rescaled images are created from the original image dataset by using *imresize* function of Matlab using bicubic interpolation. We found that the techniques can detect up-sampled images with the scaling factor larger than 1.1 rather well (with a detection rate larger than 50%). They detect perfectly (with a detection rate of about 100%) re-sampled images of size 128 with a scaling factor larger than 1.3, and re-sampled images of size 256 with the scaling factor larger than 1.2. Gallagher [8] showed that in the special case of interpolation by a factor of 2.0, there are no meaningful peaks produced in normalized frequency. Through experiments, we show that when the scaling factor is equal to 2.0, not only the technique of Gallagher, but also all other analyzed techniques did not recognize any critical periodic pattern, so the detection rate of all techniques becomes very low.

The experimental results for detecting up-sampled images of size 128 × 128 and 256 × 256 pixels are presented in Fig. 1 and Fig. 2. Since the detection techniques are based on statistics, using lager images for testing, we apparently get stronger and more accurate detection results.

In the same way of testing up-sampled images, we test down-sampled images with different scaling factors. We realized that the detection rate of the techniques in detecting down-sampled images is rather low (see Fig. 3). The reason is that down-sampling causes loss of information, thereby limiting the detection capacity of the statistical-based techniques.

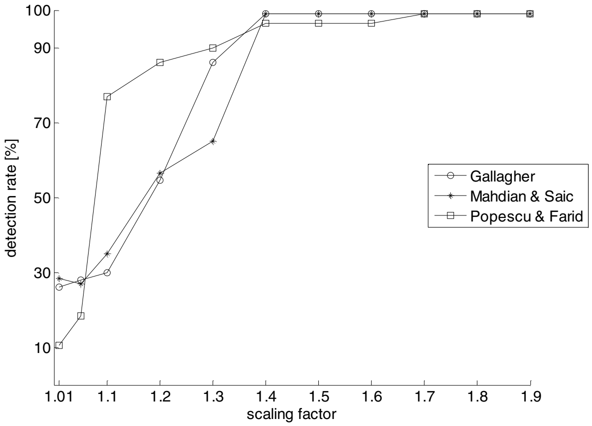


Fig. 1. Detection rate for 128 × 128 up-sampled images

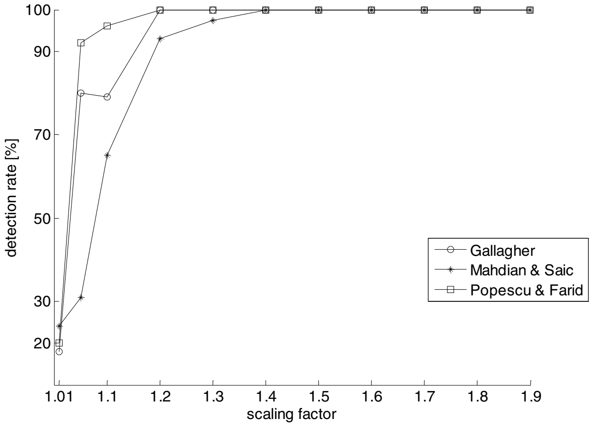


Fig. 2. Detection rate for 256 × 256 up-sampled images

Following the test with up-sampled and down-sampled images, we evaluate the detection techniques on rotated images with different angles. All rotated images were created from the 256 × 256 original image dataset by using the *imrotate* function of Matlab with bicubic interpolation. In order to reject the black parts in the corners of the rotated images, we crop the image and keep only the center part of size 196 × 196 of each rotated image for evaluation. The experimental result is presented in Fig. 4. We realize that the technique in [4] can detect rotated images (with a rotation angle larger than 5 degree) and has a detection rate is about 80%. The techniques are based on detecting the second derivatives of images [8, 9] are not robust against rotation. Since the Radon transformation is applied to the second derivatives in [9], its detection rate is higher than in [8].

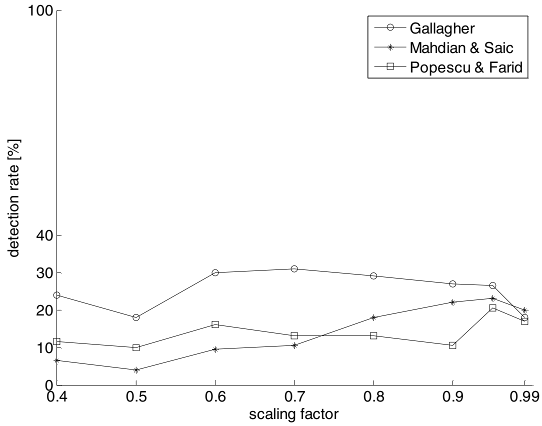


Fig. 3. Detection rate for 256 × 256 down-sampled images

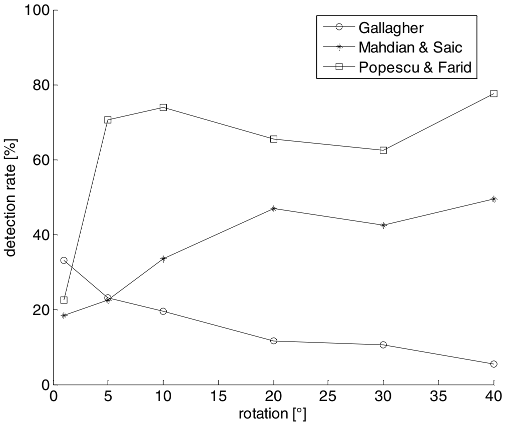


Fig. 4. Detection rate for rotated images

4 Robustness Analysis of Detection Techniques

To make tampering more convincing, post-processing is commonly applied to re-sampled images. However, post-processing often worsen the performance of detection techniques. Thus, in order to measure the robustness of the techniques, we employ different post-processing operations to the re-sampled images. We choose 200 up-sampled images with the factor of 1.2, where the detection rates were very high for all techniques. Specifically, the detection rates of the techniques for the up-sampled images are larger than 70% in the case of image size is 128 × 128 pixels and larger than 90% in the case of image size is 256 × 256 pixels.

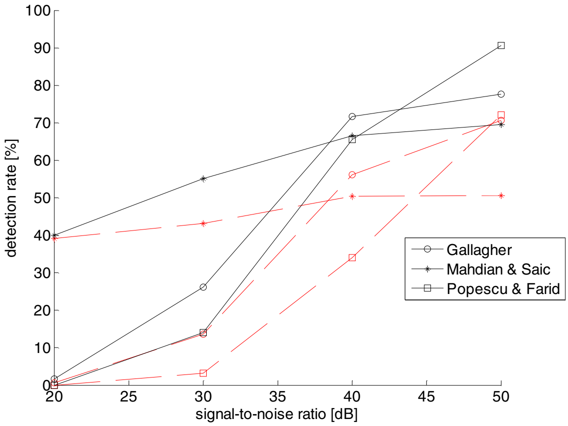


Fig. 5. Detection rate for up-sampled images with post-processing by adding Gaussian noise (solid lines and dashed lines present results for 256 × 256 images and 128 × 128 images)

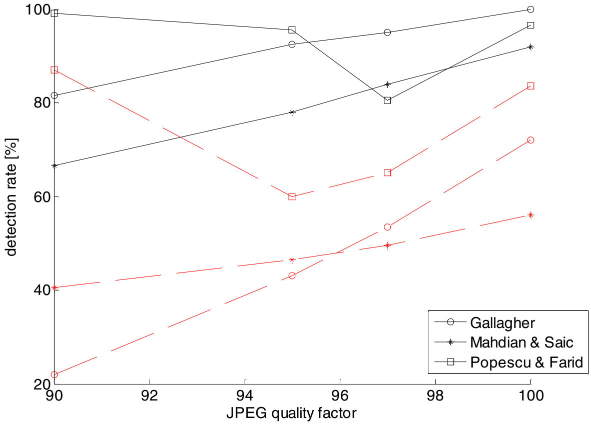


Fig. 6. Detection rate for up-sampled images with post-processing by JPEG compressing (solid lines and dashed lines present results for 256 × 256 images and 128 × 128 images)

We applied some post-processing operations such as Gaussian noise addition and JPEG compression to the up-sampled images (the scaling factor is 1.2). The detection results after the post-processing are presented in Fig. 5 and Fig. 6. While the techniques in [4, 8] are defeated in case of SNR (Signal to Noise Ratio) of the added Gaussian noise is lower than 30 dB, the technique in [9] is more robust. All techniques are more robust against adding Gaussian noise with higher SNR. The technique of Gallagher in [8] works by detecting patterns in pixel-wise differences, so it is obviously sensitive to noise, but it is more robust against JPEG compression than the technique of Mahdian and Saic [9]. However, JPEG compression creates blocking artifacts and they bring periodical peaks in normalized frequency. These peaks create false positives in the detection results and thus the detection rates are sometimes growing inversely to the quality factors. In order to quantify the robustness of the considered techniques, the test results are shown in detail in Table 1 and Table 2.

Table 1. Detection rate for up-sampled images without any post-processing [%]

	Gallagher [8]	Mahdian and Saic [9]	Popescu and Farid [4]
128 × 128 images	72.5	71.5	86.0
256 × 256 images	100	93.0	100

Table 2. Detection rate for up-sampled images with post-processing [%]

	Gallagher [8]	Mahdian and Saic [9]	Popescu and Farid [4]
SNR 20	1.5	40.0	0
SNR 30	26.0	55.0	14.0
SNR 40	71.5	66.5	65.5
SNR 50	77.5	69.5	90.5
JPEG 90	81.5	66.5	99.0
JPEG 95	92.5	78.0	95.5
JPEG 97	95.0	84.0	80.5
JPEG 100	100	92.0	96.5

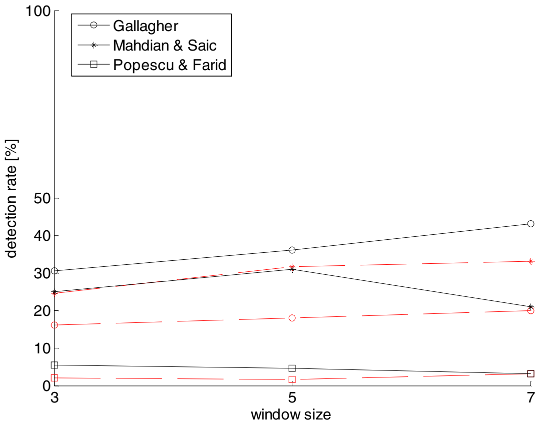


Fig. 7. Detection rate for up-sampled images with post-processing by median filtering (solid lines and dashed lines present results for 256 × 256 images and 128 × 128 images)

In light of [10], we also apply median filtering as a post-processing operation to the re-sampled images. In the original papers [4, 8, 9], the authors have not considered median filtering. However, during our experiments, we identify it as an effective attack against the re-sampling detectors. Since the median filter is non-linear, it defeats well the techniques based on detection of local linear dependency [4]. The experimental results of the techniques under test with re-sampled images are presented in Fig. 7. Some examples of applying Gaussian noise and median filter to an image selected from [17] are shown in Fig. 8 and Fig. 9. Based on the analysis, we recommend applying these operations to attack the re-sampling detection techniques. In order to satisfy the trade-off between the attack effectiveness and the quality of the attacked images, we suggest using Gaussian noise with SNR of 30 or median filtering with window size of 3. Details on the detection results are shown in Table 3. Although the techniques of Gallagher [8] and Mahdian and Saic [9] seem more robust against some attacks than the technique of Popescu and Farid [4], we noted that [8, 9] have much higher FPR than [4] with the fixed thresholds we chose for tests.

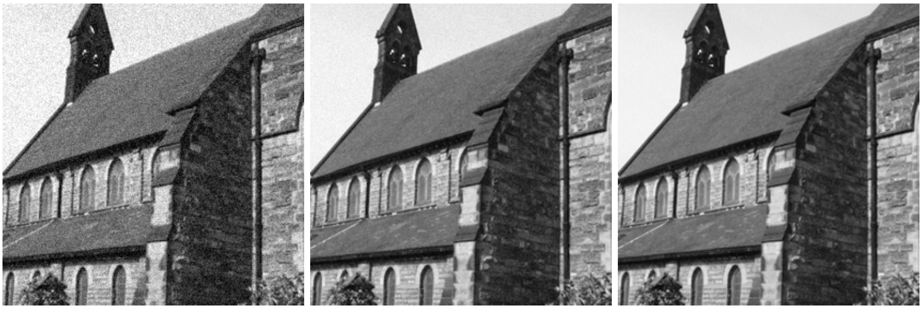


Fig. 8. An up-sampled image and adding noise by signal-to-noise ratio (SNR) of 20, 30, 40

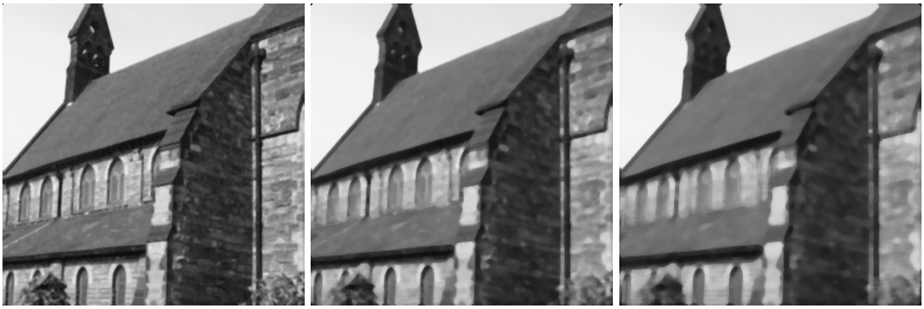


Fig. 9. An up-sampled image and median filtering by size of 3, 5, 7

Lastly, we propose a new attack by using order-statistic filtering. The filter replaces each pixel in a re-sampled image by the third largest value of the pixel among its north, east, south and west neighbours. We use this filter to attack up-sampled images with a scale factor of 1.2, and then we use the techniques [4, 8, 9] to detect the attacked images. In Table 3, we show the efficiency of the attack in comparison with other considered attacks.

A good attack not only reduces the detection rates of detection techniques, but also keeps the image quality. To quantify this factor of an attack, we compute the average difference between pairs of re-sampled images (before the attack) and attacked re-sampled images (after the attack). The difference between a pair of image can be measured by calculating the PSNR (Peak Signal to Noise Ratio). A higher PSNR normally indicates that the attacked image is of higher quality. In Table 4 we show the average difference between re-sampled dataset (without any attack) and its attacked versions. We realize that the order-statistic filter is better than a median filter for attacking the technique of [4]. It should be noted that, although median filtering is an effective attack to re-sampling detectors, it may leave evidence which can be detected and thus reveal the existence of the attack [18].

Table 3. Detection rate for up-sampled images after attack [%]

Post-processing/ Attacks	Gallagher [8]	Mahdian and Saic [9]	Popescu and Farid [4]
Gaussian noise with SNR= 30	26.0	55.0	14.0
Median filter with size= 3	30.5	25.0	5.5
Order-statistic filter	71.0	39.0	4.0

Table 4. Average difference between re-sampled images and attacked re-sampled images [dB]

Gaussian noise with SNR= 30	Median filter with size= 3	Order-statistic filter
23.3	18.4	21.0

5 Conclusion

Image re-sampling detection is an important problem in the field of image forensics. In this paper, we have experimentally tested the performance and robustness of three detection techniques of Gallagher [8], Mahdian and Saic [9], and Popescu and Farid [4] on a large image dataset. The performance and robustness of the techniques can be measured by evaluating their detection rates in tests on re-sampled images and post-processed images respectively. We have noted that all tested techniques can detect up-sampled images well, but they also easily be defeated by post-processing operations. Based on our test results, the technique of Popescu and Farid seems to be the most reliable, but it can be almost completely defeated by median filtering or by order-statistic filtering.

References

1. Sencar, H.T., Memon, N.: Overview of State-of-the-Art in Digital Image Forensics. In: Proc. WSPC (2007)
2. Farid, H.: Image forgery detection. IEEE Signal Processing Magazine 26, 16–25 (2009)
3. Mahdian, B., Saic, S.: A bibliography on blind methods for identifying image forgery. Signal Processing: Image Communication 25, 389–399 (2010)

4. Popescu, A., Farid, H.: Exposing digital forgeries by detecting traces of resampling. IEEE Transactions on Signal Processing (2005)
5. Kirchner, M.: Fast and reliable resampling detection by spectral analysis of fixed linear predictor residue. In: Proceedings of the 10th ACM Workshop on Multimedia and Security, MM&Sec 2008, vol. 11 (2008)
6. Kirchner, M., Gloe, T.: On resampling detection in re-compressed images. In: WIFS, pp.21-25 (2009)
7. Prasad, S.: On resampling detection and its application to detect image tampering. Dept. of Electrical Engineering, IIS Bangalore, India. 1325-1328 (2006)
8. Gallagher, A.C.: Detection of Linear and Cubic Interpolation in JPEG Compressed Images. In: The 2nd Canadian Conference on Computer and Robot Vision (CRV 2005), pp. 65-72 (2005)
9. Mahdian, B., Saic, S.: Blind Authentication Using Periodic Properties of Interpolation. IEEE Transactions on Information Forensics and Security 3, 529–538 (2008)
10. Kirchner, M., Boehme, R.: Hiding Traces of Resampling in Digital Images. IEEE Transactions on Information Forensics and Security 3, 582–592 (2008)
11. Stamm, M., Tjoa, S., Lin, S., Liu, J.: Anti-Forensics of JPEG Compression. In: ICASSP (2010)
12. Stamm, M., Tjoa, S., Lin, S., Liu, K.J.R.: Undetectable image tampering through JPEG compression. In: ICIP 2010 (2010)
13. Uccheddu, F., Rosa, A.D., Piva, A., Barni, M.: Detection of resampled images: performance analysis and practical challenges. In: EURASIP, pp. 1675–1679 (2010)
14. http://zoi.utia.cas.cz/files/rsmp_core.txt
15. Dempster, A., Laird, N., Rubin, D.: Maximum Likelihood from Incomplete Data via the EM Algorithm. Journal of the Royal Statistical Society. Series B (Methodological) 39, 1–38 (1977)
16. Nataraj, L., Sarkar, A., Manjunath, B.S.: Improving re-sampling detection by adding noise. Computer. 75410I-75410I-11 (2010)
17. Schaefer, G.: UCID: an uncompressed color image database. In: Proceedings of SPIE, pp. 472–480 (2004)
18. Kirchner, M., Fridrich, J.: On detection of median filtering in digital images. Computer. 754110-754110-12 (2010)

Alternative Anti-Forensics Method for Contrast Enhancement

Chun-Wing Kwok, O.C. Au, and Sung-Him Chui

The Hong Kong University of Science and Technology
Department of Electronic and Computer Engineering

Abstract. Digital image forensic researchers have proposed different robust detection schemes to classify authentic images and contrast enhanced images. The main idea of the detectors is based on the peaks and gaps introduced in the histogram after performing contrast enhancement on an image. The classifier using these peak-gap artifacts as feature achieves high accuracy result. After that, Anti-forensic researchers proposed a method to remove the peak-gap artifacts by local random dithering method which significantly reduces the accuracy of current detection scheme. In this paper, an alternative anti-forensic method is proposed to avoid the peak-gap artifacts and still have good image quality in terms of PSNR. We perform the experiment by calculating PSNR between traditional contrast enhanced images and our result images.

Keywords: Security, Anti-Forensics, Digital Forensics, contrast enhancement.

1 Introduction

Since currently photo editing software such as GIMP and Adobe Photoshop are available to many computer users, images can be easily modified. This kind of software provides lots of functionalities to enhance and manipulate images easily without any visual artifact. However, in recent years, digital images have becomes more important in many areas, such as newspapers, evidence in court which the integrity of images is important. To protect the authenticity of digital images, image forensic techniques are required.

There are two ways to authenticate an image: *1) watermarking* and *2) image forensic*. The watermarking method is an active method whereby prior information is added to the image explicitly so that authenticity of image can be easy by extracting the watermarking from the image. However, this kind of method needs the acquiring-devices containing this ability. However, this is not available in most cameras nowadays. Thus, image forensic is further developed to authenticate image "passively" which is to authenticate an image with the intrinsic fingerprint. In general, there are many digital image manipulation forensics schemes have been proposed. [4] developed a detector to check image splicing. [1] proposed a robust method to find the copy-move area within an image. [7] and [6] developed a classifier to find re-sampling in an image.

Y.Q. Shi, H.J. Kim, and F. Perez-Gonzalez (Eds.): IWDW 2011, LNCS 7128, pp. 398–410, 2012.

In [9] and [10], the blind forensic algorithms for detecting the globally and locally applied contrast enhancement have been proposed. They perform contrast enhancement detection by finding out unique peak-gap artifacts introduced into the histogram.

In this paper, we present an alternative method to attack the detection scheme developed by [9]. The idea is based on internal bit depth increment to model continuous input pixel intensity value. The reason for proposing anti-forensic method is not to allow people to fool the current detection scheme, rather we hope current forensic researchers can developer a better and more robust classification and feature later.

The rest of this paper is organized as follows: Section 2 briefly introduces the basic idea of contrast enhancement manipulation and the current forensic method for contrast enhancement, followed by the existing anti-forensic algorithms in Section 3. Then the proposed anti-forensic method are introduced in Section 4. In Section 5, the experimental settings and results are shown. Finally we draw the conclusion in Section 6.

2 Forensic Detection of Contrast Enhancement

In the prior works [9] and [10], blind forensic algorithms for detecting and estimating contrast enhancement on digital images have been proposed. Contrast enhancement is a mapping for every pixel values in an image. Let $\Omega = \{0, 1, \ldots, 255\}$ denotes the set of pixel values for 8-bit images, each pixel value $x \in \Omega$ in the authentic image is then mapped to a pixel value $y \in \Omega$ in the contrast-enhanced image by the mapping function m, such that

$$y = m(x) \tag{1}$$

In practice, m might be gamma correction mapping, the histogram equalization method which spreads the histogram to be uniform.

The authors in [9] found that the histogram of an authentic image, $h_X(x)$ is smooth since there are many factors affecting the images, e.g. CFA interpolation, lighting and shading environments, CCD noise, etc. After contrast enhancement, the histogram of output $h_Y(y)$ are changed according to the mapping m. In general, contrast enhancement should be 1-to-1 mapping as there is no sense to map two values to be the same. However, in the digital world, values need to be quantized to nearest integer value. The histogram $h_Y(y)$ of pixel values in the enhanced image can be expressed by the unaltered image's pixel value histogram $h_X(x)$ as follows,

$$h_Y(y) = \sum_{x \in \Omega} h_X(x) l(m(x) = y) \tag{2}$$

where $l(\cdot)$ denotes the indicator function. This equation shows that each value of h_Y must equal to either a single value of h_X, a sum of distinct h_X values, or none if there is no pixel intensity mapped. As a result, impulsive peaks will occur in h_Y at y values when multiple x values are mapped. Gaps are generated in h_Y at y values when no x values are mapped [9]. Such peak and gap artifacts can be visualized in Figure 1 in which the top row is an authentic image and its histogram while the bottom row is a gamma corrected image with its histogram. These artifacts will serve as the feature that is used to identify contrast enhancement operation.

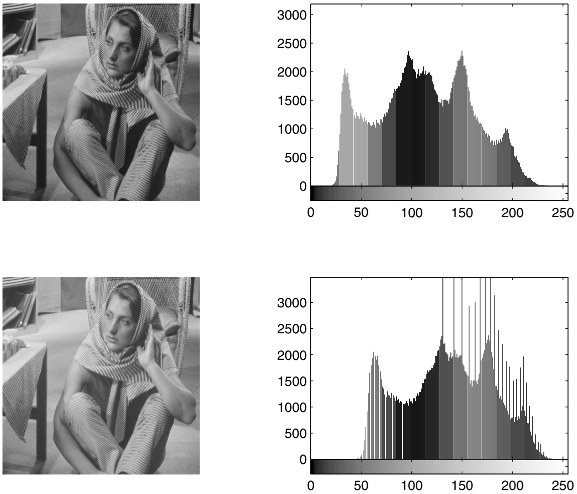


Fig. 1. Top left: original image, Top right: Histogram of original images, Bottom left: gamma corrected image with $\gamma = 0.7$, Bottom right: histogram of gamma corrected image

In [9], the authors found that histograms should be smooth which means no sudden gaps and peaks appears. Thus, the Fourier transform of the histogram should contain lots of low energy while high energy part should be small. The contrast enhancement detection algorithm proposed in [9] is summarized as follows:

1. Obtain the image's histogram $h(x)$.
2. Calculate $g(x)$ as follows,

$$g(x) = p(x) \cdot h(x) \tag{3}$$

   where $p(x)$ is a pinch off function to remove the high frequency effect caused by the high end and low end saturated images.
3. Transform $g(x)$ to the Fourier frequency domain, $G(\omega)$, and obtain the high frequency measure F according to

$$F = \frac{1}{N} \sum_{\omega} |\beta(\omega) G(\omega)| \tag{4}$$

Here the cutoff function $\beta(\omega)$ used to compute the weight of high frequency component is simply designated as

$$\beta(\omega) = \begin{cases} 1 \ |\omega| \geq c \\ 0 \ |\omega| < c \end{cases} \tag{5}$$

where c is a user specified cutoff frequency.

4. Apply the threshold test method to determine if contrast enhancement has been performed.

3 Existing Anti-Forensic Method

In this section, we briefly describe the anti-forensic method developed by [2]. They apply the basic idea of dithering and design an anti-forensic scheme to attack current forensic method. The traditional contrast enhancement operation involves two steps. First, the primary mapping function $m_0(\cdot)$ would transform the integer pixel value $x \in \Omega$ to be a real number:

$$y_0 = m_0(x) \tag{6}$$

Secondly, the resulting pixel value y is obtained by rounding y_0,

$$y = [x] \tag{7}$$

where $[\cdot]$ is the rounding function which maps a real number to the nearest integer value. Combining equation 6 and equation 7, result in $m(x) = [m_0(x)]$.

According to this formulation, the histogram of the output images can be written as,

$$h_Y(y) = \sum_{x \in \Omega} h_X(x) l(y_0 \in [y - \frac{1}{2}, y + \frac{1}{2}]) \tag{8}$$

In this formula, gaps and peaks are generated when no gray level or multiple gray levels are mapped into the unit neighboring range of y. In [2], they perform local random dithering to avoid generating gaps and peaks in the histogram. Therefore in the pixel domain,

$$y = [m_0(x) + n_x] \tag{9}$$

where $n_x \sim N(0, \sigma_x^2)$. As mentioned in the paper [2], in order to avoid detection, variance of the dithering noise should be adjusted with the original pixel values. That means variance is larger for whose pixel value has higher probability to have peaks or gaps.

4 Proposed Anti-Forensic Method

The paper [2] used the local random dithering technique when performing contrast enhancement which can remove the gaps and peaks. However, some problems are introduced: 1) the paper does not specify how to select an appropriate variance for different intensity values. 2) After local random dithering with Gaussian noise, the resulting histogram conforms the Gaussian mixture model [11].

When omitting the rounding function in equation 9,

$$h_Y = h'_X * h_N \tag{10}$$

where h_N is the Gaussian function which low-passes the histogram h'_X, the histogram after contrast enhancement. As result, the high-frequency energy will be suppressed by the noise. Instead of removing the peaks-gaps artifacts, we try to prevent from generating these artifacts. This work is inspired by camera acquisition and raw-image editing. When capturing photos using the camera, the image undergoes some image processing before being saved into the memory. Those image processing methods include noise filtering, gamma correction, contrast stretching, white balance, etc. However, even if an image undergoes contrast enhancement, the histogram of that image still be smooth. This is similar to raw-image editing, as users like to process images in raw format because the image contains more intensity precision.

In practice, m is usually monotonically increasing. We realised that most mapping should be 1-to-1 mapping, given that the input domain and output domain are continuous, since it is not likely to mapping different intensity values to a same one. The gaps and peaks are generated when we quantize the output into integer value so that there might be multiple values mapped into the same integer value.

To solve this problem, we increase the bit depth to simulate higher precision by the internal bit depth increase method (IBDI) [5]. Internal bit depth increase is currently used for video coding and it shows that it can increase the coding efficiency. In this paper, we creatively apply this method into our problem. First, we increase the bit depth for each pixel by

$$\tilde{f}_X(x,y) = f_X(x,y) + n \tag{11}$$

where f_X is the input image, $n \sim$ uniform(−0.5,0.5). We denote higher bit depth value as $\tilde{f}$. We represent the extra bit depth information in the floating point precision. The reason for choosing n to be uniform distribution is that we assume that quantization error is uniform. We decrease 1000 12-bit images to 8-bit images and calculate the error due to bit-depth decrease, the distribution is shown in Figure 2.

After increasing the bit depth of image f_X, we perform contrast enhancement as usual:

$$\tilde{f}_Y(x,y) = \tilde{m}(\tilde{f}_X(x,y)) \tag{12}$$

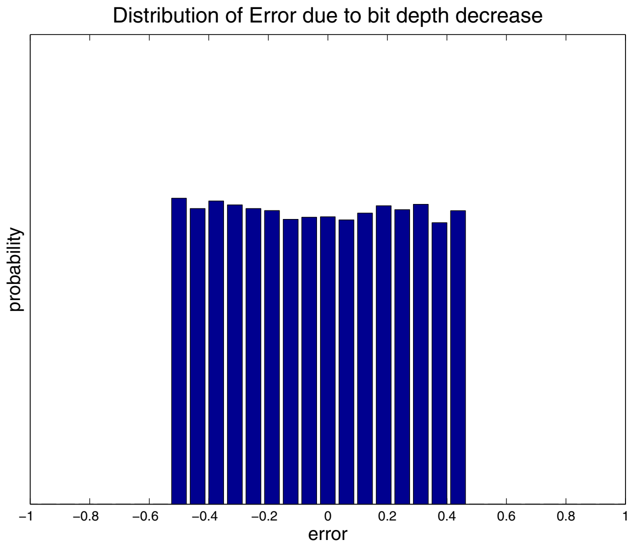


Fig. 2. The distribution of quantization error

where $\tilde{m}$ is the contrast mapping function in higher bit depth and $\tilde{f}_Y$ is the bit depth increased result image. $\tilde{m}$ can be obtained by interpolating or polynomial approximating of m.

Finally, we perform rounding to obtain the same bit depth as the input image.

$$f_Y(x,y) = [\tilde{f}_Y(x,y)] \tag{13}$$

5 Experiment and Results

5.1 Test Data Set

We test our method with a database of 1582 uncompressed and unaltered photographic images downloaded from [8] for a quantitative evaluation. The test images include natural scenes, buildings and different varieties of objects. These images are stored in TIFF format with sizes ranging from 384 × 512 to 480 × 640 pixels. In our experiment, for the color images, color are converted to gray scale images using the following equation,

$$Y = 0.299R + 0.587G + 0.114B$$

for every pixel. Then, the contrast-enhanced gray scale images are created by applying the gamma correction:

$$\tilde{m}(s) = \left[256\left(\frac{s}{256}\right)^{\gamma}\right]$$

where γ value ranges from 0.5 to 1.5 in our experiment.

5.2 SVM Configuration

For classification, we use the LIBSVM Support Vector Machine library [3] for our classifier which classifies authentic images and contrast-enhanced images. We set the SVM parameter as follows,

SVM Type	C-SVC
Kernel Type	Radial Basis function ($e^{\gamma\|u-v\|^2}$)
Error Cost	obtained from a 5-fold grid-search cross-validation process
Gamma	

5.3 Results

In Fig. 3, the top row is the authentic image and its histogram. The second row is, from left to right, the image after gamma correction with $\gamma = 0.7$ and its histogram, method in [2] with $\gamma = 0.7$ and its histogram, followed by the proposed method with $\gamma = 0.7$ and its histogram. The third row is the experiment with $\gamma = 1.2$. It is clearly shown that the F value in the original image is small compared with the contrast-enhanced image. In Fig. 3, both method in [2] and ours can remove the gaps and peaks in the histogram while our method obtains higher PSNR.

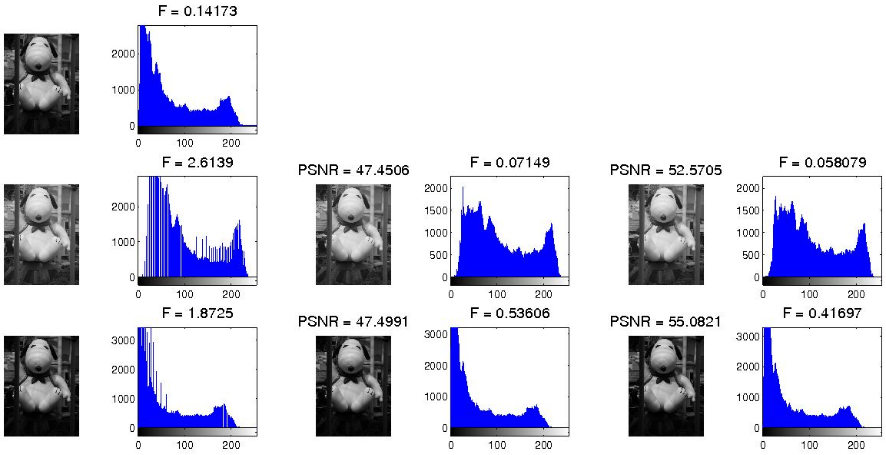


Fig. 3. Top row: original image and its histogram (F = 0.142). Middle row: gamma corrected images with $\gamma = 0.6$ and its histogram (F = 2.614), image using the method in [2] and its histogram (PSNR= 47.5dB, F = 0.071), image using proposed method and its histogram (PSNR=52.6dB, F = 0.058). Bottom row: gamma corrected image with $\gamma = 1.2$ and its histogram (F = 1.87), image using method in [2] and its histogram (PSNR = 47.5dB, F = 0.536), image using proposed method and its histogram (PSNR = 55.1dB, F = 0.417).

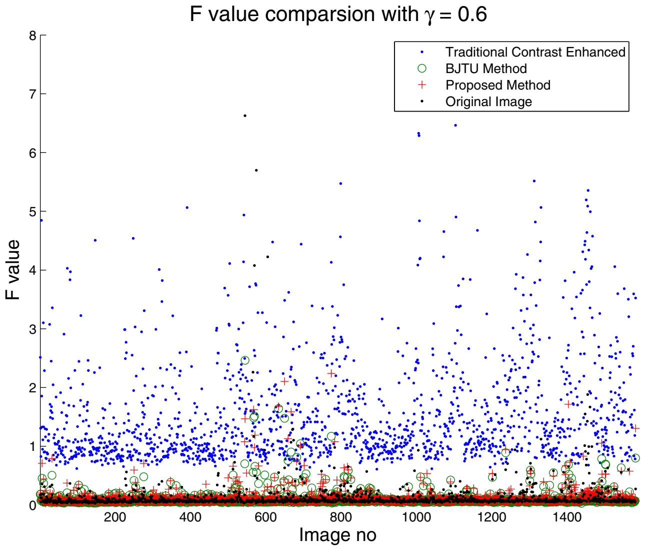


Fig. 4. F values comparison of contrast enhancement with $\gamma = 0.6$

Undetectability. We compute F value in eq. 4 for each image as a feature for classification. In Fig. 4 and Fig. 5, they show the F value of Traditional Contrast Enhancement, BJTU method, proposed method and original images with $\gamma = 0.6$ and $\gamma = 1.2$, respectively. It shows that both anti-forensic methods achieve similar F values as original images, thus it is hard to use the F value to classify the original images and the anti-forensic images.

We use ROC to show the undetectability of our anti-forensic method also. In Fig. 6, this is the ROC of the detection rate of the traditional contrast enhanced method. In Fig. 7 and Fig. 8, it is clear to see that both the BJTU method and our method decrease the accuracy of the current forensic method. The following table shows the AUC with different methods, the AUC value, the closest to 1, means the classifier can perfectly classify all data, while the AUC value which is closest to 0.5 means the classifier cannot classify based on the feature. Thus, in our experiment, the closest to 0.5, the better it is.

	average AUC									
Method	$\gamma = 0.5$	$\gamma = 0.6$	$\gamma = 0.7$	$\gamma = 0.8$	$\gamma = 0.9$	$\gamma = 1.1$	$\gamma = 1.2$	$\gamma = 1.3$	$\gamma = 1.4$	$\gamma = 1.5$
Traditional	0.991	0.991	0.987	0.981	0.979	0.976	0.984	0.989	0.992	0.993
[2]	0.751	0.516	0.555	0.565	0.552	0.484	<u>0.561</u>	0.638	<u>0.693</u>	<u>0.751</u>
Proposed	<u>0.519</u>	<u>0.490</u>	<u>0.501</u>	<u>0.480</u>	<u>0.496</u>	<u>0.493</u>	0.579	<u>0.637</u>	0.718	0.772

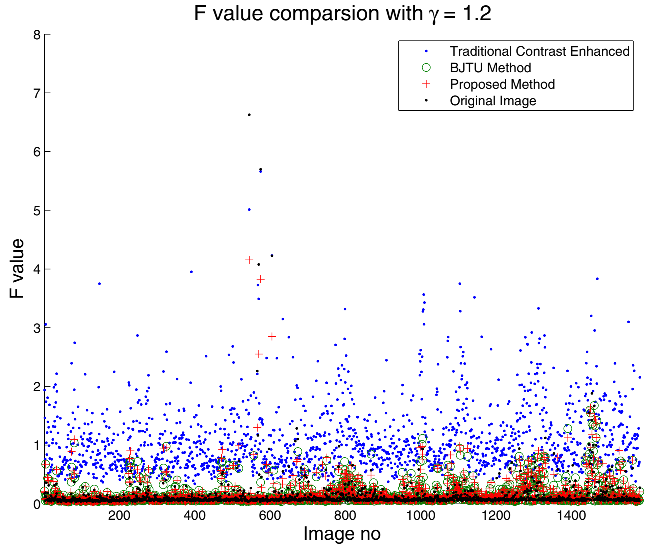


Fig. 5. F values comparison of contrast enhancement with $\gamma = 1.2$

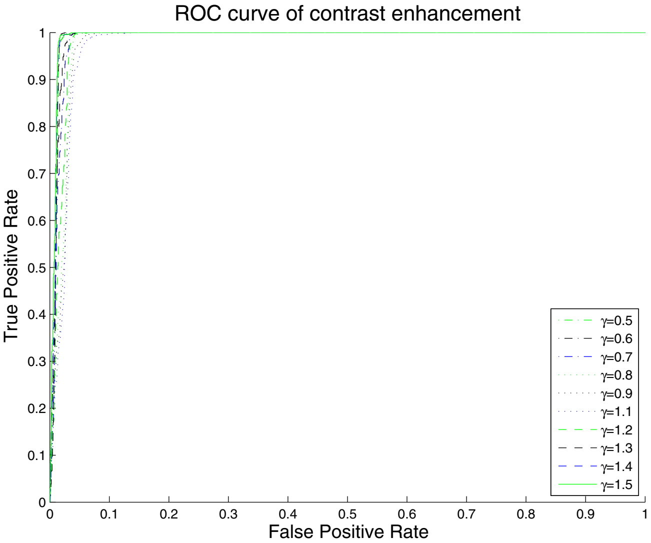


Fig. 6. ROC of contrast enhancement with $\gamma = 0.5, 0.6, 0.7, 0.8, 0.9, 1.1, 1.2, 1.3, 1.4, 1.5$

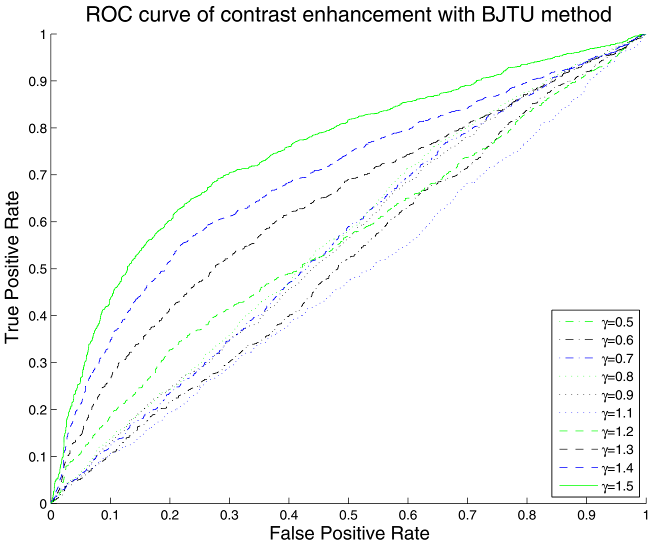


Fig. 7. ROC of contrast enhancement using BJTU method with $\gamma = 0.5, 0.6, 0.7, 0.8, 0.9, 1.1, 1.2, 1.3, 1.4, 1.5$

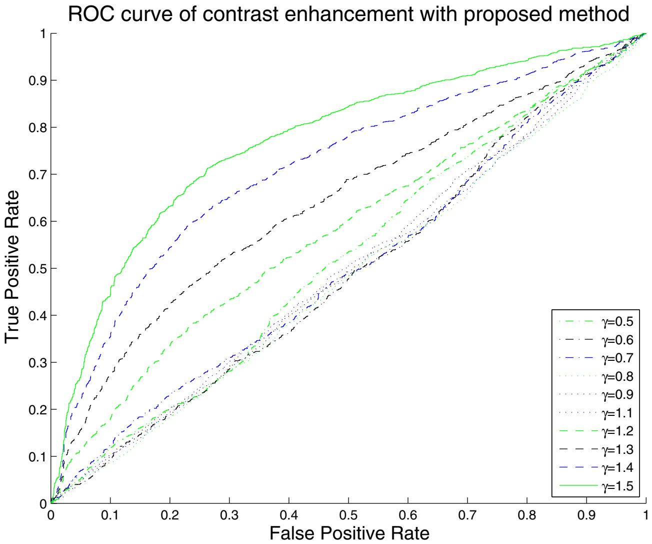


Fig. 8. ROC of contrast enhancement using proposed method with $\gamma = 0.5, 0.6, 0.7, 0.8, 0.9, 1.1, 1.2, 1.3, 1.4, 1.5$

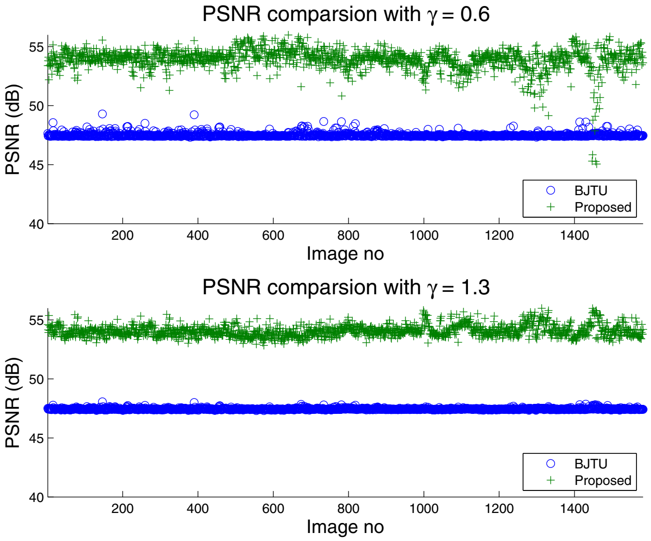


Fig. 9. ROC of contrast enhancement with $\gamma = 1.2$

Quality Comparison. PSNR is used for comparing image quality between the method in [2] and ours. We used the traditional contrast enhancement method as our reference I and anti-forensic image K as input:

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

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

The result is as follows:

	average PSNR (dB)									
Method	$\gamma = 0.5$	$\gamma = 0.6$	$\gamma = 0.7$	$\gamma = 0.8$	$\gamma = 0.9$	$\gamma = 1.1$	$\gamma = 1.2$	$\gamma = 1.3$	$\gamma = 1.4$	$\gamma = 1.5$
[2]	47.57	47.50	47.51	47.55	47.53	47.53	47.53	47.46	47.46	47.50
Proposed	54.35	53.91	54.04	54.34	54.30	54.60	54.69	54.07	54.28	54.62
Gain	+6.78	+6.41	+6.53	+6.79	+6.77	+7.07	+7.16	+6.61	+6.82	+7.12

6 Discussion

It is hard to compare the result between our method and method [2] because BJTU method can select different variances but in this experiment, we just use $\sigma = 1$ which is the same as the experiment in [2]. However, theoretically, if we want to achieve higher undetectability for the BJTU method, we can add a more severe noise to the image but it decreases the PSNR of the image also. In our method, we have better undetectability as well as having better PSNR. The reason is that the uniform we added also undergoes contrast stretching. Thus, noise level is dependent on different intensity values.

From the ROC curve, it is obvious that the undetectability decreases while the γ increases. This is because some images in the test set are over exposed, that means there is a strong peak at high intensity in the histogram. This peak will be suppressed by the pinch-off function in eq. 3. However, when the image undergoes contrast enhancement, the peak will shift to the left in the histogram, then it will not be suppressed by the function anymore. To solve this problem, we can combine both methods, which means severe noise can be added to spread out the peak.

$$f_Y = [m(f_X + n_U) + n_X] \tag{14}$$

where n_U is a uniform noise within -0.5 to 0.5 and n_X is a Gaussian noise with different variances to spread out the peak for a specific intensity.

7 Conclusion

In this paper, we have proposed an anti-forensic method to remove peak-gap artifacts in the histogram by increasing the internal bit depth of the image. Although there is an existing method to perform anti-forensic of contrast enhancement, our method has better image quality in terms of PSNR. Our proposed anti-forensic method works with various experiments with a large image database. This shows that our proposed method can result in the wrong classification in the current forensic method, which suggests that current forensic methods of contrast enhancement need to be improved.

Acknowledgement. This work has been supported in part by the Innovation and Technology Commission (GHP/048/08) of Hong Kong, China.

References

1. Bayram, S., Sencar, H., Memon, N.: An efficient and robust method for detecting copy-move forgery. In: IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2009, pp. 1053–1056. IEEE (2009)
2. Cao, G., Zhao, Y., Ni, R., Tian, H.: Anti-forensics of contrast enhancement in digital images. In: Proceedings of the 12th ACM Workshop on Multimedia and Security, MM&#38;Sec 2010, pp. 25–34. ACM, New York (2010), `http://doi.acm.org/10.1145/1854229.1854237`
3. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines (2001), software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`
4. Chen, W., Shi, Y., Su, W.: Image splicing detection using 2-d phase congruency and statistical moments of characteristic function. In: Security, Steganography and Watermarking of Multimedia Contents IX. Proceeding. of SPIE, San Jose, CA, USA (2007)

5. Chujoh, T., Noda, R.: Internal bit depth increase for coding efficiency. Proposition VCEG-AE13, ITU-T VCEG, Marrakech, Maroc (2007)
6. Mahdian, B., Saic, S.: Blind authentication using periodic properties of interpolation. IEEE Transactions on Information Forensics and Security 3(3), 529–538 (2008)
7. Popescu, A., Farid, H.: Exposing digital forgeries by detecting traces of resampling. IEEE Transactions on Signal Processing 53(2), 758–767 (2005)
8. Schaefer, G., Stich, M.: Ucid-an uncompressed colour image database. In: Proc. SPIE, Storage and Retrieval Methods and Applications for Multimedia, vol. 5307, pp. 472–480 (2004)
9. Stamm, M., Liu, K.: Forensic detection of image tampering using intrinsic statistical fingerprints in histograms. In: Proceedings: APSIPA ASC 2009: Asia-Pacific Signal and Information Processing Association, 2009 Annual Summit and Conference, pp. 563–572 (2009)
10. Stamm, M., Liu, K.: Forensic estimation and reconstruction of a contrast enhancement mapping. In: 2010 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP),, pp. 1698–1701. IEEE (2010)
11. Wang, J., Cha, B., Cho, S., Kuo, C.: Understanding benford's law and its vulnerability in image forensics. In: IEEE International Conference on Multimedia and Expo., ICME 2009, pp. 1568–1571. IEEE (2009)

Anti-Forensics of Double JPEG Compression Detection

Patchara Sutthiwan and Yun Q. Shi

Department of Electrical and Computer Engineering
New Jersey Institute of Technology, Newark, NJ, USA
{ps249,shi}@njit.edu

Abstract. In this paper, a simple yet effective anti-forensic scheme capable of misleading double JPEG compression detection techniques is proposed. Based on image resizing with bilinear interpolation, the proposed operation aims at destroying JPEG grid structure while preserving reasonably good image quality. Given a doubly compressed image, our attack modifies the image by JPEG decompressing, shrinking and zooming the image with bilinear interpolation before JPEG compression with the same quality factor as used in the given image. The efficacy of the proposed scheme has been evaluated on two prominent double JPEG detection techniques and the outcome reveals that the proposed scheme is mostly effective, especially in the cases that the first quality factor is lower than the second quality factor.

Keywords: Anti-Forensics, Counter-Forensics, Double JPEG Compression, Shrink-and-Zoom, Image Resizing, Bilinear Interpolation, Image Rotation, Low-Pass Filtering, Histogram Equalization, Tampering Detection.

1 Introduction

The advent of digital cameras as well as image editing tools has made digital images new form of statue of memory. Duplication, distribution, or tampering of such media can be easily done which calls for the necessity to be able to trace back the authenticity or history of media. Digital image forensics is a branch of research that aims to resolve the imposed problem. As sophisticated as digital image forensic schemes have been designed on the one hand, there are weak points of such schemes on the other hand. Over the past few years, anti-forensics has emerged as a relatively new branch of research. It aims at revealing the weakness of forensic technology and this may lead to improve the next generation of such a technology.

Double JPEG compression detection has been of great significance to digital image forensics, especially to image steganlysis and tampering detection. Its goal is to distinguish between JPEG images compressed once and those compressed twice. In this paper, we introduce an anti-forensic technique and we show its effectiveness on two highly effective double JPEG compression schemes.

2 Prior Art

In [1], an anti-forensic scheme to hide evidence of JPEG compression by adding adjustable noise to DCT coefficient of JPEG compressed images and removing

Y.Q. Shi, H.J. Kim, and F. Perez-Gonzalez (Eds.): IWDW 2011, LNCS 7128, pp. 411–424, 2012.

blocking artifacts was proposed. It has been shown that the anti-forensically modified images have statistical properties close to those of uncompressed images. In [2], the authors proposed de-blocking operation on top of the scheme introduced in [1] to fool forensic schemes that rely on the evidence of JPEG compression. The authors of [3] also indicated that their scheme could potentially mislead double JPEG compression schemes and extensively presented the afore-mentioned anti-forensic schemes.

3 Double JPEG Compression Detection

Double compression artifacts, also known as double quantization (DQ) artifacts, can be effectively traced by analyzing the statistical properties of DCT coefficients and are characterized by peak-and-valley pattern in a JPEG mode histogram. According to [4], double compression introduces periodic peak artifacts to a JPEG mode histogram when the ratio of second quantization step (q_2) over the first one (q_1) is not an integer; otherwise, the mode histogram of singly compressed and doubly compressed images would be indistinguishable. In JPEG compression, the expected quality of compressed images is user-defined and defined by a quality factor (QF) which is an 8×8 matrix filled up with 64 quantization steps; therefore, in the generation of a doubly compressed image, we have to specify first quality factor (QF1) and second quality factor (QF2). The aforementioned condition of periodic artifacts implies that the detection of double JPEG compression of doubly compressed images generated by QF1 = QF2 is intrinsically undetectable with this approach. Note that some new strategy has made this detection doable [5].

Several double JPEG compression detection schemes have been adopted under the passive-blind framework; however, in this paper, we pick up two rather efficient schemes to be attacked: 1) [6] proposed 324-dimensional feature vector derived from applying Markov process to horizontal, vertical, major diagonal and minor diagonal difference JPEG 2-D arrays. Herein, we abbreviate the features as MP-324. Owing to the limitation in the size of image dataset to be used, we modify MP-324 by ignoring the features generated from both diagonal difference arrays which results in the feature dimensionality reduction from 324 to 162. We therefore abbreviate the modified features as MP-162. The efficacy of image features have been evaluated by Support Vector Machine; 2) [7] utilized the probabilities of the first digit of 20 quantized JPEG AC modes to form a 180-dimensional feature vector which is abbreviated in this paper as MBFDF. The effectiveness of image features has been assessed by Fisher Linear Discriminant (FLD).

Per pair of first quality factor (QF1) and second quality factor (QF2), both double JPEG compression detection schemes have been re-evaluated using the corresponding classification settings over 1,338 pairs of singly and doubly compressed images in gray-scale of size 384x512 (landscape) or 512x384 (portrait). These images were created from UCIDv2 [8], an uncompressed image dataset, with a pair of first and second quality factors (QF1, QF2) = (a, b) where a, b ϵ {50, 60, 70, 80, 90} and a $\neq$ b. Over the generated dataset, for both schemes, we independently train and test classifiers for 20 times and report the average detection accuracies. Per iteration, SVM [9] with random data partition (5/6 for training and 1/6 for testing) has been employed to re-evaluate MP-162 on the generated dataset and tabulate the result in

Table 1. To re-evaluate MBFDF, FLD with randomly selected 1138 images for training and the 200 images for testing has been employed for 20 independent runs, the results of which appear in Table 2. Please be noted that the re-evaluated detection rates slightly differ from what have been reported in [6] and [7] potentially because of: 1) different dataset, different random partition, and different features, for MP-162; 2) different random partition, for MBFDF.

Table 1. Detection accuracy of MP-162

QF1/QF2	50	60	70	80	90
50	-	100.00	100.00	100.00	100.00
60	99.93	-	100.00	100.00	100.00
70	99.98	100.00	-	100.00	100.00
80	99.91	99.64	100.00	-	100.00
90	99.98	99.96	99.62	99.62	-

Table 2. Detection accuracy of MBFDF

QF1/QF2	50	60	70	80	90
50	-	100.00	100.00	100.00	100.00
60	100.00	-	100.00	100.00	100.00
70	100.00	100.00	-	100.00	100.00
80	99.93	99.63	100.00	-	100.00
90	99.88	99.93	99.98	99.95	-

4 Shrink-and-Zoom Attack

Obfuscating DQ artifacts is central to anti-forensics. Such deliberate attacks have created credibility gap on many forensic schemes that rely upon the trace of DQ artifacts, e.g., image tampering detection schemes derived from the DCT domain; however, in this anti-forensic work, we limit our attention to misleading double compression detection schemes of interest in the belief that the generalization of the attacks could be extended to other related forensic applications.

In this paper, we propose a simple attack on doubly compressed images which effectively fools double JPEG compression detection schemes. The rationale behind the attacks is to disrupt JPEG grid structure. The original concept is from steganalysis [10] in which the JPEG grid structure of a given JPEG image has been destroyed by calibration attack which follows the following procedures: 1) decompression; 2) cropping 4 rows and 4 columns; 3) re-compression with the same QF. Although it has been proven that calibration can effectively reduce DQ artifacts, the inconsistency in image size caused by image cropping would leave an observable trail of image modification. We then propose Shrink-and-Zoom (SAZ) attack on a doubly compressed image to suppress DQ artifacts while to maintain the original image size as well as to preserve good image quality. The procedure of SAZ attack on a given doubly compressed image is as follows:

1. JPEG decompressing a given doubly compressed image
2. Shrinking the image of size X by Y to sX by sY, where s is the degree of shrinkage and 0<s<1
3. Zooming the shrunk image back to its original size X by Y. In this paper s = 0.9 and bilinear interpolation are recommended.
4. JPEG compressing the resultant image in Step 3 with QF2

The strategy for disrupting JPEG grid structure employed in SAZ attack is basically double image resizing. Image resizing involves interpolation to estimate pixel values of the location previously non-existent. Even though there are many interpolation algorithms, we opt to employ bilinear interpolation because it not only retains reasonably good image quality but also powerfully attacks double compression detection schemes. Although image resizing has been mentioned in [11] as an alternative way to destroy JPEG grid structure, to our best knowledge, there is no anti-forensic scheme reported that this technique can fail double JPEG compression detection schemes.

We have found experimentally that the degree of shrinkage(s) and choice of interpolation have a strong connection with the effectiveness of DQ artifact elimination and image quality. Generally speaking, the smaller the s is, the worse the quality of anti-forensically modified image and the more severe the attack become; however, beyond some certain s, the effectiveness of the attack improves little while the image quality of the attacked image degrades greatly.

4.1 Image Quality Measure (PSNR)

One of the ultimate goals of anti-forensic schemes is to preserve the visual quality of the image to be attacked. Usually expressed in terms of logarithmic decibel, peak signal-to-noise ratio (PSNR) has been widely used as a measure on image quality of reconstruction of lossy image and is defined as a function of mean squared error (MSE) of two m×n gray-scale images I and J. Either I or J is a noisy approximation of the other. Please be noted that in the experimental setting presented in this paper, anti-forensically modified images is considered as a noisy approximation of doubly compressed images which is considered as a reference image.

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - J(i,j)]^2 \tag{1}$$

$$PSNR = 10\log_{10}\left(\frac{Max_I^2}{MSE}\right) = 10\log_{10}\left(\frac{255^2}{MSE}\right) \tag{2}$$

Max_I is the maximum pixel value of the image, e.g., in this paper, 8 bits are used to represent one pixel, Max_I is 255.

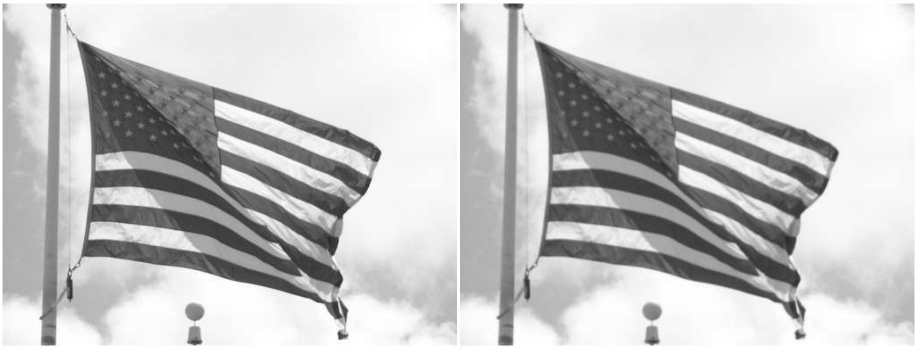


Fig. 1. (left) Doubly compressed image generated from ucid01248 using (QF1, QF2) = (60, 80); (*right*) The anti-forensically modified image generated from SAZ with s = 0.9 and bilinear interpolation. The PSNR (dB) between two images is 35.97 dB (the left picture was used as a reference image).

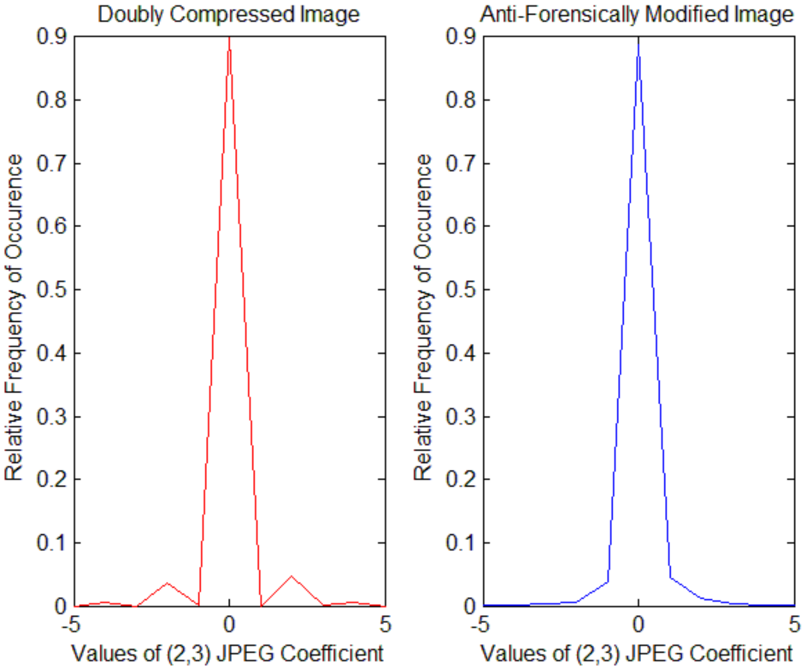


Fig. 2. Mode histograms of generated from generated from the two images in Fig. 1

In this paper, the image ucid01248 in the UCID [8] dataset was used in the demonstrations of the proposed anti-forensic techniques which depict the effectiveness of the techniques in terms of visual quality preservation and DQ artifact elimination capability. In Fig. 1, the visual difference between the two images in the figure is virtually indiscernible subjectively but also the quantitative difference measure between the two images is quite acceptable. The histogram in Fig. 2 (Left) has a peak-and-valley pattern (DQ artifacts) caused by double JPEG compression while Fig. 2 (Right) contains no such a DQ artifacts associated with double JPEG compression which indicate that SAZ suppresses DQ artifacts to a considerable degree at a given pair of quality factors. Consequently, SAZ is capable of destroying DQ artifacts while preserving decent visual quality of the resultant anti-forensically modified images.

4.2 Evaluation on Anti-Forensic Scheme

In double JPEG compression detection, doubly compressed images are considered positive instances, while singly compressed ones are considered as negative instances. True positive (TP) rate is the percentage of doubly compressed images correctly classified, while true negative (TN) rate is the percentage of singly compressed images correctly classified. To evaluate the effectiveness of the proposed anti-forensic scheme, the key measure is the rate at which a classifier, trained for detecting double JPEG compression, classifies the anti-forensically modified images as doubly compressed images; the lower the more powerful of the proposed anti-forensics scheme, The evaluation process can be briefly described as follows: 1) conduct SAZ attack on a given doubly compressed image; 2) extract image feature vector; 3) feed it to the corresponding 20 trained classifiers. Table 3 shows the relationship between s, average PSNR, TP rates after SAZ attack on doubly compressed images generated from the UCID [8].

Table 3. Average PSNR, TP rate after SAZ attack versus s

s	0.998	0.9	0.5
PSNR (dB)	35.21	32.38	28.58
TP Rate (MP-162)	48.40%	33.74%	21.90%
TP Rate (MBFDF)	19.50%	12.40%	8.60%

Please be noted that the average PSNR in dB is calculated by averaging the PSNR before converting it to dB. Table. 3 reveals that smaller s brings about more effective attack while decreases the average PSNR (dB). To maintain acceptable image quality, we choose s = 0.9.

Table 4. Detailed average TP rates of MP-162

QF1\QF2	50	60	70	80	90
50	-	100.00	100.00	100.00	100.00
60	99.87	-	100.00	100.00	100.00
70	99.87	100.00	-	100.00	100.00
80	100.00	99.60	100.00	-	100.00
90	99.96	99.96	99.42	99.64	-

Table 5. Detailed average TP rates of MP-162 after SAZ attack with s = 0.9

QF1\QF2	50	60	70	80	90
50	-	0.13	0.00	0.00	0.00
60	1.26	-	0.00	0.00	0.00
70	30.09	19.10	-	0.13	0.00
80	33.77	38.21	6.05	-	0.13
90	100.00	99.78	99.55	9.87	-

The average TP rates before SAZ attack are reported in Tables 4 and 6 in comparison with the corresponding average TP rates after SAZ attack in Tables 5 and 7. From Tables 4 to 7, it is noticeable that SAZ is more effective to mislead the trained classified when QF1<QF2 than when QF1>QF2. The intuitive explanation of this phenomenon is discussed in Section 4.3 along with a quantitative measure to support the claim.

The effectiveness of SAZ is confirmed by the low TP rates after SAZ attack as shown in Tables 5 and 7. In all cases, SAZ almost perfectly attacks the upper triangle of the tables where QF1 < QF2; however, in the lower triangle of the tables where QF1 > QF2, SAZ is less effective. This phenomenon can be intuitively explained by the following reasons: 1) when QF1 < QF2, DQ artifacts are so obvious that SAZ can powerfully distort the statistical properties of the doubly compressed images; 2) when QF1 > QF2, DQ artifacts exist less severely, hence being more difficult for SAZ to distort the statistical properties of such doubly compressed images.

Table 6. Detailed average TP rates of MBFDF

QF1\QF2	50	60	70	80	90
50	-	100.00	100.00	100.00	100.00
60	100.00	-	100.00	100.00	100.00
70	100.00	100.00	-	100.00	100.00
80	99.85	99.40	100.00	-	100.00
90	99.75	99.85	99.95	99.90	-

Table 7. Detailed average TP rates of MBFDF after SAZ attack with s = 0.9

QF1\QF2	50	60	70	80	90
50	-	1.00	0.00	0.00	0.00
60	10.55	-	0.20	0.00	0.00
70	30.80	17.95	-	0.00	0.00
80	1.60	41.20	10.00	-	0.00
90	90.65	26.80	14.80	3.20	-

4.3 Statistical Deviation by SAZ

We measure the statistical changes in feature level introduced by SAZ by comparing two statistical parameters K and Ƙ whose calculation is depicted in Fig. 3 and values are tabulated in Tables 8 to 11. To calculate both K's, a singly compressed image with QF2 is used as a reference image. K is the L1 distance between feature vectors extracted from a given doubly compressed image and the reference image, while Ƙ is the L1 distance between feature vectors extracted from the anti-forensically modified image and the reference image. That's, K represents the change with respect to the reference image in feature-level introduced by double compression, K introducing the change with respect to the reference image in feature-level introduced by SAZ. Mean values of K and Ƙ are tabulated in Tables 8 to 11.

When QF1 < QF2, Ƙ is much less than K which means that SAZ moves the statistical properties of the anti-forensically modified images closer to their corresponding singly compressed versions (the reference images). When QF1 > QF2, Ƙ is generally close to K which means that SAZ slightly distort the statistical properties of the anti-forensically modified images. SAZ is therefore generally less effective under this situation. The amount of difference between K and Ƙ indicates the degree of statistical deviation introduced by SAZ.

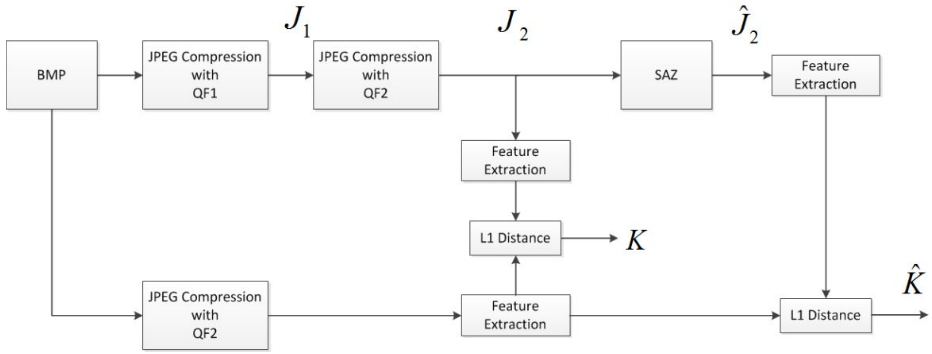


Fig. 3. Block diagram of K and $\hat{K}$

Table 8. Means of K of MP-162

QF1\QF2	50	60	70	80	90
50	-	2.97	11.14	13.26	18.26
60	1.82	-	5.39	14.85	16.05
70	1.79	1.57	-	4.40	15.06
80	1.45	1.24	1.89	-	11.82
90	0.30	0.44	0.36	1.02	-

Table 9. Means of $\hat{K}$ of MP-162

QF1\QF2	50	60	70	80	90
50	-	1.98	2.97	3.80	4.73
60	1.41	-	2.40	2.95	4.41
70	1.97	1.54	-	2.36	3.98
80	1.59	1.44	1.61	-	3.03
90	1.27	1.35	1.51	1.73	-

Table 10. Means of K of MBFDF

QF1\QF2	50	60	70	80	90
50	-	5.44	21.57	23.42	21.62
60	3.59	-	7.61	22.20	22.71
70	3.10	4.30	-	11.18	21.14
80	2.79	1.52	4.08	-	18.08
90	0.80	1.14	1.26	1.59	-

Table 11. Means of $\hat{K}$ of MBFDF

QF1\QF2	50	60	70	80	90
50	-	3.49	4.96	4.86	4.57
60	4.03	-	3.01	4.07	4.54
70	4.76	4.47	-	3.45	4.31
80	3.20	3.95	4.35	-	3.17
90	3.40	3.57	3.37	3.39	-

5 Alternatives to SAZ

As mentioned before, the ultimate goal of the anti-forensic scheme proposed in this paper is to disrupt JPEG grid structure. It has been shown in the above Sections that SAZ, based on image resizing, is one of effective methods; however, we would like to point out, without large-scale empirical validation, a few other methods that could possibly be also effective in JPEG grid disruption.

First, image rotation is able to disrupt JPEG grid structure as also having been pointed out in [11]; however, we do not recommend this operation because of the following reasons: 1) Image rotation would require image cropping to eliminate artificial image boundary and result in the inconsistency between image sizes; 2) Image rotation would make the spatial coordinate of the anti-forensically modified image nonaligned with the corresponding doubly compressed image which leads to unacceptably low PSNR.

Second, low-pass filtering can be considered as an equivalent operation of SAZ. There are of course various ways to apply low-pass filter to an image; for the sake of simplicity, we choose to convolute a 3×3 mean filter to a given doubly compressed image and then JPEG compress the filtered image with QF2.

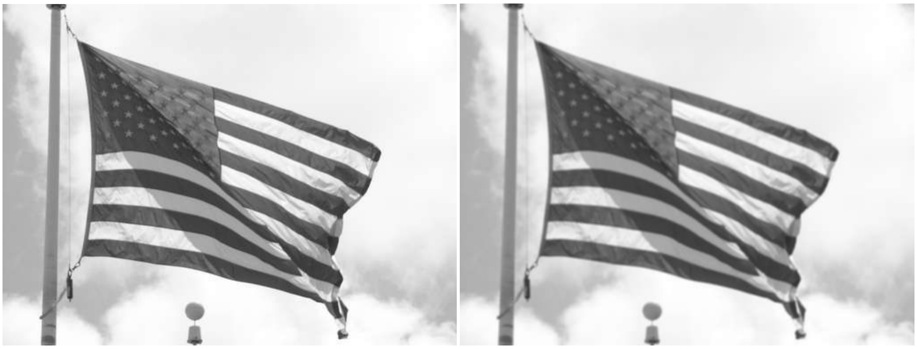


Fig. 4. (left) Doubly compressed image generated from ucid01248 using (QF1, QF2) = (60, 80); (*right*) The anti-forensically modified image generated by convoluting a 3×3 mean filter on the doubly compressed image. PSNR (dB) between two images is 34.57 dB (the right picture was used as a reference image).

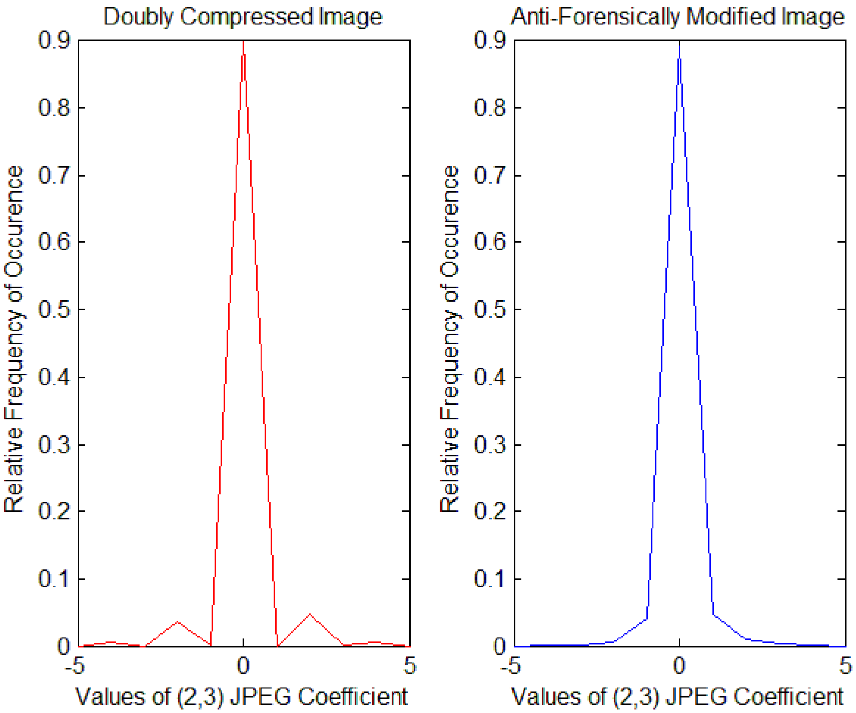


Fig. 5. Mode histograms generated from the two images in Fig. 4

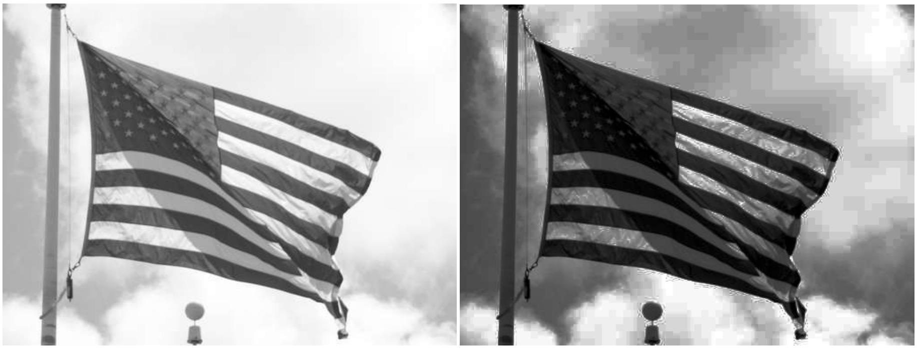


Fig. 6. (left) Doubly compressed image generated from ucid01248 using (QF1, QF2) = (60, 80); (*right*) The anti-forensically modified image generated by conducting histogram equalization. PSNR (dB) between two images is 10.63 dB (the right picture was used as a reference image).

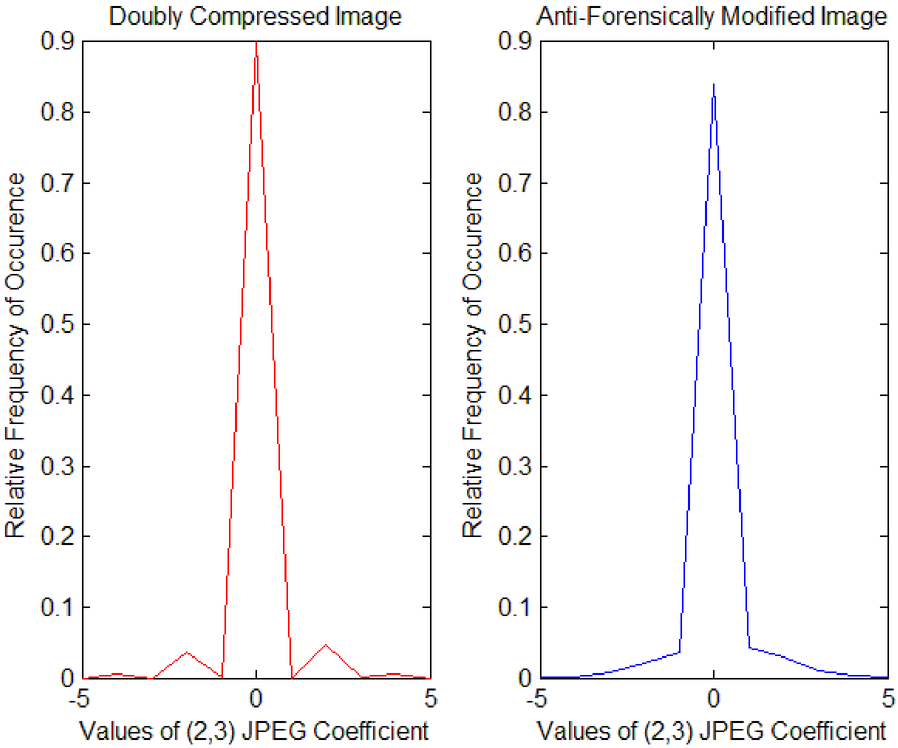


Fig. 7. Mode histograms generated from the two images in Fig. 6

Fig. 4 and Fig. 5 confirm the effectiveness of the usage of a low-pass filter not only to preserve image quality but also to eliminate DQ artifacts; under the same circumstances, the resultant PSNR suggests that image quality retained by convoluting a 3×3 mean filter to the corresponding doubly compressed image is slightly worse than that retained by SAZ. We may also visually see that the obtained anti-forensically modified image in Fig. 4 (Right) is a little more blurry than that in Fig. 1. This is, however, in no ways, a definite conclusion as other choices of low-pass filter may bring off a higher PSNR.

Third, intensity transformations, such as histogram equalization and logarithmic transformation, can also destroy DQ artifact. They globally change the relationship among pixel intensity in the given image; in some sense, depending on the degree of change introduced to a compressed image, such operations create a seemingly new uncompressed image with similar visual semantic content with virtually no DQ artifacts. In Fig. 6 and Fig. 7, we demonstrate the effectiveness of applying histogram equalization to the given doubly compressed image. As expected, PSNR of the two images in Fig. 6 is unacceptably low, only 10.63 dB, but DQ artifacts have been removed to some extent as shown in Fig. 7. Although intensity transformation may work very effectively in hiding the traces of DQ artifacts, we do not recommend this approach such it is most likely to yield a noticeably change in the attacked images.

6 Discussion and Concluding Remarks

6.1 SAZ in a Nutshell

In this paper, we introduce an anti-forensic operation capable of misleading two highly effective JPEG double compression detection schemes [6] and [7]. There are two ultimate goals of anti-forensic schemes: 1) image quality preservation; 2) obfuscating forensic artifacts. The proposed attack relies upon image resizing with bilinear interpolation and is called Shrink-and-Zoom (SAZ). The efficacy of SAZ is assessed by the rate at which a classifier, trained for double JPEG compression detection, classifies anti-forensically modified images as doubly compressed images. That is, the lower such a rate, the more powerful the attack is. To operate SAZ, the scaling factor s is the key parameter that controls not only the image quality but also TP rates after the attack. We choose s = 0.9 which yield a reasonable balance between PSNR and TP rates after the attack.

The performance gap between TP rates between before and after SAZ attack also indicates how effective the attack is; that is, the lower TP rate after the attack than TP rate before the attack, the more effective the attack is. Although in most cases, the TP rates after the attack are lower than those before the attack, it has been observed that TP rate after the attack is slightly higher than TP rate before the attack for only two combinations of QF1 and QF2 and in such cases the attack can be deemed ineffective.

When TP rate after SAZ attack is lower than TP rate before the attack, the attack moves the statistical properties of anti-forensically modified images closer to those of singly compressed images. On the other hand, when TP rate after the attack is higher than TP rate before the attack (as said before, this is rare), the attack moves the statistical properties of anti-forensically modified images further away from those of singly compressed images. The relationship between TP rates before and after the attack implies the direction toward which the attack moves the statistical properties of features, while the difference between K and $\hat{K}$indicates the degree of statistical deviation the attack brings forth.

6.2 The Connection between Anti-Forensic Schemes of Double JPEG Compression Detection and the Practicality of Image Tampering Detection Schemes

Image tampering detection has been a hot research topic over the past few years. Many such schemes, such as [12], [13], and [14], have been proposed to measure irregularity caused by image tampering from observed JPEG coefficients and evaluated over some public datasets, e.g., [15] and [16]. These schemes generally perform fairly well within the datasets used; however, its practicality is still questionable as it has been reported in [14] that the accuracy of testing real-life tampered images, of which the ground truths are known but the processing histories done to the images are not, collected from the Internet on the classifiers trained over [16] is much lower than that evaluated by using the testing images in the used dataset.

What has been presented all along this paper could partly explain why image tampering detection schemes derived from JPEG coefficients have performed poorly in reality. Assuming no further malicious attack, image resizing and image rotation seem to be frequent choices of non-malicious image processing operation that makes a given image suitable for being displayed in an Internet browser while image enhancement operations like histogram equalization as well as low-pass seem to be done relatively much less. However, any of such non-malicious image operations is likely to be applied to any digital image before its distribution over the Internet and it can potentially obfuscate the traces of tampering artifacts left in the JPEG coefficients. In other words, for compressed domain methods, tampered images generated from JPEG images are susceptible to many common image processing operations; consequently, the practicality of such methods is limited. SAZ as well as some other methods discussed in Section 5 is most likely to be able to confuse image tampering detection schemes that rely on JPEG information.

References

[1] Stamm, M.C., Tjoa, S.K., Lin, W.S., Liu, K.J.R.: Anti-Forensics of JPEG Compression. In: IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP), pp. 1694–1697. IEEE Press, New York (2010)

[2] Stamm, M.C., Tjoa, S.K., Lin, W.S., Liu, K.J.R.: Undetectable Image Tampering Through JPEG Compression. In: IEEE International Conference on Image Processing (ICIP), pp. 2109–2112. IEEE Press, New York (2010)

[3] Stamm, M.C., Liu, K.J.R.: Anti-Forensics of Digital Image Compression. IEEE Transactions on Information Forensics and Security 6, 1050–1065 (2011)

[4] Popescu, A.C.: Statistical Tools for Digital Image Forensics. PhD Thesis, Darmouth College, Hanover, NH, USA (advised by H. Farid) (December 2004)

[5] Huang, F., Huang, J., Shi, Y.Q.: Detecting Double JPEG Compression with the Same Quantization Matrix. IEEE Transactions on Information Forensics and Security 5, 848–856 (2010)

[6] Chen, C., Shi, Y.Q., Su, W.: A Machine Learning Based Scheme for Double JPEG Compression Detection. In: International Conference on Pattern Recognition (ICPR), pp. 1–4. IEEE Press, New York (2008)

[7] Bin, L., Shi, Y.Q., Huang, J.: Detecting Doubly Compressed JPEG Images by Using Mode Based First Digit Features. In: IEEE Workshop on Multimedia Signal Processing (MMSP), pp. 730–733. IEEE Press, New York (2008)

[8] Schaefer, G., Stich, M.: UCID: an Uncompressed Color Image Database. In: Proc. SPIE, Storage and Retrieval Methods and Applications for Multimedia, San Jose, USA, pp. 472–480 (2004)

[9] Chang, C.C., Lin, C.J.: LIBSVM : a library for support vector machines. ACM Transactions on Intelligent Systems and Technology (2011), Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

[10] 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)

[11] 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)

[12] Sutthiwan, P., Shi, Y.Q., Dong, J., Tan, T., Ng, T.T.: New Developments in Color Image Tampering Detection. In: IEEE International Symposium on Circuits and Systems, pp. 3064–3067. IEEE Press, New York (2010)

[13] Sutthiwan, P., Shi, Y.Q., Su, W., Ng, T.T.: Rake Transform and Edge Statistics for Image Forgery Detection. In: Workshop on Content Protection and Forensics, IEEE International Conference on Multimedia and Expo., pp. 1463–1468. IEEE Press, New York (2010)

[14] Sutthiwan, P., Shi, Y.Q., Zhao, H., Ng, T.-T., Su, W.: Markovian Rake Transform for Digital Image Tampering Detection. In: Shi, Y.Q., Emmanuel, S., Kankanhalli, M.S., Chang, S.-F., Radhakrishnan, R., Ma, F., Zhao, L. (eds.) Transactions on DHMS VI. LNCS, vol. 6730, pp. 1–17. Springer, Heidelberg (2011)

[15] Columbia DVMM Research Lab: Columbia Image Splicing Detection Evaluation Dataset (2004), http://www.ee.columbia.edu/ln/dvmm/downloads/AuthSpliced-DataSet/AuthSplicedDataSet.htm

[16] CASIA Tampered Image Detection Evaluation Database (2010), http://forensics.idealtest.org

(r, n)-Threshold Image Secret Sharing Methods with Small Shadow Images*

Xiaofeng Wang[1,**], Zhen Li[1], Xiaoni Zhang[1], Shangping Wang[1], and Jing Chen[2]

[1] Xi'an University of Technology, Xi'an, Shaanxi, 710048, P.R. China
[2] Xi'an University of Architecture and Technology, Xi'an, Shaanxi, 710055, P.R. China
xfwang66@sina.com.cn

Abstract. Image secret sharing is a steganography communication approach which provides the functionality of tolerance data corruption or loss. In this paper, we combined existing methods [5] and [8], proposed two improved (2, 4)-threshold image secret sharing methods, the method I is based on multi-secret sharing mode, and method II is based on priority sharing mode. In improved methods, Huffman coding and the difference matrix are used to reduce the size of shadow images, image blocks are used to limit error-propagation. Comparing with the existing methods, the characters of the proposed methods are as follows: (1) the sizes of generated shadow images are smaller; (2) method II provided a lossless secret image recovery approach with smaller shadow images. The experimental results show that the sizes of shadow images are only 30% (method I) and 15% (method II) of the original secret image.

Keywords: (r, n)-Threshold, Image secret sharing, Huffman coding, Difference matrix.

1 Introduction

With the development of computer and multimedia technologies, the exchanges of multimedia information have become very extensive. At the same time, information eavesdropping is very easy. The loss or damage of some sensitive information such as government documents, financial information, military intelligence, will cause serious losses. Therefore, how to provide a secure data transmission method is an urgent issue. As one of the secret communication methods, traditional data encryption technology cannot meet the needs of multimedia information secure transmission while the steganography communication technology emerges in recent years.

Steganography communication is achieved by information hiding techniques, which is becoming a hot research topic in recent years because of its advantage in the field of information security.

* This work was supported by the National Natural Science Foundation of China under Grant No.61075007 and No.61074077; the Natural Science Foundation Research Plan of Shaanxi Province of China under Grant No.2009JM8004-5 and No.11JK0900.

** Corresponding author.

Y.Q. Shi, H.J. Kim, and F. Perez-Gonzalez (Eds.): IWDW 2011, LNCS 7128, pp. 425–438, 2012.

Information hiding [1] is a technique that hiding the secret information into carrier information such as images, texts, voice and videos. Only the legitimated participants can recover the secret information from the stego-carriers. The purpose of information hiding is preserving the confidentiality of the secret information and providing a secure way for information transmission. On the other hand, it is necessary that information hiding does not affect visual or auditory effect of the stego-carriers.

Image is a typical multimedia form, as an effective method to implement information hiding, image digital watermarking has been developed in recent years. However, traditional image watermarking technology embed the secret information into a single carrier image, once the carrier image is destroyed, secret information will lose forever. Image secret sharing [2] is a mechanism that does not suffer above mentioned problem. The study of image secret sharing started from the concept of Visual cryptography [2]. Visual cryptography can be considered as a development of the cryptology unconditional security protocol----secret sharing is used in the field of the digital image authentication. The basic idea behind visual cryptography is to split a secret image into n shares, each of them is called a shadow image, and then send shadow images to n legitimate receivers. In the retrieval phase, a subset of the legitimate receivers printed the shadow images that they held on the transparent film and then overlap with those films, and then the secret image can be retrieved. By using this way, the subsets of legitimate receivers are able to see the secret image which is retrieved through the visual directly, while the illegal participants have nothing information about the secret image.

In recent years, steganography communication has been concerned greatly, and emerged many image secret sharing schemes which adapt to the needs of various applications. The methods of image secret sharing can be divided into two categories, one is image pixel-based and another is based on image blocks.

Shamir [2] is one of the typical pixel-based methods of Image Secret Sharing. Another typical method is proposed by Thien [3]. It is a (r, n)-threshold image secret sharing scheme. In their schemes, the $N\times N$ secret image is divided into several shares, each share has r pixels, and each pixel belongs to one and only one share. For j-th share, r-1 degree polynomial is constructed as follows:

$$q_j(x) = \left(a_0 + a_1 x + \cdots + a_{r-1} x^{r-1}\right) \bmod 251 \tag{1}$$

Here $a_0, a_1, \cdots, a_{r-1}$ are the r pixels of the j-th share. By this way, the size of each generated shadow image is $1/r$ of the original secret image.

Lin and Tsai [4] proposed an approach based on Shamir (r, n)-threshold image secret sharing. In their scheme, n shadow images were generated by computing the pixel values of the given secret image. These shadow images are then hidden in n user-selected camouflage images. Furthermore, an image watermarking technique is employed to embed fragile watermark signals into the camouflage images by using parity-bit checking. Their method provides the capabilities of steganography and authentication, simultaneous.

In the application of image secret sharing, shadow images are usually used as watermark signals which are embedded into the camouflage images. To achieve a high-quality visual effect, the smaller shadow images are necessary. In order to decrease the size of the shadow image, scholars have tried lots of different approaches. Wang and Su [5] proposed a method that computed the difference image of the secret image, and compress the difference image by using Huffman coding, then achieve (r, n)-threshold image secret sharing. The experimental results shown that generated shadow image is about 40% smaller than that of the method in Thien [3].

Another category of image secret sharing method is image block-based approach. Typical schemes can be seen from [6], [7], and [8].

In [6], Li Bai proposed a reliable (r, n)-threshold image secret sharing method that incorporates two r-out-of-n secret sharing schemes: i) Shamir's secret sharing scheme and ii) matrix projection secret sharing scheme [9]. By using their methods, the size of generated shadow image is $1/r+1/n$ of the secret image, which is bigger than that of the method of Thien [3]. Wang et al. [7] proposed a scalable image secret sharing scheme by using three sharing styles, namely the multi-secret mode, priority mode and progressive mode. These modes are designed to show diverse restoration effects for secret image. In their schemes, the size of each shadow image is half of the original secret image. Subsequently, Lin et al. [8] changed the shadow generation method of [7], the size of each shadow image is $(2n\text{-}r)/n^2$ of the original secret image.

In order to convenient for steganography communication, small shadow images are necessary. However, it is a challenging by using small shadow images to recover the original secret image. In this paper, based on image secret sharing method [5] and [8], we proposed two improved (2, 4)-threshold image secret sharing methods, which are based on multi-secret sharing mode and priority sharing mode, respectively. In multi-secret sharing mode, the secret image was partitioned into sub-images, and computed difference image for every sub-image, then compressed difference images by using Huffman coding, and generated shadow images. In priority sharing mode, instead of dividing secret image into sub-images, the secret image was partitioned into bit-level images, then compressed the bit-level images by taking the smallest pixel value of every 2×2 pixel block. The experimental results show that the shadow image sizes of improved methods are only 30% (method I) and 15% (method II) of the original secret image.

The remainder of this paper is organized as follows: improved methods are described in section 2 and section 3, and experimental results are shown in section 4. Finally, a brief conclusion is summarized in section 5.

2 The Proposed Method I: Multi-secret Sharing Mode

Inspired by the idea of literature [5] and [8], in this section, we presented an improved image secret sharing scheme based on Multi-Secret Sharing Mode.

2.1 Overall Structure

Proposed method consists of three phases: image pre-processing phase, shadow image generation phase, and secret image recovery phase (see Fig.1).

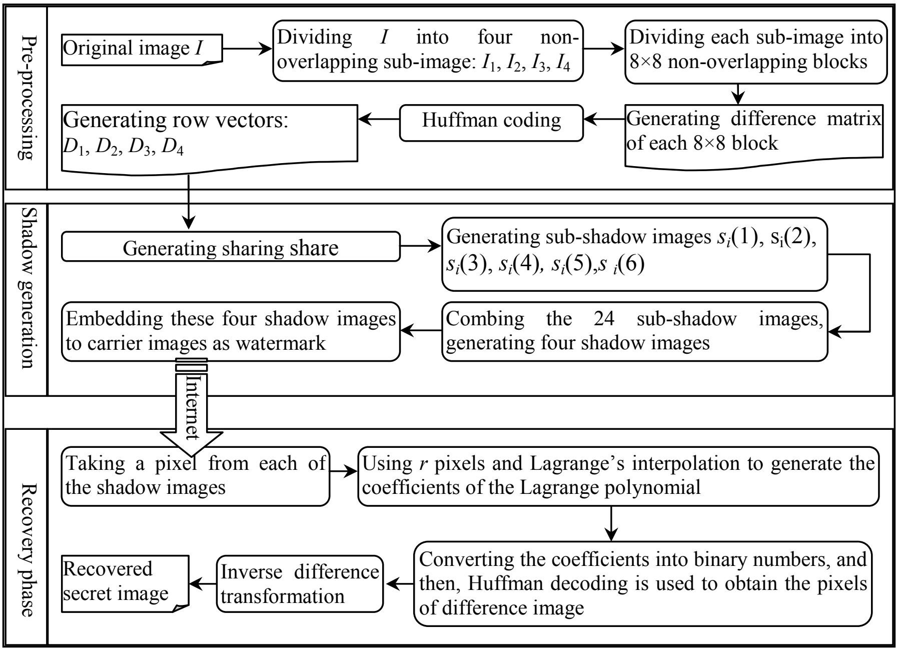

Pre-processing
Original image I
Dividing I into four non-overlapping sub-image: I_1, I_2, I_3, I_4
Dividing each sub-image into 8×8 non-overlapping blocks
Generating difference matrix of each 8×8 block
Huffman coding
Generating row vectors: D_1, D_2, D_3, D_4
Shadow generation
Generating sharing share
Generating sub-shadow images $s_i(1)$, $s_i(2)$, $s_i(3)$, $s_i(4)$, $s_i(5)$, $s_i(6)$
Combing the 24 sub-shadow images, generating four shadow images
Embedding these four shadow images to carrier images as watermark
Internet
Recovery phase
Taking a pixel from each of the shadow images
Using r pixels and Lagrange's interpolation to generate the coefficients of the Lagrange polynomial
Converting the coefficients into binary numbers, and then, Huffman decoding is used to obtain the pixels of difference image
Inverse difference transformation
Recovered secret image

Fig. 1. The structure of multi-secret sharing mode

2.2 Algorithm Description

(1) Image pre-processing

① Using the multi-secret sharing mode [7], dividing the original secret image I into n parts $P_1, P_2, \cdots, P_n$, and such that:

$$\begin{cases} \bigcup\limits_i P_i = P, & 1 \le j \le n, \\ P_i \bigcap P_j = \varphi, & 1 \le i \ne j \le n, \\ \left|P_i\right| = \dfrac{1}{n}\left|P\right|, & 1 \le i \le n, \end{cases} \tag{2}$$

For example, for (2, 4)-threshold (r=2, n=4) scheme, the image I (the size is N×N) is divided into four sub-images I_1, I_2, I_3, I_4 (the size is N/2×N/2). Fig. 2 (b) and (c) show two separation modes of Fig.2 (a).

② Dividing every sub-image $I_k\left(k=1,2,3,4\right)$ into 8×8 image blocks E_u^k, u=1,..., (N/16)× (N/16).

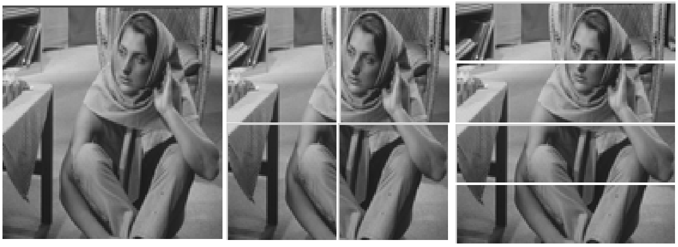


Fig. 2. (a) Original image; (*b*) Separation mode I; (*c*) Separation mode II

③ Computing difference matrix $diff_{ij}^{k}$ for each 8×8 image block:

$$diff_{ij}^{k}=\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{18} \\ a_{21} & a_{22} & \cdots & a_{28} \\ \vdots & \vdots & \vdots & \vdots \\ a_{81} & a_{82} & \cdots & a_{88} \end{bmatrix} \qquad k=1,2,3,4,\quad i,j=1,2,\cdots,N/16 \tag{3}$$

Here,

$$a_{st}=\begin{cases} E_u^k(s,t) & s=1,t=1 \\ E_u^k(s,t)-E_u^k(s-1,t) & s\neq 1,t=1 \\ E_u^k(s,t)-E_u^k(s,t-1) & others \end{cases} \qquad s,t=1,2,\cdots,8 \tag{4}$$

④ Combining $diff_{ij}^{k}$ to generate matrix $Diff_k$:

$$Diff_k=\begin{bmatrix} diff_{11}^{k} & diff_{12}^{k} & \cdots & diff_{1,N/16}^{k} \\ diff_{21}^{k} & diff_{22}^{k} & \cdots & diff_{2,N/16}^{k} \\ \vdots & \vdots & \vdots & \vdots \\ diff_{N/16,1}^{k} & diff_{N/16,2}^{k} & \cdots & diff_{N/16,N/16}^{k} \end{bmatrix} \quad (k=1,2,3,4) \tag{5}$$

⑤ Re-arranging elements of *Diff*$_k$ by row to generate a row vector, and compressing it by using Huffman coding, then converting the results into 0~255 decimal numbers, denoted as $D_k\ (k=1,2,3,4)$.

The purpose of using difference matrix is to reduce the size of shadow images. However, once an error occurred in the shadow image, it will cause error spread in the recovery phase. To overcome this problem, we divide the secret image into non-overlapping 8×8 image blocks; by this way, the error propagation is limited within a single image block.

(2) The shadow generation algorithm

We use (2, 4)-threshold scheme [8] to generate shadow images. The algorithm steps are as follows:

① Generating block-shadow images: Taking every four pixels, in sequence, from D_k $(k=1,2,3,4)$ as a sharing share, denoted as $\{a_0, a_1, a_2, a_3\}$, then construct Lagrange polynomial for each sharing share:

$$q_k^i(x) = (a_0 + a_1 x + \cdots + a_3 x^3) \bmod 2^t \tag{6}$$

where t=8, i=1,...,||D_k||/4, ||·|| is norm operation. Use $q_k^i(x)$ as one pixel of $s_k(x)$, generate six block-shadow images $s_k(x)$, $x = 1, 2, \cdots, 6$.

② Combining the 24 block-shadow images which obtained from step ① to generate four shadow images SH_1, SH_2, SH_3, SH_4, the algorithm is described in Table 1.

Table 1. The algorithm to generating shadow images SH_1, SH_2, SH_3, SH_4

Shadow images	D_1	D_2	D_3	D_4
SH_1	$s_1(1), s_1(2), s_1(3)$	$s_2(1)$	$s_3(1)$	$s_4(1)$
SH_2	$s_1(4)$	$s_2(2), s_2(3), s_2(4)$	$s_3(2)$	$s_4(2)$
SH_3	$s_1(5)$	$s_2(5)$	$s_3(3), s_3(4), s_3(5)$	$s_4(3)$
SH_4	$s_1(6)$	$s_2(6)$	$s_3(6)$	$s_4(4), s_4(5), s_4(6)$

(3)Secret image recovery algorithm

According to (2, 4)-threshold scheme, the secret image can be reconstructed by anyone who obtains at least two shadow images from the four generated shadow images.

① Taking one pixel in sequence from each of the r (2≤r≤4) shadow images (they are a subset of $\{q_k^i(1), q_k^i(2), \cdots, q_k^i(6)\}$), and using Lagrange's interpolation to compute the coefficients $a_0, \cdots, a_3$ of formula (6).

② Converting the coefficients $a_0, \cdots, a_3$ into binary numbers, then use Huffman decoding to reconstruct the difference images $D_1', D_2', \cdots, D_r'$, the size is $N/2$×$N/2$.

③ Taking the inverse-difference transformation for reconstructed r difference images $D_1', D_2', \cdots, D_r'$, then obtain the sub-images $I_1', I_2', \cdots, I_r'$. The inverse-difference transformation is shown as follows:

$$I_i'(j,k) = \begin{cases} D_i'(j,k) & j=1, k=1 \\ \sum_{j=1}^{j} D_i'(j,k) & j \neq 1, k=1 \\ \sum_{k=1}^{k} D_i'(j,k) & others \end{cases} \tag{7}$$

Here, $i = 1,2,...,r;\quad j,k = 1,2,...,N/2$.

3 The Proposed Method II: Priority Sharing Mode

The image secret sharing scheme based on bit-level decomposition was originally proposed by Lakac et al. [10-12]. Wang et al. [7] followed their idea proposed the priority sharing mode for image secret sharing. In order to provide a loss-less reconstruction method, we improved the method of literature [7], present an image secret sharing method with loss-less priority sharing mode and smaller shadow images, and the algorithm steps are as follows:

(1) Image pre-processing

① Dividing the secret image *I* into *n* bit-level images $P_0, P_1, \cdots, P_{n-1}$ by using bit-level decomposition technique [7], and satisfy:

$$\begin{cases} \bigcup_i B(P_i) = B(I) & 1 \le i \le n \\ B(P_i) \bigcap B(P_j) = \varnothing & 1 \le i \ne j \le n \end{cases} \tag{8}$$

Here $B(I) = \{b_{m-1}, \cdots, b_1, b_0\}$ is the set of bit-level images for a certain pixel in *I*, *m* is the depth of a pixel, b_0 is the least significant bit, and b_{m-1} is the most significant bit. Base upon the characteristics of a pixel value, the bits in *B*(*I*) follows a partial ordering $b_{m-1} > b_{m-2} \ldots > b_0$, where $b_i > b_j$ means b_i is more significant than b_j.

Taking (2, 4)-threshold scheme and 8-level decomposition for example. Dividing the original image *I* into $P_0, P_1, P_2, P_3, P_4, P_5, P_6, P_7$, and combing them to generate four combined bit-level images I_1, I_2, I_3, I_4. Fig.4 and Fig.5 show two combined models, the depth of each pixel in Fig.4 and Fig.5 is only 2. Fig.4 shows combined model I: $B(I_1) = \{b_7, b_6\}, B(I_2) = \{b_5, b_4\}, B(I_3) = \{b_3, b_2\}$, $B(I_4) = \{b_1, b_0\}$. Fig.5 shows combined model II: $B(I_1) = \{b_7, b_3\}, B(I_2) = \{b_6, b_2\}$, $B(I_3) = \{b_5, b_1\}$, $B(I_4) = \{b_4, b_0\}$.

Fig. 3. Original image (128×128)

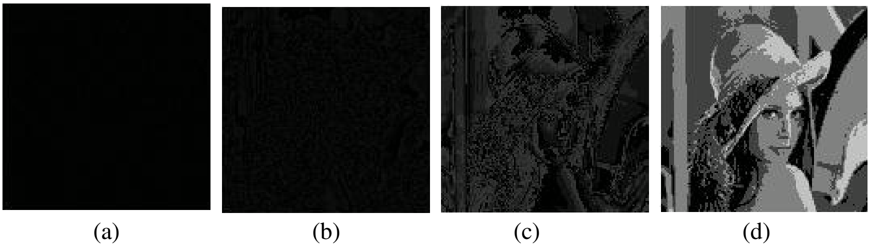

(a) (b) (c) (d)

Fig. 4. Combined model I: (*a*)-(*d*) are combined bit-level images $I_1,...,I_4$

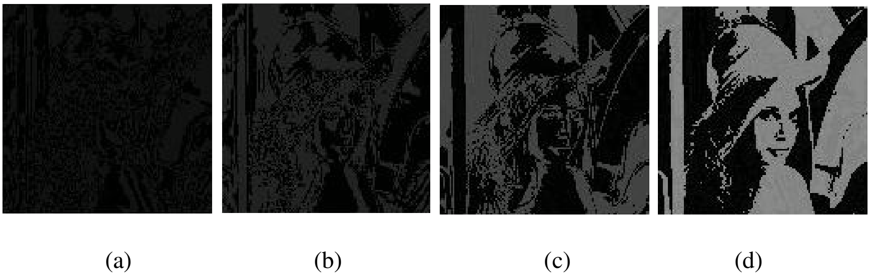

(a) (b) (c) (d)

Fig. 5. Combined model II: (*a*)-(*d*) are combined bit-level images $I_1,...,I_4$

② Dividing $B(I_i)$ (i=1,2,3,4) into 2×2 image blocks, denote as:

$$g^i_{jk} = \begin{bmatrix} b_{2j-1,2k-1} & b_{2j-1,2k} \\ b_{2j,2k-1} & b_{2j,2k} \end{bmatrix} \quad (j,k=1,2,\ldots,N/2) \tag{9}$$

We denote the minimum value of g^i_{jk} as m^i_{jk}, that is $m^i_{jk} = \min\{b_{2j-1,2k-1}, b_{2j-1,2k}$ $b_{2j,2k-1}, b_{2j,2k}\}$, and define matrix M^i (the size is $N/2\times N/2, i$=1,2,3,4) that are constructed by m^i_{jk} , and reconstruct images IM^i by M^i . Defining matrix D^i that would be used in recovery phase, the elements of D^i are as follows:

$$d^i_{jk} = \begin{bmatrix} b_{2j-1,2k-1} - m^i_{jk} & b_{2j-1,2k} - m^i_{jk} \\ b_{2j,2k-1} - m^i_{jk} & b_{2j,2k} - m^i_{jk} \end{bmatrix} \quad (j,k=1,2,\ldots,N/2) \tag{10}$$

③ Dividing IM^i (i=1, 2, 3, 4) into 8×8 image blocks E^i_u , u=1, ..., N/16×N/16, and computing the difference matrix $diff^i_{jk}$ for each 8×8 image block by formula (11):

$$diff^i_{jk} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{18} \\ a_{21} & a_{22} & \cdots & a_{28} \\ \vdots & \vdots & \vdots & \vdots \\ a_{81} & a_{82} & \cdots & a_{88} \end{bmatrix} (i=1,\ 2,\ 3,\ 4) \tag{11}$$

Here, $$a_{st}=\begin{cases} E_u^i(s,t) & s=1,t=1 \\ E_u^i(s,t)-E_u^i(s-1,t) & s\neq 1,t=1 \\ E_u^i(s,t)-E_u^i(s,t-1) & others \end{cases} \quad (s,t=1,2,...,8) \tag{12}$$

④ Combining $diff_{jk}^i$ to generate matrix $Diff_i$:

$$Diff_i=\begin{bmatrix} diff_{11}^i & diff_{12}^i & \cdots & diff_{1,N/16}^i \\ diff_{21}^i & diff_{22}^i & \cdots & diff_{2,N/16}^i \\ \vdots & \vdots & \vdots & \vdots \\ diff_{N/16,1}^i & diff_{N/16,2}^i & \cdots & diff_{N/16,N/16}^i \end{bmatrix} (i=1,2,3,4) \tag{13}$$

⑤ Re-arranging elements of $Diff_i$ by row to generate a row vector, and compressing it by Huffman coding, then converting the output into 0~255 decimal numbers, noted as $C(I_1),C(I_2),C(I_3),C(I_4)$.

(2) The shadow generation algorithm

This algorithm involving two steps: first, for each $C(I_i)$ (*i*=1,2,3,4), (4,6)-threshold scheme is used to generate six sub-shadow images, then combining the obtained 24 sub-shadow images to generate four shadow images SH_1,SH_2,SH_3,SH_4.

① Taking every four pixels, in sequence, from $C(I_i)$ (*i*=1,2,3,4) as a sharing share, denoted as $\{a_0,a_1,a_2,a_3\}$, then construct Lagrange polynomial for each sharing share:

$$q_k^i(x)=\left(a_0+a_1x+\cdots+a_3x^3\right) \bmod 2^t \tag{14}$$

Here, $k=1,2,3,4$, *i*=1,..., ||*C*(*I_i*)||/4, ||·|| is norm operation. Take $q_k^i(x)$ as one pixel of $s_k(x)$, generate six sub-shadow images $s_k(x)$, $x=1,2,...,6$.

② Combining the 24 sub-shadow images obtained from step ①, then generate four shadow images SH_1,SH_2,SH_3,SH_4, the algorithm is described in Table 2.

Table 2. The algorithm of generating shadow images SH_1,SH_2,SH_3,SH_4

Shadow image	$C(I_1)$	$C(I_2)$	$C(I_3)$	$C(I_4)$
SH_1	$s_1(1),s_1(2),s_1(3)$	$s_2(1)$	$s_3(1)$	$s_4(1)$
SH_2	$s_1(4)$	$s_2(2),s_2(3),s_2(4)$	$s_3(2)$	$s_4(2)$
SH_3	$s_1(5)$	$s_2(5)$	$s_3(3),s_3(4),s_3(5)$	$s_4(3)$
SH_4	$s_1(6)$	$s_2(6)$	$s_3(6)$	$s_4(4),s_4(5),s_4(6)$

(3) Secret image reconstruction algorithm

① Taking one pixel in sequence from each of the *r* (2≤*r*≤4) shadow images (they are a subset of $\{q_k^i(1), q_k^i(2), ..., q_k^i(6)\}$), and using Lagrange's interpolation to recover the coefficients $a_0, ..., a_3$ in formula (14).

② Converting the coefficients $a_0, ..., a_3$ into binary numbers, then use Huffman decoding to recover the difference bit-level images $G_1', G_2', ..., G_r'$, their size is *N/2*×*N/2*.

③ Using inverse-difference transformation shown in formula (15) to obtain the bit-level images $M_1', M_2', ..., M_r'$.

$$M_i'(j,k) = \begin{cases} G_i'(j,k) & j = 1, k = 1 \\ \sum_{j=1}^{j} G_i'(j,k) & j \neq 1, k = 1 \\ \sum_{k=1}^{k} G_i'(j,k) & others \end{cases} \tag{15}$$

Here $i = 1,2,...,r;\ j,k = 1,2,...,N/2$.

④Accumulating M_i' and the corresponding matrix D^i to obtain I_i' by formula (16), and obtain the recovered secret image I' by formula (17). Recovered secret image is lossless since the action of D^i .

$$\begin{cases} I_i'(j,k) = M_i'(j,k) + D^i(j,k) \\ I_i'(j,k+1) = M_i'(j,k) + D^i(j,k) \\ I_i'(j+1,k) = M_i'(j,k) + D^i(j+1,k) \\ I_i'(j+1,k+1) = M_i'(j,k) + D^i(j+1,k+1) \end{cases} \tag{16}$$

Here, $i = 1,2,...,r;\ j,k = 1,2,...,N/2$.

$$I' = I_1' + I_2' + ... + I_r' \tag{17}$$

4 Experimental Results

4.1 The Experimental Results for Method I

In this section, some experimental results are given to show the validity of the proposed method. In experiments, we use 128×128 gray-level image as the secret image (see Fig.6 (*a*)), and divide the secret image into four sub-images (see Fig.6 (*b*)).

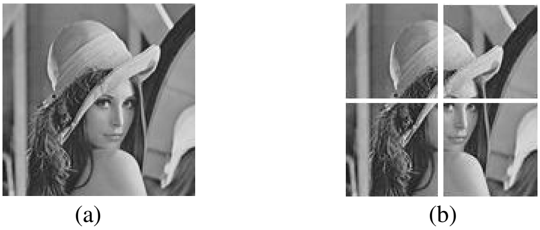

(a) (b)

Fig. 6. (a) Original secret image; (*b*) Four sub-images

By using the algorithm of section 2.1.2, four shadow images are generated (see Fig.7 (*a*), (*b*), (*c*), (*d*)), the size of each shadow image is 14×192, which is only 30% of the original secret image.

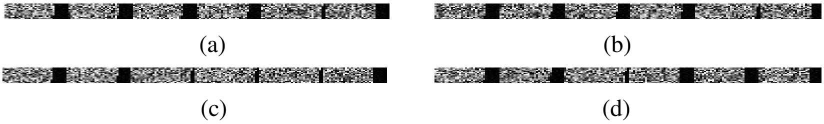

(a) (b)
(c) (d)

Fig. 7. Four shadow images generated by using the multi-secret sharing mode

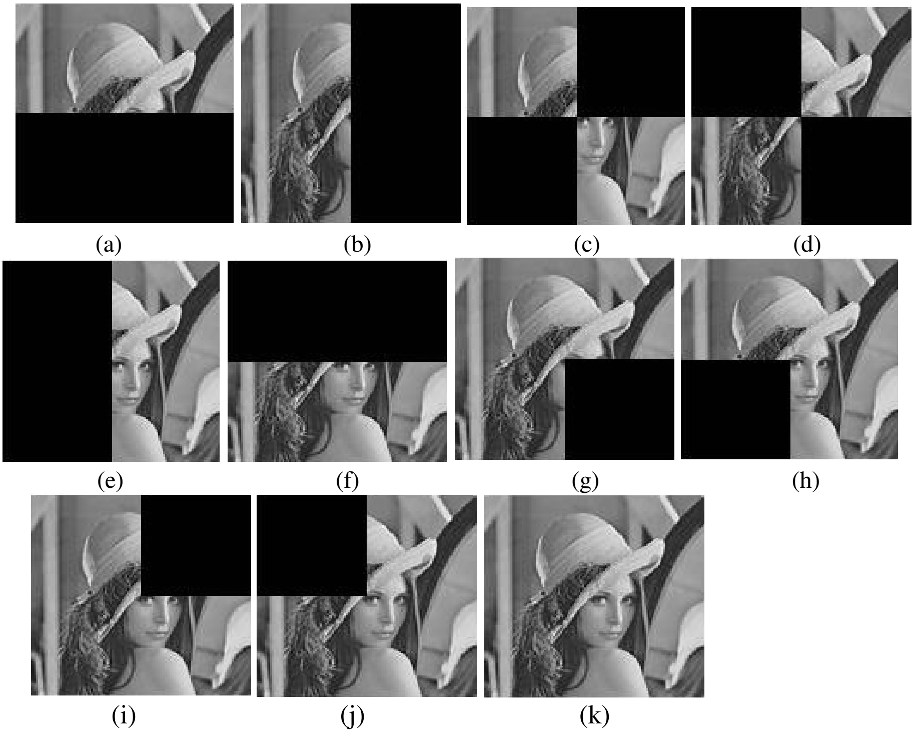

(a) (b) (c) (d)
(e) (f) (g) (h)
(i) (j) (k)

Fig. 8. Reconstructed images by using different numbers of shadow images in multi-secret sharing mode, where (*a*)-(*f*) are recovered images by using two shadow images, (*g*)-(*j*) are recovered images by using three shadow images, and (*k*) is recovered image by using four shadow images

Fig.8 shows the recovered secret images by the multi-secret sharing mode. Fig.8 (*a*)-(*f*) are images reconstructed from any two of the four shadow images in Fig.7. They are the half of the secret image. The images reconstructed from any three of the shadow images of Fig.7 are shown in Figs.8 (*g*)-(*j*), and each one is three-quarters of the secret image. Fig.8 (*k*) is the image reconstructed using all of the shadow images of Fig.7, and is a lossless version of the original secret image. As can be seen from the experimental results, the number of recovered sub-images is proportional to the number of used shadow images.

4.2 The Experimental Results for Method II

We still use 128×128 gray-level image as the secret image. First, we divide the secret image into bit-level images by model I in section 2.2 (see Fig.4). We compress every four neighboring pixels of each combined bit-level image into one pixel by formula (9). The size of the compressed images is one quarter of the original secret image. Fig.9 (*a*)-(*d*) are the results by compressing Fig.4 (*a*)-(*d*).

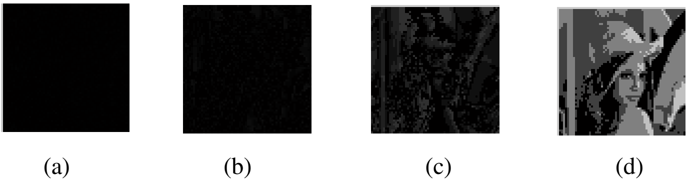


Fig. 9. (*a*)-(*d*) are compressed bit-level images by compressing Fig.4 (*a*)-(*d*), respectively

The shadow images SH_1, SH_2, SH_3, SH_4 generated by the shadow generation algorithm described in section 2.2 are shown in Fig.10, the size of each shadow image is 6×192, which is only 15% of the original secret image.

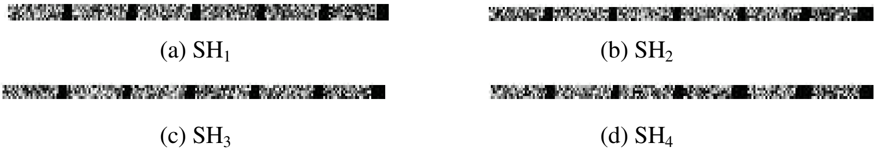


Fig. 10. Shadow images generated by using the priority sharing mode

Fig.11 shows the recovered secret images by using priority sharing mode. Fig.11 (*a*)-(*f*) are reconstructed images by using any two of SH_1, SH_2, SH_3, SH_4; Fig.11(*g*)-(*j*) are reconstructed images by using any three of SH_1, SH_2, SH_3, SH_4; Fig.11(*k*) is the reconstructed image by using four shadow images SH_1, SH_2, SH_3, SH_4. As can be seen from Fig.11, the quality of the reconstructed images are different, it is related to the shadow images number and their significance. When the shadow images that are used to reconstruct the secret image have higher significance, a higher resolution reconstructed image can be obtained.

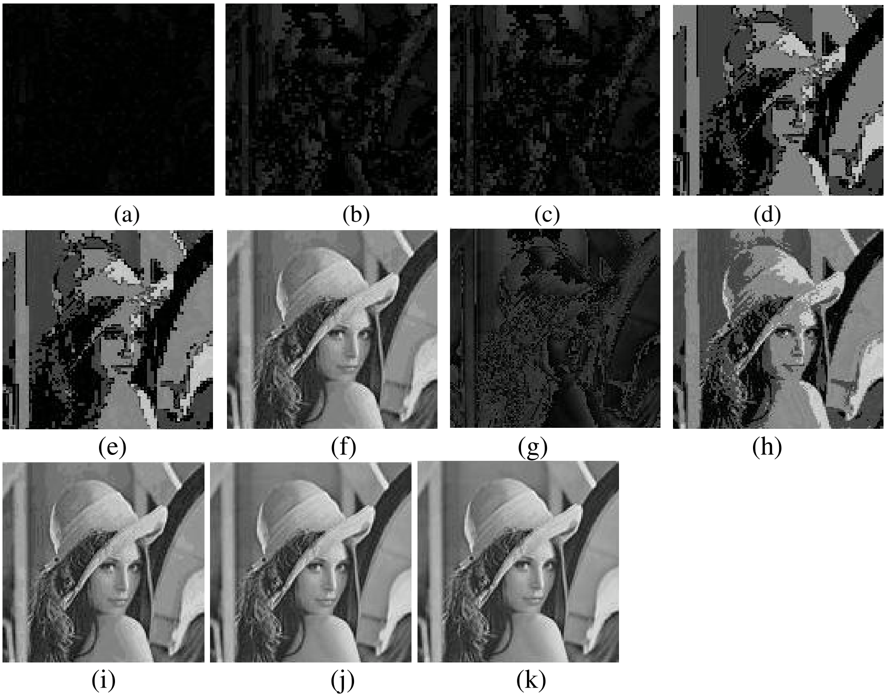


Fig. 11. The reconstructed images by using priority sharing mode. (*a*)-(*f*) are reconstructed images by using two shadow images, (*g*)-(*j*) are reconstructed images by using three shadow images, and (*k*) is reconstructed image by using four shadow images

Comparing the shadow image sizes of the proposed methods with that of the existing methods [3], [5], and [8], the results are list in Table 3. As can be seen from Table 3, comparing with the method [3], [5], and [8], the sizes of the shadow images generated by proposed methods are smaller. They are only 30% (method I) and 15% (method II) of the original secret image.

Table 3. The size of the shadow images generated by proposed methods and the existing methods

Methods	Secret image	Shadow image (bits)
Thien [3]	128×128	128×64=8192
Wang'07 [5]	128×128	4915
Wang'10 [8]	128×128	32×192=6144
Proposed method I (Multi-secret sharing mode)	128×128	14×192=2688
Proposed method II (Priority sharing mode)	128×128	6×192=1152

5 Conclusions

Image secret sharing techniques have wide application prospects in the image content protection, digital rights management, Steganography communication, etc. In this paper, using priority sharing mode and multi-secret sharing mode [7], we have proposed two (2, 4)-threshold image secret sharing methods. Comparing with [5] and [8], our methods provide smaller shadow images. In image secret sharing scheme, the size of shadow images is important. Small shadow images are necessary for saving storage space and transmission time, it is also required for hiding shadow images in host images, and storing shadow images on distributed web-based servers.

References

1. Wang, Y., Zhang, T., Huang, J.: Information Hiding-Theory and Technology. Tsinghua University Press, Beijing (2006)
2. Naor, M., Shamir, A.: Visual Cryptography. In: De Santis, A. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 1–12. Springer, Heidelberg (1995)
3. Thien, C.C., Lin, C.: Secret image sharing. Computer & Graphic 226(5), 765–770 (2002)
4. Lin, C., Tsai, W.H.: Secret image sharing with steganography and authentication. The Journal of system and software 73(3), 405–414 (2004)
5. Wang, R.Z., Su, C.H.: Secret image sharing with smaller shadow images. Pattern Recognition Letters 27(6), 551–555 (2006)
6. Li, B.: A Reliable (*k*, *n*) Image Secret Sharing Scheme. In: Proceedings: Second International Symposium on Dependable Autonomic and Secure Computing, Indianapolis, Indiana, USA, pp. 31–36 (2006)
7. Wang, R.Z., Shyu, S.J.: Scalable secret image sharing. Signal Processing: Image Communication 22(4), 363–373 (2007)
8. Lin, Y.Y., Wang, R.Z.: Scalable Secret Image Sharing with Smaller Shadow Images. IEEE Signal Processing Letters 17(3), 316–319 (2010)
9. Li, B.: A strong ramp secret sharing scheme using matrix projection. In: Proceedings: 2006 International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2006), New York, USA, pp. 652–656 (2006)
10. Lukac, R., Plataniotis, K.N.: A cost-effective encryption scheme for color images. Real–Time Image 11(5-6), 454–464 (2005)
11. Lukac, R., Plataniotis, K.N.: Image representation based secret sharing. Communications of the CCISA, Special Issue on Visual Secret Sharing 11(2), 103–114 (2005)
12. Lukac, R., Plataniotis, K.N.: Bit-level based secret sharing for image encryption. Pattern Recognition 38(5), 767–772 (2005)

An Algorithm for *k*-Anonymity-Based Fingerprinting

Sebastian Schrittwieser[1], Peter Kieseberg[2], Isao Echizen[3], Sven Wohlgemuth[3], Noboru Sonehara[3], and Edgar Weippl[2]

[1] Vienna University of Technology, Austria
sebastian.schrittwieser@tuwien.ac.at
[2] SBA-Research, Austria
{pkieseberg,eweippl}@sba-research.org
[3] National Institute of Informatics, Japan
{iechizen,wohlgemuth,sonehara}@nii.ac.jp

Abstract. The anonymization of sensitive microdata (e.g. medical health records) is a widely-studied topic in the research community. A still unsolved problem is the limited informative value of anonymized microdata that often rules out further processing (e.g. statistical analysis). Thus, a tradeoff between anonymity and data precision has to be made, resulting in the release of partially anonymized microdata sets that still can contain sensitive information and have to be protected against unrestricted disclosure. Anonymization is often driven by the concept of k-anonymity that allows fine-grained control of the anonymization level. In this paper, we present an algorithm for creating unique fingerprints of microdata sets that were partially anonymized with k-anonymity techniques. We show that it is possible to create different versions of partially anonymized microdata sets that share very similar levels of anonymity and data precision, but still can be uniquely identified by a robust fingerprint that is based on the anonymization process.

Keywords: k-anonymity, fingerprinting, generalization, algorithm.

1 Introduction

When releasing microdata containing sensitive information such as medical health data for research purposes, a tradeoff between data privacy and data quality has to be made. On the one hand, completely anonymized data records are often too generalized to be useful for further processing (e.g. statistical analysis), on the other hand, anonymization is desirable and often even demanded to achieve regulatory compliance such as Directive 95/46/EC in the European Union or the Health Insurance Portability and Accountability Act (HIPAA), which regulates the processing of medical health data in the United States. k-anonymity [8] is a technique that allows defining a fine-grained level of anonymity for microdata. Although partially anonymized data is usually protected by special usage restrictions, both technical and organizational misuse (e.g. unauthorized disclosure) can not be eliminated. Fingerprinting is a passive form of security, meaning

Y.Q. Shi, H.J. Kim, and F. Perez-Gonzalez (Eds.): IWDW 2011, LNCS 7128, pp. 439–452, 2012.

that it can help to identify the source of disclosure after it took place. In this paper, we take our idea of using k-anonymity techniques for fingerprinting partially anonymize microdata presented in [6] and introduce an algorithm for the generation of a batch of partially anonymized microdata sets that share the same levels of anonymity and similar levels of data precision while being uniquely identifiable by a robust fingerprint. A typical application scenario for our approach is the release of partially anonymized microdata sets to multiple receivers (e.g. research institutions, universities, etc.). Instead of delivering identical copies, for each receiver the microdata is anonymized in a slightly different way. The anonymization process inherently generates a fingerprint that is contained in every single record of a given set and unique for each set. Thus, in the case of unauthorized data disclosure, it is possible to identify the source of the leak based on the fingerprint.

The rest of the paper is structured as follows. Section 2 summarizes the concept of k-anonymity and our idea of fingerprinting partially anonymized microdata presented in [6]. Our proposed algorithm is explained in Section 3.1. In Section 4, we evaluate security aspects, including the problem of inference attacks, and finally, we conclude in Section 5.

2 Related Work

2.1 *k*-Anonymity

Sweeney [5,8] showed that even after removing uniquely identifying attributes (e.g., name or social security number) from medical health data, people can still be identified by so-called quasi-identifiers (QI), i.e. attributes such as ZIP code, birthdate, and sex that can be linked in order to break anonymization. Her introduced concept of k-anonymity is a widely adopted anonymization technique. The k-anonymity criterion is satisfied, if each record is indistinguishable from at least *k-1* other records with respect to the quasi-identifiers. Hence, quasi-identifying attributes must have the same values within an equivalence class, so that it is impossible to uniquely link a person to a specific record within the class. By raising the value of k, high levels of anonymity can be achieved with this anonymization technique, however, to maintain significance of the data, often lower anonymization levels need to be chosen.

Over the past years, several improvements to the idea of k-anonymity were proposed. ℓ-diversity tries to enhance anonymity in cases where little diversity in sensitive attributes occurs by requiring that each equivalence class has at least ℓ well-represented values for each sensitive attribute [3]. Another improvement to k-anonymity is t-closeness that requires that the distribution of a sensitive attribute in any equivalence class is close to the distribution of the attribute in the original table [2]. Both, ℓ-diversity and t-closeness are additional, tightened criteria to k-anonymity, but do not replace its original idea. Our proposed algorithm is explained using the original k-anonymity criterion. However, an adaption to ℓ-diversity and t-closeness would not require any modifications of the algorithm.

Generalization is the main strategy for achieving k-anonymity. Hereby, data granularity is reduced to unify the data records. Figure 1 shows possible generalizations for two different quasi-identifiers. The characters a and b define the two quasi-identifiers *sex* and *birthdate* that have different levels of generalization (described by the numbers 0 to 2).

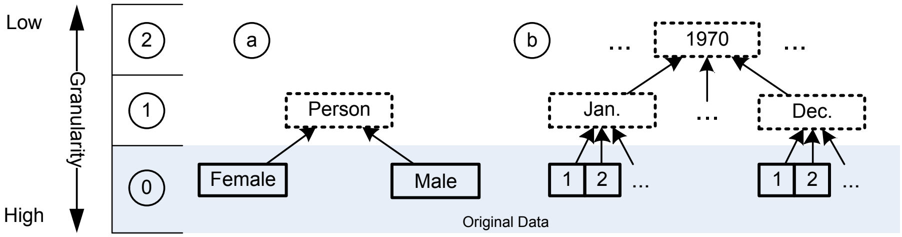


Fig. 1. Generalization strategies for two quasi-identifiers

2.2 Data Precision Metrics

Generalization almost always results in the loss of information. To be able to judge the quality of a resulting generalized data set, metrics that make this loss of information measurable need to be devised. We will use this definition of quality for generating the classes of equivalent generalization strategies, since we target at providing all recipients with almost the same data quality.

Samarati Metric. The Samarati Metric [4] is defined by simply adding the generalization levels of the quasi-identifiers. This approach has the positive effect that it is very intuitive and easy to calculate, but has a serious drawback: This metric only counts the generalization steps without taking the total number of possible generalizations for an identifier into account.The generalization from {female, male} to {person} results in a much higher information loss than generalizing a date (e.g. birthdate) from a timestamp-granularity to the day-granularity. In the Samarati Metric both operations may yield the same information loss.

Precision Metric[7]. contrary to the Samarati Metric, not every generalization possesses the same weight, but this is calculated with respect to the maximum generalization depth possible for a quasi-identifier. In case an identifier possesses more than one level of generalization, the weight of generalizing is defined by

$$\frac{\text{Number of levels generalized}}{\text{Number of possible generalization levels}}$$

Example: For the quasi-identifier *birthdate* (original granularity: timestamp), three levels of generality are defined: {day, month, year}. Thus generalizing birthdate from *timestamp*-level to *day*-level results in a weight of $\frac{1}{3}$. Contrary, generalizing the quasi-identifier *sex* from {female, male} to {person} still results in a weight of 1. One drawback of the Precision Metric lies in the fact that it is completely unrelated to the actual data (a drawback that is valid for the Samarati Metric as well).

*Modified Discernability Metric DM**. This is a slightly modified version of the Discernability Metric and was proposed by El Emam et al. in [1]: N denotes the number of classes and n_i the number of elements of the i-th class. Then the Modified Discernability Metric DM* is defined by $DM^* = \sum_{i=1}^{N} n_i^2$.

The big advantage of this metric lies in the fact that the sizes of the resulting classes are incorporated into the data quality.

Our research has shown that in general a perfect metric does not exist and that the practicability of the results heavily depends on the source data, the types of attributes and the generalization patterns. In real life scenarios, the choice for a specific data precision metric has to be based on these parameters.

2.3 Algorithm for Finding the Optimal Solution

All possible generalization strategies can be depicted by the lattice diagram shown in Figure 2. A node in the diagram is generated by generalizing one quasi-identifier by one level, i.e. the diagram shows all generalizations that can be derived by generalizing by one level at once and their relations. For example, a_1 refers to the second generalization step of the quasi-identifier a.

El Emam et. al. [1] proposed an algorithm for calculating the optimal generalization strategy (i.e. the generalization with the highest data precision) with respect to a given metric.

The algorithm finds all k-anonymous nodes for each generalization-strategy. The algorithm uses a technique called *predictive tagging* for reducing the workload by being able to tag nodes in the lattice diagram without directly calculating their level of anonymity: Since a metric fulfills the axiom of monotony, the two statements hold true:

– If a node in the diagram is k-anonymous, then all nodes above it in the diagram are at least k-anonymous too, i.e. by further generalizing a k-anonymous classification, the k-anonymity is not lost.
– If a node in the diagram is not k-anonymous, then all nodes below it in the diagram can not be k-anonymous.

When all generalization-strategies are evaluated, the globally lowest node is chosen.

2.4 Fingerprinting

The term watermarking defines techniques that add visible or hidden information (e.g. a copyright notice) to the target data. The important aspect in this definition is that adding a watermark modifies the target data. This modification can either be visible (e.g. a text overlay in an image file) or invisible (e.g. by implementing steganographic techniques) to the user. In both cases, the watermark information is additional data that is combined with the target data. In contrast to watermarking, the definition of fingerprinting is not consistent among the research community. There exist at least two definitions. The first

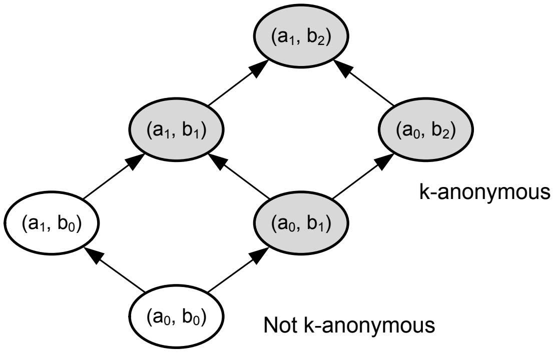


Fig. 2. Generalization strategies for two quasi-identifiers

one describes fingerprinting as a subtype of watermarking where a unique watermark (i.e. the fingerprint) is added to each copy of the target data. The second definition distinguishes fingerprinting from watermarking by the source of the fingerprint. While in watermarking, information is actively added, fingerprinting uses intrinsic properties of the data to uniquely differentiate the copies of the data. In both definitions, however, the uniqueness of the fingerprint is the key concept that enables a data owner to uniquely link a data customer to a specific file. Our schema is based on the idea of extracting unique fingerprints from the data structure, thus follows the second definition.

In the past, the concept of fingerprinting microdata was discussed by Willenborg and de Waal [9] and Willenborg and Kardaun [10]. In both approaches, fingerprints are built from combinations of identifying variables in the microdata records and are used for identifying specific records in a set of microdata.

2.5 Extracting Unique Fingerprints from Generalization Patterns

This subsection summarizes our previous work on devising a fingerprinting technique based on k-anonymity [6]. This technique aims at enabling the identification of a source microdata set based on the analysis of a single record. Thus, the fingerprint is the same for every record in a given set and unique for each set. The scenario for our method is the release of sensitive microdata (e.g. medical health records) to multiple receivers (e.g. universities). The idea is that each receiver gets a differently anonymized data set that can be uniquely identified by a single record based on this fingerprint.

When sensitive microdata sets are anonymized using k-anonymity techniques, the choice of generalization levels for the quasi-identifiers influences the value of k and the quality of the anonymized data. In general, data quality decreases and k increases with higher generalization levels as generalization removes information expressiveness of the data and more records look the same with respect to the quasi-identifiers, resulting in larger equivalence classes.

The fingerprints in our approach are formed by the generalization levels of quasi-identifiers used for anonymization. That generalization levels are chosen differently for each released microdata set in a way, that the data quality values of the datasets are all roughly the same. Our proposed algorithm identifies similar anonymization strategies; strategies that generate anonymized datasets that meet a specified level of k and share similar data quality. Figure 3 explains our approach of generating anonymized data sets with different generalization strategies. All candidate data sets above the defined threshold of k are compared and clustered with respect to data quality. Data sets from one cluster are then released to different data receivers. The generalization strategy used for a dataset can be extracted from every single record, thus allowing the identification of the source data set.

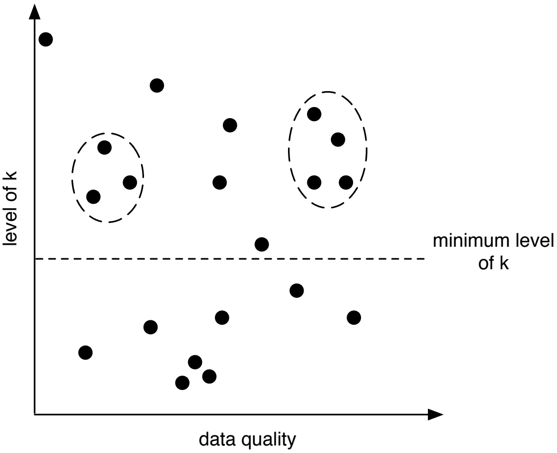


Fig. 3. Clustering similar anonymization strategies

Table 1. Original data and two anonymized sets ($k = 2$)

Original data				First Set			Second Set		
name	sex	birthdate	disease	sex	birthdate	disease	sex	birthdate	disease
Bob	m	19.03.1970	chest pain	F	1970	chest pain	P	03.1970	chest pain
Dave	m	20.03.1970	short breath	M	1970	short breath	P	03.1970	short breath
Alice	f	18.04.1970	obesity	F	1970	obesity	P	04.1970	obesity
Eve	f	21.04.1970	short breath	M	1970	short breath	P	04.1970	short breath

3 Approach

In this section we propose an algorithm for the generation of unique fingerprints for microdata sets and an algorithm for the identification of the source of a data leak.

3.1 Algorithm for Generating Fingerprints

Our approach is based on El Emam's algorithm for calculating the optimal solution (see Section 2.3). Since we need all k-anonymous solutions instead of just the optimal one (i.e. the one with minimum information loss), we cannot rule out the more general solutions of a solution found, but still have to calculate the data precision for each of these nodes.

1. Define a minimum k for the k-anonymity criterion, the minimum and maximum levels of data loss l_{min} and l_{max} and the data precision metric to be used.
2. Define the generalization strategies for each identifier.
3. Calculate the lattice diagram derived from all possible generalizations.
4. Choose a node at middle height and decide whether it is at least k-anonymous.
   (a) In case it is not, rule out all nodes below in the lattice diagram.
   (b) In case it is, mark all nodes above the chosen one as possible solutions.
5. Start with step four for the remaining subgraph, similar to the original algorithm.
6. In case no subgraph is left, start by choosing another initial node at middle height and proceed with step four until all nodes are evaluated.
7. For each at least k-anonymous solution, calculate data precision and the actual k. Remove all solutions with data precision outside the bounds of l_{min} and l_{max}.
8. Classify and cluster the solutions by their data precision.
9. Create "similar" microdata sets based on results in one cluster and distribute them to the recipients.

Following we give further details on some of the steps:

Data Precision Metrics and Maximum Levels (Step 1). All recipients shall be provided with (roughly) the same level of data precision. Thus, the method of measuring data precision can have a great impact on the definition of the classes of equivalent anonymization strategies (with respect to data quality). Table 2 gives an overview on the perceived data precision of the data from Table 1 based on all identified generalization techniques (see Figures 1 and 2), with respect to the three metrics discussed in Section 2.2. Additionally, it must be decided beforehand, how big the maximum tolerable data loss is.

Table 2. Impact of different metrics

Node	Sam	Prec	DM*	k
(a_0, b_0)	0	0	4	1
(a_1, b_0)	1	1	4	1
(a_0, b_1)	1	0.5	8	2
(a_0, b_2)	2	1	8	2
(a_1, b_1)	2	1.5	8	2
(a_1, b_2)	3	2	16	4

Eliminating Nodes (Step 4). Contrary to the algorithm for finding the optimal solution as proposed in 2.3, the nodes above the chosen one cannot be ruled out in case of 4.b (i.e. the node represents a k-anonymous generalization), since we need all solutions, not only the optimal one.

Clustering the Solutions (Step 8). In step eight the solutions are clustered by data loss. As discussed in Section 2.2, the data precision metric can have a great impact on the measured data loss and thus on the resulting clustering. Some metrics lead to finer grained results, thus no real clustering can be achieved by using the equality-function (see the case using the precision metric in Table 2). Here, a reasonable form of discretization (e.g. rounding or defining intervals) needs to be applied in order to generate usable classes.

Final Distribution (Step 9). Before distribution, a list containing all assignments from generalization patterns to recipients is generated. We will later refer to this list as the *pattern-list*.

In section 3.3 we provide a detailed step-by-step-walkthrough of this algorithm.

3.2 Principle of the Identification of Data Leaks

In this section we introduce the algorithm for detecting data leaks (see Figure 4 for an illustration). The scenario is based on a data holder encountering data samples in the wild that he gave to other organizations in an anonymized (and fingerprinted) form.

1. The original data holder encounters some leaked data samples.
2. The underlying generalization pattern of this data is extracted.
3. The pattern is compared to the pattern list.
   (a) In case the pattern directly matches the pattern of the data given to a receiver, this receiver is identified as the source of disclosure.
   (b) Otherwise, calculate the minimum of receivers that possess the knowledge (i.e. data) to be able to generate the encountered data set structure.

Figure 4 explains the approach.

Example Original Data D_0 is anonymized using two different patterns, (a_1, b_1) and (a_0, b_2) thus generating the anonymized data sets D_1 and D_2 respectively. Recipient R_1 is provided with D_1 and R_2 with D_2. If, for example, a data record of the form (a_0, b_2) is disclosed (i.e. in our examples this would be a data record where "sex" is provided at original granularity and "birthdate" at year-granularity), it is easy to identify user U_2 as source of the leaked data, since user U_1 would not be able to provide this much detail on the QI "sex".

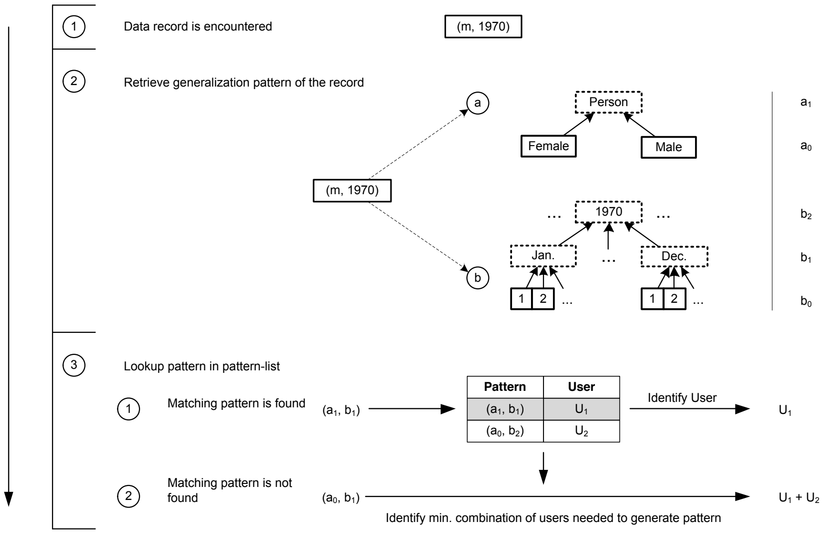

1
Data record is encountered
(m, 1970)
2
Retrieve generalization pattern of the record
(m, 1970)
a
b
Person
Female
Male
1970
Jan.
Dec.
1
2
...
a1
a0
b2
b1
b0
3
Lookup pattern in pattern-list
1
Matching pattern is found
(a1, b1)
Pattern
User
(a1, b1)
U1
(a0, b2)
U2
Identify User
U1
2
Matching pattern is not found
(a0, b1)
U1 + U2
Identify min. combination of users needed to generate pattern

Fig. 4. Identifying the source of data leakage based on data-patterns

3.3 Step-by-Step Description

For this example, we use the microdata from Table 1. In step one we define the side constraints $k = 2$, $l_{min} \geq 1$, and $l_{max} \leq 2$ (minimal and maximal level of data loss) and choose the Samarati metric for measuring data precision. In step two and three we use the generalization strategy from Figure 1, yielding to the lattice diagram shown in Figure 5.

Analogous to El Emam's algorithm, we then choose a node of middle height as starting node for step four, e.g. (a_1, b_0). This generalization does not fulfill the $2 - anonymity$ criterion, thus we can rule out all nodes below (a_1, b_0) (in our example, this applies to one node only: (a_0, b_0)), because they would not fulfill the criterion as well (Step 4a). We now take the resulting subgraph of the

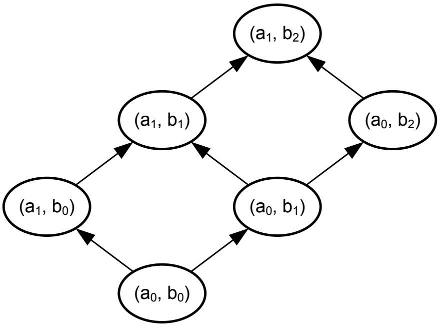


Fig. 5. Lattice diagram after step three

Table 3. Generalization (a_1, b_0)

Sex (a)	Birthdate (b)
p	19.03.1970
p	20.03.1970
p	18.04.1970
p	21.04.1970

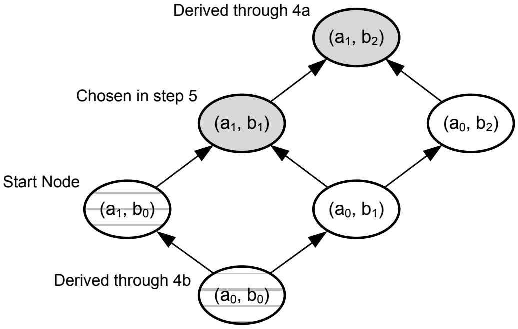


Fig. 6. Lattice diagram after the first generalization path

chosen generalization path (Step 5) and apply Step 4 to node (a_1, b_1), which fulfills $k = 2$, and thus (by applying Step 4b) the node (a_1, b_2) fulfills the 2-anonymity criterion as well. This leads to the intermediate lattice diagram shown in Figure 6.

As no subgraph is left, we arrive at Step 6 and again choose a starting node (a_0, b_1). It does provide 2-anonymity, so we can continue with Step 4b and and mark node (a_0, b_2) as a possible solution. The resulting lattice diagram is shown in Figure 7.

Since we now have traversed through the entire lattice diagram, we can proceed with Step 7 and calculate the data precision and the actual k for each node that fulfills the 2-anonymous criterion (see Table 4).

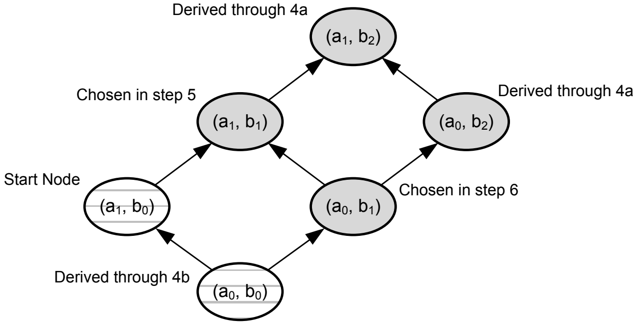


Fig. 7. Lattice diagram after the second generalization path

Table 4. Data loss and k for solutions

Node	Data Loss	k
(a_0, b_1)	1	2
(a_0, b_2)	2	2
(a_1, b_1)	2	2
(a_1, b_2)	3	4

We remove node (a_1, b_2) from the solution set, since the data precision is out of our defined bounds for the data loss l_{min} and l_{max}. The remaining candidate microdata sets are shown in Table 5.

Table 5. Generalization results

(a_0, b_1)		(a_1, b_1)		(a_0, b_2)	
Sex	Birthdate	Sex	Birthdate	Sex	Birthdate
m	03.1970	p	03.1970	m	1970
m	03.1970	p	03.1970	m	1970
f	04.1970	p	04.1970	f	1970
f	04.1970	p	04.1970	f	1970

Finally, by clustering the remaining data sets by data precision, we derive *generalization clusters*. All solutions in such a cluster will be treated as being equivalent with respect to the data quality provided. In some cases (e.g. when the classes are too small), we have to use ranges instead of exact values as classification criteria. In our example we derived two clusters: $C_1 = \{(a_0, b_1)\}$ with data precision of 1 and $C_2 = \{(a_1, b_1), (a_0, b_2)\}$ with the data precision of 2.

In case the data needs to be sent to two receivers, we choose cluster C_2 and send data anonymized with strategy (a_1, b_1) to the first recipient and data anonymized with strategy (a_0, b_2) to the second one. Additionally, a list containing tuples of data receivers and generalization patterns used is stored by the provider. When data is disclosed, the provider can determine the generalization levels of the quasi-identifier attributes and then compare it to generalization the patterns stored. Thus, it is possible to identify the original owner of the data based on the unique generalization variant.

4 Evaluation

In this chapter we focus on the evaluation of the following aspects:

1. Number of possible fingerprints
2. Robustness of the fingerprints
3. Removing the fingerprint by utilization of complementary releases

Number of Fingerprints. When utilizing this approach for fingerprinting, it is very important that the owner of the original data is able to generate enough fingerprints, i.e. there must exist enough suitable anonymization patterns so that each recipient can be given a unique dataset. Unfortunately this number depends on various parameters like number of quasi-identifiers and generalization levels, the anonymization level k, the data precision metric in use, and last but not least the actual data itself. Therefore, an exact number cannot be given without analyzing the source data. Still an upper bound can be retrieved by plainly looking at the quasi-identifiers and their respective generalization strategies.

Be N the number of quasi-identifiers and n_i the number of generalization levels of the i-th QI. Then F is an upper bound for the number of different fingerprints with

$$F = \prod_{i=1}^{N} n_i$$

Robustness of a Fingerprint. In this paragraph we discuss how fingerprints devised by our method can be removed or changed in a way to avoid detection of the disclosing participant. Since the actual granularity of the data is used for fingerprinting, every removal must incorporate changing the generalization patterns. Since the attacker is not in the possession of finer grained data, the only reasonable way would lie in further generalization of one or more quasi-identifiers (actually the attacker could also "invent" finer granulations than he possesses, thus faking the data. Still, this would result in wrong, thus worthless data). This approach results in two major drawbacks:

1. In order to get undetectable, the disclosing party must be in the possession of the generalization strategy that was used for the other recipients, i.e. if U_1 received data with the pattern (a_1, b_2) and U_2 received (a_2, b_1), reducing U_2's granularity to (a_3, b_1) does not avoid U_2's detection.
2. Even if detection is avoided by the disclosing user, the data quality is reduced significantly. In our example, U_1 would at least need to generalize the data to the form (a_2, b_2) which is much more general than the data U_1 would be able to disclose.

Removing Fingerprints through Complementary Releases. Collusion attacks utilizing complementary releases can pose a severe threat to the anonymization of the data sets. Still, our fingerprint can lead to detection of the leaking parties, although it can not be guaranteed anymore.

The following examples shows detection of two collaborating leaking parties. The quasi-identifiers birthdate and sex are generalized like in the previous examples and zip code is granulated on two levels.

Revelation of the record {`03.1970, p, 1015, short breath`} reveals data sets one and two as sources: The month of birth could only be extracted from data set one, the zip code in this granulation only from data set two.

Table 6. Original data and three generalizations

data set	name	sex	birthdate	zip code	disease
original data set	Bob	m	18.03.1970	1004	chest pain
	Dave	m	19.03.1970	1015	short breath
	Alice	f	20.04.1970	1004	obesity
	Eve	f	21.04.1970	1015	short breath
anonymized set 1	-	p	1970	1004	chest pain
	-	p	1970	1015	short breath
	-	p	1970	1004	obesity
	-	p	1970	1015	short breath
anonymized set 2	-	p	03.1970	100X	chest pain
	-	p	03.1970	101X	short breath
	-	p	04.1970	100X	obesity
	-	p	04.1970	101X	short breath
anonymized set 3	-	m	1970	100X	chest pain
	-	m	1970	101X	short breath
	-	f	1970	100X	obesity
	-	f	1970	101X	short breath

5 Conclusion

In this paper, we introduced an algorithm for fingerprinting partially anonymized microdata sets. The main idea is to take the different generalization patterns that can be used to achieve k-anonymity. Our approach is based on an algorithm for finding the optimal solution for the k-anonymity criterion and generates groups of microdata sets that share similar levels of anonymity and data precision.

In contrast to previous work, our fingerprinting method does not aim at identifying a single record from a database, but the identification of the source data set by just analyzing one record out of it. Every single record of the data set stores the same fingerprint that is unique and identifying for the data set it was take from. A typical application scenario for our approach is the release of microdata sets for research purposes to multiple receivers (e.g. universities). In the case of data disclosure, the source can be identified by even a single record.

Future work will focus on the threat of collusion attacks. We aim at constructing collusion-free data sets, i.e. data sets that do not allow to thwart the k-anonymity criterion by combining them.

Acknowledgements. This work was funded by the National Institute of Informatics (NII) International Internship Program, COMET K1, FFG - Austrian Research Promotion Agency and by the Austrian Science Fund (FWF): P12345-B51.

References

1. El Emam, K., Dankar, F., Issa, R., Jonker, E., Amyot, D., Cogo, E., Corriveau, J., Walker, M., Chowdhury, S., Vaillancourt, R., et al.: A globally optimal k-anonymity method for the de-identification of health data. Journal of the American Medical Informatics Association 16(5), 670 (2009)
2. Li, N., Li, T., Venkatasubramanian, S.: t-closeness: Privacy beyond k-anonymity and l-diversity. In: IEEE 23rd International Conference on Data Engineering, ICDE 2007, pp. 106–115. IEEE (2007)
3. Machanavajjhala, A., Kifer, D., Gehrke, J., Venkitasubramaniam, M.: l-diversity: Privacy beyond k-anonymity. ACM Transactions on Knowledge Discovery from Data (TKDD) 1(1), 3 (2007)
4. Samarati, P.: Protecting respondents' identities in microdata release. IEEE Transactions on Knowledge and Data Engineering 13, 1010–1027 (2001)
5. Samarati, P., Sweeney, L.: Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical Report SRI-CSL-98-04, Computer Science Laboratory, SRI International (1998)
6. Schrittwieser, S., Kieseberg, P., Echizen, I., Wohlgemuth, S., Sonehara, N.: Using Generalization Patterns for Fingerprinting Sets of Partially Anonymized Microdata in the Course of Disasters. In: Workshop on Resilience and IT-Risk in Social Infrastructures, RISI 2011 (2011)
7. Sweeney, L.: Achieving k-anonymity privacy protection using generalization and suppression. International Journal of Uncertainty Fuzziness and Knowledge-Based Systems 10(5), 571–588 (2002)
8. Sweeney, L., et al.: k-anonymity: A model for protecting privacy. International Journal of Uncertainty Fuzziness and Knowledge Based Systems 10(5), 557–570 (2002)
9. Willenborg, L., De Waal, T.: Statistical disclosure control in practice. Springer (1996)
10. Willenborg, L., Kardaun, J.: Fingerprints in Microdata Sets. In: Joint ECE/EUROSTAT Work Session on Statistical Data Confidentiality. Working Paper No. 10 (1999)

Contribution of Non-scrambled Chroma Information in Privacy-Protected Face Images to Privacy Leakage

Hosik Sohn[1], Dohyoung Lee[2], Wesley De Neve[1],
Konstantinos N. Plataniotis[2], and Yong Man Ro[1]

[1] Image and Video Systems Lab, Department of Electrical Engineering,
Korea Advanced Institute of Science and Technology
{sohnhosik,wesley.deneve}@kaist.ac.kr, ymro@ee.kaist.ac.kr
[2] Multimedia Lab, The Edward S. Rogers Sr.,
Department of Electrical and Computer Engineering, University of Toronto
dohyoung.lee@utoronto.ca, kostas@comm.utoronto.ca

Abstract. To mitigate privacy concerns, scrambling can be used to conceal face regions present in surveillance video content. Given that lightweight scrambling tools may not protect chroma information in order to limit bit rate overhead in heterogeneous usage environments, this paper investigates how the presence of non-scrambled chroma information in face regions influences the effectiveness of automatic and human face recognition (FR). To that end, we apply three automatic FR techniques to face images that have been privacy-protected by means of a layered scrambling technique developed for Motion JPEG XR, testing the effectiveness of automatic FR and layered scrambling using various experimental conditions. In addition, we investigate whether agreement exists between the judgments of 32 human observers and the output of automatic FR. Our experimental results demonstrate that human observers are not able to successfully recognize face images when simultaneously visualizing scrambled luma and non-scrambled chroma information. However, when an adversary has access to the coded bit stream structure, the presence of non-scrambled chroma information may significantly contribute to privacy leakage. By additionally applying layered scrambling to chroma information, our experimental results show that the amount of privacy leakage can be substantially decreased at the cost of an increase in bit rate overhead, and with the increase in bit rate overhead dependent on the number of scalability layers used.

Keywords: Chroma information, face recognition, Motion JPEG XR, privacy protection, scalability, scrambling, video surveillance.

1 Introduction

Present-day video surveillance systems often come with high-quality video capture capabilities and plenty of computational resources, making it possible to detect and recognize human faces with increasing rates of success. This ability has recently raised privacy concerns [1]. To mitigate these privacy concerns, scrambling can be

Y.Q. Shi, H.J. Kim, and F. Perez-Gonzalez (Eds.): IWDW 2011, LNCS 7128, pp. 453–467, 2012.

leveraged to conceal the identity of face images in video content originating from surveillance cameras.

The past few years have witnessed the development of a wide range of content-based tools for protecting privacy in video surveillance systems. The authors of [2] propose two scrambling techniques to conceal regions-of-interest (ROIs) in video sequences compliant with MPEG-4 Visual: a transform-domain technique that pseudo-randomly flips the sign of selected transform coefficients and a codestream-domain technique that inverts selected bits of codewords in a compressed bit stream. Descrambling privacy-protected ROIs can only be done by authorized users (*i.e.*, users in possession of a secret key), thus ensuring privacy preservation. To hide identity revealing features (e.g., faces or vehicle tags), the authors of [3] propose and evaluate a format-independent encryption scheme that randomly permutes pixel values in each macroblock before compression. The permutation-based encryption scheme tolerates lossy compression and is also robust to transcoding (without requiring access to secret keys). In [4], we introduce a layered scrambling technique that aims at concealing the identity of face images in video sequences scalably encoded with Motion JPEG XR. To that end, we make use of Random Sign Inversion (RSI), Random Permutation (RP), and Random Level Shift (RLS) to scramble the different types of luma subbands that make up face regions. Also, in [5], we present a video surveillance system that makes use of the Scalable Video Coding (SVC) extension of H.264/AVC. To ensure privacy protection, we detect face regions and scramble these regions by means of RSI, making use of a different secret key for each layer in the SVC bit stream. Finally, it is worth mentioning that the authors of [6] introduce a framework for joint encryption and compression for the purpose of content access control. To that end, video data are modified in the frequency domain by means of selective bit scrambling, block shuffling, and block rotation of the transform coefficients and motion vectors.

In general, content-based tools for privacy protection alter the visual information present in privacy-sensitive regions. However, altering visual information typically breaks the effectiveness of coding tools such as entropy coding, thus leading to bit rate overhead. Consequently, content-based tools for privacy protection need to find a proper balance between the level of security offered on the one hand and the amount of bit rate overhead on the other hand. As a result, to limit bit rate overhead, content-based tools for privacy protection may only scramble luma information, leaving chroma information unprotected [4, 7-11].

In this paper, we study and quantify the influence of the presence of non-scrambled chroma information on the effectiveness of automatic and human FR. To that end, following the objective evaluation methodology of [12], we apply three automatic FR techniques to face images that have been privacy-protected in the luma domain by means of a layered scrambling technique developed for Motion JPEG XR [4], testing the effectiveness of automatic FR and layered scrambling using various experimental conditions. In addition, we extend the objective evaluation methodology of [12] in order to investigate whether agreement exists between the judgments of 32 human observers and the output of automatic FR. Note that the objective evaluation methodology of [12] was not available yet at the time of designing and testing the layered scrambling technique for Motion JPEG XR presented in [4].

Note that, besides FR, perception-based security metrics such as Luminance Similarity Score (LSS) and Edge Similarity Score (ESS) can be used to assess the level of privacy protection offered by scrambling [13], as well as more advanced visual security metrics (e.g., the metric outlined in [7], making use of local features). However, these metrics are general in nature and are thus not able to take advantage of domain-specific information (e.g., face information). As a result, these general security metrics may overestimate the level of protection offered by privacy-preserving tools. Therefore, when a privacy-threatening tool such as (automatic) FR can be applied, we believe it is better to make use of face recognition rates in order to assess the level of protection offered by a particular tool for privacy protection.

Our experiments demonstrate that human observers are not able to successfully recognize face images when simultaneously visualizing scrambled luma and non-scrambled chroma information. However, when an adversary has access to the coded bit stream structure, the presence of non-scrambled chroma information may significantly contribute to privacy leakage. By additionally applying layered scrambling to chroma information, our experiments show that the amount of privacy leakage can be substantially decreased at the cost of an increase in bit rate overhead, and with the increase in bit rate overhead dependent on the number of scalability layers used.

This paper is organized as follows. In Section 2, we briefly review layered scrambling for Motion JPEG XR. In Section 3, we describe our experimental setup. We subsequently present and discuss experimental results in Section 4. In Section 5, we detail the costs and benefits of layered scrambling of chroma information. Finally, we provide concluding remarks and directions for future research in Section 6.

2 Layered Scrambling for Motion JPEG XR

The video surveillance system studied in this paper makes use of Motion JPEG XR to encode video content captured by surveillance cameras. Motion JPEG XR offers a low-complexity solution for the intra coding of high-resolution video content, while at the same time offering quality and scalability provisions that are similar to the quality and scalability provisions of Motion JPEG 2000 and the Scalable High Intra Profile of H.264/AVC SVC [14-15]. The aforementioned features allow designing video surveillance systems that are able to facilitate real-time monitoring in diverse usage environments, ranging from desktop PCs connected to a wired network to mobile devices connected to a wireless network.

In [4], to facilitate privacy protection, we apply a layered scrambling technique to the luma subbands that make up face regions [4]. Specifically, we apply RLS, RP, and RSI to the DC, LP, and HP luma subbands of face regions, respectively. That way, it is possible to trade-off the visual importance of subbands against the amount of coded data in the subbands, the level of security offered by a particular scrambling tool, the impact of a particular scrambling tool on the coding efficiency, and the computational complexity of a particular scrambling tool. For more details regarding layered scrambling for Motion JPEG XR, we would like to refer the reader to [4]. Note that the layered scrambling technique under consideration can be applied to any coding format that allows organizing its transform coefficients in subbands (layered scrambling in H.264/AVC could for instance be realized by taking advantage of data partitioning).

3 Experimental Setup

To construct sets of training, gallery, and probe face images, we collected 3070 frontal face images of 68 subjects from the 'talking' image set of CMU PIE [16]. This resulted in the use of 68 gallery, 340 training, and 2662 probe face images. Further, assuming that an adversary does not have access to an implementation of layered scrambling for Motion JPEG XR, we only scrambled the probe face images. Consequently, the probe face images represent privacy-protected face images that appear in video content originating from surveillance cameras.

To encode face images, we inherited the settings previously used in [4] for the "ATM" video sequence [17]. This implies that we encoded face images with a spatial resolution of 192×192 and with a quantization parameter (QP) value set to 20.

To investigate the privacy-preserving nature of layered scrambling for Motion JPEG XR, we made use of the following automatic FR techniques: Principal Component Analysis (PCA) [18] and Fisher's Linear Discriminant Analysis (FLDA) [19], which both make use of global features, and Local Binary Patterns (LBP) [20], which makes use of local features. When using PCA- and FLDA-based FR, we followed the recommendations made by the FERET protocol to normalize face images [21]. When using LBP-based FR, we followed the preprocessing method of [20]. Also, when using LBP-based FR, we sampled eight binary patterns on a circle of radius two for each 16×16 region. Further, we applied layered scrambling after geometrical alignment, assuming that face detection is accurate and that eye coordinates are known.

We plotted FR results on a Cumulative Match Characteristic (CMC) curve [21]. To facilitate a fair comparison, we adopted the best found correct recognition rate (BstCRR) for PCA- and FLDA-based FR [22]. BstCRR reports the highest true positive recognition rate by means of an exhaustive search, varying the dimension of the feature vectors. On the other hand, we obtained the recognition rate for LBP-based FR for feature vectors with a maximum dimensionality of 8496 (the 144 blocks are represented by 59 features each).

Besides objective assessments, we also conducted subjective assessments with 32 human observers. For each of the experimental settings used, we presented three scrambled probe face images of different subjects to the human observers. Given a probe face image and a set of twelve gallery face images, we subsequently asked the human observers to select the gallery face image that is most similar to the given probe face image. The human observers were also able to indicate that a suitable match could not be found. Further, we allowed the human observers to study the probe face images at different zoom levels. In addition, we enhanced the contrast of the probe face images. This reflects a real-world scenario in which an attacker has complete control over the scrambled probe face images.

4 Assessment of Chroma-Induced Privacy Leakage

In this section, we present our objective and subjective assessments, studying and quantifying privacy leakage induced by the presence of non-scrambled chroma information in face images.

4.1 Notations

Table 1 introduces a number of notations used throughout the remainder of this paper. *DC*, *LP*, and *HP* denote a DC, LP, and HP subband, respectively. A first subscript is used to denote the incremental use of several subbands. Specifically, S_1, S_2, and S_3 represent the use of *DC*, *DC+LP*, and *DC+LP+HP*, respectively. A second subscript is used to denote the presence of luma and/or chroma channels. Finally, a prime is used to indicate the use of scrambling. As an overall example, $S'_{3,Y}$ indicates that the DC, LP, and HP subbands of the luma channel have been scrambled: $S'_{3,Y} = DC'_Y + LP'_Y + HP'_Y$.

Table 1. Summary of notations used

Notation	Explanation
DC, LP, and *HP*	DC, LP, and HP subband
S_3, S_2, and S_1	*DC+LP+HP, DC+LP,* and *DC*
Subscripts (Y, Co, and Cg)	Luma and chroma channels (Y, Co, and Cg)
Prime (′)	Scrambled image data

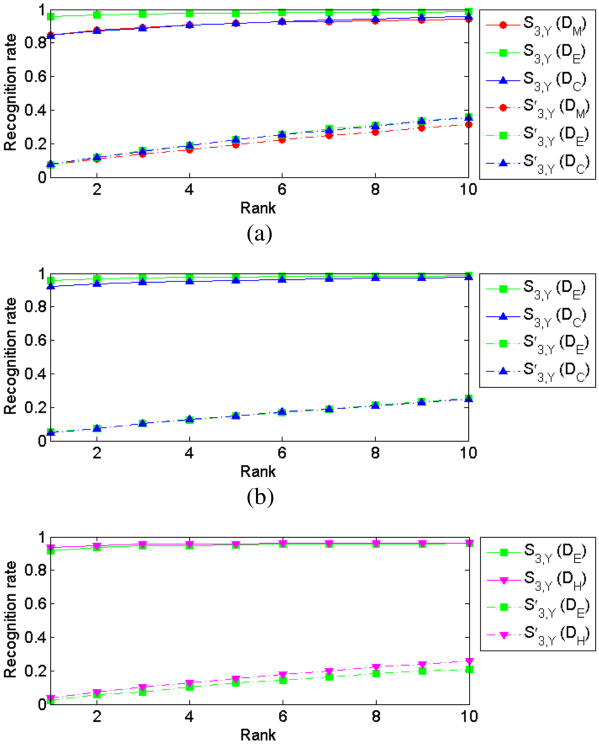
4.2 Objective Assessments

Our objective assessments consist of four experiments: 1) distance measurement for automatic FR applied to privacy-protected probe face images; 2) automatic FR applied to scrambled luma information; 3) automatic FR applied to scrambled luma and non-scrambled chroma information; and 4) automatic FR applied to non-scrambled chroma information.

Distance Measurement – The effectiveness of automatic FR depends on the metric used for measuring the distance between the feature vector of a probe face image and the feature vectors of the gallery face images. In this experiment, we investigate the influence of the following distance metrics on the effectiveness of automatic FR: Euclidean distance, Mahalanobis distance, Cosine distance, and Chi-square distance (denoted as D_E, D_M, D_C, and D_H, respectively).

When making use of non-scrambled probe face images, Fig. 1 shows that PCA-, FLDA-, and LBP-based FR are most effective when the Mahalanobis, Euclidean, and Chi-square distance metric are used, respectively (PCA-, FLDA-, and LBP-based FR achieve a rank 1 recognition rate of 84%, 96%, and 95%, respectively). These observations independently confirm results previously presented in the scientific literature [19],[20],[24]. However, when making use of scrambled probe face images, our experimental results indicate that PCA-based FR is most effective when making use of the Euclidean distance metric. Consequently, in the remainder of our experiments, we make use of the Euclidean distance metric for PCA- and FLDA-based FR, and we make use of the Chi-square distance metric for LBP-based FR. Note that, hereinafter, the grey-shaded area in each figure represents the set of recognition rates that yield an ideal or asymptotical level of privacy protection, which is the probability of success of random guessing.

Scrambled Luma Information – Fig. 2 allows studying the effectiveness of FR when only protecting the luma subbands of probe face images, assuming that an adversary is not able to take advantage of the possible presence of non-scrambled chroma information in the privacy-protected probe face images.



(a)

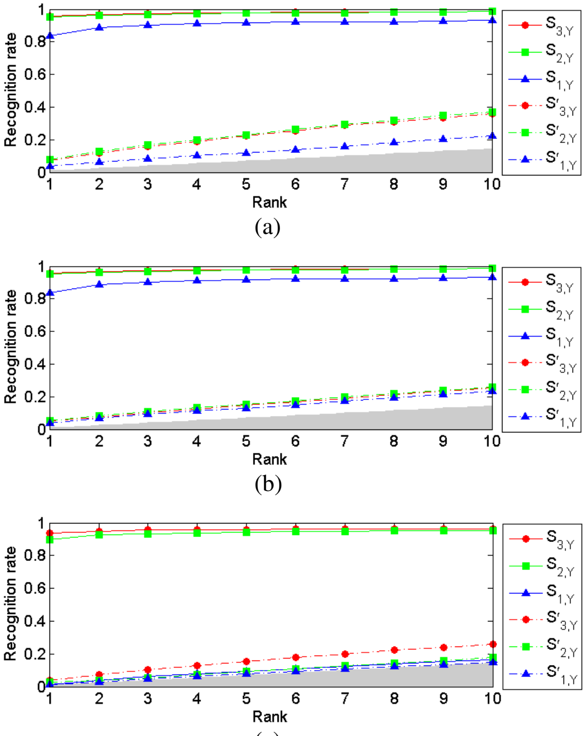
(b)

(c)

Fig. 1. Influence of distance measurement on FR effectiveness: (a) PCA, (b) FLDA, and (c) LBP

When making use of non-scrambled probe face images, all rank 1 recognition rates are higher than 83%, except when LBP-based FR is applied to $S_{1,Y}$, (*i.e.*, when LBP-based FR is applied to the luma information stored in the DC subbands of a face region). Specifically, for LBP-based FR, the rank 1 recognition rate decreases from 94% for $S_{1,Y}$ to 1.2% for $S_{1,Y}$. The substantial decrease in effectiveness of LBP-based

FR can be attributed to the fact that distinctive pixel information in local regions is almost completely eliminated in DC subbands. When scrambling the luma subbands of probe face images, the overall effectiveness of FR decreases significantly. The rank 1 recognition rates obtained for PCA-, FLDA-, and LBP-based FR are lower than 7.6%, 6.1%, and 3.8%, respectively. Consequently, the results presented in Fig. 2 show that layered scrambling is for instance highly effective in preserving privacy when making use of grayscale video surveillance.



(a)

(b)

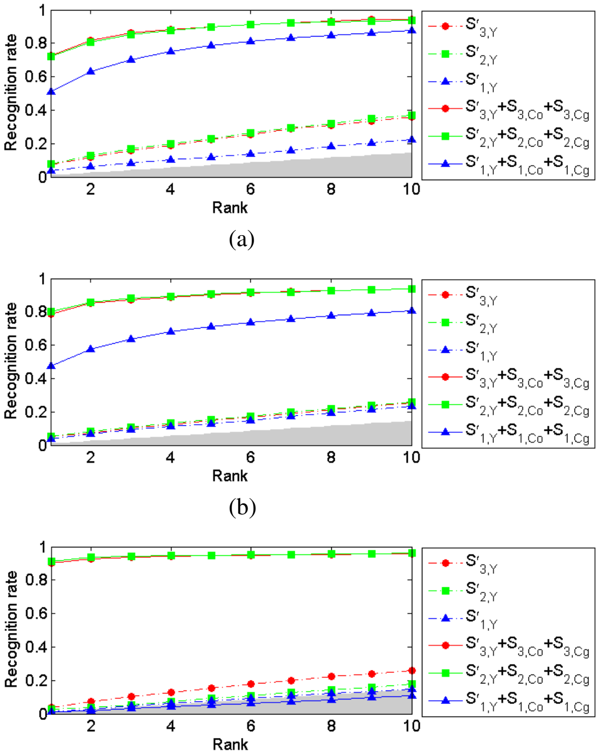
(c)

Fig. 2. Influence of layered scrambling on FR effectiveness: (a) PCA, (b) FLDA, and (c) LBP. Note that FR only makes use of luma information.

Scrambled Luma and Non-scrambled Chroma Information – To limit bit rate overhead in heterogeneous usage environments, the layered scrambling technique proposed in [4] only protects the luma subbands of face regions. In this experiment, we investigate whether layered scrambling is still effective when the scrambled luma channel and the non-scrambled chroma channels are simultaneously used for the purpose of automatic FR, assuming that an adversary has access to the compressed bit stream structure, and thus to the non-scrambled chroma information. Indeed, previous research has demonstrated that the additional use of chroma information is capable of increasing the overall effectiveness of FR [25].

To take advantage of non-scrambled chroma information, we adopted feature-level fusion [23]. Specifically, we fused feature vectors extracted from the scrambled luma and the non-scrambled chroma channels of face images by means of concatenation. Note that JPEG XR by default makes use of the YCoCg color space. Also, note that we did not subsample the chroma channels during encoding (*i.e.*, we made use of the YCoCg 4:4:4 color format).

As shown in Fig. 3, the recognition rates significantly increase when automatic FR is able to make use of both scrambled luma information and non-scrambled chroma information, compared to the recognition rates obtained when automatic FR is only able to make use of scrambled luma information. In particular, the rank 1 recognition rate is at least 44% for all FR techniques used, except when LBP-based FR is applied to DC subbands. This implies that the presence of non-scrambled chroma information can significantly decrease the effectiveness of scrambling when an adversary has access to the compressed bit stream structure.



(a)

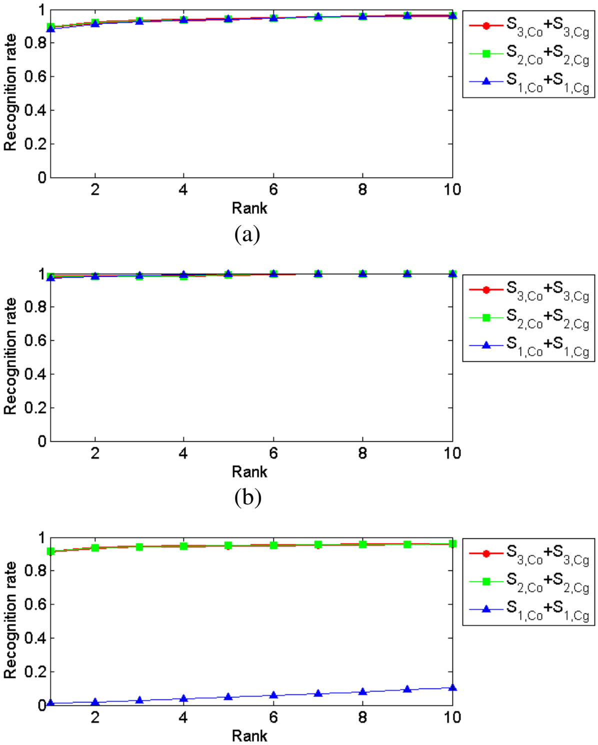
(b)

(c)

Fig. 3. Influence of scrambled luma information and non-scrambled chroma information on FR effectiveness: (a) PCA, (b) FLDA, and (c) LBP

Non-scrambled Chroma Information – Since non-scrambled chroma information is available to an adversary aware of the compressed bit stream structure, it is also important to investigate the effectiveness of automatic FR when only making use of non-scrambled chroma information. Fig. 4 plots the recognition rates obtained for automatic FR only making use of non-scrambled chroma information. We can observe that the rank 1 recognition rate is always higher than 88% for all FR techniques used, except when LBP-based FR is applied to DC subbands. This again implies that the presence of non-scrambled chroma information can significantly decrease the effectiveness of scrambling when an adversary has access to the compressed bit stream structure.

To summarize, for video surveillance applications requiring a high level of privacy protection, the results presented in Fig. 3 and Fig. 4 indicate that both luma and chroma information needs to be scrambled.



(a)

(b)

(c)

Fig. 4. FR effectiveness when only making use of non-scrambled chroma information: (a) PCA, (b) FLDA, and (c) LBP

4.3 Subjective Assessments

Our subjective assessments consist of two experiments: 1) human FR applied to non-scrambled chroma information and 2) human FR applied to scrambled luma and non-scrambled chroma information. In addition, we make use of complementary objective assessments to study whether agreement exists between 32 human observers and the output of automatic FR. The complementary objective assessments make use of experimental settings that are identical to the experimental settings used in our subjective assessments (for each of the subjective experiments, we used three probe face images and twelve gallery face images). Consequently, we report experimental results by means of example face images, Subjective Recognition Rates (SRR), and Objective Recognition Rates (ORR). Subjective recognition rates are obtained by counting the number of human observers reporting a correct identification over the total number of trials (since three probe face images are used for each parameter setting, the total number of trials for each parameter setting is three times the total number of human observers), while objective recognition rates are obtained by counting the number of correctly identified probe face images over the total number of probe face images at rank 1. Note that objective recognition rates are obtained for PCA-based FR (PCA-based FR had the highest overall effectiveness in Section 4.2).

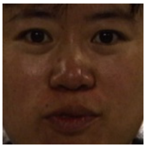
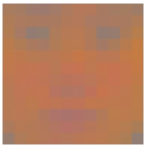
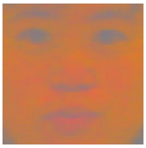
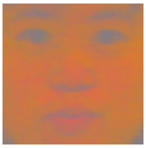
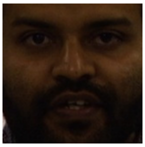
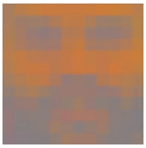
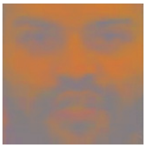
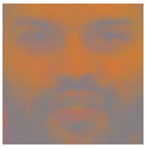
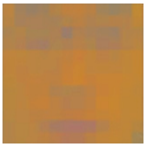
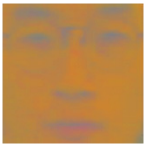
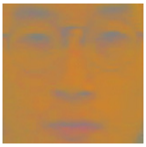
Subbands used	Original face images	$S_{1,Co} + S_{1,Cg}$	$S_{2,Co} + S_{2,Cg}$	$S_{3,Co} + S_{3,Cg}$
Example face images				
				
				
SRR / ORR	·	0.59 / 1.0	0.94 / 1.0	0.96 / 1.0

Fig. 5. Subjective and objective results for non-scrambled chroma information

Non-scrambled Chroma Information – This experiment assumes that an adversary has access to the compressed bit stream structure. Fig. 5 shows the subjective recognition rates obtained for probe face images only consisting of non-scrambled chroma information (this is, we did not visualize luma information for the probe face images shown in Fig. 5). We can observe that the subjective recognition rate is equal to 96% for S_3, 94% for S_2, and 59% for S_1. When not scrambling chroma information, the 32 human observers were able to correctly identify the face images by means of visual clues such as skin color information, the shape of a face, the presence of four corners in the face images, and even slight differences in face orientation. In addition, we can observe that PCA-based FR is able to achieve perfect objective recognition rates, regardless of the different types of subbands used.

Scrambled Luma and Non-scrambled Chroma Information – Again assuming that an adversary has access to the compressed bit stream structure, Fig. 6 shows the subjective recognition rates obtained for probe face images with scrambled luma and non-scrambled chroma information. From both the subjective recognition rates and the example face images shown, we can observe that scrambled luma information significantly hampers the identification of probe face images when simultaneously visualizing scrambled luma and non-scrambled chroma information. On the other hand, we can observe that PCA-based FR is able to achieve perfect recognition rates (see ORR in Fig. 6). The latter can be attributed to the presence of non-scrambled chroma information and the use of a limited number of gallery face images.

To summarize, the experimental results presented in Fig. 5 and Fig. 6 indicate that, when an adversary has access to the compressed bit stream structure, non-scrambled chroma information can be successfully used to identify probe face images that have only been privacy-protected in the luma domain.

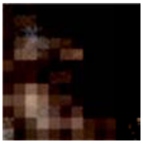
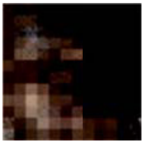
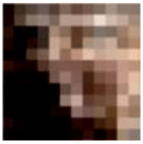
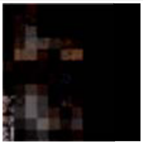
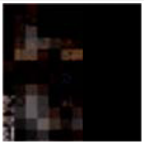
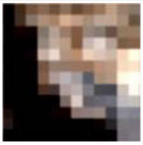
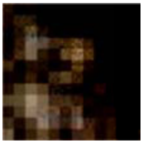
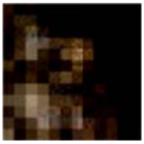
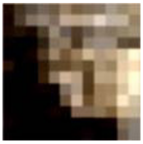
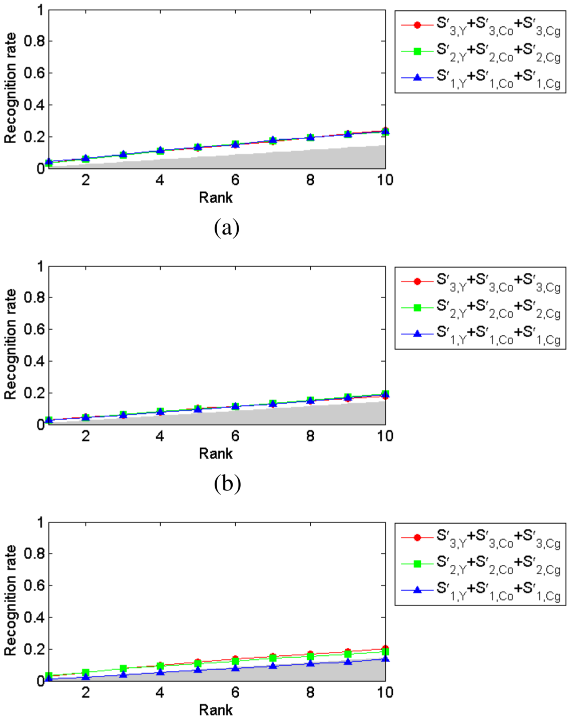
Subbands used	Original face images	$S'_{1,Y}+S_{1,Co}+S_{1,Cg}$	$S'_{2,Y}+S_{2,Co}+S_{2,Cg}$	$S'_{3,Y}+S_{3,Co}+S_{3,Cg}$
Example face images				
				
				
SRR / ORR	·	0.03 / 1.0	0.0 / 1.0	0.0 / 1.0

Fig. 6. Subjective and objective results for scrambled luma and non-scrambled chroma information

5 Discussion

Given that the mere scrambling of luma information already results in significant distortion of the facial information present in a face region, human observers were not able to correctly recognize face images when simultaneously visualizing scrambled luma and non-scrambled chroma information. However, as shown by our objective and subjective assessments (see Fig. 4 and Fig. 5), an adversary aware of the compressed bit stream structure can take advantage of the presence of non-scrambled chroma information in probe face images. Consequently, for video surveillance applications requiring a high level of privacy protection, both the luma and the chroma channels need to be scrambled at the cost of a higher bit rate overhead. In this paper, we therefore propose to apply layered scrambling to both the luma (Y) and the chroma channels (Co and Cg).



(a)

(b)

(c)

Fig. 7. FR effectiveness when making use of scrambled luma and chroma information: (a) PCA, (b) FLDA, and (c) LBP.

Fig. 7 allows studying the effectiveness of automatic FR applied to probe face images with scrambled luma and chroma channels. The resolution of the probe face images used was fixed to 192×192 and the QP value was set to 20. For all FR techniques used, we can observe that the rank 1 recognition rates obtained are lower than 4.1%, demonstrating that a high level of privacy protection can be achieved by scrambling both the luma and the chroma channels.

Fig. 8 shows that scrambling both the luma and chroma channels results in a total bit rate overhead of 7.0%, 18.1%, and 46.4% for S_3, S_2, and S_1, respectively, using a value of 8 for the RLS parameter L (see [4]). Compared to the case in which only luma information is scrambled, the increase in bit rate overhead is 5.4%, 14.1%, and 36.9% for S_3, S_2, and S_1, respectively. Note that we measured the overhead relative to the size of the face image (and not to the size of the whole image, which includes both the face image and background information), thus assuming a worst case scenario (dependent on the scenario, face regions may make up a large part of the video content). Further, Fig. 8 also presents bit rate overhead numbers when using 4:2:0 sub-sampling. Note that sub-sampling decreases the level of privacy protection, given the lesser amount of data available for scrambling. For instance, in the ideal case where all transform coefficients are non-zero and where the RLS parameter L is set to 8, the use of chroma sub-sampling reduces the total number of combinations required to break the protection of 10 MBs from 3.6×10^{722} to 1.7×10^{360}.

The results above show that, although scrambling chroma information increases the level of privacy protection, it comes with a higher bit rate overhead. The latter may be of concern when multiple video streams need to be simultaneously delivered in diverse usage environments.

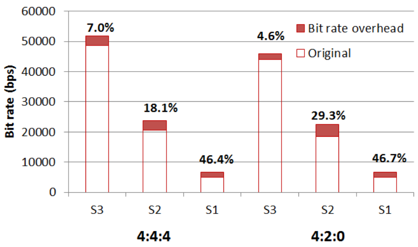


Fig. 8. Bit rate overhead when making use of scrambled luma and chroma information: (a) 4:4:4 color format and (b) 4:2:0 color format

6 Conclusions and Directions for Future Research

This paper studied and quantified the influence of non-scrambled chroma information on the effectiveness of automatic and human FR. To that end, we made use of three automatic FR techniques to test the effectiveness of a layered scrambling technique developed for Motion JPEG XR, taking into account the following four experimental conditions: 1) distance measurement for automatic FR applied to privacy-protected probe face images; 2) automatic FR applied to scrambled luma information; 3) automatic FR applied to scrambled luma and non-scrambled chroma information; and 4) automatic FR applied to non-scrambled chroma information. In addition, we investigated whether agreement exists between the judgments of 32 human observers and the output of automatic FR for the following two experimental conditions: 1) human

FR applied to non-scrambled chroma information and 2) human FR applied to scrambled luma and non-scrambled chroma information.

Our results show that human observers were not able to successfully recognize face images when simultaneously visualizing scrambled luma and non-scrambled chroma information. However, when an adversary has access to the coded bit stream structure, the presence of non-scrambled chroma information may significantly contribute to privacy leakage (rank 1 recognition rates > 88% when applying automatic FR to non-scrambled chroma information). Consequently, for video surveillance applications requiring a high level of privacy protection, our results indicate that both luma and chroma information needs to be scrambled. We therefore applied layered scrambling to both luma and chroma information, showing that a higher level of privacy protection can be achieved (rank 1 recognition rates < 4.1%) at the cost of an increase in bit rate overhead of 36.9% (DC), 14.1% (DC+LP), and 5.4% (DC+LP+HP), measuring overhead relative to the case where only luma information is scrambled.

In order to compile a benchmark for tools for privacy protection, future research will focus on identifying additional worst case scenarios. This benchmark could for instance be used to design a more effective evaluation framework for privacy preservation. This benchmark could also be used to design more effective tools for privacy protection. Finally, we plan to evaluate the use of layered scrambling in coding formats other than Motion JPEG XR.

Acknowledgement. This research was supported by the National Research Foundation (NRF) of Korea (grant: NRF-D00070).

References

1. Bowyer, K.W.: Face recognition technology: security versus privacy. IEEE Society on Social Implications of Technology 23, 9–19 (2004)
2. Dufaux, F., Ebrahimi, T.: Scrambling for Privacy Protection in Video Surveillance Systems. IEEE Transactions on Circuits and Systems for Video Technology 18, 1168–1174 (2008)
3. Carrillo, P., Kalva, H., Magliveras, S.: Compression Independent Reversible Encryption for Privacy in Video Surveillance. EURASIP Journal on Information Security 2009, Article ID 429581, 13 pages (2009)
4. Sohn, H., De Neve, W., Ro, Y.M.: Privacy Protection in Video Surveillance Systems: Analysis of Subband-Adaptive Scrambling in JPEG XR. IEEE Transactions on Circuits and Systems for Video Technology 21, 170–177 (2011)
5. Sohn, H., Anzaku, E.T., De Neve, W., Ro, Y.M., Plataniotis, K.N.: Privacy Protection in Video Surveillance Systems Using Scalable Video Coding. In: IEEE International Conference on Advanced Video and Signal Based Surveillance, pp. 424–429 (2009)
6. Zeng, W., Lei, S.: Efficient frequency domain video scrambling for content access control. In: Proceedings of ACM International Conference on Multimedia, pp. 285–294 (1999)
7. Tong, L., Dai, F., Zhang, Y., Li, J.: Visual security evaluation for video encryption. In: Proceedings of ACM International Conference on Multimedia, pp. 835–838 (2010)
8. Dufaux, F., Ebrahimi, T.: H.264/AVC video scrambling for privacy protection. In: Proceedings of IEEE International Conference on Image Processing, pp. 1688–1691 (2008)

9. Sohn, H., De Neve, W., Ro, Y.M.: Region-of-Interest Scrambling for Scalable Surveillance Video using JPEG XR. In: Proceedings of ACM International Conference on Multimedia, pp. 861–864 (2009)
10. Rodrigues, J.-M., Puech, W., Bors, A.: A Selective Encryption for Heterogeneous Color JPEG Images Based on VLC and AES Stream Cipher. In: Proceedings of European Conference on Colour in Graphics, Imaging and Vision, pp. 34–39 (2006)
11. Zou, J., Ward, R.K., Qi, D.: A new digital image scrambling method based on Fibonacci numbers. In: Proceedings of International Symposium on Circuits and Systems, III-965-8 (2004)
12. Dufaux, F., Ebrahimi, T.: A Framework for the Validation of Privacy Protection Solutions in Video Surveillance. In: IEEE International Conference on Multimedia & Expo., pp. 66–71 (2010)
13. Mao, Y., Wu, M.: A joint signal processing and cryptographic approach to multimedia encryption. IEEE Transactions on Image Processing 15(7), 2061–2075 (2006)
14. Tran, T.D., Liu, T., Topiwala, P.: Performance comparison of leading image codecs: H.264/AVC Intra, JPEG2000, and Microsoft HD Photo. In: Proceedings of SPIE, pp. 66960B.1–66960B.14 (2007)
15. Srinivasan, S., Tu, C., Regunathan, S.L., Sullivan, G.J.: HD Photo: A new image coding technology for digital photography. In: Proceedings of SPIE, pp. 66960A.1–66960A.19 (2007)
16. Sim, T., Baker, S., Bsat, M.: The CMU pose, illumination, and expression database. IEEE Transactions on Pattern Analysis and Machine Intelligence 25, 1615–1618 (2003)
17. IVY Lab Video Surveillance Dataset, `http://ivylab.kaist.ac.kr/demo/vs/dataset.htm`
18. Turk, M.A., Pentland, A.P.: Eigenfaces for recognition. Journal of Cognitive Neuroscience 3, 71–86 (1991)
19. Belhumeur, P.N., Hesphanha, J.P., Kriegman, D.J.: Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. IEEE Transactions on Pattern Analysis and Machine Intelligence 9, 711–720 (1997)
20. Ahonen, T., Hadid, A., Pietikainen, M.: Face description with local binary patterns: Application to face recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 28, 2037–2041 (2006)
21. Phillips, P.J., Wechsler, H., Huang, J., Rauss, P.: The FERET database and evaluation procedure for face recognition algorithms. Image and Vision Computing Journal 16, 295–306 (1998)
22. Wang, J., Plataniotis, K.N., Lu, J., Venetsanopoulos, A.N.: On solving the face recognition problem with one training sample per subject. Pattern Recognition 39, 1746–1762 (2006)
23. Jain, A.K., Nandakumar, K., Ross, A.: Score normalization in multimodal biometric systems. Pattern Recognition 38, 2270–2285 (2005)
24. Perlibakas, V.: Distance measures for PCA-based face recognition. Pattern Recognition Letters 25, 1421–1430 (2004)
25. Choi, J.Y., Ro, Y.M., Plataniotis, K.N.: Color face recognition for degraded face images. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 39, 1217–1230 (2009)

Perceptual Image Hashing via Wave Atom Transform

Fang Liu and Lee-Ming Cheng

Department of Electronic Engineering, City University of Hong Kong,
83 Tat Chee Avenue, Kowloon Tong, Hong Kong
f.liu@student.cityu.edu.hk, itlcheng@cityu.edu.hk

Abstract. This paper presents a perceptual image hashing algorithm based on wave atom transform, which can authenticate the content preserved images and distinguish the content altered ones in authentication module. Wave atoms are employed since it has significantly sparser expansion and better feature extraction capability than traditional transforms, like wavelet and DCT. In addition, a randomized pixel modulation based on RC4 algorithm is performed to ensure the security. According to the experimental results, the proposed scheme is sensitive to content-alteration attacks with the resiliency to content-preserving operations, such as JPEG compression, Gaussian noise and contrast enhancement. Moreover, compared with some other image hashing algorithms, the proposed approach also achieves better performance at the aspect of robustness.

Keywords: Image hashing, Authentication, Robustness, Wave atom transform.

1 Introduction

Nowadays, the popularity of image editing tools has led to an enormous growth of image illegal use, such as image forgery and unauthorized utilization. A traditional solution to prevent this is to generate a hash using some standard cryptographic hash functions, and form a digital signature by some public key encryption algorithms [1]. This kind of hash functions is very sensitive and even one bit change in the message will result in significant changes in the hash value. However, it is the sensitivity that makes these functions not applicable to digital images. Since images will also be considered as the identical one even if they have undergone some content-preserving manipulations, like JPEG compression, contrast operation or median filtering.

Recently, many perceptual image hashing schemes have been proposed [2-11]. The core idea of image hashing is to construct a hash by extracting the characteristics of human perception in images, and use this hash to authenticate or retrieve an image without considering the various variables or formats of this image. This kind of schemes takes the changes of human perception into account and ignores the perceptually unnoticeable changes. And they have drawn a lot of attentions owing to the outstanding performance against some common signal processing operations.

The state of the art techniques of image hashing schemes are roughly classified into four categories, including image statistics based approach [2-4], relation based

Y.Q. Shi, H.J. Kim, and F. Perez-Gonzalez (Eds.): IWDW 2011, LNCS 7128, pp. 468–478, 2012.

schemes [5, 6], low-level image feature extraction [7, 8], and preservation of coarse image representation [9-11]. The methodology used in [2-4] is to construct the hash by selecting some invariant statistics characteristics of images in order to obtain great robustness, such as mean, variance and higher moments of intensity values of image blocks. In relation based schemes, the invariant relationships of coefficients in DCT, wavelet or other transforms are employed to generate the hash, such as the invariant relation of DCT coefficients in the same position of different blocks before and after JPEG compression [5], and the invariant relation of a parent and child node located at the multiple scales in wavelet decomposition [6]. Approaches based on image feature extraction [7, 8] usually extract hash features by detecting the salient image feature points, and possess good robustness. Schemes on preservation of coarse image representation extract hash features by making use of the coarse information of the whole image, such as the coefficients of low frequency in DCT [9] or Fourier transform [10], the low-rank decomposition of nonnegative matrix factorization (NMF) [11], and so forth.

At present, it is known that there are plenty of image hashing schemes using wavelet, DCT and other transforms. However it is expected that wave atom transform can achieve better performance than other transforms in image hashing. Demanet and Ying introduced wave atoms in 2007 [12], which are a recent addition to the repertoire of mathematical transforms of computational harmonic analysis. They have been proved to have a dramatically sparser expansion of wave equations than traditional transforms, which come either as an orthonormal basis or a tight frame of directional wave packets, and are particularly well suitable for representing oscillatory patterns in images. Motivated by these characteristics, this paper finds out the feasibility of wave atom transform applied in image hashing.

The rest of this paper is structured as follows. Section 2 shows a brief overview of wave atom transform. The proposed algorithm is described in Section 3. The experiments and analysis are presented in Section 4, whereas the conclusions are giving in Section 5.

2 Wave Atom Transform

Demanet and Ying introduced wave atoms as a variant of 2-D wavelet packets in 2007 [12], which can adapt to arbitrary local directions of a pattern, and also can sparsely represent anisotropic patterns aligned with the axes. Oscillatory functions and oriented textures in wave atoms have been proved to have a dramatically sparser expansion compared to some other fixed standard representations like Gabor filters, wavelets, and curvelets. Wave atoms interpolate precisely between Gabor atoms [13] and directional wavelets [14]. The period of oscillations of each wave packet is related to the size of essential support via parabolic scaling, *wavelength* ~ (*diameter*)^2.

Wave atoms can be constructed from tensor products of adequately chosen 1-D wave packets. Let $\psi^{j}_{m,n}(x)$ represent a 1-D wave packet, where $j, m \geq 0,$ and $n \in Z,$ centered around $x_{j,n} = 2^{-j}n$ in space and $\pm w_{j,m} = \pm\pi 2^{j} m$ in frequency respectively, with $C_1 2^j \leq m \leq C_2 2^j$. The basis function is defined combining dyadic scaled and translated versions of $\hat{\psi}^0_m$ in frequency domain as the following

$$\psi^j_{m,n}(x) = \psi^j_m(x - 2^{-j}n) = 2^{j/2}\psi^0_m(2^{-j}x - n) \tag{1}$$

where

$$\psi^0_m(w) = e^{-iw/2}[e^{i\alpha_m}g(\epsilon_m(w - \pi(m + 1/2)) + e^{-i\alpha_m}g(\epsilon_m(w - \pi(m + 1/2)))] \tag{2}$$

with $\alpha_m = \pi/2\,(m + 1/2)$, $\epsilon_m = (-1)^m$ and g a real-value compactly-support C^∞ bump function such that $\sum_m |\psi^0_m(w)|^2 = 1$.

For each wave $w_{j,m}$ at scale 2^{-j}, the coefficient $c_{j,m,n}$ is treated as a decimated convolution. Discretize the sample u at $x_k = kh$, $h= 1/N$, $k=1,\cdots,N$, and the discrete coefficients $c^D_{j,m,n}$ are calculated by utilizing a reduced inverse FFT inside an interval of size $2^{j+1}\pi$, centered around the origin.

$$c^D_{j,m,n} = \sum_{k=2\pi\left(-2^j/_2+1:1:2^j/_2\right)} e^{i2^{-jnk}} \sum_{p\in 2\pi Z} \overline{\hat{\psi}^j_m(k + 2^j p)}\,\hat{u}(k + 2^j p) \tag{3}$$

The 2-D orthonormal basis functions with four bumps are formed by individually utilizing products of 1-D wave packets in frequency plane. Let $\mu = (j, \boldsymbol{m}, \boldsymbol{n}) = (j, m_1, m_2, n_1, n_2)$, the basis function is modified as

$$\phi^+_\mu(x_1, x_2) = \psi^j_{m1}(x_1 - 2^{-j}n_1)\psi^j_{m2}(x_2 - 2^{-j}n_2) \tag{4}$$

A dual orthonormal basis can be established from the "Hilbert-transformed" wavelet packets as

$$\phi^-_\mu(x_1, x_2) = H\psi^j_{m1}(x_1 - 2^{-j}n_1)H\psi^j_{m2}(x_2 - 2^{-j}n_2) \tag{5}$$

By combining eq. (4) and eq. (5), basis functions with two bumps are provided in frequency domain, and directional wave packets oscillate in one single direction.

$$\phi^{(1)}_u = (\phi^+_u + \phi^-_u)/2\ , \quad \phi^{(2)}_u = (\phi^+_u - \phi^-_u)/2 \tag{6}$$

$\phi^{(1)}_u$ and $\phi^{(2)}_u$ are denoted as ϕ_u together, and form the wave atoms frame.

3 Proposed Algorithm

In this section, a wave atoms based image hashing scheme is proposed. According to the previous research [15], it has been found that the wave atom coefficients in the middle frequency sub-bands are robust to common signal processing attacks. This property is also useful when wave atoms are applied to image hashing. Besides, in order to enhance the security of the proposed scheme, a randomized pixel modulation (RPM) [16] is used. Before taking the wave atom transform, all pixels in spatial domain are randomly modulated using a pseudo-random secret key stream based on RC4. The details of the proposed algorithm are shown below.

3.1 Hash Generation

The block diagram of the hash generation is shown in Fig. 1(a) & (b) and the detailed procedures are described as follows:

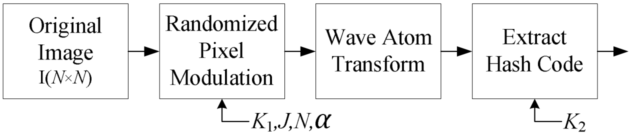


Fig. 1(a). Hash extraction module

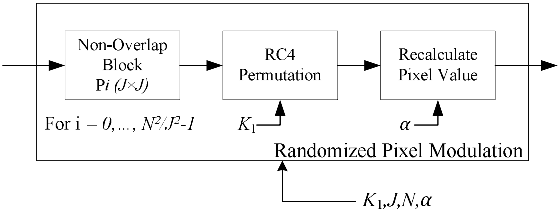


Fig. 1(b). Randomized pixel modulation

Step 1: Let I denote the input image of size $N \times N$. Divide I into a number of non-overlapping blocks, each with dimension $J \times J$. Thus, N^2/J^2 blocks are generated. Let P_i denote these blocks, where $i = 0, \cdots, N^2/J^2 - 1$. In this implementation, each block is divided into 16 × 16 pixels.

Every block is randomly permutated by the sequence, which is obtained from sorting the pseudo-random numbers generated by RC4 algorithm and governed by the secret key K_1. Let $P_i(x, y)$ represent the gray value of a pixel at spatial domain location (x, y) in the block P_i. Denote $S_i(m)$ as the new sequence of each block, where i is the block index and m is the index of a particular element S_i. In order to make the image hash code dependent on the secret key, the RPM transform [16] is employed to modulate every pixel in every block as the following

$$P_i(x, y)' = P_i(x, y) + \alpha \times S_i(m) \tag{7}$$

where $P_i(x, y)'$ is the new pixel value and $1 \leq x, y \leq J, 1 \leq m \leq J^2$.

Step 2: By going through wave atom transform, the image is decomposed into five scale bands; each scale band has different frequency. For every scale band, there are a number of sub-blocks which consists of a great deal of wave atom coefficients. Among these scale bands, the third scale band is selected to compute the hash code, since middle frequency scale coefficients are more robust than high frequency ones, and also more fragile than low frequency ones. Since the energy of wave atom coefficients captures most information of main image features, the intermediate hash can be computed by exploring the mutual relationship between these sub-blocks.

Step 3: Denote $C(j,m_1,m_2,n_1,n_2)$ as wave atom coefficients, where j is the scale, and m_1,m_2,n_1,n_2 represent the phase. Assign an index i for each sub-block in the third scale band. Let E_i be the energy of the i-th block.

For all non-empty blocks in the third scale band

$$E_i = \sum_{q=1}^{l_2} \sum_{p=1}^{l_1} \mathrm{C}(j,m_1,m_2,p,q)^2 \tag{8}$$

where l_1 and l_2 represent the length and width of the sub-block respectively.

To ensure that the information of features used to generate the hash code cannot be exposed, a random permutation based on RC4 is applied to E_i, and the new sequence $\mathrm{E}_i{}'$ is generated.

Let the total number of non-empty blocks in the third scale band be t. The energy difference between each two blocks is used to generate one hash bit. The intermediate hash can be calculated as follows:

$$h^{(i)} = \begin{cases} 1, & if\ \mathrm{E}_i{}' > \mathrm{E}_{i+1}{}' \\ 0, & Otherwise \end{cases} \tag{9}$$

where $i \in [1,\cdots,t-1]$.

Step 4: To increase the security of hash code, a pseudo-random sequence generated by the secret key K_2 is employed to XOR the intermediate hash h based on RC4 algorithm and generates the final hash H.

3.2 Image Authentication

The image authentication module, which is employed to authenticate the received image, is shown in Fig. 2. Using the same parameters N, K_1, K_2, α and J, the system calculates the received image's hash code and makes the comparison with the original hash code. The image authentication procedure is described as follows:

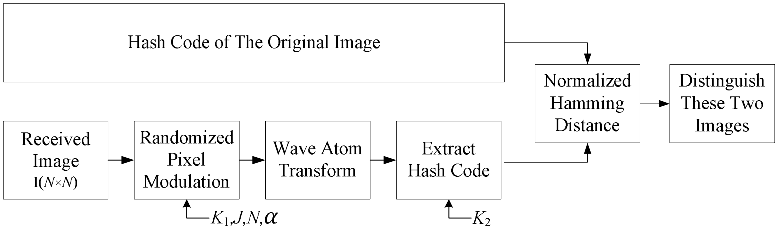

Hash Code of The Original Image
Normalized Hamming Distance
Distinguish These Two Images
Received Image I(N×N)
Randomized Pixel Modulation
Wave Atom Transform
Extract Hash Code
K1,J,N,α
K2

Fig. 1. Image authentication module

Step 1: The received image goes through the same steps as described in Section 3.1 to obtain the hash code H'.

Step 2: The normalized hamming distance *d* of these two hash code is then computed as the following.

Denote the i-th values of H and H' as $H(i)$ and $H'(i)$ respectively, and L as the length of the image hash.

$$d(H,H') = 1/L \sum_{i=1}^{L} \delta(H(i),H'(i)) \tag{10}$$

where

$$\delta\big(H(i),H'(i)\big) = \begin{cases} 0, H(i) = H'(i) \\ 1, H(i) \neq H'(i) \end{cases} \tag{11}$$

Step 3: Denote ϑ as a threshold to distinguish the two images. If the calculated normalized hamming distance is larger than ϑ, the image I' is considered as tampered or even a different image, which is unauthentic. Otherwise the image I' will be authenticated.

4 Experiments and Analysis

In order to test the proposed algorithm for image authentication, 21 gray-scale images of size 512 × 512 are used as the original test images, and the total numbers of images for content-preserving operations and content-alteration attacks are 819 and 609, respectively. In this paper, the normalized hamming distance *d* is used as the metric.

4.1 Content-Preserving Experimental Analysis

It is important to notice that a good perceptual image hashing scheme can authenticate perceptually identical images and distinguish the perceptually different images. In this section, some content-preserving signal processing operations are applied to illustrate the performance of proposed scheme. Table 1 shows the average normalized hamming distance of 21 original images under those operations based on different α. The parameter α in eq. (7) is used to enhance the security of hashing code such that the new pixel value depends on both the original pixel value and the secret key. Without knowing the secret key, an attacker cannot extract the hash code accurately, and cannot create a forged image.

Note that *d* is expected to approach zero for the perceptual identical images and approach 0.5 for different images. From Table 1, it can be observed that the values of *d* between the original and processed images are small and less than 0.1 for all cases, except rotation manipulations. Since the coefficients in the third scale band of wave atom transform cannot be changed greatly without change the content of the image. However rotation manipulation will change the positions of image pixels and destroy the correlations of original wave atoms, thus result to a larger *d*. Therefore, if the threshold ϑ is selected as 0.1, the proposed scheme can authenticate the images which go through all kinds of content-preserving operations presented in Table 1 except rotation.

Table 1. Average normalized hamming distance under different operations

Image	Parameter	Normalized Hamming Distance *d*				
		α =0.1	α = 0.2	α = 0.3	α = 0.4	α =0.5
JPEG (quality factor)	5	0.04210	0.03589	0.03589	0.04141	0.03658
	15	0.02208	0.01311	0.01863	0.02830	0.03451
	25	0.01863	0.00690	0.01380	0.02484	0.03796
	50	0.01380	0.00621	0.01449	0.02692	0.03451
	85	0.01449	0.00138	0.00966	0.02346	0.04693
Gaussian Noise (standard variance)	5	0.02415	0.02139	0.02346	0.03175	0.03865
	10	0.04486	0.04210	0.04417	0.04831	0.05521
	15	0.05659	0.04969	0.05176	0.05728	0.06556
	20	0.07384	0.07108	0.06901	0.07039	0.07453
Salt and Pepper Noises Addition (noise density)	0.01	0.02484	0.02484	0.01794	0.03520	0.04072
	0.04	0.04003	0.02622	0.04279	0.05521	0.05314
	0.06	0.05107	0.05521	0.04969	0.06625	0.06763
	0.08	0.06004	0.05452	0.05797	0.05935	0.07246
	0.1	0.06418	0.06418	0.06694	0.06832	0.07108
Gaussian Low Pass Filtering (Standard Variance, Window)	0.5,3	0.01311	0.00276	0.01104	0.02484	0.03451
	3,3	0.01794	0.00759	0.01311	0.02415	0.03382
	0.5,5	0.01311	0.00276	0.01104	0.02484	0.03451
	1.5,5	0.02415	0.01380	0.01656	0.02484	0.03313
	3,5	0.02761	0.01725	0.02001	0.02692	0.03382
Median Filtering (filter size)	3×3	0.02553	0.01932	0.02622	0.03313	0.04555
	5×5	0.03313	0.02968	0.03520	0.03934	0.04900
	7×7	0.04624	0.03589	0.03865	0.04279	0.04969
	9×9	0.05452	0.04555	0.04693	0.04831	0.05107
Histogram Equalization		0.05521	0.04900	0.04900	0.04900	0.05728
Contrast Change	10%	0.01656	0.00207	0.00759	0.02139	0.03244
	20%	0.02070	0.00897	0.01173	0.02139	0.03382
	-10%	0.01311	0.00276	0.01104	0.02484	0.03451
	-20%	0.01242	0.00345	0.01173	0.02553	0.03520
Cropping	20%	0.09386	0.08627	0.08765	0.08489	0.08351
Scaling	50%	0.00276	0.00345	0.00690	0.00966	0.01042
Laplacian Sharpening (operator)	0.1	0.01380	0.02415	0.03244	0.04210	0.05176
	0.3	0.01311	0.02346	0.03175	0.04141	0.05107
	0.6	0.01311	0.02346	0.03175	0.04141	0.05107
	0.8	0.01311	0.02346	0.03175	0.04141	0.05107
	1	0.01242	0.02277	0.03106	0.04072	0.05038
Rotation	5°	0.14631	0.14700	0.14700	0.14010	0.14424
	10°	0.16701	0.16218	0.15804	0.15114	0.15114
	15°	0.23395	0.22774	0.22498	0.21946	0.22084
	20°	0.26501	0.25880	0.25742	0.25052	0.24914

4.2 Content-Alteration Experimental Analysis

To prove the capability of detecting image content alterations, 609 images are used to test in total, and only several tampered versions of image Lena have been shown in Fig. 3. Table 2 shows the normalized hamming distances between the hash of Lena and tampered versions of Lena in different values of α.

It is observed that the distances d between the hash of Lena and tampered versions of Lena are normally larger than the distances between the original images and content-preserving processed ones. However, the distances decrease with the increase of α. Note that the randomness in the hash is increased with the value of α, but the capability of the tampering detection is also decreased. From eq. (7), it can be seen that the new pixel value $P_i(x,y)'$ is affected by the value of α. It can be explained that the larger value of α causes a large offset of $P_i(x,y)'$. In other words, the new pixel value $P_i(x,y)'$ will be less influenced by the original pixel value $P_i(x,y)$ of the original image itself. Therefore, the capability of tampering detection is degraded.

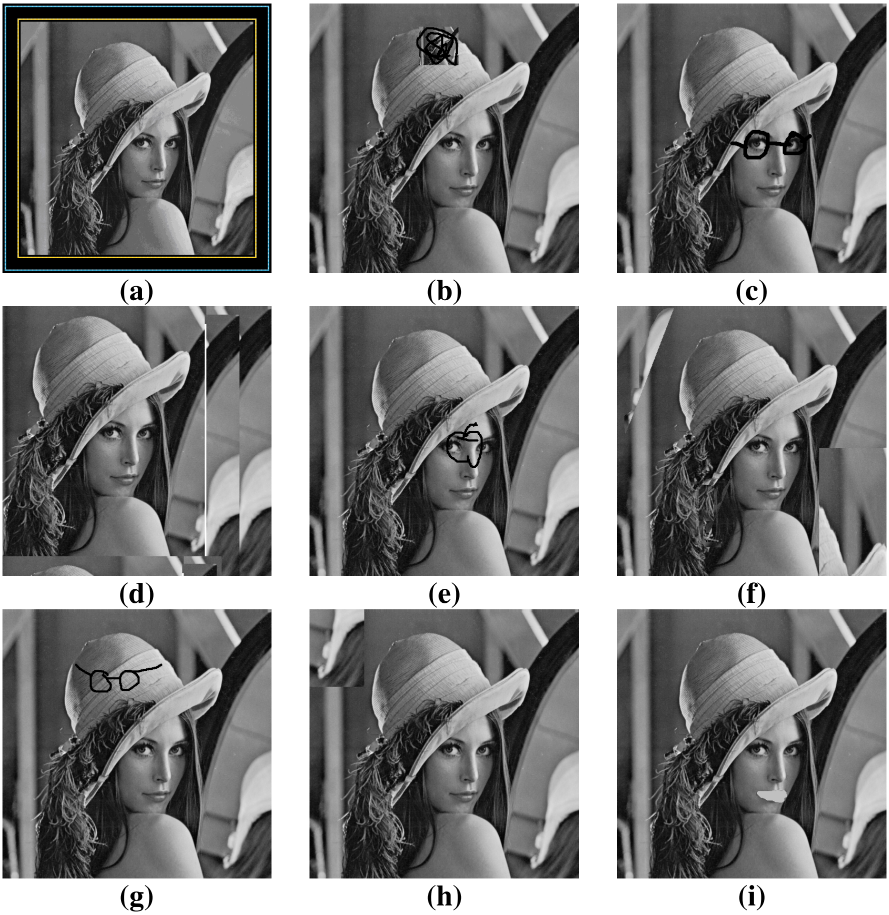


Fig. 2. Tampered versions of image Lena

Table 2. Normalized hamming distance against different tamperings

Image		Normalized Hamming Distance *d*				
		α =0.1	α = 0.2	α = 0.3	α = 0.4	α =0.5
Malicious Attack	**(a)**	0.39130	0.33333	0.34783	0.30435	0.28986
	(b)	0.10145	0.08696	0.08696	0.05797	0.05797
	(c)	0.13043	0.10145	0.10145	0.11594	0.14493
	(d)	0.30435	0.24638	0.18841	0.17391	0.14493
	(e)	0.14493	0.11594	0.10145	0.08696	0.08696
	(f)	0.15942	0.17391	0.14493	0.11594	0.10145
	(g)	0.15942	0.10145	0.13043	0.13043	0.14493
	(h)	0.13043	0.07246	0.07246	0.05797	0.07246
	(i)	0.10145	0.11594	0.08696	0.07246	0.05797

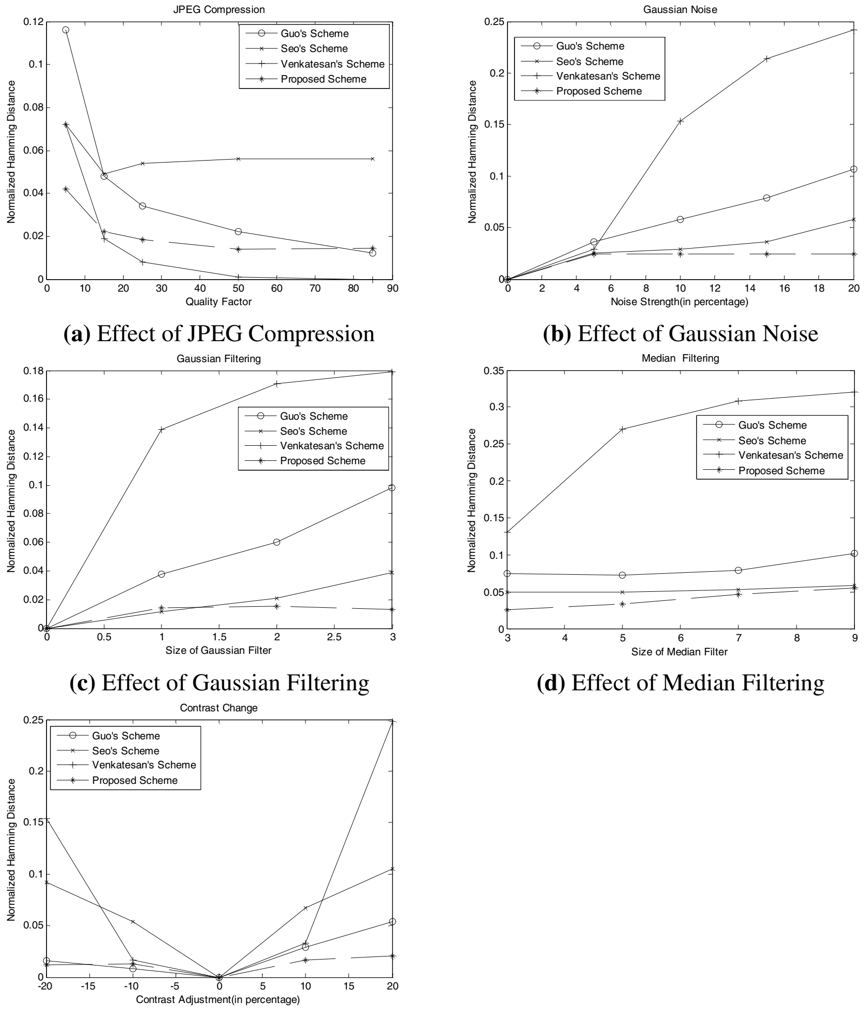
It is noteworthy that the threshold ϑ is a tradeoff to evaluate the robustness and the capability of the tampering detection. By comparing these two tables, it can also be observed that when the value of ϑ decreases, the capability to detect the malicious tampering will be increased, but the sensitivity of content-preserving operations becomes relatively higher. $\vartheta = 0.1$ is found to be an optimal value to discriminate the content-preserving and content-alteration operations. Taking all criteria into consideration, we come to the conclusion that if α and ϑ are chosen as 0.1 and 0.1 respectively, the proposed algorithm is robust to common signal processing manipulations as shown in Table 1 and also able to detect all malicious tampered images as shown in Fig.3. In addition, the security of the proposed scheme has also been guaranteed.

4.3 Comparisons with Related Schemes

Moreover, three image hashing algorithms proposed by Guo et al. [17], Seo et al.[18] and Venkatean et al. [3] have been compared with the proposed scheme in which α is chosen as 0.1. Guo et al proposed a content based image hashing scheme via wavelet and radon transform, while Seo's scheme and Venkatean's scheme is only based on the radon transform and wavelet transform respectively. Fig. 4 shows the performance of these image hashing schemes in terms of normalized hamming distance.

As shown in Fig. 4(a), the robustness performance of proposed scheme is better than Guo's scheme and Seo's scheme but a little worse than Venkatean's scheme under JPEG compression, where the normalized hamming distance of the proposed scheme is kept below 0.05. With the increase of Gaussian noise strength in Fig. 4(b), the proposed scheme still keeps greater robustness, and performs the best among the four methods. Considering the effect of Gaussian filtering, Median filtering and contrast change, the performance of our proposed method is better than the other three where the normalized hamming distances are all below 0.1, whereas in other schemes, the normalized hamming distance is above the threshold of 0.1 in some cases.

To conclude, the simulation results reveal that our scheme is superior to the schemes proposed by Guo et al., Seo et al. and Venkatean et al. The use of third scale band of wave atom transform enables the algorithm to extract invariant features from images, which are generally robust against image manipulations.



(a) Effect of JPEG Compression

(b) Effect of Gaussian Noise

(c) Effect of Gaussian Filtering

(d) Effect of Median Filtering

(e) Effect of Contrast Change

Fig. 3. Normalized hamming distance between the original images and modified images

5 Conclusion

In this paper, we have proposed a perceptual hashing scheme for image authentication based on wave atom transform and randomized pixel modulation. The proposed algorithm can authenticate the images which have undergone common content preserved image processing operations, such as JPEG compression, histogram equalization, low-pass filtering, median filtering, Gaussian noise and contrast enhancement. It is simultaneously sensitive to malicious tampering with the guaranty of system security. Instead of using traditional transform like wavelet, DCT or other transform, we propose to employ wave atom transform for the sparser expansion and better characteristics to extract texture features when compared with others. The comparison results also show that the proposed scheme achieves better performance than the schemes proposed by Guo et al [17], Seo et al [18] and Venkatean et al [3].

References

1. Schneier, B.: Applied Cryptography. John Wiley & Sons, Inc., USA (1996)
2. Schneider, M., Chang, S.F.: A robust content-based digital signature for image authentication. In: Proc. IEEE Int. Conf. Image Processing, Lausanne, Switzerland, vol. 3, pp. 227–230 (1996)
3. Venkatesan, R., Koon, S.M., Jakubowski, M.H., Moulin, P.: Robust image hashing. In: Proc. IEEE Int. Conf. Image Processing, Vancouver, BC, Canada, vol. 3, pp. 664–666 (2000)
4. Kailasanathan, C., Naini, R.S., Ogunbona, P.: Image authentication surviving acceptable modifications. In: Proc. IEEE-EURASIP Workshop on Nonlinear Signal Image Processing, Baltimore, MD (2001)
5. Lin, C.Y., Chang, S.F.: A robust image authentication system distinguishing JPEG compression from malicious manipulation. IEEE Trans. on Circuits and Systems for Video Technology 11(2), 153–168 (2001)
6. Lu, C.S., Liao, H.Y.M.: Structural digital signature for image authentication: an incidental distortion resistant scheme. IEEE Trans. Multimedia 5(2), 161–173 (2003)
7. Bhattacharjee, S., Kutter, M.: Compression tolerant image authentication. In: Proc. International Conference on Image Processing, Chicago, USA, vol. 1(7), pp. 435–438 (1998)
8. Monga, V., Evans, B.L.: Perceptual image hashing via feature points: Performance evaluation and tradeoffs. IEEE Trans. Image Process. 15(11), 3452–3465 (2006)
9. Fridrich, J., Goljan, M.: Robust hash functions for digital watermarking. In: Proc. IEEE Int. Conf. Information Technology: Coding and Computing, Las Vegas, NV, pp. 178–183 (2000)
10. Swaminathan, A., Mao, Y., Wu, M.: Robust and secure image hashing. IEEE Trans. Inf. Forens. Sec. 1(2), 215–230 (2006)
11. Monga, V., Mihcak, M.K.: Robust and secure image hashing via non-negative matrix factorizations. IEEE Trans. Inf. Forens. Sec. 2(3), 376–390 (2007)
12. Demanet, L., Ying, L.: Wave atoms and sparsity of oscillatory patterns. Applied and Computational Harmonic Analysis 23(3), 368–387 (2007)
13. Mallat, S.: A Wavelet Tour of Signal Processing, 2nd edn. Academic Press, Orlando (1999)
14. Antoine, J.P., Murenzi, R.: Two-dimensional directional wavelets and the scale-angle representation. Signal Processing 52, 259–281 (1996)
15. Leung, H.Y., Cheng, L.M.: Robust watermarking scheme using wave atoms. EURASIP Journal on Advances in Signal Processing 2011, Article ID 184817, 9 pages (2011), doi:10.1155/2011/184817
16. Ahmed, F., Siyal, M.Y., Vali, U.A.: A secure and robust hash-based scheme for image authentication. Signal Processing 90(5), 1456–1470 (2010)
17. Guo, X.C., Hatzinakos, D.: Content Based Image Hashing Via Wavelet and Radon Transform. In: Ip, H.H.-S., Au, O.C., Leung, H., Sun, M.-T., Ma, W.-Y., Hu, S.-M. (eds.) PCM 2007. LNCS, vol. 4810, pp. 755–764. Springer, Heidelberg (2007)
18. Seo, J.S., Haitsma, J., Kalker, T., Yoo, C.D.: A robust image fingerprinting system using the rado transform. Signal Processing: Image Communication 19(4), 325–339 (2004)

Witsenhausen's Counterexample and Its Links with Multimedia Security Problems

Pedro Comesaña[1,2], Fernando Pérez-González[1,2], and Chaouki T. Abdallah[1,⋆]

[1] Electrical and Computer Engineering Department, University of New Mexico,
Albuquerque, NM, USA
pcomesan@unm.edu

[2] Signal Theory and Communications Department, University of Vigo,
36310, Vigo, Spain

Abstract. Witsenhausen's counterexample was proposed more than four decades ago in order to show that affine control strategies are not optimal for systems with non-classical information patterns. Finding the optimal solution to Witsenhausen's problem however remains an open problem. Recently, the stochastic control community has re-discovered Costa's Dirty Paper result as a potential solution to Witsenhausen's problem. In this paper the similarities and differences between Witsenhausen's scenario and multimedia security problems are reviewed, and the historical evolution of the solutions to Witsenhausen's problem compared with those proposed for watermarking detection.

Keywords: Control Theory, Dirty Paper Coding, Multimedia Security, Quantization-Based Techniques, Watermark Detection, Witsenhausen's counterexample.

1 Introduction

Control theory is a multidisciplinary field of research where Engineering, Mathematics and Physics interplay. The goal of control design is to modify the input of a dynamical system (which may be thought of as a physical plant) in order to make the system's output follow a reference value. The combined system is composed of a physical plant, and a controller (usually implemented in software), which is the part of the system in charge of modifying the plant's input. Whenever randomness is involved in the system input or dynamics, stochastic control is typically used. Please see [12] for a detailed discussion of such concepts.

[⋆] Research supported by the European Union under project REWIND (Grant Agreement Number 268478), the European Regional Development Fund (ERDF) and the Spanish Government under projects DYNACS (TEC2010-21245-C02-02/TCM) and COMONSENS (CONSOLIDER-INGENIO 2010 CSD2008-00010), the Galician Regional Government under projects "Consolidation of Research Units" 2009/62, 2010/85 and SCALLOPS (10PXIB322231PR), and the Iberdrola Foundation through the Prince of Asturias Endowed Chair in Information Science and Related Technologies.

Y.Q. Shi, H.J. Kim, and F. Perez-Gonzalez (Eds.): IWDW 2011, LNCS 7128, pp. 479–493, 2012.

In real applications, several different controlled systems (with the corresponding controllers) could co-exist. If those controllers are allowed to make decisions without communicating between them or communicating with a centralized controlling entity, the resulting problem is usually denoted Decentralized Control. One of the most interesting cases in that scenario is that where the different controllers observe different input signals. It is within this particular framework that Witsenhausen proved more than four decades ago that the optimal control strategy, even when linear systems with quadratic performance objective and Gaussian noise are considered, can not be an affine function of the state [36].

Although the non-optimality of affine strategies has been formally proven, Witsenhausen's work does not establish what the optimal solution is. Indeed, this problem has received great attention, and even today the optimal solution for Witsenhausen's problem remains elusive. Interestingly, in the last years Grover *et al.* have pointed out links between distributed control, in particular one of the most promising approaches to Witsenhausen's problem, and Costa's dirty paper coding [21,22].

The objective of our work is to present Witsenhausen's problem from a media security perspective. A review of the solutions proposed for this problem are introduced and compared for the first time with those proposed for different multimedia security applications; similarities and differences between Witsenhausen's counterexample and multimedia security problems are pointed out.

The remaining of this paper is organized as follows: after briefly introducing our notation in Sect. 1.1, Witsenhausen's problem is formally presented in Sect. 2. Digital watermarking classical problems are reviewed in Sect. 3, and the solutions proposed to Witsenhausen's problem are summarized and compared with watermarking detection strategies in Sect. 4. Then, Witsenhausen's problem is compared with two classical problems in multimedia security: authentication (in Sect. 5) and reversible watermarking (in Sect. 6). Finally, the conclusions of this work are presented in Sect. 7.

1.1 Notation

We 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 L-dimensional 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. The power of signal $\mathbf{X}$ is denoted by $\sigma_X^2 \triangleq \mathrm{E}\{X_i^2\}$, being valid for any i, as the components of the considered vectors are assumed to be i.i.d..

2 Witsenhausen's Counterexample

The general objective in stochastic control is to minimize the expected value of a target function, for given noise distributions [12]. The most basic scenario is based on what is referred to as the *classical information pattern*, which assumes that all actions performed by the controllers are based on the same data, and that any data available at a given time will be also available at all later times. In that framework, and if linear systems, quadratic objective criteria and Gaussian noise are considered, the optimal solution was proven to be an affine function of the system's state.

In 1968 Hans S. Witsenhausen [36] showed by means of a counterexample, that affine control functions are no longer optimal solutions to problems where the information pattern is not classical. This counterexample is based on the modification by a first controller of a variable x (which in most of his paper is assumed to follow a Gaussian distribution) by adding a variable w. The resulting variable $y = x + w$ passes through a Gaussian channel; we denote by n the realization of that Gaussian channel, and by $z = y + n$ the output variable, which in turn feeds into a second controller that aims to provide an estimate $\hat{y}$ of y based only on z with an estimation error $q = y - \hat{y}$. This framework does not follow a classical information pattern, since x is known only to the first controller, but not to the second. The proposed target function is

$$k^2\mathrm{E}\{W^2\} + \mathrm{E}\{Q^2\},$$

i.e., the sum of a weighted version of the variance of the signal introduced by the first controller (that is, $k^2\mathrm{E}\{W^2\}$) and the variance of the estimation error at the second controller $\mathrm{E}\{Q^2\}$ (assuming that both W and Q are zero-mean).

Witsenhausen derived the optimal *affine* solution, and compared its achieved target function value with that obtained by using $w = \sigma_X \mathrm{sign}(x) - x$, and $q = y - \sigma_X \tanh(\sigma_X z)$. He was then able to show that the value of the target function achieved by this strategy is strictly smaller than that obtained by the optimal affine solution thus proving that affine solutions for control problems with non-classical information patterns are no longer optimal.

Nevertheless, as Witsenhausen established in his paper, the solution proposed as counterexample is itself far from being optimal. Since the publication of the original paper, a large number of papers in the stochastic control field have been published in an attempt to derive the optimal controlling strategies for Witsenhausen's counterexample. An optimal solution for a general framework has not yet been found, but the proposed schemes have considerably reduced the value of the target function resulting from Witsenhausen's original strategy. Some of these proposals are reviewed in the following sections, then compared with different strategies designed for dealing with multimedia security problems.

2.1 Vector Witsenhausen's Problem

Although Witsenhausen's original counterexample used scalar signals, Grover and Sahai have introduced the vector version of that problem [20], where the signal entering the first controller $\mathbf{x}$ is modified by the addition of a signal $\mathbf{w}$, yielding $\mathbf{y}$. The observation noise $\mathbf{n}$ is added to $\mathbf{y}$, so that the second controller must provide an estimate of $\mathbf{y}$, which we denote by $\hat{\mathbf{y}}$, based just on $\mathbf{z} = \mathbf{y} + \mathbf{n}$. Similarly to the scalar case, the target function in the vector scenario is given by

$$k^2\mathrm{E}\{||\mathbf{W}||^2\} + \mathrm{E}\{||\mathbf{Q}||^2\}.$$

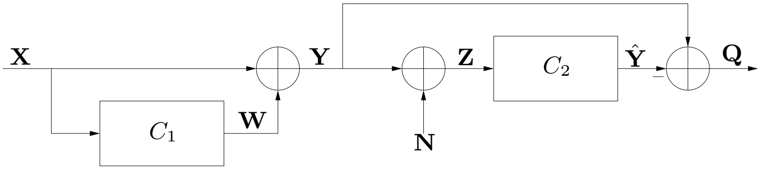


Fig. 1. Block-diagram of Witsenhausen's multidimensional problem

As we discuss later, this extension to the multidimensional case allows for the use of more complex coding strategies that lead to target function reduction.

A block-diagram of Witsenhausen multidimensional problem is found in Fig. 1.

3 Classical Problems in Digital Watermarking

In digital watermarking, two major problems are typically distinguished: one-bit watermarking (a.k.a. *zero-bit watermarking*, or *watermark detection problem*) and *multibit watermarking* (a.k.a. *watermark decoding problem*).[1] In the first problem, the receiver side tries to determine whether a given watermark, which is *a priori* known, is present or not at the available signal. This then is a binary hypothesis problem. A block-diagram of the watermark detection problem is shown in Fig. 2. On the other hand, in the multi-bit watermarking problem the presence of a watermark is assumed, and the objective is to determine which message, from a finite set of possibilities, has been embedded at the transmitter side; consequently, it is a multiple hypothesis problem.

Due to their different natures, the measures used for quantifying the goodness of one-bit and multi-bit watermarking are also different. For one-bit watermarking the probability of false-positive (or false-alarm) and the probability of false-negative (or missed-detection) are used, while in the multi-bit watermarking, the probability of decoding error (or some related measure, as the Bit Error Rate) is typically used. Considering these measures, and the imperceptibility constraints the watermark detection and decoding problems are formalized as

$$\min_{\mathbf{W}:\mathrm{E}\{||\mathbf{W}||^2\}\leq D_e, P_{fp}\leq P_{fp}^{target}} P_{fn}, \text{ and}$$
$$\min_{\mathbf{W}:\mathrm{E}\{||\mathbf{W}||^2\}\leq D_e} P_e,$$

respectively, where P_{fp} stands for the false positive probability, P_{fn} for the false negative probability, P_e for the probability of decoding error, and D_e for the maximum allowed mean embedding distortion. Note that the watermark detection problem may be defined from its dual counterpart, i.e. by fixing a target false negative probability and minimizing the false positive probability.

[1] Other watermarking problems, such as those of steganography, authentication, or reversible watermarking, may be regarded as subclasses where additional constraints or modified versions of the target function are considered.

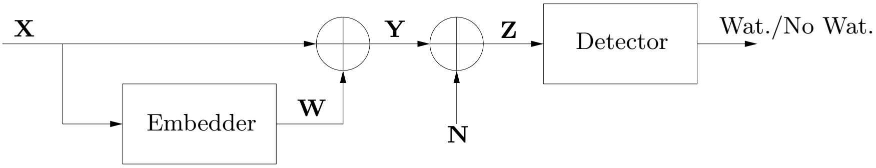


Fig. 2. Block-diagram of watermark detection problem

Since the birth of digital watermarking in the late 1990's, a large number of different strategies have been proposed for dealing with both problems. The significant advances achieved so far have dramatically improved the performance of early watermarking schemes, to the point that a well-grounded theory is now available.

Although the block-diagrams for Witsenhausen's counterexample and the watermark detection problem are rather similar, one wonders how deep this similarity actually is, and what are the main differences between both problems. The target of the next section is to explore these similarities and differences.

3.1 Similarities and Differences of Witsenhausen's Counterexample with Watermark Detection

Although at first, the two problems may seem completely different, the fact is that the problems share some common traits:

- Both schemes have a non-classical information pattern. Specifically, in watermarking detection, the embedder observes the original host signal $\mathbf{x}$, while the detector must make its decision based solely on the received attacked signal, $\mathbf{z}$. Similarly, in Witsenhausen's counterexample, the first controller observes $\mathbf{x}$, while the second controller estimates its output based only on the observation of $\mathbf{z}$.
- Another similarity stems from the constraint on the watermark variance; this constraint is explicit in the watermark detection problem, and implicit in Witsenhausen's counterexample. Indeed, in the latter case the watermark variance (i.e., the variance of the signal introduced by the first controller) can not be arbitrarily large, as this would increase the score of the target function yielding a *de facto* non-feasible point. In other words, it is the target function itself that constrains the considered embedding functions to use a reduced watermark variance. This constraint on the watermark variance will strongly influence which codes are good for transmitting the desired information from the embedder (or the first controller) to the detector (or the second controller).
- In both cases, the transmitted signals are sent through an additive white Gaussian channel. As the channel noise distribution is the same in both cases, the shape of the used codes will share some geometrical characteristics that make them suitable for coping with such noise distribution.

- Both scenarios deal with zero-rate problems, meaning that no additional information, besides that used for estimating the signal produced by the first controller or the presence of the watermark, is transmitted. This the main reason for focusing our comparison of multimedia security and Witsenhausen's counterexample on watermark detection techniques (and in the next sections, on other zero-rate multimedia security problems). A remarkable exception to the zero-rate nature of Witsenhausen's counterexample can be found in [21], where Grover and Sahai consider the use of Costa-based schemes for conveying additional information.
- In both cases, the host signal may be interpreted as an interfering factor. This is now obvious and widely recognized within the watermarking research community. In fact, one of the major drawbacks of early embedding schemes (both for detection and decoding watermarking) was the interference due to the host signal, which made the reliable transmission of information very hard. One of the first mechanisms devised for dealing with this problem was to introduce the watermark in reduced-dimensionality domains, in an attempt to improve the signal-to-noise (SNR) ratio achieved in the original domain, but at the expense of a reduced available payload.

  Similarly, in Witsenhausen's counterexample, the larger the variance of the host signal $\mathbf{x}$, the more difficult it is for the second controller to estimate the transmitted signal $\mathbf{y}$. As a degenerate case illustrating this fact, one may think of a zero-variance host signal; in such case both controllers could use trivial strategies (e.g., $\mathbf{w} = 0$ and $\hat{\mathbf{y}} = 0$) that provide a null-score of the non-negative target function, and, consequently, an optimal solution. As the variance of the host signal is increased, the design of suitable strategies become harder and the target function increases.

Concerning the differences between both problems, probably the most obvious is the fact that in Witsenhausen's counterexample, an estimate of the watermarked signal must be provided. Nevertheless, if one considers this estimate as a two-step procedure (as is done in state-of-the-art schemes), where the watermarked signal estimate is itself based on a preliminary estimate of the codeword used at the first controller, this difference is not so distinctive as one could initially think. Indeed, this first stage of the estimate in Witsenhausen's counterexample may be seen as a characteristic shared with the watermark detection problem, as in both cases one is looking for the codeword used at the embedder. In Witsenhausen's problem however, there is only one set (codebook) of possible codewords as opposed to the watermark decoding problem, where there is a codebook for each possible transmitted message.

Consequently, although both problems seem to be rather different at first sight, the fact is that strong links between them exist. These connections constitute the reason why techniques used to solve each problem are quite similar. Since the watermark detection problem is the one in multimedia security that shares more characteristics with Witsenhausen's scenario, in the next section we review some of the strategies proposed in the literature for dealing with

Witsenhausen's counterexample, and analyze their relationship with well-known watermark detection methods.

4 A Review of Existing Techniques for Witsenhausen's Problem

The first strategies proposed for minimizing Witsenhausen's target function date back to Witsenhausen's original paper [36]. The first of them, is the non-optimal affine solution. In that case, $y = \lambda x$, with λ an appropriate real constant. The multidimensional version of this embedding strategy has been used for a long time in watermarking, both for detection and decoding problems, where it is known as *Multiplicative Spread-Spectrum*. For example, Barni *et al.* analyze in [4] a watermarking scheme with embedding function $y_i = x_i(1 + \lambda b_i)$, both for watermark detection and decoding, where b_i the message to be hidden. This was not the first time that such a strategy was used in a watermarking context; Cox *et al.* [10] had employed it several years before, although just for multibit watermarking.

In the counterexample given by Witsenhausen, the signal produced by the first controller is constructed as $w = \sigma_X \text{sign}(x) - x$, so $y = \sigma_X \text{sign}(x)$. The result resembles a sign-quantization strategy. This strategy may be interpreted as an example of the well-known Quantization Index Modulation (QIM) proposed by Chen and Wornell [8], where only one index is used and the quantizer has only two possible levels that are symmetrical (antipodal) about the origin. Although this is clearly not the most general configuration of QIM, this embedding strategy also fits within the QIM framework. Additionally, and although this strategy was sufficient in Witsenhausen's scenario for improving the results achieved by the optimal affine strategy, from the watermarking perspective it has the serious drawback of the large embedding distortion that it requires (as the watermarked signal is binary antipodal). To the best of our knowledge, this embedding strategy has never been used in the multimedia security field.

A non-affine strategy similar to that proposed by Witsenhausen, but with improved performance, was put forward by Bansal and Basar [3]. In this case $w = \sigma_X \sqrt{2/\pi}\text{sign}(x) - x$, but a binary antipodal quantizer is used. Furthermore, this work proves that affine solutions may still be optimal even in non-classical information patterns scenarios, if the target function does not depend on the product of the control variables. Finally, they also considered a generalized control strategy to exploit the benefits from both linear and non-linear strategies, proposing the use of $w = \epsilon \text{sign}(x) + \lambda x - x$.

An interesting result illustrating the hardness of the search for the optimal solution to Witsenhausen's counterexample is due to Papadimitriou and Tsitsiklis [33]. The authors proved that the discretized version of Witsenhausen's problem is fundamentally intractable, as it is an NP-complete problem therefore ustifying the lack of progress in the search for the optimal solution. Additionally, they relate the complexity of the discrete problem with that of the continuous one, proving the nonexistence of realistic algorithms for the continuous case.

In [11] Deng and Ho used ordinal optimization to study Witsenhausen's counterexample. Specifically, they implicitly deal with the problem of the large embedding distortion induced by the use of binary antipodal quantizers. By introducing multilevel quantizers, the authors were able to reduce the quantization error, or in other words, the variance of the signal introduced by the first controller. As long as the quantization levels are far enough for ensuring their correct estimation at the detector, an increased number of quantization levels reduces Witsenhausen's cost.

This idea is further explored at [25] by Lee *et al.*. In that work the authors study the effect of the number of quantization levels (if the quantization levels are broken down, the first stage of Witsenhausen's target function is reduced, but the difficulty of estimating y at the second controller increases), the quantization boundary values, and the quantized values. Even more interestingly, by using numerical methods, they prove that by considering piecewise linear functions, instead of pure step functions, the target function may be decreased. This last result clearly links to another well-known concept in watermarking: *Distortion Compensation* (DC). The basic idea behind Distortion Compensation is that by adding back a part of the quantization error to the quantized signal, the quantization error variance (i.e., the watermark variance in the watermarking application, or the variance of the signal introduced by the first controller in Witsenhausen's scenario) is reduced, allowing to inflate the used quantizer, and consequently providing increased robustness against channel attacks. This idea, which was proposed for the first time by Costa [9] in a purely information theoretic scenario, has been exploited in the watermarking field by the DC-QIM [8] and Scalar Costa Scheme (SCS) [13] schemes. Remarkably, this link was overlooked in the stochastic control literature at [25] (where DC quantization schemes were implicitly proposed for the first time in that scenario), and later discovered by Grover and Sahai (see, for example, [21]), whose works we will discuss below. Finally, a hierarchical search numerical method is proposed in [25] for the computation of the parameters of the proposed scheme (number of quantization levels, quantization boundaries, and value of the quantization levels).

Similar results showing the convenience of using sloped step functions (or similarly distortion-compensated quantization strategies) have also been obtained by Baglietto *et al.* [2] and Li *et al.* [26]. Baglietto *et al.*'s methodology is based on constraining the control functions to have some parameterized fixed structure, denoted by nonlinear approximation networks; stochastic approximation is used for solving the resulting nonlinear programming problem. On the other hand, the approach followed by Li *et al.* is based on discretizing the problem and formulating it as a potential game; the authors solve it by using the learning algorithm known as *Fading Memory Joint Strategy Fictitious Play with Inertia* [30].

In recent years, Grover and Sahai have published a series of interesting articles on Witsenhausen's counterexample, where they establish connections between this problem and open problems in communications, such as those of the cognitive radio channel, the multiple access channel with partial state information at the encoder, state masking [32] or Dirty Paper Coding (DPC) [9] (although surprisingly they do not mention the links with existing data hiding methods, as the aforementioned DC-QIM and SCS). Indeed, in [21] the authors interpret Witsenhausen's problem as a particular case of the wireless communication problem which they name *Assisted Interference Suppression.* Furthermore, in [21] the authors also propose several solutions to Witsenhausen's vector problem, where the most suitable alternative is chosen depending on the working-point (defined by σ_X^2 and k^2):

- $\mathbf{w} = -\mathbf{x}$, so $\mathbf{y} = \mathbf{0}$. In this case, the estimation at the second controller is trivial, and the variance of the estimation error is null. Therefore, only the first stage of Witsenhausen's cost will be non-null (specifically, $k^2\sigma_X^2$). This strategy makes sense for small values of k^2 and small values of σ_X^2.
- $\mathbf{w} = \mathbf{0}$, so $\mathbf{y} = \mathbf{x}$. This may be considered to be the counterpart of the previous scheme, as the first stage of Witsenhausen's target function is zero. The estimation at the second controller will be an MMSE estimation (i.e., Wiener's filter) of $\mathbf{y}$ given $\mathbf{z}$, yielding a Witsenhausen's cost of $\sigma_X^2\sigma_N^2/(\sigma_X^2 + \sigma_N^2)$. This strategy makes sense for large values of k^2.
- A randomized nonlinear controller based on the quantization of $\mathbf{x}$ by using a random codebook of square radius per dimension equal to $\sigma_X^2 - \sigma_W^2$. This is a *pure* quantizer, in the sense that distortion compensation is not considered. Due to its connections with the rate-distortion function, and noticing that the cardinality of the quantizer is chosen in order to avoid decoding mistakes, the authors name this strategy *Joint Source-Channel Coding* (JSCC). This approach makes sense for small values of k^2 and large values of σ_X^2.
- Dirty Paper Coding based strategy [9]. Similarly to the previous scheme, a random codebook is used, although in this case the square radius per dimension is equal to $\alpha^2\sigma_X^2 + \sigma_W^2$, where α stands for the distortion compensation parameter. Interestingly, the authors consider both the case where DPC is used for conveying additional information to that about $\mathbf{y}$ (related to the previously explained multi-bit watermarking problem), and the case where the quantization aims at aiding the estimation of $\mathbf{y}$. Therefore, no additional information is transmitted, yielding a Costa's zero-rate problem, related to watermark detection problem. The estimation stage is based on first quantizing the received signal $\mathbf{z}$ with the considered codebook (as done in conventional DPC schemes), followed by a second stage where an MMSE estimator is applied to estimate the self-noise at the first controller. The quantization error at the first controller scaled by $(1 - \alpha)$, given the total quantization noise at the second controller (which assuming that the codeword used at the first controller is correctly determined, is the sum of the self-noise and the channel AWGN). This two-step procedure reveals one of the similarities between Witsenhausen's counterexample and watermark detection problems as outlined in Sect. 3.1.
- Marginal improvements are also achieved by implementing the first controller as a two-stage process in which $\mathbf{X}$ is first scaled down to reduce its power, and then DPC is applied.

Additionally, it is important to point out that in [21] the authors show that the performance that can be achieved by using Witsenhausen's multidimensional version problem, is strictly better than that achieved by using its scalar counterpart, due to the advantages brought about by multidimensional codes.

Grover *et al.* also proposed in [19] the use of lattice-based quantization strategies for performing the quantization. Their analysis is based on considering the packing and covering radii, following an approach similar to that in [14].

Although lattice-based quantization strategies to multimedia security have been typically used for mult-ibit watermarking, where their optimality was proven by Erez and Zamir [15], there are also examples where they have been used for detection purposes. One of the most relevant contributions in this direction is due to Liu and Moulin [27], where the authors derive the error exponents of watermarking detection both for Additive Spread Spectrum and QIM. They assume that lattices are used in the QIM case and that the distortion compensation parameter takes the value proposed by Costa [9], being the detection region a hypershpere. Other work where quantization-based schemes were suggested for dealing with one-bit watermarking is [34], where Pérez-Freire *et al.* proposed to quantize the correlation of a series of pseudorandom-sequences and the original host signal (i.e., the projection of $\mathbf{x}$ onto a pseudorandomly generated subspace) without considering distortion compensation. Furthermore, they propose several detection regions in order to determine if the received signal $\mathbf{z}$ is watermarked or not.

Finally, we would like to mention that strategies which are not based on quantizing the host signal were proposed in the watermarking literature for reducing the host signal interference. As an example, in [6] Cannons and Moulin's effectively reduce the host signal interference thanks to the exploitation of a hash of the original signal available at the detector. Given that the hash provides information about the original host signal belonging to a given subset of the signal space, it allows to condition the probability density function (pdf) considered by the detector.

Another strategy for reducing host signal interference not based on quantization, is the Improved Spread Spectrum technique due to Malvar and Florencio [29], which was initially intended for decoding watermarking scenarios. In contrast, no control strategies similar to [6] or [29] have been suggested for dealing with Witsenhausen's counterexample.

5 Links between Witsenhausen's Counterexample and Authentication

Multimedia authentication methods may be basically divided into two main categories: those that complement the digital content under analysis with an additional authentication code used for checking the authenticity (e.g. Image Messages Authentication Codes [37], Approximate Message Authentication Codes [18], or Noise Tolerant Message Authentication Code [5]), and those that embed the information required for checking whether the considered contents is a fake or not, in the content itself by using watermarking techniques. As we are interested in the comparison with Witsenhausen's counterexample, we focus in this work on the second category. For those methods, and similarly to zero-bit watermarking, the authentication problem is a binary hypothesis test, where the binary decision tries to determine if a given content was modified or not. In any case, as long as the authenticating methods are based on estimating the codeword used at the embedder, most of the similarities pointed to above between Witsenhausen's counterexample and watermark detection still hold.

These similarities are even more pronounced for some particular authentication methods, such as the one due to Martinian *et al.* [31]. In that work, the authentication process is split into two steps: first, the codeword used at the embedder is estimated, and then a reconstruction of the original host signal is produced. This estimate of the original host signal $\mathbf{x}$ (instead of $\mathbf{y}$, as in Witsenhausen's counterexample) is required to be free from the effects of any modifications by the editor, so it will be effectively defined by the encoder. Furthermore, the set of possible modifications on the watermarked signal $\mathbf{y}$ is constrained to verify a so-called reference channel model. For the case of Gaussian host and channel and quadratic distortion measures, Martinian *et al.* use a Gaussian random codebook for quantizing the original host signal, including distortion compensation.

A scheme conceptually similar to that in [31], but without the final estimation stage, was proposed by Fei *et al.* [16]. There, the authors define the *admissible set*, a deterministic set that characterizes the legitimate modifications that an authenticated signal may be subjected to in order to be considered authentic, similarly to the reference channel in [31]. Additionally, Fei *et al.*'s scheme is also based on the quantization of the original host signal, although in this case the use of distortion compensation is not considered, and lattice-based quantization instead of random quantization is used. This difference raises a security problem since a periodic structure, such that of the lattice, should not be used for determining the authenticity of a content, since as soon as an attacker has access to any authenticated object he/she could produce as many falsely authenticated contents as he/she wishes. This of course is not the case for random-coding based schemes, where the observation of a centroid does not supply any information about the rest of the codewords in the considered . On the other hand, this lack of structure renders random-coding-based schemes difficult to implement in practice. The solution proposed by Fei *et al.* relies on two nested lattices, where the coarse lattice is actually used for quantizing the original host signal, and in a second stage a point of the fine lattice within the Voronoi region of the chosen coarse-lattice centroid is pseudo-randomly selected depending on a secret key and the coarse-lattice centroid itself. In doing so, the security of the resulting scheme can be increased, in the sense that an attacker observing other watermarked contents but who is not aware of the secret key, could not produce a falsely authenticated content.

Interestingly, both embedding authentication methods just reviewed reveal a subtle similarity and a inherent difference between Witsenhausen's scenario and authentication applications. On one hand, Witsenhausen's solutions can be seen to include a reference channel. This is the case for example, for DPC-based techniques where the distortion compensation parameter value is determined by considering a certain channel distribution (AWGN) and a given noise variance. If the actual channel at the output of the first controller were different, then estimation errors could arise; for example, if the variance of the AWGN were underestimated when computing the distortion compensation parameter, the total noise variance might be larger than the maximum allowed one for guaranteeing an error-free decoding. On the other hand, one does not need to take into account the security constraints in Witsenhausen's counterexample, as an attacker does not exist in that framework.

6 Links between Witsenhausen's Counterexample and Reversible Watermarking

Since reversible watermarking also aims to estimate a signal which is not known at the receiver side, one may think that this is the multimedia security problem that is more similar to Witsenhausen's problem. Nevertheless, several characteristics of reversible watermarking, and especially of the scenarios where its use was proposed, give rise to some important differences:

- All reversible watermarking schemes we surveyed are multi-bit. This probably owes to the applications reversible watermarking was designed for, e.g. medical or military images [17], in contrast to copyright applications, where watermark detection is typically used.
- Similarly to [31], in reversible watermarking the decoder tries to estimate the original host signal $\mathbf{x}$, instead of the watermarked signal $\mathbf{y}$ (as is the case in Witsenhausen's problem).
- All studied reversible watermarking schemes consider a discrete host alphabet, in contrast to Witsenhausen's Gaussian-distributed x. This is not a trivial difference, as it will strongly determine the feasible solutions in each case.
- Most of the studied reversible watermarking schemes do not consider that the watermarked content could be further corrupted by noise, i.e. they assume that $\mathbf{n} = \mathbf{0}$. Again, this assumption may make sense in most reversible watermarking applications. One exception to this lack of channel noise consideration is found in [17], although it is studied just from a empirical point of view.
- Most of the studied schemes are constrained to provide perfect estimates of the host signal, i.e. $\hat{\mathbf{X}} = \mathbf{X}$. Remarkably, this is not the case in [35], where a non-null distortion between the host signal and its estimate is considered.
- Most of the studied schemes are based on performing a lossless coding of the Least Significative Bits (or of some kind of prediction error) and use the remaining room for sending the additional data (e.g. [17], [1], [24]). Nevertheless, this strategy was proven to be non-optimal [23].

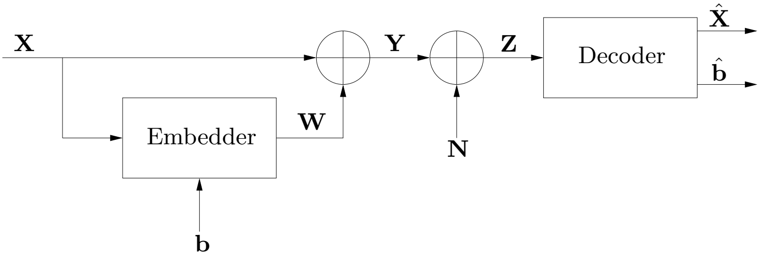


Fig. 3. Typical block-diagram of reversible watermarking

Concerning the similarities, some reversible watermarking schemes using quantization methods exist (e.g. [7]), although this is not the most general approach to this problem. Additionally, embedding distortion constraints are typically considered. Concerning this last point, although at first sight it could seem to be obvious, note that given that the target is just the estimation of the original host signal at the decoder, the watermarked signal may be arbitrarily far from the original host. Indeed, methods where the distortion embedding constraint is not considered due to this fact have been proposed [28].

A typical block-diagram of reversible watermarking illustrating some of the characteristics mentioned above can be found in Fig. 3.

7 Conclusions

Similarities and differences between multimedia security problems and Witsenhausen's counterexample have been revealed. Although the definitions and application domains of these problems are intrinsically different, the similarities we have spotted (especially for authentication and watermark detection) explain why similar solutions had been independently proposed for both problems.

One of the main conclusions one can derive from this comparison is the fact that dirty paper coding is suitable for reducing the host interference (or state interference in control) in a range of scenarios much wider than the one initially proposed by Costa. This fact seems to encourage the use of dirty paper based techniques for multimedia security applications where it has not been used so far, as for example reversible watermarking, robust hashing or active forensics. Although it could well be the case that DPC were not optimal in those scenarios, as it is the case in Witsenhausen's counterexample, if the gain with respect to conventional approaches were as large as for Witsenhausen's problem, DPC would be worth considering.

References

1. Alattar, A.M.: Reversible watermark using the difference expansion of a generalized integer transform. IEEE Transactions on Image Processing 13(8), 1142–1156 (2004)

2. Baglietto, M., Parisini, T., Zoppoli, R.: Numerical solutions to the Witsenhausen counterexample by approximating networks. IEEE Transactions on Automatic Control 46(9), 1471–1477 (2001)
3. Bansal, R., Basar, T.: Stochastic teams with nonclassical information revisited: When is an affine law optimal. IEEE Transactions on Automatic Control 32(6), 554–559 (1987)
4. Barni, M., Bartolini, F., De Rosa, A., Piva, A.: Optimum decoding and detection of multiplicative watermarks. IEEE Transactions on Signal Processing 51(4), 1118–1123 (2003)
5. Boncelet, C.G.: The NTMAC for authentication of noisy messages. IEEE Transactions on Information Forensics and Security 1(1), 35–42 (2006)
6. Cannons, J., Moulin, P.: Design and statistical analysis of a hash-aided image watermarking system. IEEE Transactions on Image Processing 13(10), 1393–1408 (2004)
7. Celik, M.U., Sharma, G., Tekalp, A.M., Saber, E.: Reversible data hiding. In: Proceeding of the IEEE International Conference on Image Processing, vol. 2, pp. 157–160 (December 2002)
8. 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)
9. Costa, M.H.M.: Writing on dirty paper. IEEE Transactions on Information Theory 29(3), 439–441 (1983)
10. Cox, I.J., Kilian, J., Leighton, F.T., Shamoon, T.: Secure spread spectrum watermarking for multimedia. IEEE Transactions on Image Processing 6(12), 1673–1687 (1997)
11. Deng, M., Ho, Y.C.: An ordinal optimization approach to optimal control problems. Automatica 35(2), 331–338 (1999)
12. Dorato, P., Abdallah, C., Cerone, V.: Linear Quadratic Control: An Introduction. Krieger Publishing Company, Malabar (2000)
13. Eggers, J.J., Bäuml, R., Tzschoppe, R., Girod, B.: Scalar Costa Scheme for information embedding. IEEE Transactions on Signal Processing 51(4), 1003–1019 (2003)
14. Erez, U., Listsyn, S., Zamir, R.: Lattices which are good for (almost) everything. IEEE Transactions on Information Theory 51(10), 3401–3416 (2005)
15. Erez, U., Zamir, R.: Achieving $\frac{1}{2}\log(1+\mathrm{SNR})$ on the AWGN channel with lattice encoding and decoding. IEEE Transactions on Information Theory 50(10), 2293–2314 (2004)
16. Fei, C., Kundur, D., Kwong, R.H.: Analysis and design of secure watermark-based authentication systems. IEEE Transactions on Information Forensics and Security 1(1), 43–55 (2006)
17. Fridrich, J., Goljan, M., Du, R.: Lossless data embedding – new paradigm in digital watermarking. EURASIP Journal on Applied Signal Processing (2), 185–196 (2002)
18. Ge, R., Arce, G.R., Di Crescenzo, G.: Approximate message authentication codes for n-ary alphabets. IEEE Transactions on Information Forensics and Security 1(1), 56–67 (2006)
19. Grover, P., Park, S.Y., Sahai, A.: The finite-dimensional witsenhausen counterexample. IEEE Transactions on Automatic Control (submitted)

20. Grover, P., Sahai, A.: A vector version of Witsenhausen's counterexample: A convergence of control, communication and computation. In: Proceedings of the 47th IEEE Conference on Decision and Control (CDC), pp. 1636–1641 (December 2008)
21. Grover, P., Sahai, A.: Witsenhausen's counterexample as assisted interference supression. International Journal on Systems, Control and Communications 2(1/2/3), 197–237 (2010)
22. Grover, P., Wagner, A., Sahai, A.: Information embedding meets distributed control. IEEE Transactions on Information Theory (2010) (submitted)
23. Kalker, T., Willems, F.: Capacity bounds and constructions for reversible datahiding. In: Proceedings of the IEEE International Conference on Digital Signal Processing, vol. 1, pp. 71–76 (April 2002)
24. 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)
25. Lee, J.T., Lau, E., Ho, Y.C.: The Witsenhausen counterexample: A hierarchical search approach for nonconvex optimization problems. IEEE Transactions on Automatic Control 46(3), 382–397 (2001)
26. Li, N., Marden, J.R., Shamma, J.S.: Learning approaches to the Witsenhausen counterexample from a view of potential games. In: Proceedings of the 48th IEEE Conference on Decision and Control, pp. 157–162 (December 2009)
27. Liu, T., Moulin, P.: Error exponents for one-bit watermarking. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 3, pp. 65–68 (April 2003)
28. Macq, B.: Lossless multiresolution transform for image authenticating watermarking. In: Proceedings of EUSIPCO, pp. 533–536 (September 2002)
29. Malvar, H.S., Florencio, D.A.F.: Improved spread spectrum: A new modulation technique for robust watermarking. IEEE Transactions on Signal Processing 51(4), 898–905 (2003)
30. Marden, J.R., Arslan, G., Shamma, J.: Joint strategy fictitious play with inertia for potential games. IEEE Transactions on Automatic Control 54(2), 208–220 (2009)
31. Martinian, E., Wornell, G.W., Chen, B.: Authentication with distortion criteria. IEEE Transactions on Information Theory 51(7), 2523–2542 (2005)
32. Merhav, N., Shamai, S.: Information rates subject to state masking. IEEE Transactions on Information Theory 53(6), 2254–2261 (2007)
33. Papadimitriou, C.H., Tsitsiklis, J.: Intractactable problems in control theory. SIAM J. Control and Optimization 24(2), 639–654 (1986)
34. Pérez-Freire, L., Comesaña, P., Pérez-González, F.: Detection in quantization-based watermarking: Performance and security issues. In: Delp III, E.J., Wong, P.W. (eds.) Proceedings of SPIE. Security, Steganography, and Watermarking of Multimedia Contents VII, vol. 5681, pp. 721–733. SPIE, San Jose (2005)
35. Willems, F., Kalker, T.: Reversible embedding methods. In: Proceedings of the 40thAllerton Conference on Communications, Control and Computing, pp. 1462–1471 (2002)
36. Witsenhausen, H.S.: A counterexample in stochastic optimum control. SIAM J. Control 6(1), 131–147 (1968)
37. Xie, L., Arce, G.R., Graveman, R.F.: Approximate image message authentication codes. IEEE Transactions on Multimedia 3(2), 242–252 (2001)

Data Forensics Constructions from Cryptographic Hashing and Coding

Giovanni Di Crescenzo[1] and Gonzalo Arce[2]

[1] Telcordia Technologies, Piscataway, NJ, USA
giovanni@research.telcordia.com
[2] University of Delaware, Newark, DE, USA
arce@ece.udel.edu

Abstract. Data forensics needs techniques that gather digital evidence of data corruption. While techniques like error correcting codes, disjunct matrices and cryptographic hashing are frequently studied and used in practical applications, very few research efforts have been done to rigorously evaluate and combine benefits of these techniques for data forensics purposes. In this paper we formulate unifying algorithm, data and security models that allow to evaluate and prove the security guarantees provided by direct forensic encoding constructions from these techniques and suitable combinations of them. We rigorously clarify the different security guarantees provided by using these techniques (alone or in some standard or novel combinations) for both data at rest and data in transit. Our most novel construction provides a forensic encoding scheme that allows to detect if any errors were introduced by corrupted data senders, does not allow data intruders to detect whether the data was encoded or not, and requires no data expansion in a large-min-entropy data model, as typical in multimedia data.

1 Introduction

The area of data forensics studies the design and analysis of techniques to gather some digital evidence of events related to computer or network related malicious behaviour. Typical scenarios for data forensics include data at rest and data in transit. In addition to computer memory or hardware faults, data at rest can be modified as a result of computer virus infections, unauthorized modifications by malicious computer intruders, and undesired modifications by data owners. Real-life application scenarios include changes to computer files detected by anti-virus systems, e-mail and multimedia data stored at third-party servers, and storage in a cloud in the recently emerging cloud-computing paradigm. In addition to being subject to communication errors, data in transit can be modified as a result of unauthorized modifications by malicious men-in-the-middle, and undesired modifications by data senders. Here, real-life application scenarios include downloading software applications over the Internet, which often allow clients to check for application authenticity, and multimedia data services, including RSS feeds and publish/subscribe protocols.

In all these scenarios, both detecting and correcting data corruptions are goals of fundamental importance as part of data forensics. In this paper, we study forensic data encoding/decoding schemes that process data so to allow the detection and correction

Y.Q. Shi, H.J. Kim, and F. Perez-Gonzalez (Eds.): IWDW 2011, LNCS 7128, pp. 494–509, 2012.

of corruptions on both data at rest and data in transit. In particular, we focus on data modifications carried out by attackers of two different types: storage intruders and corrupted data senders, the latter type being an especially novel type in the case of error correction. Carrying out this type of investigation requires the solution of a number of challenges: formally defining suitable algorithmic models that capture both scenarios and related data models and security notions, reformulating known techniques from coding theory and cryptography and evaluating results obtained from them in these models, and finally providing new constructions with enhanced security guarantees.

Our Contribution. We propose a unifying modeling approach that allows to formally define and analyze forensic encoding constructions for both data at rest and data in transit, to be protected about both data intruders and corrupted data senders. In this model, we can first of all formulate suitable generalizations of formal security notions in the literature, such as: error detection against data intruders and error corrections against data intruders. We can then further define new notions that have apparently not been considered in the literature, such as conditional error detection against corrupted data senders (a generalization of a known error detection notion, where the condition restricts the type of errors which a corrupted data sender can generate; e.g., two very different images); error correction against corrupted data senders and encoding indistinguishability against data intruders (requiring that a data intruder is not able to distinguish an data item encoded by a honest user from one that was not encoded at all). This model allows us to formally define forensics encoding schemes from well-known tools from coding theory and cryptography and formally prove their security guarantees, as follows:

1. using arbitrary error correcting codes to define a forensic encoding scheme that satisfies error correction against data intruders;
2. using superimposed codes to define a forensic encoding scheme (where a sender uses secret data) that satisfies error correction and encoding indistinguishability against data intruders; and
3. using cryptographic (collision-resistant) hash functions to define a forensic encoding scheme that satisfies error detection against corrupted data senders.

Although all of these constructions can be considered folklore, and variations of them are often considered in the multimedia security, watermarking, steganography, and cryptography literatures, they were apparently never evaluated in a unified security model which clarifies their different properties with respect to error detection, error correction and encoding indistinguishability. Our paper clarifies their differences and may help clarifying associated security claims in future, higher-level, constructions.

We then target two stronger security goals with two additional constructions. Using a natural combination of superimposed codes and cryptographic hash functions, our fourth scheme provably achieves error detection against corrupted data senders, error correction against data intruders, and encoding indistinguishability against data intruders. Combinations of error correcting codes and cryptographic hash functions were proposed in several papers, including [5,8,6,4], targeting different security guarantees than ours. The closest constructions from the literature are from [16,14], where the authors also combine superimposed codes and cryptographic hash functions to encode data items stored in different types of storage data structures, but without providing a proof of security for their targeted error correction property. Our construction, defined for a linear array storage data structure, clarifies what security properties (error correction and more) are achievable by combining these two primitives, and naturally extends to the data structure types in [16,14]. Our fifth construction is a novel scheme based on a combination of universal hash functions and cryptographic (collision-resistant and pseudo-random) hash functions, and when applied to a large-min-entropy data model, can be proved to simultaneously achieve conditional error detection against corrupted data senders, encoding indistinguishability against data intruders, and zero message expansion (i.e., the data length remains unchanged after the encoding procedure). Large-min-entropy data models have often been used in the provable-security steganography literature (starting with [17]). Although several schemes in the watermarking and steganography literatures target similar goals to ours, we are not aware of a construction that was shown to provably achieve these security goals with zero message expansion. A summary of the properties achieved by our constructions can be found in Figure 1.

Related areas where these data forensics notions are of interest include coding theory, digital watermarking, software download, program checking, memory correctness checking, combinatorial group testing, and authenticated data structures. Some previous work (see, e.g., [6,16,8]) already includes detailed discussions of these areas.

Scheme name	Error detection	Error correction	Data intrusion	Sender corruption	Encoding indistinguishability	Zero data expansion
fS_1	yes	yes	yes	no	no	no
fS_2	yes	yes	yes	no	yes	no
fS_3	yes	no	no	yes	yes	no
fS_4	yes	yes	no	yes	yes	no
fS_5	yes	no	no	yes	yes	yes

Fig. 1. The table characterizes our results on the 5 schemes in the paper. Columns 3 and 4 in the table describe in which adversary models the results in columns 1 and 2 are obtained.

2 Model and Security Notions

In this section we present a number of basic definitions, adversary models, data models, and security notions that are of interest in the rest of the paper. In particular, we introduce new formal definitions of forensic encoding/decoding schemes, of a large-min-entropy data model that specifically captures a variety of multimedia data items, and of security notions, including error detection and correction properties against corrupted data senders, and encoding indistinguishability against data intruders.

Basic definitions. Let A be a (possibly probabilistic) polynomial-time algorithm (briefly, an *efficient* algorithm). The notation $A^{O(\cdot)}$ denotes an *oracle algorithm*, defined as an algorithm A that can make a polynomial number of queries to and receive associated answers from an oracle $O(\cdot)$. By $y \leftarrow z$ we denote assignment, by $y \leftarrow S$ the random choice of y from set S, and by $y \leftarrow A(x_1, x_2, \ldots)$ the (possibly probabilistic) process of running algorithm A on input $x_1, x_2, \ldots$ and any random coins, giving y as output.

Algorithm models. We propose a model of forensic encoding schemes that captures two main application scenarios: a sender communicating the data item to a receiver (the communication scenario), and a client storing a data item at a server's storage area, possibly using the client's own smaller storage area (the storage scenario). For both scenarios, a forensic encoding scheme can be described by two algorithms: a *forensic encoding* algorithm, that performs data preprocessing before storing or communicating the data in a way that later allows forensic analysis, and a *forensic decoding* algorithm, that performs the forensic analysis over a (potentially corrupted) data item at a later stage. On input a cryptographic key k and a string m, the forensic encoding algorithm fEnc returns a pair (m', tag), Here, the string m models the data item; the string tag models a tag for the data item and the value m' models a processed version of the data item m. The string m' is intended to be publicly accessible (including by adversaries) while the string tag is not (e.g., it is kept secret by the client in the storage scenario). Both values possibly include information helpful for forensic analysis. On input a cryptographic key k, an error parameter v, and strings m'', tag, the forensic decoding algorithm fDec returns an output out. Here, out models details of the forensics analysis and the string m'' models a potentially corrupted version of string m'.

Binary Data Model. In all of our results, we simply model a *data item* m as a sequence of bits $m_1, \ldots, m_n$, where n denotes the *length* of the item.

Large-min-entropy Binary Data Model. In this model we consider digital data items that capture various applications, including transmission and storage of multimedia files. In our main result, we use a stronger characterization of multimedia data items, where a data item can be considered as as a sequence of bits generated by a random variable and divided into *data blocks*, and each data block has multiple distinct yet application-equivalent representations (e.g., images or video samples that, although different, appear equal from the point of view of an application or adversary). More formally, we say that the data item m has a (r, p)*-dense representation* if m can be written as a sequence $B_1, \ldots, B_p$ of data blocks, where data block B_i can be written as a sequence of bits $m_{(i-1)n/p+1}, \ldots, m_{i \cdot n/p}$, for $i = 1, \ldots, p$, (we assume for simplicity that p divides n), and each block B_i is drawn from a distribution D_i of support size equal to r; i.e., B_i has r denoted as $B_{i,0}, B_{i,1}, \ldots, B_{i,r-1}$, satisfying the following additional three properties:

1. *(distinctness)*: $B_{i,0}, \ldots, B_{i,r-1}$ are distinct;
2. *(representation equivalence)*: $B_{i,0}, \ldots, B_{i,r-1}$ are equally likely or a multimedia application or an adversary see them as equivalent;
3. *(representation index efficiency)*: there exists an efficient (in n) algorithm rI that, given as input $(i, B_{i,j})$, returns the index j of the block within its possible representations;
4. *(representation change efficiency)*: there exists an efficient (in n) algorithm rC that, given as input $(j, i, B_{i,j})$, returns an alternative representation $B_{i,j'}$ of the same block, for any $j' \in \{1, \ldots, r-1\} \setminus \{j\}$.

In the rest of the paper, we will only use data items that have an (r, p)-dense representation, for $r = 2$ and various values of p. Our data model is validated by the fact that typical multimedia files (i.e., images and video) are expected to have an (r, p)-dense representation, for reasonably large values of parameters r, p.

Adversary models. We consider adversaries that attempt to corrupt data items by introducing data modifications or errors in a way that such errors evade any forensics attempts. Specifically, we consider three main adversary goals, all being often considered in the coding theory and cryptography literature:

1. preventing the *detection* of the existence of data modification;
2. preventing the *correction* of all data modifications; and
3. *distinguishing* whether data was preprocessed by the sender's encoding algorithm.

Furthermore, two main types of adversaries will be considered, both being often considered in the coding theory and cryptography literature:

1. adversaries that act as *intruders* into the receiver's public storage, and
2. adversaries that act as *corrupted senders* and modify the sender's program.

By mixing and matching adversary goals with adversary types, we can formulate 5 security notions (one of the 6 feasible ones is not technically meaningful) of interest in our studies of forensic encoding schemes. The first three notions, corresponding to the three adversary goals in conjunction with the intruding adversaries, are adaptations of notions already considered in the areas of error detection, error correction and steganography, respectively. The remaining two notions, corresponding to the detection and correction adversary goals in conjunction with the corrupted senders adversary type, are new but can be seen as careful combinations of the previous notions with adversary modeling done in the area of cryptographic hashing.

Given two n-bit data items m and m', we measure their difference using the Hamming distance (i.e., the number of bits in which they differ). The binary vector e, defined as having a vector component $= 1$ if x and x' differ in this component and 0 otherwise, is also called the *error vector* of m, m'. Note that $e = m \oplus m'$, where $\oplus$ denotes bitwise logical XOR. Then, an intruder adversary can be considered as an adversary that changes m into m' by returning a v-weight error vector e (i.e., a vector e with at most v 1's). Instead, a sender corrupting adversary can be considered as an adversary that returns m and m' such that the error vector $e = m \oplus m'$ has weight at most v. For sake of greater generality, and to align our adversary model with the cryptography literature, we also allow the adversary to adaptively issue queries to a forensic encoding oracle both before and after returning its corruption, as just described. Specifically, the adversary can compute a new query based on the previous ones and the received answers, ask the new query to the oracle, and continue until he is ready to issue the corruption.

Security Notions. We now define 5 security notions for arbitrary forensic encoding schemes. The first 2 notions model the capability of detecting and correcting errors introduced on the data item by an adversary acting as an intruder.

Definition 1. *Let pair of algorithms* $fS = (\mathrm{fEnc}, \mathrm{fDec})$ *denote a forensics encoding scheme, where both* fEnc *and* fDec *run in time polynomial in* n*.*

We say that scheme fS *satisfies* (t, ϵ)-error detection against data intruders *if for any oracle algorithm* A *running in time* t*, experiment* $\mathsf{Exp1}^{\mathrm{fEnc},\mathrm{fDec},A}(1^n)$ *(defined below) returns 1 with probability at most* ϵ*.*

We say that scheme fS satisfies (t, v, ϵ)-error correction against data intruders *if for any oracle algorithm A running in time t, experiment* $\mathsf{Exp2}^{\text{fEnc},\text{fDec},A}(1^n, v)$ *(defined below) returns 1 with probability at most ϵ.*

```
$\mathsf{Exp1}^{\text{fEnc},\text{fDec},A}(1^n)$
$k \leftarrow K$;
$m \leftarrow M$;
$(m', tag) \leftarrow \text{fEnc}(k, m)$
$m'' \leftarrow A^{\text{fEnc}(\cdot)}(m')$
$out \leftarrow \text{fDec}(k, m'', tag)$
if $out = 1 \wedge m' \neq m''$ then return: 1
if $out = 0 \wedge m' = m''$ then return: 1
return: 0.
```

```
$\mathsf{Exp2}^{\text{fEnc},\text{fDec},A}(1^n, v)$
$k \leftarrow K$;
$m \leftarrow M$;
$(m', tag) \leftarrow \text{fEnc}(k, m)$
$m'' \leftarrow A^{\text{fEnc}(\cdot)}(m')$
$out \leftarrow \text{fDec}(v, k, m'', tag)$
if $d(m', m'') > v$ then return: 0
if $out \neq m' \oplus m''$ then return: 1
    else return: 0.
```

The next security notion models the capability of encoding data items without allowing an adversary acting as an intruder to distinguish an encoded data item from one that was not encoded.

Definition 2. *Let pair of algorithms $fS = (\text{fEnc}, \text{fDec})$ denote a forensics encoding scheme, where both* fEnc *and* fDec *run in time polynomial in n.*

We say that scheme fS satisfies (t, ϵ)-encoding indistinguishability against data intruders *if for any oracle algorithm A running in time t,*

$$\text{Prob}[\,\mathsf{Exp3}^{\text{fEnc},\text{fDec},A}(1^n) = 1\,] - \text{Prob}[\,\mathsf{Exp4}^{\text{fEnc},\text{fDec},A}(1^n) = 1\,]| \leq \epsilon,$$

where experiments $\mathsf{Exp3}, \mathsf{Exp4}$ *are defined below.*

```
$\mathsf{Exp3}^{\text{fEnc},\text{fDec},A}(1^n)$
$k \leftarrow K$;
$m \leftarrow M$;
$(m', tag) \leftarrow \text{fEnc}(k, m)$
$out \leftarrow A^{\text{fEnc}(\cdot)}(m')$
return: $out$.
```

```
$\mathsf{Exp4}^{\text{fEnc},\text{fDec},A}(1^n)$
$k \leftarrow K$;
$m \leftarrow M$;
$out \leftarrow A^{\text{fEnc}(\cdot)}(m)$
return: $out$.
```

The final security notions model the capability of conditionally detecting [7] and correcting errors introduced on the data item by an adversary acting as a corrupted sender.

Definition 3. *Let ρ be a predicate and let algorithms $fS = (\text{fEnc}, \text{fDec})$ denote a forensics encoding scheme, where ρ,* fEnc *and* fDec *run in time polynomial in n.*

We say that scheme fS satisfies (ρ, t, ϵ)-conditional error detection against data sender corruption *if for any oracle algorithm A running in time t, the probability that experiment* $\mathsf{Exp5}^{\text{fEnc},\text{fDec},A}(1^n)$ *(defined below) returns 1 is at most ϵ.*

We say that scheme fS satisfies (t, v, ϵ)-error correction against data sender corruption *if for any oracle algorithm A running in time t, the probability that experiment* $\mathsf{Exp6}^{\text{fEnc},\text{fDec},A}(1^n, v)$ *(defined below) returns 1 is at most ϵ.*

```
$\mathsf{Exp5}^{\mathrm{fEnc},\mathrm{fDec},A}(v)$
1. $k \leftarrow K$;
2. $(m, m'') \leftarrow A^{\mathrm{fEnc}(\cdot)}(1^n)$
3. $(m', tag) \leftarrow \mathrm{fEnc}(k, m)$
4. $out \leftarrow \mathrm{fDec}(k, m'', tag)$
5. if $out = 1 \wedge \rho(m', m'') = 1$ then return: 1
6. return: 0.
```

```
$\mathsf{Exp6}^{\mathrm{fEnc},\mathrm{fDec},A}(v)$
1. $k \leftarrow K$;
2. $(m, m'') \leftarrow A^{\mathrm{fEnc}(\cdot)}(1^n)$
3. $(m', tag) \leftarrow \mathrm{fEnc}(k, m)$
4. $out \leftarrow \mathrm{fDec}(v, k, m'', tag)$
5. if $d(m', m'') > v$ then return: 0
6. if $out \neq m' \oplus m''$ then return: 1
       else return: 0.
```

We note that although we have modeled the adversary as having access to the $\mathrm{fEnc}(k, \cdot)$ oracle, our definitions directly extend to the case where the adversary also has access to the $\mathrm{fDec}(k, \cdot)$ oracle (see, e.g., [1]).

3 Basic Constructions from Coding and Cryptography

We consider three basic constructions using known primitives from coding theory (specifically, arbitrary error correcting codes and superimposed codes) and cryptography (specifically, collision-resistant hashing). Although giving essentially folklore results, these schemes show that the definitions from Section 2 introduce a unifying framework to express and evaluate known (and, later, new) results for data forensic encoding.

3.1 Forensic Encoding from Error Correction Codes

Background. We assume familiarity with error correcting codes. We denote as (eccEnc, eccDec) an error correcting code, where eccEnc is the encoding algorithm and eccDec is the decoding algorithm, and by v we denote the number of corrected (bit) errors in an n-bit data item.

Construction, Scenario and Properties. We can define a forensic encoding scheme $\mathrm{fS}_1 = (\mathrm{fEnc}_1, \mathrm{fDec}_1)$ directly from (eccEnc, eccDec). Specifically, the fEnc_1 algorithm, on input a cryptographic key k and a data item m, returns a pair (m', tag), computed as follows: $m' = \mathrm{eccEnc}(m)$ and tag is empty. The fDec_1 algorithm, on input a cryptographic key k, an error parameter v, and strings m'', tag, returns an output out containing the output of the correction algorithm eccDec on input m''. (Note that neither fEnc_1 nor fDec_1 use k.)

Such a scheme could be used in the following client-server scenario. Given data item m, the client uses the forensic encoding algorithm fEnc_1 to generate m' and send it to the server, who stores it. Later, assume an adversary intrudes into the server's storage and is able to introduce up to v errors in m', changing it into m''. At any later time, when retrieving m'' from the server, the client can run algorithm fDec_1 to correct up to v errors in m'' and thus recover m.

As a direct consequence of this discussion and of the properties of error correcting codes, we obtain the following

Theorem 1. *If (eccEnc,eccDec) is an error correcting code correcting up to v errors on n-bit data items. Then the forensic encoding scheme $\mathrm{fS}_1 = (\mathrm{fEnc}_1, \mathrm{fDec}_1)$ satisfies (t, v, ϵ)-error correction against data intruders, for $\epsilon = 0$ and any t polynomial in n.*

3.2 Forensic Encoding from Superimposed Codes

Background. Superimposed codes are frequently studied in the group testing literature. They are a type of systematic codes, in that the encoding algorithm adds a tag to the input message, which is otherwise left unchanged. The tag is computed by matrix product between a special matrix, sometimes also called a disjunct or disjunct matrix, and the data item, seen as a vector. A matrix $M_{s \times n}$ is a *v-disjunct* if for each set V of v column indices, and each column index $j \notin V$, there exists a row index i such that $M(i,j) = 1$ and $M(i,j) = 0$ for all $j \in V$. We note that in a v-disjunct matrix, for any v columns, there exists another column that is not covered by these v columns. Moreover, if $\mathcal{S}$ is the family of subsets of $\{0,\ldots,n-1\}$ having the columns of M as incidence vectors, then $\mathcal{S}$ has the property that for any v members, there exists another member that is not contained in the union of these v members. This property is then used to show that the decoding algorithm, given the tag and the modified data, can uniquely recostruct the original data item. In [12] it was proved that v-disjunct matrices satisfy $s = \Omega(\min(v^2 \log_v n, n))$, and in [23] the authors give one of the most recent efficiently computable constructions, satisfying $s = O(\min(v^2 \log n, n))$. There is an extensive literature on v-disjunct matrices and their applications (i.e., non-adaptive combinatorial group testing algorithms [10]); we refer to reader to [23,10,11] and references therein.

In what follows, we denote as (sEnc, sDec) a v-superimposed code for n-bit data items, where sEnc is the encoding algorithm and sDec is the decoding algorithm.

Construction, Scenario and Properties. We can define a forensic encoding scheme $\mathrm{fS}_2 = (\mathrm{fEnc}_2, \mathrm{fDec}_2)$ directly from (sEnc, sDec), as follows. The fEnc_2 algorithm, on input a cryptographic key k and a data item m, returns a pair (m', tag), computed as follows: $m' = m$ and $tag =$ sEnc(m). The fDec_2 algorithm, on input a cryptographic key k, an error parameter v, and strings m'', tag, returns an output out containing the output of the correction algorithm sDec on input m'' and tag. (Note that neither fEnc_2 nor fDec_2 use k.)

Such a scheme could be used in the following client-server scenario. Given data item m, the client uses the forensic encoding algorithm fEnc_2 to generate tag, keeps tag private and sends $m' = m$ to the server, who stores it. Later, assume an adversary intrudes into the server's storage and is able to introduce up to v errors in m', changing it into m''. At any later time, when retrieving m'' from the server, the client can run algorithm fDec_2 on input m'' and tag to correct up to v errors in m'' and thus recover m. As a direct consequence of this discussion and of the properties of superimposed codes, we obtain the following

Theorem 2. *Assume (sEnc, sDec) is a v-superimposed code for n-bit data items. The forensic encoding scheme* $\mathrm{fS}_2 = (\mathrm{fEnc}_2, \mathrm{fDec}_2)$ *satisfies (t, v', ϵ)-error correction and (t, ϵ)-encoding indistinguishability against data intruders, for $v' = v$, $\epsilon = 0$ and any t polynomial in n.*

3.3 Forensic Encoding from Collision-Resistant Hashing

Background. Collision-resistant hash functions are at the center of an active research area in the cryptography literature. We now briefly recall their formal definition and known facts about them.

Let $\mathcal{H} = \{H_\lambda\}_{\lambda \in \mathcal{N}}$, where H_λ is a set of functions $h_\lambda : \{0,1\}^\lambda \times \{0,1\}^{p(\lambda)} \rightarrow \{0,1\}^\sigma$, λ is a security parameter, p is a polynomial and σ is constant with respect to λ. We say that $\mathcal{H}$ is a *family of keyed hash functions*, if it is a family of polynomial-time (in λ) samplable and computable functions, and we denote as $t_n(H)$ an upper bound on the running time of hash functions in H_λ on inputs of length n. As keyed hash functions compress an arbitrarily large input to a fixed-size output, each output has a very large set of preimages; yet, the collision-intractability property states that any efficient algorithm (even when given oracle access to the hashing function) can find two preimages of the same output only with small probability. A formal definition follows.

Definition 4. *Let $\mathcal{H} = \{H_\lambda\}_{\lambda \in \mathcal{N}}$ be a family of keyed hash functions. For any $t, \epsilon > 0$, we say that $\mathcal{H}$ is (t, q, ϵ)-*collision-intractable *if for any oracle algorithm A running in time t and making q queries to its oracle, it holds that*

$$\text{Prob}[\, h \leftarrow H_\lambda; k \leftarrow \{0,1\}^\lambda; (x_1, x_2) \leftarrow A^{h(k,\cdot)}(1^\lambda) \, : \, Coll_k^h(x_1, x_2) = 1 \,] \, \leq \, \epsilon.$$

where $Coll_k^h(x, y) = 1$ if and only if $(x \neq y) \, \wedge \, (h(k, x) = h(k, y))$.

Constructions of collision-intractable hash functions in the literature are based on either the computational intractability of number-theoretic problems (see, e.g., [19,3]), or more general complexity-theoretic assumptions (see, e.g., [24]), or heuristic finite functions with very high efficiency but only conjectured collision intractability (see, e.g., [20])), which, although unproven to be collision-intractable, are much more efficient and currently used by real-life cryptographic systems and products. In the light of recent cryptanalysis results on previous heuristic proposals, researchers have proposed a number of new functions and a standard process was recently started [21]. In practice, keyed hash functions, although represented in constant space, are often assumed to have a pseudo-randomness property stating that they are indistinguishable from random functions, as we now formally recall.

Definition 5. *Let $\mathcal{H} = \{H_\lambda\}_{\lambda \in \mathcal{N}}$ be a family of keyed hash functions and let F be the set of functions with domain $\{0,1\}^{p(\lambda)}$ and codomain $\{0,1\}^\sigma$. For any $t, \epsilon > 0$, we say that $\mathcal{H}$ is (t, q, ϵ)-*pseudo-random *if for any oracle algorithm A running in time t and making q queries to its oracle, it holds that*

$$\text{Prob}[\, h \leftarrow H_\lambda; k \leftarrow \{0,1\}^\lambda : A^{h(k,\cdot)}(1^\lambda) = 1 \,] - \text{Prob}[\, r \leftarrow F : A^{r(\cdot)}(1^\lambda) = 1 \,] \leq \epsilon.$$

Construction, Scenario and Properties. We can define a forensic encoding scheme $\text{fS}_3 = (\text{fEnc}_3, \text{fDec}_3)$ directly from any hash function $h \in H_\lambda$, as follows. The fEnc_3 algorithm, on input key k and a data item m, returns a pair (m', tag), computed as follows: $m' = m$ and $tag = h(k, m)$. The fDec_3 algorithm, on input key k, an error parameter v, and strings m'', tag, returns a bit denoting whether $tag = h(k, m'')$.

Such a scheme could be used in the following client-server scenario. Assume an adversary wants to efficiently generate m_0, m_1 such that, acting as a client, he can store $m = m_0$ with the server and later claim that he actually stored $m' = m_1$. The server expects to receive a pair (m, tag) and, to prevent the adversary's attack, will store this pair only if equality $h(k, m) = tag$ holds. In that case, even a non-corrupted client that uses the forensic encoding algorithm fEnc_3 to generate (m, tag), can send (m, tag) to the server and be sure that the pair is stored (as the equality $h(k, m) = tag$ holds).

As a direct consequence of this discussion and of the properties of collision-resistant hash functions, we obtain the following

Theorem 3. *Assume $\mathcal{H}$ is a (t, q, ϵ)-collision-intractable family of keyed hash functions, for any polynomial q. The forensic encoding scheme $\mathrm{fS}_3 = (\mathrm{fEnc}_3, \mathrm{fDec}_3)$ satisfies (t', ϵ')-error detection against data senders, for $t' = t$, $\epsilon' = \epsilon$ and (t'', ϵ'')-encoding indistinguishability against data intruders, for $\epsilon'' = 0$ and any t'' polynomial in n.*

4 Error Correction against Sender Corruption

In this section we show a forensic encoding scheme based on a suitable combination of collision-intractable hashing and superimposed codes, that satisfies error correction against sender corruptions.

An informal description. The encoding algorithm fEnc computes one hash tag for each row in the disjunct matrix $M_{s,n}$ associated with the superimposed code, as follows. The value of the j-th bit in the i-th matrix row is used to select (or not) the bit in the j-th position of the data item, for $j = 1, \ldots, n$ and $i = 1, \ldots, s$. Then, all selected data bits are concatenated into a string L_i and the i-th hash tag tag_i is obtained by running the collision-intractable hash function h on input L_i, for $i = 1, \ldots, s$. The algorithm outputs the pair (m, tag), where $tag = (tag_1, \ldots, tag_s)$.

The decoding algorithm fDec_4 recomputes hash tags $tag''_1, \ldots, tag''_s$ similarly as done in fEnc_4, but this time using the received data item m'' as input. Then it compares each received tag tag_i with the recomputed tag tag''_i, for each $i = 1, \ldots, s$, so to reconstruct a characteristic s-bit vector e of differences among hash tags (i.e., for $i = 1, \ldots, s$, $e[i] = 1$ if $tag_i \neq tag''_i$ and 0 otherwise). Finally, it finds all columns in the disjunct matrix M that are covered by e, and returns these columns' indices as the positions where m'' differs from m.

A formal description. The description of algorithms fEnc_4 and fDec_4 follows (here, the symbol | denotes string concatenation).

The algorithm fEnc$_4$: On input k, m, and a description of the q-disjunct matrix M and of the family $\mathcal{H} = \{H_\lambda\}_{\lambda \in \mathcal{N}}$ of a collision-intractable hash function, do as follows:

```
1. Randomly choose $h$ from $H_\lambda$;
2. for $i = 1, \ldots, s$,
     let $L_i$ = empty string
     for $j = 1, \ldots, n$,
       if $M(i, j) = 1$ then set $L_i = L_i \,|\, x[j]$
     set $tag_i = h_\lambda(k, L_i)$
3. Return: $m' = m$ and $tag = (h; tag_1, \ldots, tag_s)$.
```

The algorithm fDec$_4$*:* On input v, k, m'', tag, do the following:

```
1. Write $tag$ as $(h; tag_1, \ldots, tag_s)$
2. for $i = 1, \ldots, s$,
       let $L''_i$ = empty string and $u_i = 0$
       for $j = 1, \ldots, n$,
           if $M(i,j) = 1$ then set $L''_i = L''_i \,|\, m''[j]$
       set $tag''_i = h_\lambda(k, L''_i)$
       if $tag''_i \neq tag_i$ then set $u_i = 1$
3. set $u = (u_1, \ldots, u_s)$
4. let $e$ be the $\leq q$-weight $n$-bit vector such that
       $e[i] = 1$ if the $i$-th column in $M$ is covered by $u$, for $i = 1, \ldots, n$
5. Return: $out = e$.
```

The above construction satisfies the following

Theorem 4. *Assume $\mathcal{H}$ is a family of (t, q, ϵ)-collision intractable keyed hash function, for any polynomial q and (sEnc,sDec) is a v-superimposed code. The forensic encoding scheme* $\mathrm{fS}_4 = (\mathrm{fEnc}_4, \mathrm{fDec}_4)$ *satisfies (t', v', ϵ')-error correction against data sender corruptions, for $t' = t + O(s \cdot t_n(H))$, $v' = v$, and $\epsilon' = \epsilon$, and (t'', ϵ'')-encoding indistinguishability against data intruders, for any t'' polynomial in n and $\epsilon' = 0$.*

The encoding indistinguishability property of fS_4 claimed in Theorem 4 follows directly from the fact that algorithm fEnc returns $m' = m$.

The main observation to prove the error correction property of fS_4 claimed in Theorem 4 is that, under the contradiction assumption that the adversary is successful, we obtain one of the following two consequences: (1) collisions are found in h_λ, or (2) the vector u can be shown to be equal to the matrix product of the disjunct matrix M times the error vector e. In the first case, we obtain a contradiction of the main property of the hash function h_λ. In the second case, one can use the definition of disjunct matrix and obtain that the error correction property of this matrix implies an analogue error correction property for the forensic encoding scheme. Now we proceed with a more formal sketch of proof for this property.

Let $\mathsf{Coll}(A)$ be the event defined as follows: "For some $i \in \{1, \ldots, s\}$, it holds that $L_i \neq L''_i$ and $h_\lambda(L_i) = h_\lambda(L''_i)$, where L_i (resp., L''_i) is a string computed while running fEnc_4 on input m (resp., fDec_4 on input m''), and m, m'' are the strings returned by oracle algorithm A." Given an algorithm A that makes event $\mathsf{Coll}(A)$ true, we can obtain an algorithm A' that, with the same probability, violates the collision-intractability property of $\mathcal{H}$. This gives us the following

Lemma 1. *If the family $\mathcal{H}$ of hash functions satisfies (t, q, ϵ)-collision-intractability, for any arbitrary polynomial q, then for any algorithm A running in time t', it holds that* $\mathrm{Prob}[\,\mathsf{Coll}(A)\,] \leq \epsilon$,, *where $t' = t + O(s \cdot t_n(H))$.*

If event $\mathsf{Coll}(A)$ does not happen, for all $i \in \{1, \ldots, s\}$, the condition $L_i \neq L''_i$ implies $h_\lambda(L_i) \neq h_\lambda(L''_i)$, where L_i (resp., L''_i) is a string computed while running fEnc on input the string m (resp., fDec on input the string m'') returned by A. By construction of fDec_4, each component $u[i]$ of vector u is equal to 1 if and only if $tag_i \neq tag''_i$, which always happens when $h_\lambda(L_i) \neq h_\lambda(L''_i)$. This, in turn, always happens when $L_i \neq L''_i$, as we assumed that $\mathsf{Coll}(A)$ is false, which implies that for at least one value j such that $M(i,j) = 1$, the j-th bit of the error vector $e = m \oplus m''$ is equal to 1. Thus, the vector u can be written as $M \cdot e$, this matrix product being over the semiring $(\{0,1\}, \vee, \wedge)$. By the properties of v-superimposed codes, we obtain that the nonzero components in e have the same indices as the columns in M that are covered by u, and thus all v errors can be corrected, proving the following

Lemma 2. *If M is a v-disjoint matrix then, for any oracle algorithm A, it holds that* $\text{Prob}[\,\mathsf{Exp6}^{\text{fEnc,fDec},A}(v) = 1|\overline{\mathsf{Coll}(A)}\,]) = 0$.

Using the results obtained via these lemmas, one can then write

$$\begin{aligned}\epsilon' &= \text{Prob}[\,\mathsf{Exp6}^{\text{fEnc,fDec},A}(v) = 1\,]\\ &\leq \text{Prob}[\,\mathsf{Coll}(A)\,] + \text{Prob}[\,\mathsf{Exp6}^{\text{fEnc,fDec},A}(v) = 1|\overline{\mathsf{Coll}(A)}\,],\\ &\leq \epsilon + (\text{Prob}[\,\mathsf{Exp6}^{\text{fEnc,fDec},A}(v) = 1|\overline{\mathsf{Coll}(A)}\,] \leq \epsilon + 0 = \epsilon\end{aligned}$$

5 Conditional Error Detection and Encoding Indistinguishability

In this section we show a forensic encoding scheme using pseudo-random and collision-intractable hash functions, universal hash functions, and data representation algorithms in the large min entropy data model. The properties achieved by the scheme include zero message expansion, encoding indistinguishability and conditional error detection against sender corruptions.

An informal description. Let $p = p_1 + p_2$ and let $m_{1,0}, \ldots, m_{p,0}$ denote the p data blocks of a $(2,p)$-dense data item m. Also, let ρ be the predicate that, on input (m, w), returns 1 if data item w can be written as $w = m_{1,b(1)}|\cdots|m_{p,b(p)}$ for some bits $b(1), \ldots, b(p)$, and 0 otherwise.

The encoding algorithm fEnc$_5$ divides a $(2,p)$-dense message m into p data blocks $m_{1,0}, \ldots, m_{p,0}$, and uses the representation change algorithm rC to compute their alternative representations $m_{1,1}, \ldots, m_{p,1}$. Then, it randomly chooses a p_1-bit nonce u and uses the collision-intractable and pseudo-random hash function h^{cr} to expand the random key k into a much longer pseudo-random string R. For $i = 1, \ldots, p$, this latter string is used to randomly choose a 2-to-1 universal hashing function h^{un}_i which maps two (n/p)-bit representations $m_{i,0}, m_{i,1}$ of the same block to the same $(n/p - 1)$-bit value, denoted as Z_i. Now the collision-intractable hash function h^{cr} is used to compute a p_2-bit string $htag$ for the compressed message $z = Z_1|\cdots|Z_p$. Then the p bits $t(1), \ldots, t(p)$ in strings u and $htag$ are embedded into the original message m by using them as selectors of the representation associated with each data block, and computing an alternate representation $m' = m_{1,t(1)}|\cdots|m_{p,t(p)}$ of m. Finally the algorithm fEnc$_5$ returns the strings m' and $tag = \emptyset$ as output.

On input a string m'', the decoding algorithm fDec$_5$ uses the representation index algorithm rI to find $t''(1), \ldots, t''(p)$ such that $m'' = m''_{1,t''(1)}|\cdots|m''_{p,t''(p)}$ and uses the representation change algorithm rC to compute the alternative representations $m''_{p_1+1,1-t''(p_1+1)}, \ldots, m''_{p,1-t''(p)}$. Then it sets $u = t''(1)|\cdots|t''(p_1)$ and $htag'' =$

$t''(p_1+1)|\cdots|t''(p)$ and recomputes R from u and k exactly as fEnc$_5$ does. For $i=1,\ldots,p$, using R as random string, it randomly chooses 2-to-1 universal hashing function h_i^{un} which maps the two (n/p)-bit representations $m''_{i,0}, m''_{i,1}$ of the same block to the same $(n/p-1)$-bit value, denoted as Z''_i, Then, it sets $z'' = Z''_1|\cdots|Z''_p$ and checks whether $h^{cr}(z'') = htag''$. If yes, it returns: 1; otherwise, it returns: 0.

A formal description. By $\mathcal{H}^{cr} = \{H_\lambda^{cr}\}_{\lambda\in\mathcal{N}}$ we denote a family of collision-intractable and pseudo-random hash functions from $\{0,1\}^*$ to $\{0,1\}^\sigma$, for some constant σ (e.g., $\sigma = 160$). By $\mathcal{H}^{un} = \{H_\lambda^{un}\}_{\lambda\in\mathcal{N}}$ we denote a family of 2-to-1 universal hash functions from $\{0,1\}^*n/p$ to $\{0,1\}^{n/p-1}$. The formal description of fEnc$_5$ and fDec$_5$ follows.

The algorithm fEnc$_5$*:* On input key k and n-bit data item m, do as follows:

```
1. Randomly choose $h^{cr}$ from $H_\lambda^{cr}$ and $u \in \{0,1\}^{p_1}$;
2. write $m$ as $m = m_{1,0}|\cdots|m_{p,0}$, where the $m_{i,0}$'s are equal-length data blocks
3. set $R = h^{cr}(k,u)$ and use $R$ as random string in the following steps;
4. for $i = 1,\ldots,p$,
       let $m_{i,1} = \mathrm{rC}(0,i,m_{i,0})$;
       randomly choose $h_i^{un}$ from $H_\lambda^{un}$ such that $Z_i = h_i^{un}(m_{i,0}) = h_i^{un}(m_{i,1})$
5. set $z = Z_1|\cdots|Z_p$ and $htag = h^{cr}(z)$, where $|htag| = p_2$;
6. write $u|htag$ as $t_1|\cdots|t_p$, where $t_i \in \{0,1\}$, for $i = 1,\ldots,p$;
7. set $m' = m_{1,t(1)}|\cdots|m_{p,t(p)}$ and $tag = \emptyset$;
8. return: $(m', tag)$.
```

The algorithm fDec$_5$*:* On input k, m'', tag, do the following:

```
1. Write $m''$ as the concatenation of $p$ data blocks $m''_{1,\cdot}, \ldots, m''_{p,\cdot}$.
2. for $i = 1,\ldots,p$,
       let $t''_i = \mathrm{rI}(i, m''_{i,\cdot})$
       let $m''_{i,1-t''(i)} = \mathrm{rC}(t''(i), i, m''_{i,t''(i)})$
3. set $u'' = t''(1)|\cdots|t''(p_1)$ and $htag'' = t''(p_1+1)|\cdots|t''(p)$
4. set $R'' = h^{cr}(k,u'')$ and use $R''$ as random string in the following steps
5. for $i = 1,\ldots,p$,
       randomly choose $h_i^{un}$ from $H_\lambda^{un}$ such that $Z''_i = h_i^{un}(m''_{i,0}) = h_i^{un}(m''_{i,1})$
6. set $z'' = Z''_1|\cdots|Z''_p$
7. if $htag'' = h^{cr}(z'')$ then return: 1 else return: 0.
```

The above construction satisfies the following

Theorem 5. *Assume $\mathcal{H}^{cr}$ is a family of (t_i, q, ϵ_i)-collision intractable and (t_r, q, ϵ_r)-pseudo-random hash functions, for any polynomial q, and assume $\mathcal{H}^{un}$ is a family of 2-to-1 universal hash functions. Then the forensic encoding scheme* fS$_5$ = (fEnc$_5$, fDec$_5$) *satisfies zero data expansion; (t', ϵ')-indistinguishability against data intruders, for any polynomial t', and $\epsilon' = \epsilon$; and (t', ϵ')-conditional error detection against data sender corruptions, for any polynomial t', and $\epsilon' = \epsilon_r + 2^{-n/(p-p_1)}$.*

The zero data expansion property of scheme fS$_5$ is verified by inspection.

The encoding indistinguishability property against data intruders of scheme fS$_5$ directly follows by inspection of the string m' output by algorithm fEnc$_5$ and by using the representation equivalence property of the dense data items used.

The overall structure of the proof of the conditional error detection property is that, under the contradiction assumption that the adversary is successful in breaking this property, we obtain one of the following three consequences: (1) h^{cr} is a indistinguishable from a random function; (2) collisions are found in h^{un}, or (3) collisions are found in $h^{cr}(k,\cdot)$. In the first case we obtain a contradiction of the pseudo-randomness property of h^{cr}. Assuming the statement in the first case is not true, the properties of h^{un} imply that the second case only happens with very low probability. In the third case we obtain a contradiction of the collision-intractability property of $h^{cr}(k,\cdot)$.

A sketch of the formal proof for this property goes as follows. Note that experiment $\mathsf{Exp}5$ returns 1 for fS_5 whenever an efficient adversary A returns an m'' such that $\rho(m',m'')=1$ and fDec_5 returns 1 on input k,m'',tag, where m' is the output returned by fEnc_5. We start by considering the case $htag = htag''$. In this case, if experiment $\mathsf{Exp}5$ returns 1, the condition $\rho(m',m'')=1$ implies that $z \neq z''$, and the condition $htag = htag''$ implies that $h^{cr}(k,z) = h^{cr}(k,z'')$. This implies that A can produce a collision (z,z'') for $h^{cr}(k,\cdot)$, thus violating the collision-intractability property of $\mathcal{H}^{cr}$. We then consider the case $htag \neq htag''$. In this case, if experiment $\mathsf{Exp}5$ returns 1, it holds that $htag'' = h^{cr}(k,z'')$. Then the condition $\rho(m',m'')=1$ implies that $z \neq z''$, and thus either the condition $htag'' = h^{cr}(k,z'')$ holds with (very low) probability $2^{-n/(p-p_1)}$ (by the definition of universal hash functions) or the hash function $h^{cr}(k,z'')$ can be distinguished by A from a random function with probability higher than ϵ_r, thus violating the pseudo-randomness property of $\mathcal{H}^{cr}$.

Multimedia files and the large-min-entropy data model. We briefly discuss why we believe that multimedia files are well modeled using the previously defined large-min-entropy data model. We assume that the data is sparse in some basis, such that the data is sufficiently redundant in the observation space. Natural images, for instance, have effective sparse representations in the discrete Fourier domain or the wavelet domain [15]. Consequently, consecutive pixels in the spatial domain carry significant correlation that can be exploited in constructing tag embedding procedures that are imperceptible to the naked eye. Indeed, many methods of tag embedding exist in the literature, the overarching goal being achieving zero message expansion while preserving visual appearance of the original data or even making the tag imperceptible to the eye [2]. Least significant bit (LSB) data embedding is a simple example where the tag is embedded by modifying a sequence of LSB of a particular pixel in consecutive non-overlapping windows of an image. While visually imperceptible, LSB embedding can be statistically detected by computer analysis of the data. Randomization in the selection of pixels in the windows, and other more elaborated methods, have been proposed to make the tag embedding not only visually imperceptible but detectable with no more than very small probability. The embedding can also be performed by perturbed quantization. In this case, the tag is embedded in the final quantization step associated with lossy compression procedures that have negligible visual effect on the data representation. The detection of tags in this latter approach is harder. A thorough exposition of these techniques and analysis of their properties and characteristics is found in [2].

6 Conclusions

We formulated unifying algorithm, data and security models to formally prove security guarantees provided by forensic encoding constructions from known techniques and suitable combinations of them. We rigorously clarified the different security guarantees provided by using these techniques (alone or in some standard or novel combinations) for both data at rest and data in transit. Our most novel construction provides a forensic encoding scheme with error detection at zero data expansion in the large-min-entropy data model. This scheme can be extended to provide error correction using a suitable combination with superimposed codes. For both schemes, more rigorous statements relating the features of the data model to the level of error detection and correction would be of interest, and are thus left as a future work item.

References

1. Bellare, M., Goldreich, O., Mityagin, A.: The power of verification queries in message authentication and authenticated encryption. Cryptology ePrint Archive: Report 2004/309
2. Cox, I., Miller, M., Bloom, J., Fridrich, J., Kalker, T.: Digital Watermarking and Steganography, 2nd edn. Morgan Kaufmann Publishers (2008)
3. Damgård, I.B.: Collision Free Hash Functions and Public Key Signature Schemes. In: Price, W.L., Chaum, D. (eds.) EUROCRYPT 1987. LNCS, vol. 304, pp. 203–216. Springer, Heidelberg (1988)
4. De Bonis, A., Di Crescenzo, G.: Combinatorial Group Testing for Corruption Localizing Hashing. In: Fu, B., Du, D.-Z. (eds.) COCOON 2011. LNCS, vol. 6842, pp. 579–591. Springer, Heidelberg (2011)
5. Di Crescenzo, G., Ge, R., Arce, G.: Design and Analysis of DBMAC: an Error-Localizing Message Authentication Code. In: Proceedings of IEEE GLOBECOM 2004 (2004)
6. Di Crescenzo, G., Jiang, S., Safavi-Naini, R.: Corruption-Localizing Hashing. In: Backes, M., Ning, P. (eds.) ESORICS 2009. LNCS, vol. 5789, pp. 489–504. Springer, Heidelberg (2009)
7. Di Crescenzo, G., Ostrovsky, R., Rajagopalan, S.: Conditional Oblivious Transfer and Timed-Release Encryption. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 74–89. Springer, Heidelberg (1999)
8. Di Crescenzo, G., Vakil, F.: Cryptographic hashing for virus localization. In: Proceedings of the 2006 ACM CCS Workshop on Rapid Malcode, WORM 2006, pp. 41–48 (2006)
9. Dorfman, R.: The detection of defective members of large populations. Ann. Math. Statist. 14, 436–440 (1943)
10. Du, D.Z., Hwang, F.K.: Combinatorial Group Testing and its Applications. World Scientific (2000)
11. Dyachkov, A.G., Rykov, V.V.: A survey of superimposed code theory. Problems Control & Inform. Theory 12(4), 1–13 (1983)
12. Dyachkov, A.G., Rykov, V.V.: Bounds on the length of disjunctive codes. Problemy Peredachi Informatsii (Problems of Information Transmission) 18(3), 7–13
13. Erdös, P., Frankl, P., Füredi, Z.: Families of finite sets in which no set is covered by the union of r others. Israel J. of Math. 51, 75–89 (1985)
14. Fang, J., Jiang, Z., Yiu, S., Hui, C.: Hard Disk Integrity Check by Hashing with Combinatorial Group Testing. In: Proc. of CSA 2009 (2009)
15. Gonzalez, R., Woods, R.: Digital Image Processing, 3rd edn. Prentice Hall, New Jersey (2008)

16. Goodrich, M., Atallah, M., Tamassia, R.: Indexing Information for Data Forensics. In: Ioannidis, J., Keromytis, A.D., Yung, M. (eds.) ACNS 2005. LNCS, vol. 3531, pp. 206–221. Springer, Heidelberg (2005)
17. Hopper, N.J., Langford, J., von Ahn, L.: Provably Secure Steganography. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 77–92. Springer, Heidelberg (2002)
18. MacWilliams, F.J., Sloane, N.J.A.: The Theory of Error-Correcting Codes. North-Holland, New York (1977)
19. Merkle, R.C.: A Certified Digital Signature. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 218–238. Springer, Heidelberg (1990)
20. NIST. Secure Hash Signature Standard (SHS) (FIPS PUB 180-2). United States of America, Federal Information Processing Standard (FIPS) 180-2, August 1 (2002)
21. NIST, Cryptographic Hash Algorithm Competition, `http://csrc.nist.gov/groups/ST/hash/sha-3/index.html`
22. Kautz, W.H., Singleton, R.R.: Nonrandom binary superimposed codes. IEEE Trans. on Inform. Theory 10, 363–377 (1964)
23. Porat, E., Rothschild, A.: Explicit Non-adaptive Combinatorial Group Testing Schemes. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) ICALP 2008, Part I. LNCS, vol. 5125, pp. 748–759. Springer, Heidelberg (2008)
24. Russell, A.: Necessary and Sufficient Conditions for Collision-Free Hashing. Journal of Cryptology 8(2) (1995)

Author Index