Hyoung-Joong Kim
Yun-Qing Shi
Mauro Barni (Eds.)

LNCS 6526

Digital Watermarking

9th International Workshop, IWDW 2010
Seoul, Korea, October 2010
Revised Selected Papers

Springer

Lecture Notes in Computer Science 6526

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

Hyoung-Joong Kim Yun-Qing Shi
Mauro Barni (Eds.)

Digital Watermarking

9th International Workshop, IWDW 2010
Seoul, Korea, October 1-3, 2010
Revised Selected Papers

Springer

Volume Editors

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

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

Mauro Barni
University of Siena
Department of Information Engineering
53100 Siena, Italy
E-mail: barni@dii.unisi.it

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-18404-8 e-ISBN 978-3-642-18405-5
DOI 10.1007/978-3-642-18405-5
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2010942729

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

On behalf of the Technical Program Committee, we thank all authors, Program Committee members, reviewers, sponsors, invited speaker, and assistants. This 9^{th} workshop was possible due to the help and devotion of those who love IWDW (International Workshop on Digital Watermarking). Their invaluable time, concerns and efforts will not be forgotten. Invited speaker, Alex Kot deliver an invaluable special talk on forensic analysis: "Image Forensics Tools Using Statistical Features."

Since its first event, IWDW has been a forum to check the current state of the art, exchange new ideas, and discuss open problems. This year, 42 papers were submitted from Austria, China, India, Iran, Korea, Japan, Russia, Singapore, UK, USA. Of these 26 papers were accepted. The Selected papers are well balanced and diverse. Topics include robust watermarking, reversible watermarking, fingerprinting, steganography and steganalysis, visual encryption, digital forensics, and so on. Some data hiding and multimedia security technologies are solid and quite saturated, but other technologies are just beginning to sprout or sprouting. Target media include audio, image, video, movie, and 3D content.

We hope that the papers in this volume be a catalyst for further research in this field.

October 2010

M. Barni
H.J. Kim
Y.-Q. Shi

Organization and Committees

General Chair

Lim, Jongin	Korea University, Korea

Technical Program Chairs

Kim, Hyoung Joong	Korea University, Korea
Shi, Yun Q	New Jersey Institute of Technology, USA
Barni, Mauro	University of Siena, Italy

Publicity Committee and Secretariat

Hwang, Hee Joon	Korea University, Korea
Kim, Jung Min	Korea University, Korea
Sachnev, Vasiliy	Korea University, Korea
Murasova, Yulia	Korea University, Korea
Dong, Keming	Korea University, Korea

Local Arrangements Committee

Tsomko, Elena	Korea University, Korea

International Publicity Liasons

Ho, Tony	University of Surrey, UK
Westfeld, Andreas	Dresden University of Technology, Germany

Technical Program Committee

Abhayaratne, Charith	University of Sheffield, UK
Barbier, Johann	CELAR, D'epartement de Cryptologie, La Roche Marguerite, France
Bloom, Jeffey	Thomson, USA
Cayre, Francois	TEMICS / IRISA, France
Chen, Wen	Dialogic, USA
Cheng, Lee-Ming	City University of Hong Kong, Hong Kong
Dittman, Jana	Magdeburg, Otto von Guericke University, Germany
Doërr, Gwenaël	University College London, UK

Dugelay, Jean-Luc	Institut EURECOM, France
Goljan, Miroslav	Binghamton University, USA
Huang, Jiwu	Sun Yat-sen University, China
Jeon, Byeungwoo	Sungkyunkwan University, Korea
Kalker, Ton	Hewlett Packard Laboratories, USA
Kankanhalli, Mohan	National University of Singapore, Singapore
Ker, Andrew	Oxford University, UK
Kot, Alex	Nanyang Technological University, Singapore
Kuo, C.-C. Jay	University of Southern California, USA
Kutter, Martin	Swiss Federal Institute of Technology, Switzerland
Lagendijk, Inald	Delft University of Technology, The Netherlands
Lee, Heung Kyu	KAIST, Korea
Li, Chang-Tsun	University of Warwick, UK
Lu, Zheming	University of Freiburg, Germany
Martin, Keith	Royal Holloway, University of London, UK
Memon, Nasir	Polytechnic University, USA
Ni, Jiang Qun	Sun Yat-sen University, China
Ni, Zhicheng	WorldGate Communications, USA
Pan, Jeng-Shyang	National Kaohsiung University of Applied Sciences, Taiwan
Pérez–González, Fernando	University of Vigo, Spain
Pitas, Ioannis	University of Thessaloniki, Greece
Piva, Alessandro	University of Florence, Italy
Ro, Yong-Man	Information and Communications University, Korea
Sadeghi, Ahmad-Reza	Ruhr-University Bochum, Germany
Sakurai, Kouichi	Kyushu University, Japan
Schaathun, Hans Georg	University of Surrey, UK
Voloshynovskiy, Sviatoslav	University of Geneva, Switzerland
Waller, Adrian	Thales Research and Technology, UK
Wang, Shuozhong	Shanghai University, China
Xiang, Shijun	Jinan University, China
Zhang, Hongbin	Beijing University of Technology, China
Zou, Dekun	Thomson, USA

Table of Contents

Passive Detection of Paint-Doctored JPEG Images

Yu Qian Zhao[1], Frank Y. Shih[2], and Yun Q. Shi[3]

[1] School of Info-Physics and Geomatics Engineering, Central South University, Changsha, Hunan 410083, China
[2] Computing Sciences, New Jersey Institute of Technology, Newark, NJ 07102, USA
[3] Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102, USA
zhaocsu@163.com, shih@njit.edu, shi@adm.njit.edu

Abstract. Image painting is an image doctoring method to remove particular objects. In this paper, a novel passive detection method for paint-doctored JPEG images is proposed when the doctored image is saved in an uncompressed format or in the JPEG compressed format. We detect the doctored region by computing the average of sum of absolute difference images between the doctored image and a resaved JPEG compressed image at different quality factors. There are several advantages of the proposed method: first, it can detect the doctored region accurately even if the doctored region is small in size; second, it can detect multiple doctored regions in the same image; third, it can detect the doctored region automatically and does not need any manual operation; finally, the computation is simple. Experimental results show that the proposed method can detect the paint-doctored regions efficiently and accurately.

Keywords: Image forensic; Sum of absolute difference; Painted-doctored image; JPEG compression.

1 Introduction

Nowadays, with the advent of low-cost and high-resolution digital cameras, high-performance computers, and the availability of many powerful image processing software, digital images can be doctored easily. As a result, images no longer hold the unique stature as an exact recording of events. Undoubtedly, this brings some difficulty for people to prove the authenticity of digital images. Therefore, image forensics technologies are becoming increasingly important.

There are two kinds of image forensic authenticity techniques, which are known as active one [1-2] and passive one [3-6]. The former is mainly about digital watermarking or signature. The latter is also called blind digital image forensics technique, which has been widely studied in recent years, including five categories [7]: 1) pixel-based techniques, 2) format-based techniques, 3) camera-based techniques, 4) physically based techniques, and 5) geometric-based techniques.

JPEG is a commonly-used compression standard for photographic images. It is based on block splitting and discrete cosine transform (DCT), hence leaving blocking artifact. Li et al. [8] proposed a blocking artifact grid (BAG) extraction method to

H.-J. Kim, Y. Shi, and M. Barni (Eds.): IWDW 2010, LNCS 6526, pp. 1–11, 2011.

detect doctored JPEG images, including paint-doctored JPEG images. Luo et al. [9] presented a blocking artifact characteristics matrix (BACM), which exhibits regular symmetrical shape for original JPEG images, and applied it to expose digital forgeries by detecting the symmetry change of BACM. The method is only efficient for cropped and recompressed images, and a SVM classifier must be trained. Lin et al. [4] developed a method for detecting doctored JPEG images by examining the double quantization effect hidden among the DCT coefficients, and computing the block posterior probability map (BPPM) by Bayesian approach, but this method is not effective on non-JPEG image. To measure inconsistencies of blocking artifact, Ye et al. [10] presented a fast quantization table estimation algorithm based on histogram power spectrum of DCT coefficients. But the suspicious area must be first selected for evaluation, which is actually difficult. Chen et al. [11] proposed a machine learning based scheme to distinguish between double and single JPEG compressed images. The method can't detection local tampered image region. Farid [12] proposed the JPEG ghost to detect whether the part of an image was initially compressed at a lower quality than the remaining of the image.

Image painting is a usually used method to conceal particular objects [13-14]. If a JPEG image is doctored by image painting, the JPEG block lattice of the doctored region of the original image will be eliminated. In fact, the painted region can be treated as an uncompressed region. In this paper, we propose a blind digital image forensics method for detecting paint-doctored JPEG image, which is saved in an uncompressed format or JPEG compressed format. The doctored region can be detected by computing and observing the sum of absolute difference images between the doctored image and the resaved JPEG images at different quality factors.

The rest of the paper is organized as follows. In Section 2, we propose the methods for detecting paint-doctored JPEG images. In Section 3, we present experimental work, and compare our method with Li et al.'s [8]. Finally, conclusion is made in Section 4.

2 Methodology

Two different image painting-doctored scenarios are considered in this paper. In the first case, the paint-doctored image is saved as a non-JPEG image such as a BMP image. In the second case, the paint-doctored image is saved as a JPEG image. In this section we first conduct the simulations for these two different scenarios. Afterwards, the proposed algorithm is presented.

If a JPEG image is doctored by painting and then saved in an uncompressed format, the doctored region does not undergo JPEG compression and the remaining region of the doctored image undergoes single JPEG compression. If the paint-doctored image is saved in JPEG format, the doctored region and the remaining region of the doctored image undergo single and double JPEG compression, respectively. To detect the paint-doctored region, we conduct simulation for JPEG compression on a set of data at first. Let these data, denoted by T, be taken column by column say from left to right from a 512×512 uncompressed tank image shown in Fig. 1(a). We doubly quantize T by a quantization step q_1 firstly to obtain T_{q_1}, and subsequently by quantization step q to obtain $T_{q_1 q}$, and then calculate the average of sum of absolute difference,

denoted by D_1, between T_{q_1q} and T_{q_1}. On the other hand, we quantize T by quantization step q to obtain T_q, and calculate the average of sum of absolute difference D_2 between T_q and T. Let $q_1 = 29$. The dashed and solid curves in Fig. 1(b) show the average of sum of absolute difference D_1 and D_2, respectively, as a function of q ranging from 1 to 50 in increments of 1. From Fig. 1(b), it can be clearly observed that the difference between curves D_1 and D_2 is obvious for most quantization steps q.

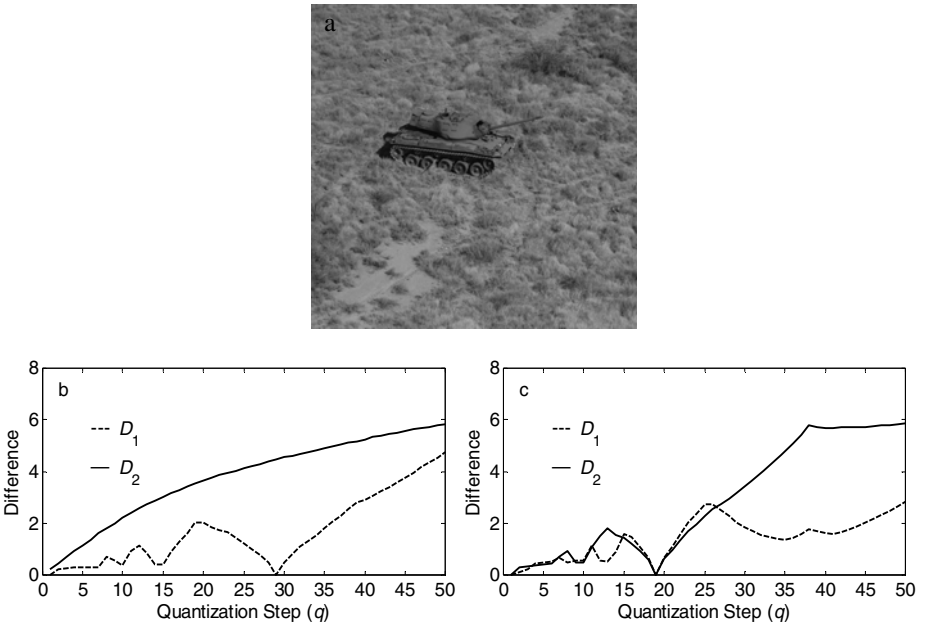


Fig. 1. Algorithm simulation. (a) An uncompressed 512×512 image from which the simulated data set T is taken column by column from left to right, (b) curve D_1 show the average of sum of absolute difference between data T quantized by q_1 = 29, and T first quantized by q_1 = 29, following quantized by q in the range of [1,50]; curve D_2 show the average of sum of absolute difference between data T, and T quantized by q in the range of [1,50], and (c) curve D_1 show the average of sum of absolute difference between data T first quantized by q_1 = 29 following quantized by q_2 = 19, and T first quantized by q_1 = 29 following quantized by q_2 = 19 then quantized by q in the range of [1,50]; curve D_2 show the average of sum of absolute difference between data T quantized by q_2 = 19, and T first quantized by q_2 = 19 following quantized by q in the range of [1,50].

Similarly, let data set T be firstly quantized by q_1, subsequently quantized by q_2 to obtain data set $T_{q_1q_2}$, lastly quantized by q to obtain data set $T_{q_1q_2q}$. We calculate average of the sum of absolute difference, denoted by D_1, between $T_{q_1q_2}$ and $T_{q_1q_2q}$. Differently, let data set T be firstly quantized by q_2 to obtain data set T_{q_2}, and subsequently quantized by q to obtain data set T_{q_2q}. We calculate the average of sum of

absolute difference D_2 between T_{q_2} and T_{q_2q}. Let q_1 = 29 and q_2 = 19. The dashed and solid curves in Fig. 1(c) show the average of sum of absolute difference D_1 and D_2, respectively, as a function of q ranging from 1 to 50 in increments of 1. From Fig. 1(c), it can also be clearly observed that the difference between curves D_1 and D_2 is obvious for some quantization steps q, such as from 30 to 50.

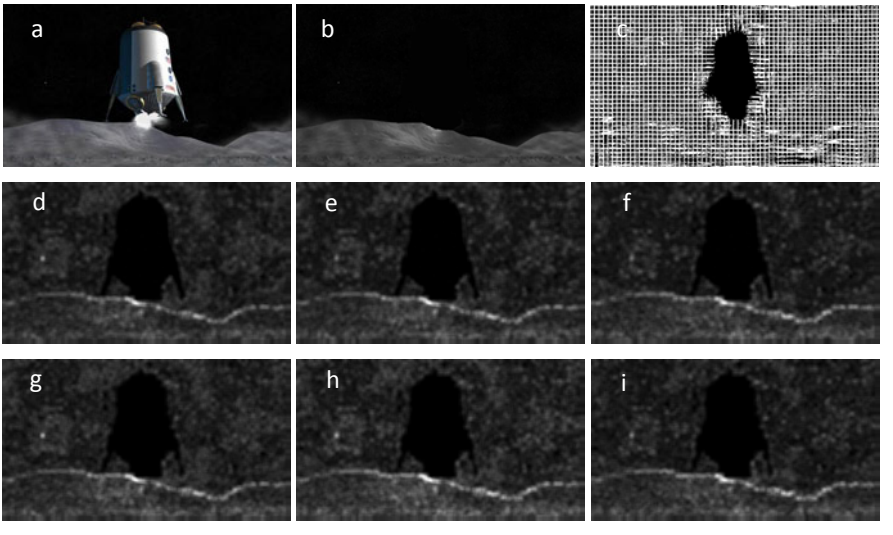


Fig. 2. Detection of the painted image which is doctored by filling black color. (a) Original image, (b) doctored image, (c) extracted BAG image from (b) saved in BMP format by Li et al.'s method [8], (d)-(f) the difference images generated from (b) saved in BMP format by our proposed method with the resaved quality factors 45, 50 and 55, respectively, and (g)-(i) the difference images generated from (b) saved in JPEG format by our proposed method with the resaved quality factors 45, 50 and 55, respectively.

According to the above analyses, we proposed a paint-doctored image detection method which consists of the following steps.

(1) JPEG compression

We resave the to-be detected image f at JPEG quality factor of Q, and obtain a JPEG compressed image f_Q.

(2) Computing absolute difference for every pixel value

Absolute difference between f and f_Q is computed for every pixel value as:

$$F_Q(x, y) = \left| f(x, y) - f_Q(x, y) \right| \tag{1}$$

(3) Averaging the sum of absolute difference across $b \times b$ image block for every pixel value

Since the absolute difference image $F_Q(x, y)$ is not so clear to differentiate the doctored region, especially for small detected region from the authentic region, we

calculate the average of sum of absolute difference across $b \times b$ image block for every pixel value by:

$$D_Q(x, y) = \frac{1}{b^2} \sum_{m=0}^{b-1} \sum_{n=0}^{b-1} F_Q(x+m, y+n) \tag{2}$$

where $b \times b$ denotes the size of block. The larger the b is, the more obvious the detected doctored region is, but the more obscure the detected edge of the doctored region is, and the more the computing time is. $D_Q(x, y)$ denotes the average of sum of the absolute difference between f and f_Q across the block for every pixel value.

(4) Repeating the above steps for another different resaved JPEG quality factor Q if it is needed

In general, if a region is obviously brighter or darker than the rest region of the sum of absolute difference image, the corresponding region of the to-be-detected image is considered as a paint-doctored region.

For color images, we can detect the paint-doctored region by applying eqs. (1) and (2) for three color channels of RGB, and then calculating the average of them for every pixel. Therefore, eqs. (1) and (2) will be replaced by eqs. (3) and (4), respectively.

$$F_Q(x, y) = \frac{1}{3} \sum_{i=1}^{3} \left| f(x, y, i) - f_Q(x, y, i) \right| \tag{3}$$

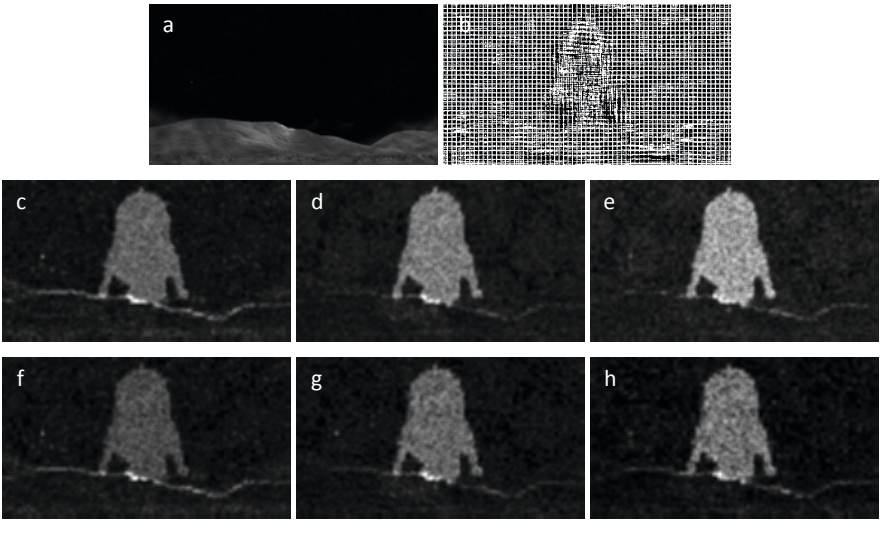


Fig. 3. Detection of a painted image which is doctored by filling random dark color. (a) Doctored image, (b) extracted BAG image from (a) saved in BMP format, (c)-(e) the difference images generated from (a) saved in BMP format by our proposed method with the resaved quality factors 70, 80 and 90, respectively, and (f)-(h) the difference images generated from (a) saved in JPEG format by our proposed method with the resaved quality factors 70, 80 and 90, respectively.

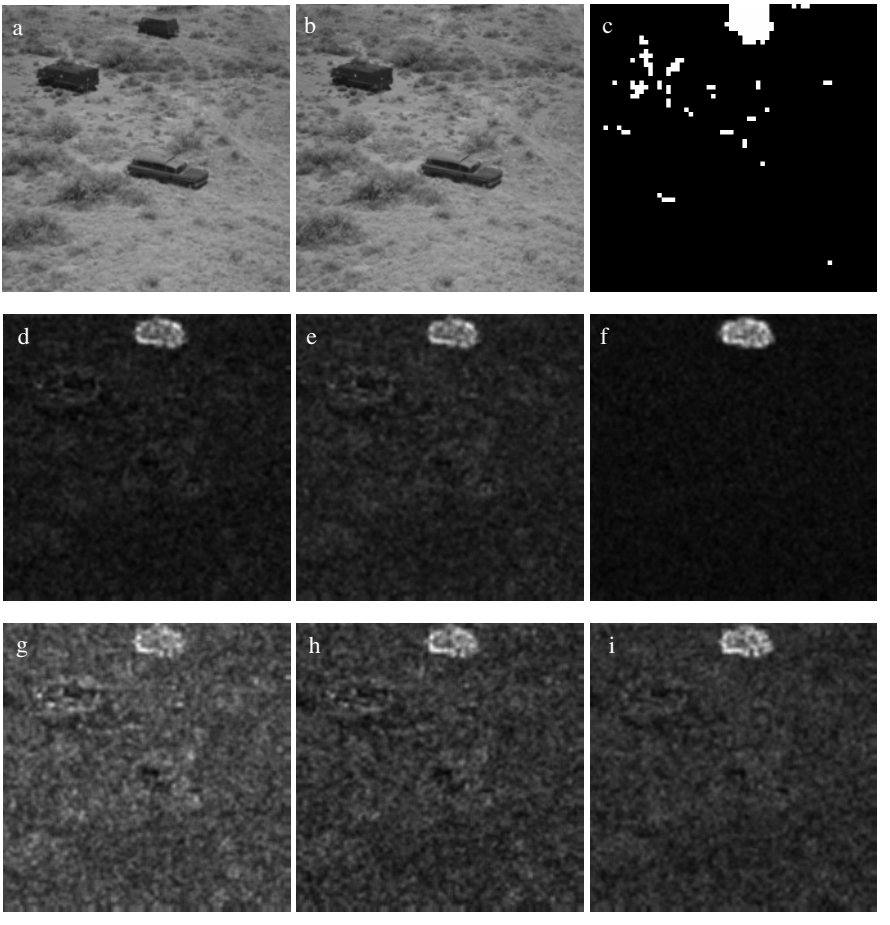


Fig. 4. Detection of painted image. (a) Original image, (b) doctored image, (c) extracted marked BAG image from (b) which is saved in BMP format by Li et al.'s method [8], (d)-(f) the difference images generated from (b) which is saved in BMP format by our proposed method with the resaved quality factors ranging from 80 to 90 in a step of 5, and (g)-(i) the difference images generated from (b) which is saved in JPEG format by our proposed method with the resaved quality factors ranging from 70 to 80 in a step of 5.

$$D_Q(x,y) = \frac{1}{3}\sum_{i=1}^{3}\frac{1}{b^2}\sum_{m=0}^{b-1}\sum_{n=0}^{b-1} F_Q(x+m, y+n, i) \tag{4}$$

where $i = 1, 2, 3$ denotes three color channels of RGB, $f_Q(x,y,i)$ denotes the pixel value of the resaved image of channel i with JPEG quality factor Q, and $f(x,y,i)$ denotes the pixel value of to-be detected image of channel i.

3 Experiments and Analysis

Image painting is a popularly-used method to remove the objects. In [8], the doctored region can be detected if there exists a large blank in the BAG image. However, the doctored image must be kept in an uncompressed image format, such as BMP, TIF and PNG, because the BAGs are obtained from all 8×8 blocks of the originally JPEG compressed image. If the doctored image is kept in the JPEG format, a new JPEG block lattice across the entire doctored image will be generated, and therefore, the method will fail to detect the doctored region.

To testify the efficiency of our proposed method, we first use the same original image as used in [8], which is shown in Fig. 2(a), and conduct the same two doctoring methods as presented in [8]. We choose b=10 in eqs. (2) and (4) for all our experiments. Fig. 2(b) is the doctored images from Fig. 2(a) with the spacecraft removed by filling the area with black color by using brush tool in Photoshop. When the doctored image is saved in an uncompressed format, the BAG image can be extracted by [8] as shown in Fig. 2(c). Fig. 2(d)-(f) show the difference images generated by our proposed method with the resaved quality factors 45, 50 and 55, respectively. When the doctored image is saved in the JPEG compressed format, the difference images

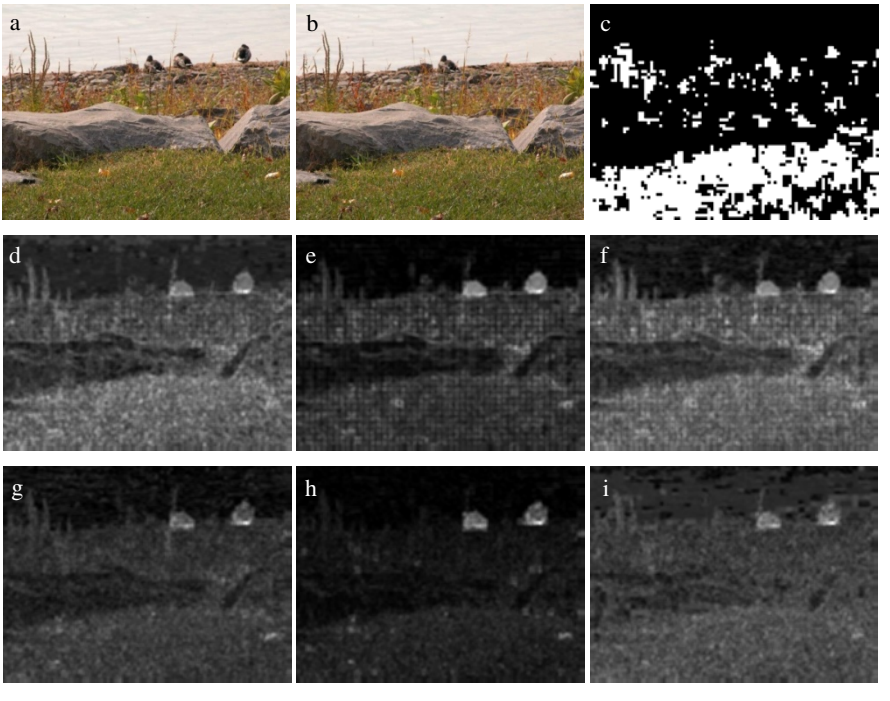


Fig. 5. Detection of painted color image. (a) Original image, (b) doctored image, (c) extracted marked BAG image, (d)-(f) the difference images generated from (b) saved in BMP format by our proposed method with the resaved quality factors ranging from 60 to 70 in a step of 5, and (g)-(i) the resulting difference images generated from (b) saved in JPEG format by our proposed method with the resaved quality factors ranging from 64 to 68 in a step of 2.

generated by our proposed method are shown in Fig. 2(g)-(i) with the resaved quality factors 45, 50 and 55, respectively. Comparing the BAG image with the difference images achieved by our proposed method, it is obviously that our proposed method can detect the doctored region accurately with more details such as the bottom brackets of the spacecraft, even if the doctored image is saved in the JPEG compressed format. Besides, the detected doctored region in the BAG image by [8] is smaller than the actual doctored region in the original image.

Fig. 3(a) is another doctored image from Fig. 2(a), which is doctored by removing the spacecraft with random dark color by using filter tool in Photoshop. When the doctored image is saved in BMP format, the extracted BAG image is shown in Fig. 3(b), and the difference images generated by our method are shown in Fig.3 (c)-(e) with the resaved quality factors 70, 80 and 90, respectively. Shown in Fig. 3(f)-(h) are the difference images generated from Fig. 3(a) saved in JPEG format by our method with the resaved quality factors 70, 80 and 90, respectively.

Fig. 4(a) is the original JPEG image, and Fig. 4(b) is the doctored image in which the conveyance on the top is concealed by painting. Fig. 4(c) shows the marked BAG image generated from the doctored image saved in BMP format. Although the doctored region can be detected, some noise is present in the non-doctored region, which causes disturbance/error in identifying the doctored region. Fig. 4(d)-(f) are the difference images for the doctored image saved in BMP format by our proposed method with quality factors ranging from 80 to 90 in a step of 5. Fig. 4(g)-(i) are the difference images for the doctored image saved in JPEG format with quality factors ranging from 70 to 80 in a step of 5. It is easy and clear for one to judge that the given image (Fig. 4(b)) has been doctored and, moreover, where the doctored region is.

Shown in Fig. 5(a) is the original color image from [15] which is JPEG compressed at quality factor of 66. Fig. 5(b) is the doctored image in which two birds are concealed by painting. Fig. 5(c) is the marked BAG image extracted from the doctored image saved in BMP format by the method in [8], from which it is very difficult to judge the doctored regions. Shown in Fig. 5(d)-(f) are the difference images for the doctored image saved in BMP format with the resaved quality factors ranging from 60 to 70 in a step of 5. Shown in Fig. 5(g)-(i) are the difference image for the doctored image saved in JPEG format at quality factor of 94 with the resaved quality factors ranging from 64 to 68 in a step of 2.

Fig. 6(a) is an original color image from [16] which is JPEG compressed at quality factor of 5 out of 12 (Photoshop). Fig. 6(b) is the doctored image in which two persons are removed by the exemplar-based image inpainting method proposed in [13]. Fig. 6(c) is the marked BAG image extracted from the doctored image saved in BMP format by the method in [8], from which it is difficult to distinguish the doctored region. Shown in Fig. 6(d)-(f) are the difference images for the doctored image saved in BMP format with the resaved quality factors ranging from 85 to 95 in a step of 5. Shown in Fig. 6(g)-(i) are the difference images for the doctored image saved at JPEG quality factor of 10 out of 12 with the resaved quality factors ranging from 88 to 92 in a step of 2. Obviously, from the results of our proposed method, it is easy to judge the doctored region no matter the doctored image is saved in BMP format or saved in JPEG format, even if the doctored region is small in size.

To further confirm the validity of our proposed algorithm, we have randomly chose 1000 uncompressed color images from [16] and conducted JPEG compression on

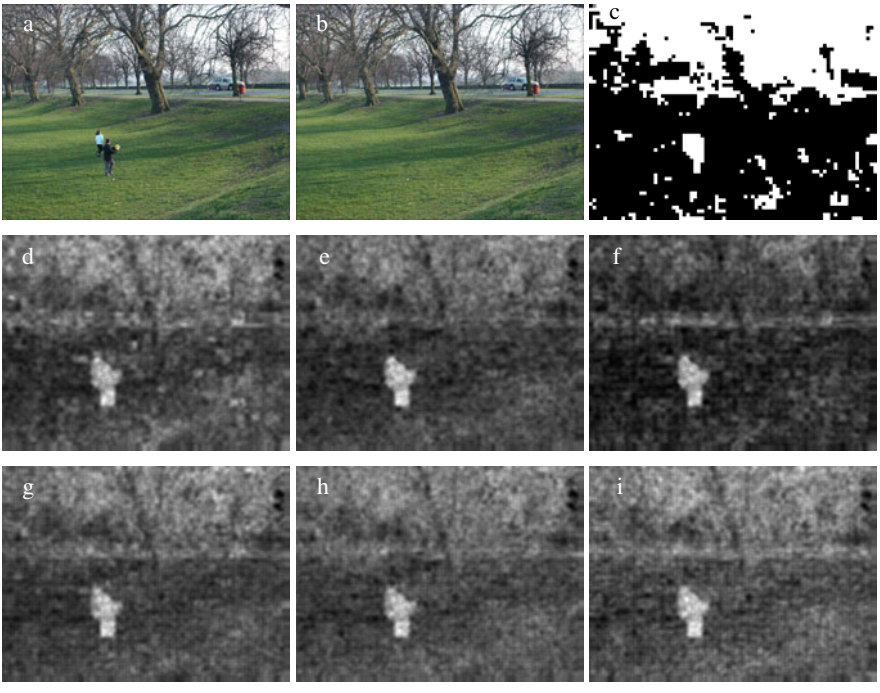


Fig. 6. Detection of an exemplar-based inpainting [13] color image. (a) Original image, (b) inpainting doctored image, (c) extracted marked BAG image, (d)-(f) the difference images generated from (b) saved in BMP format by our proposed method with the resaved quality factors ranging from 85 to 95 in a step of 5, and (g)-(i) the difference images generated from (b) saved in JPEG format by our proposed method with the resaved quality factors ranging from 88 to 92 in a step of 2.

them at quality factor of 70 (out of 100, Matlab). Subsequently, we remove a central circle region of radius 40 from every obtained JPEG image by the exemplar-based image inpainting method proposed in [13]. In our experimental works of detecting the paint-doctored region by using our proposed method, we set the false alarm rate (an authentic image incorrectly classified as paint-doctored) as 1%. The achieved detection accuracy is 99.5% and 95.9%, respectively, as the paint-doctored images are saved in BMP format and in the JPEG format with the quality factor 90 (out of 100).

With MATLAB 7.9 environment on a ThinkPad T400 notebook PC (2.53 GHz CPU, 2.00 GB memory), for a detected color image with the size of 384×512, our proposed method costs 14.45s to yield four difference images with b = 10 in eq. (4), and the BAG extraction method [8] costs 30.08s. It is obviously that our proposed method is computationally simpler than the BAG extraction method.

4 Conclusion

Image forensics is becoming increasingly important with the rapid development of digital techniques. In this paper, a novel passive detection method for paint-doctored

JPEG images is proposed. The doctored regions can be detected by calculating the averaged sum of absolute difference images between the to-be-examined image and its resaved JPEG images at different quality factors. This method works no matter what type of image format (i.e., the standard JPEG compressed or non-standard JPEG compressed) the original image and the paint-doctored image assume. Although our proposed method is related to the original JPEG quality factor, it does not need to estimate the primary quantization matrix [17-18]. There are several advantages of the proposed method. First, it can detect paint-doctored JPEG images when the doctored image is saved in an uncompressed format or JPEG compressed format. Second, it can detect the doctored region accurately even if the doctored region is small in size. Third, it can detect multiple doctored regions in the same image. Fourth, the computation is simple. Finally, it can detect the doctored region automatically and does not need any manually operation. This paper also compares the proposed method with the BAG extraction method in [8]. Experimental results have demonstrated that our proposed method can detect the paint-doctored region efficiently and accurately if the paint-doctored image is saved in BMP format or in JPEG format whose quality factor is larger than the original JPEG quality factor. The main disadvantage of the proposed method is that it cannot detect the paint-doctored region if the paint-doctored JPEG image is saved in smaller JPEG quality factor than the original quality factor.

Acknowledgments

This work is supported by Hunan Provincial Natural Science Foundation of China (09JJ3119), the Planned Hunan Provincial Science and Technology Project of China (2009FJ3015), and China Postdoctoral Science Foundation Specially Funded Project (200902482). The authors would like to thank Dr. Weihai Li for giving them good suggestion, and thank the anonymous reviewers for their valuable comments.

References

1. Lu, C.S., Liao, H.Y.M.: Structural digital signature for image authentication: an incidental distortion resistant scheme. IEEE Transactions on Multimedia 5(2), 161–173 (2003)
2. Celik, M., Sharma, G., Saber, E., Tekalp, A.: Hierarchical watermarking for secure image authentication with localization. IEEE Transactions on Image Processing 11(6), 585–595 (2002)
3. Li, B., Shi, Y.Q., Huang, J.: Detecting double compressed JPEG Image by using mode based first digit features. In: IEEE International Workshop on Multimedia Signal Processing, MMSP 2008, Queensland, Australia, pp. 730–735 (2008)
4. Lin, Z., He, J., Tang, X., Tang, C.K.: Fast, automatic and fine-grained tampered JPEG image detection via DCT coefficient analysis. Pattern Recognition 42(11), 2492–2501 (2009)
5. Mahdian, B., Saic, S.: Using noise inconsistencies for blind image forensics. Image and Vision Computing 27(10), 1497–1503 (2009)
6. Cao, H., Kot, A.C.: Accurate detection of demosaicing regularity for digital image forensics. IEEE Trans. on Information Forensics and Security 4(4), 899–910 (2009)
7. Farid, H.: A survey of image forgery detection. IEEE Signal Processing Magazine 26(2), 16–25 (2009)

8. Li, W., Yuan, Y., Yu, N.: Passive detection of doctored JPEG image via block artifact grid extraction. Signal Processing 89(9), 1821–1829 (2009)
9. Luo, W., Qu, Z., Huang, J., Qiu, G.: A novel method for detecting cropped and recompressed image block. In: Proc. ICASSP 2007, IEEE International Conference on Acoustics, Speech and Signal Processing, vol. 2, pp. II-217–II-220 (2007)
10. Ye, S., Sun, Q., Chang, E.C.: Detecting digital image forgeries by measuring inconsistencies of blocking artifact. In: Proc. of IEEE International Conference on Multimedia and Expo, pp. 12–15 (2007)
11. Chen, C., Shi, Y.Q., Su, W.: A machine learning based scheme for double JPEG compression detection. In: IEEE International Conference on pattern Recognition, ICPR 2008, Florida, USA, pp. 1–4 (2008)
12. Farid, H.: Exposing digital forgeries from JPEG ghosts. IEEE Transactions on Information Forensics and Security 4(1), 154–160 (2009)
13. Criminisi, A., Perez, P., Toyama, K.: Region filling and object removal by exemplar-based inpainting. IEEE Transactions on Image Processing 13(9), 1200–1212 (2004)
14. Bertalmio, M., Vese, L., Sapiro, G., Osher, S.: Simultaneous structure and texture image inpainting. IEEE Transactions on Image Processing 12(8), 882–889 (2003)
15. Olmos, A., Kingdom, F.A.A.: McGill Calibrated Colour Image Database (2004), `http://tabby.vision.mcgill.ca`
16. Schaefer, G., Stich, M.: UCID - An Uncompressed Colour Image Database, Technical Report. School of Computing and Mathematics, Nottingham Trent University, U.K (2003)
17. Lukas, J., Fridrich, J.: Estimation of primary quantization matrix in double compressed JPEG images. In: Proceedings of Digital Forensic Research Workshop, Cleveland, OH, USA, pp. 5–8 (2003)
18. Fu, D., Shi, Y.Q., Su, W.: A generalized Benford's law for JPEG coefficients and its applications in image forensics. In: Proceedings of SPIE, Security, Steganography and Watermarking of Multimedia Contents IX, San Jose, USA, p. 65051L (2007)

Detecting Digital Image Splicing in Chroma Spaces

Xudong Zhao[1], Jianhua Li[1], Shenghong Li[1], and Shilin Wang[2]

[1] Department of Electronic Engineering, Shanghai Jiao Tong University
[2] School of Information Security Engineering, Shanghai Jiao Tong University
Shanghai, P.R. China 200240

Abstract. Detecting splicing traces in the tampering color space is usually a tough work. However, it is found that image splicing which is difficult to be detected in one color space is probably much easier to be detected in another one. In this paper, an efficient approach for passive color image splicing detection is proposed. Chroma spaces are introduced in our work compared with commonly used RGB and luminance spaces. Four gray level run-length run-number (RLRN) vectors with different directions extracted from de-correlated chroma channels are employed as distinguishing features for image splicing detection. Support vector machine (SVM) is used as a classifier to demonstrate the performance of the proposed feature extraction method. Experimental results have shown that that RLRN features extracted from chroma channels provide much better performance than that extracted from R, G, B and luminance channels.

Keywords: image splicing, chroma spaces, RLRN, SVM.

1 Introduction

"Photograph lost its innocence many years ago". It can be dated to 1860s when photographs have already been manipulated [1]. With the help of cheap and high resolution digital cameras and powerful photo editing software, forged images are appearing with a fast growing frequency and sophistication. Over the past few years, digital image forensics has emerged to regain some trust to digital images. Digital watermarking has been proposed as an active detecting approach. In order to detect image forgeries, watermark must be inserted at the time of imaging, however most digital cameras do not have this function due to cost and imaging quality consideration. In contrast, passive approaches for image forensics do not need any watermark or prior knowledge, and it gains much attention. In this paper, we focus on passive image forgeries detection method.

Researchers have recently make effort on digital image forgeries detection and several methods have been proposed. In [2] a blind splicing detection approach based on a natural image model is proposed. The natural image model consists of statistical features including moments of characteristic functions of wavelet sub-bands and Markov transition probabilities of difference 2-D arrays. This

H.-J. Kim, Y. Shi, and M. Barni (Eds.): IWDW 2010, LNCS 6526, pp. 12–22, 2011.

method achieved 91.8% detecting accuracy over [3]. Image region duplication is a commonly used image tampering where one part of the image is copyed and pasted to another part. In [4] [5], SIFT features was proposed to detect image region duplications. SIFT features from Different image regions are compared and their correlations are used to output a map indicating region with high possibility to be duplicated from another region. The proposed method can handle cases when a region is scaled or rotated before pasted to a new location. A blind image forgery detection method using noise inconsistencies is proposed in [6]. The proposed method capable of dividing an investigated image into various partitions with homogenous noise levels, and the detection of various noise levels in an image may signify image forgery. Lighting inconsistence in an image is proposed as an evidence of image tampering in [7]. However, the propose method failed in detecting splicing part with the same light direction. Most digital cameras employ a single CCD with a color filter array (CFA), and interpolate the missing color samples to obtain a true color image. Correlations introduced by the CFA interpolation are likely to be destroyed when tampering an image. Popescu and Farid develop a method of detecting image forgeries by using the CFA interpolation [8]. In [9] and [10], an improved bicoherence based features was proposed, and a model of image splicing to explain the effectiveness of bicoherence for image splicing detection was also proposed, the detecting rate over [3] reach 72%. Dong et al in [11] analyzed the discontinuity of image pixel correlation and coherency caused by image splicing in terms of multi-order moments of the characteristic function of image run-length histogram and Edge based statistic moments. It achieved 76.52% detecting accuracy over [3].

Nowadays, most of the detecting methods in image forensics are focus on RGB color space or luminance channel, however, useful information in other color spaces and correlations between different channels are often neglected. Wang et al. in [12] investigated the detecting efficiency of YCbCr color space over [13] based on co-occurrence matrix, and the experimental results showed that features extracted from Cb and Cr channels demonstrated much better performance than that extracted from luminance channel. In this paper, four directional (0°, 45°, 90° and 135°) run-length run RLRNs extracted from gray level run-length pixel number matrix of de-correlated channel are used as distinguishing features to separate spliced images from authentic ones. And we find that, RLRNs are much more sensitive to image splicing in chroma spaces than that in luminance and RGB spaces, and this phenomenon is analyzed. Finally, necessary comparisons are made to show the effectiveness of proposed detecting work.

The rest of this paper is organized as follows. The proposed approach is described in section 2. In section 3, the experimental works and results are reported. Finally, conclusion and discussion are provided in Section 4.

2 Capturing Image Splicing in Chroma Spaces

It is known that, image splicing will introduce abrupt changes (i.e. sharp edges) of image pixel intensity, capturing these abrupt intensity changes is the key in

image splicing detection work. Image forgers usually do their tampering in a certain color space (most of time, RGB), and they will certainly try to cover manipulated traces. For this reason, detecting forgery traces in the manipulated color space is usually a tough work. In this paper, four directional Run-Length Run-Number (RLRN) vectors are used to capture splicing traces and chroma spaces are introduced as an effective way of detecting image splicing. In order to reduce the unnecessary influence caused by smooth image area, image de-correlation is first applied as a preprocessing step before detecting work.

2.1 Image Preprocessing and Feature Extraction

One of the difficulties in image forgery detection is how to isolate the suspicious part from the image background. In this paper, image de-correlation is used as a preprocessing step to reduce the influence caused by the diversity of image content. Image de-correlation can be obtained by subtracting neighbor pixel value from the original pixel [14], the computations are as follows:

$$\begin{aligned} E_h(i,j) &= |x(i,j+1) - x(i,j)| \\ E_v(i,j) &= |x(i+1,j) - x(i,j)| \\ E_d(i,j) &= |x(i+1,j+1) - x(i,j)| \\ E_{-d}(i,j) &= |x(i+1,j-1) - x(i,j)| \end{aligned} \tag{1}$$

where $x(i,j)$ is the image gray value at the position of (i,j), $E_h(i,j)$, $E_v(i,j)$, $E_d(i,j)$ and $E_{-d}(i,j)$ indicate de-correlated image along horizontal vertical, diagonal and counter-diagonal directions respectively. An example of a spliced image's E_d in Y, Cb and Cr channels is given in Fig. 1. It can be seen that most of image information is removed except for abrupt changes of pixels, i.e. edges. Image splicing introduces discontinuity in the spliced region and it can be captured by image de-correlation.

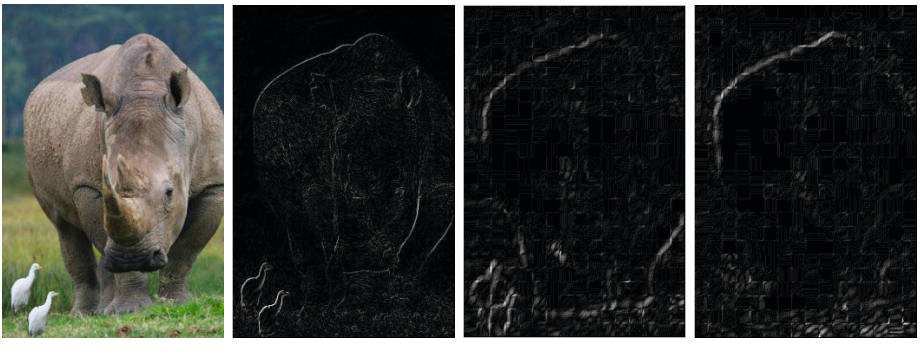


Fig. 1. A spliced image and its de-correlated images along diagonal direction in Y, Cb and Cr channels respectively

Galloway in [15] first proposed the use of a run-length matrix for texture feature extraction. A run is defined as a string of consecutive pixels which have the same gray level intensity along a specific orientation (typically in 0°, 45°, 90°, and 135°). For a given image, a run-length matrix $p_\theta(m,n)$ is defined as the number of runs with gray level m and and run length n along θ direction. In the run-length matrix of image, the number of short runs dominates total number of runs, in order to give equal emphasis on all lengths of run, a transformed run-length matrix is given as follow [16]:

$$\tilde{p}_\theta(m,n) = p_\theta(m,n) \cdot n \tag{2}$$

where $\tilde{p}_\theta(m,n)$ is the variation of $p_\theta(m,n)$, and it gives emphasis on long runs. Run-length run-number vector (RLRN) which is the sum distribution of $\tilde{p}_\theta(m,n)$ with run length n, defined as:

$$p_{\theta r}(n) = \sum_{m=1}^{M} \tilde{p}_\theta(m,n) \tag{3}$$

where M is the number of gray levels. $p_{\theta r}(n)$ we defined in (3) is a variation of Run-Length Run-Number Vector in [16], for the simplicity reason we name it RLRN directly. And $p_{\theta r}(n)$ is employed as the feature extraction method in our work. Image splicing will change the correlation between neighbor pixels, i.e. the local texture information, and the change will captured by the run-length matrix based statistics. A concrete feature extraction procedure is shown in Fig. 2. A color image is first transformed into YCbCr color space, and then image decorrelation in four directions is implemented in every single channel according to (1). Finally RLRN features extracted from the four 2D arrays on every single channel. The detailed procedure is shown in Fig. 3. The number of short runs occupies most of total number of runs, i.e. most information of run-length matrix can be represented by short runs. In our work, the first 15 RLRNs of every decorrelated image are extracted as features, and there are total 60 D features in four directions in every channel.

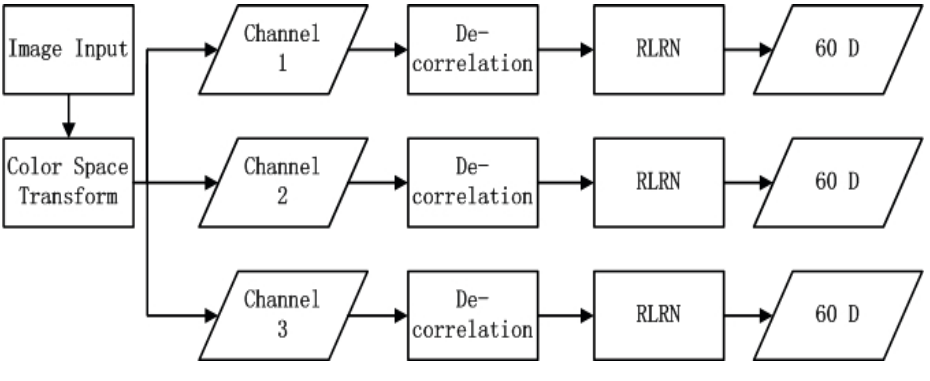


Fig. 2. Diagram of RLRN features extraction

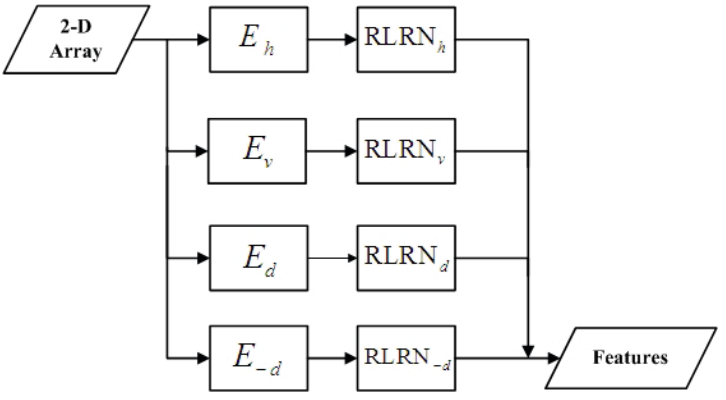


Fig. 3. Features extraction in single channel

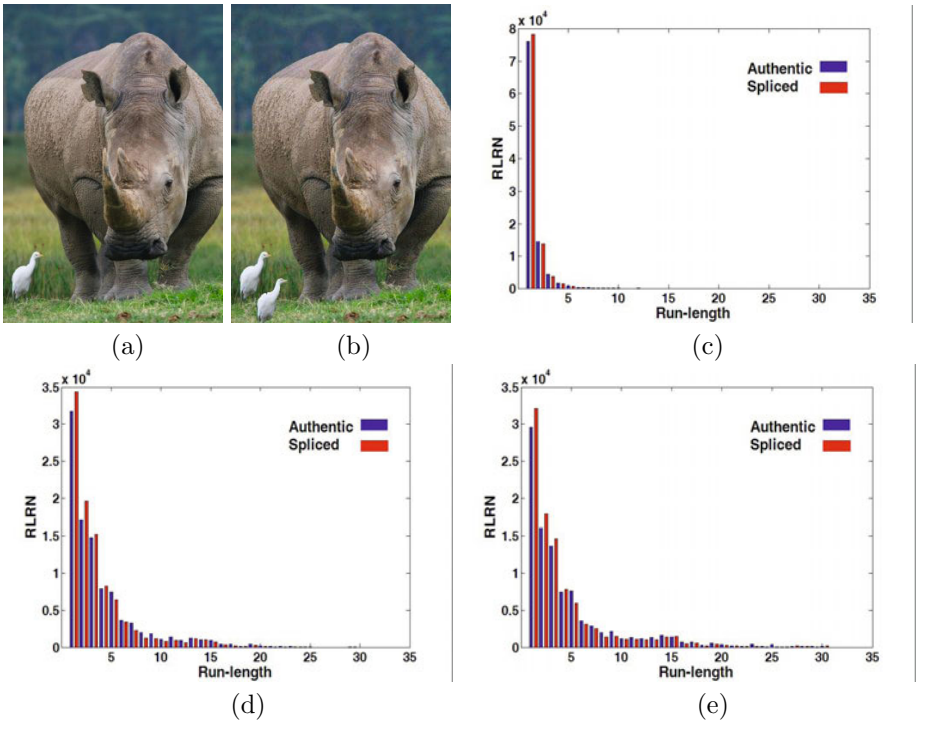


Fig. 4. RLRN compare between authentic and spliced images in YCbCr color space. (a) original image, (b) spliced image, (c) Y channel, (d) Cb channel, (e) Cr Channel.

2.2 Chroma Spaces

Chroma spaces also refer to color difference spaces, i.e. color components where brightness (luma) is removed [17]. It is standard to form two chroma spaces by subtracting luma from blue (B-Y) and by substracting luma from red (R-Y). Commonly used transformation from RGB to Y (luma) is given as:

$$Y = 0.299R + 0.587G + 0.114B \tag{4}$$

and the corresponding transformation from RGB to luma-chroma space is defined as:

$$\begin{pmatrix} Y \\ B-Y \\ R-Y \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.117 \\ -0.299 & -0.587 & 0.886 \\ 0.701 & -0.587 & -0.114 \end{pmatrix} \cdot \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{5}$$

Luma-Chroma spaces usually refer to YIQ, YUV, YCbCr and YCC color models, all these models are the slight variations of (5). YCbCr is introduced as image splicing detecting space in this paper. Since human eyes are more sensitive

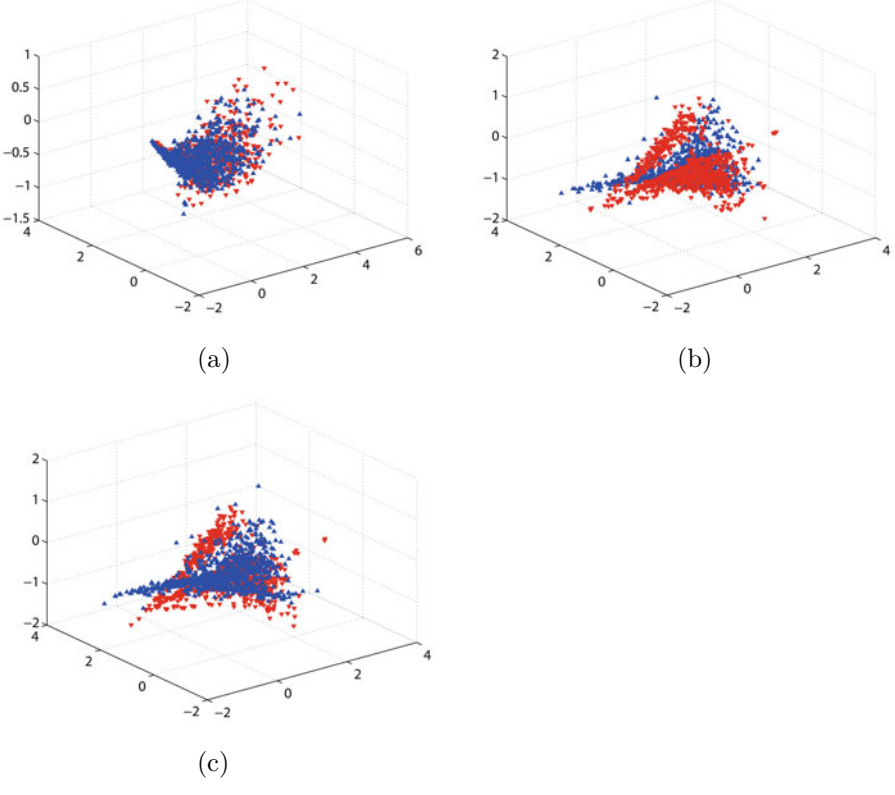


Fig. 5. Scatter distributions of authentic (blue △) and spliced images (red ▽) in Y, Cb and Cr channels. (a) Y channel, (b) Cb channel, (c) Cr channel.

to luminance than to chroma, even if spliced image which looks natural, some tampering traces will be left in chroma channel [12]. An example is shown in Fig. 4. Figure 4(a) is an authentic image and its tampered counterpart is shown in Fig. 4(b), the tampered region is relative small and almost without visual artifact. RLRNs of (a) and (b) in Y, Cb and Cr channels are shown in Fig. 4(c), Fig. 4(d) and Fig. 4(e), where blue bars represent authentic image and red bar represent spliced one. It is clear that RLRN in chroma channels (Cb and Cr) is much easier to separate tampered images from authentic ones than that in luma channel (Y).

In order to demonstrate the universal effectiveness of chroma spaces in detecting image splicing, the first three principal components of RLRN features extracted in Y, Cb and Cr channels over image dataset [13] are scatter plotted in Fig. 5(a), Fig. 5(b) and Fig. 5(c) respectively, where blue "△" represent authentic images and red "▽" represent spliced images. As is shown in Fig. 5, it is clear that, the scatter distribution of spliced images overlaps much of that of authentic ones in Y channel, however, there is a much smaller overlap in either Cb or Cr channel. That is to say RLRN features in Cb and Cr channels demonstrate much better classification performance than that in Y channel.

3 Experiments and Results

To demonstrate the effectiveness of proposed features, experiments and results are presented in this section.

3.1 Image Dataset

CASIA image tampering detection dataset [13] is used in our experimental work. This dataset has 800 authentic and 921 tampered images, it covers a variety of images. All tampered images in this dataset are made by splicing operation without postprocessing (i.e. only crop-and-paste operation). Images in this dataset are all in JPEG format with dimensions of 384×256 (or 256×384). Another color image dataset provided by Digital Video and Multimedia Lab (DVMM) of Columbia University [18] [19] is also used in our experimental work. There are totally 363 images in this dataset, 183 of them are authentic images, and 180 are spliced ones. Authentic images are taken with four different kinds of cameras, tampered images are all made by crop-and-paste operation. Images in this dataset are all in high resolution and umpressed TIFF format with dimensions ranging from 757×568 to 1152×768. These images mainly contain indoor scenes, some images are taken outdoors on a cloudy day.

3.2 Classifier and Results

Support vector machine (SVM) is a supervised machine learning method, and it is widely used for classification and regression. In this paper, LIBSVM [20] is used as classifier and the Radial Basis Function (RBF) is selected as kernel

Table 1. Classification results of R, G, B, Gray, Y, Cb and Cr channels

	Image Dataset 1			Image Dataset 2		
	TP	TN	Accuracy	TP	TN	Accuracy
R	56.3%	83.7%	70.9%	75.3%	75.0%	76.1%
	(0.04)	(0.02)	(2.40)	(0.07)	(0.08)	(5.12)
G	51.8%	83.2%	68.5%	72.1%	77.8%	74.9%
	(0.04)	(0.03)	(2.59)	(0.06)	(0.08)	(4.57)
B	57.2%	83.7%	71.3%	65.9%	78.7%	73.1%
	(0.04)	(0.02)	(2.37)	(0.08)	(0.09)	(5.47)
Gray	60.4%	81.7%	71.8%	70.7%	79.0%	74.7%
	(0.04)	(0.04)	(2.94)	(0.08)	(0.07)	(5.58)
Y	53.3%	83.1%	69.2%	76.1%	76.9%	76.3%
	(0.04)	(0.03)	(2.48)	(0.07)	(0.09)	(5.61)
Cb	91.7%	96.5%	94.3%	81.8%	82.6%	82.1%
	(0.02)	(0.02)	(1.20)	(0.07)	(0.07)	(3.76)
Cr	91.8%	97.1%	94.7%	80.2%	89.8%	85.0%
	(0.02)	(0.02)	(1.19)	(0.08)	(0.05)	(4.67)

function of SVM. Grid searching is employed to select the best parameters C and γ for classification. 6-fold cross-validation is employed in classification, and it is repeated 30 times for each parameter group (C, γ) then the performance is measured by the average classification accuracy across the 30 runs.

Experimental results of proposed features in RGB, gray and YCbCr color spaces are presented in Table 1 where TP (true positive) represents the detection rate of authentic images, TN (true negative) represents the detection rate of spliced images, and accuracy is the average detection rate. Standard deviation among 30 random tests is shown in parentheses. In the following table, image dataset 1 indicates CASIA dataset and image dataset 2 stands for DVMM dataset.

From the experimental results shown above, it can be seen that:

- RLRN in Cb and Cr channels perform best among RGB, YCbCr and gray spaces over both image datasets. Detecting accuracies in Cb and Cr channels over image dataset 1 reach as high as 94.3% and 94.7% respectively, and that over image dataset 2 reach 82.1% and 85.0%.
- Detecting accuracies of RLRN in commonly used R, G, B channels over both image datasets vary within three percent range. This is mainly due to strong correlations which are introduced in the color filter array (CFA) interpolation process.
- Performance of RLRN in Y channel is approximately the same as that in RGB space over both image datasets. Y channel also refers to luma channel which is the linear combination of R, G and B channels. Detecting performance in Y channel isn't superior to that in R, G and B channels.

Table 2. Comparison between proposed RLRN features and [12] proposed gray level co-occurrence matrix features over image dataset 1. "Dim" means dimensionality of features.

	Proposed Features		Features of [12]	
	Accuracy	Dim	Accuracy	Dim
Cb	94.3%	60	88.6%	100
Cr	94.7%	60	90.5%	50

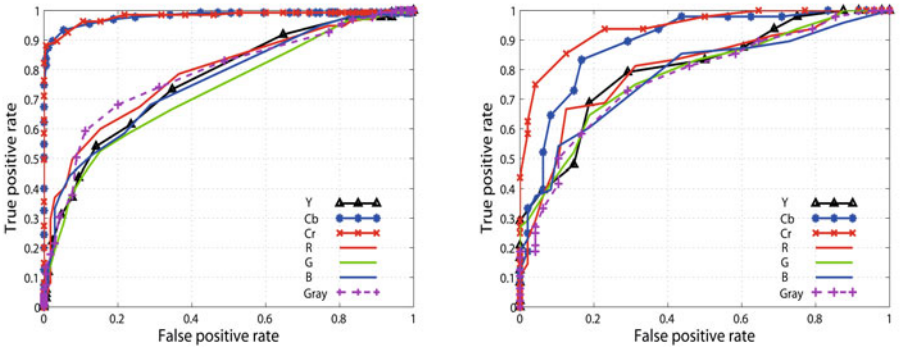


Fig. 6. Comparison of RLRN in RGB, YCbCr and Gray color spaces. (a) CASIA1 image dataset, (b) DVMM image dataset.

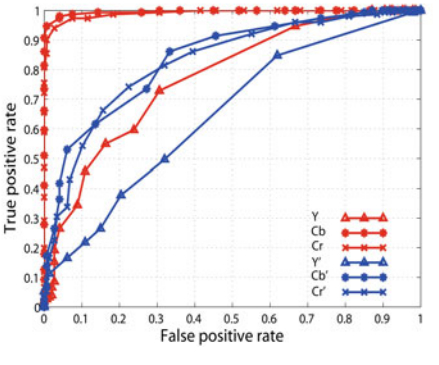


Fig. 7. Comparison between proposed 60 D features (red) and 61 D features (blue) of [11] in YCbCr color space

– Chroma RLRN features over image dataset 1 demonstrate better performance than that over image dataset 2, i.e. image format (JPEG or TIFF) has influence on proposed image splicing detection. The relationship between image formats and RLRN features will be further studied in the future work.

Detection rates of proposed chroma features are also shown in Table 2 in comparison with the best detection rates reported in [12].

Receiver operating characteristics (ROC) curves in YCbCr, RGB and Gray spaces over image dataset 1 and image dataset 2 are compared in Fig. 6 (a) and Fig. 6(b) respectively. Comparison between proposed feature and [11]'s features in YCbCr space is shown in Fig. 7, where red curves represent proposed features and blue curves represent features of [11].

4 Conclusions

Detecting image splicing in chroma spaces based on RLRN features is proposed in this paper. Run-length matrix is a way of describing texture information of images, in natural images, the number of short runs dominates total number of runs, in order to give emphasis on long runs in an image, a variation of the traditional run-length matrix is defined, which is given in formula (2). RLRN extracted from the modified run-length matrix reflects the distribution of run length in the image, and it is introduced as distinguishing features in this paper. Run-length matrix is not isotropy, and neither is RLRN, then four RLRNs with different directions are extracted and treated as final features. Chroma space refers to color difference space where luma information is removed. YCbCr which is one of luma-chroma spaces, is introduced in our detecting work. We find that RLRN is much more sensitive to image splicing in chroma spaces than that in luma space or RGB space, and the reason is analyzed. SVM is employed as a classifier to test the effectiveness of proposed method. Experimental results have shown that performance of RLRN in chroma channel (Cb or Cr) outperforms that in luma and RGB channels over two different image datasets. Detecting rate reaches 94% over image dataset 1, and 85% over image dataset 2. Further research on the relationship between color spaces and feature extraction method is necessary. Color space which is optimal space in the sense of image splicing detection will be studied.

Acknowledgements

This research work is funded by the National Natural Science Foundation of China (Grant No. 61071152, 60702043), 973 Program (Grant No. 2010CB731403) of China, and Shanghai Educational Development Foundation. Credits for the use of the CASIA Image Tempering Detection Evaluation Database (CAISA TDED) V1.0 are given to the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Science, Corel Image Database and the photographers. http://forensics.idealtest.org.

References

1. Photo tampering throughout history, http://www.cs.dartmouth.edu/farid/research/digitaltampering
2. 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)
3. Ng, T.T., Chang, S.F., Sun, Q.: A data set of authentic and spliced image blocks. Tech. Rep., DVMM, Columbia University (2004), http://www.ee.columbia.edu/ln/dvmm/downloads/AuthSplicedDataSet/photographers.htm
4. 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 (PACIIA 2008) (2008)
5. Pan, X., Lyu, S.: Detecting image region duplication using SIFT features. In: 2010 Acoustics Speech and Signal Processing, ICASSP 2010 (2010)
6. Mahdian, B., Saic, S.: Using noise inconsistencies for blind image forensics. Image and Vision Computing 27, 1497–1503 (2009)
7. Johnson, M.K., Farid, H.: Exposing digital forgeries by detecting inconsistencies in lighting. In: ACM Proceedings of the 7th Workshop on Multimedia and Security, pp. 1–10 (2005)
8. Popescu, A.C., Farid, H.: Exposing digital forgeries in color filter array interpolated images. IEEE Transactions Signal Processing 53(10), 3948–3959 (2005)
9. 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)
10. Ng, T.T., Chang, S.F.: A model for image splicing. In: 2004 International Conference on Image Processing (ICIP 2004), pp. 1169–1172 (2004)
11. 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)
12. Wang, W., Dong, J., Tan, T.: Effective image splicing detection based on image chroma. In: 2009 International Conference on Image Processing, ICIP 2009 (2009)
13. CASIA Tampering Detection Dataset, http://forensics.idealtest.org
14. Zou, D., Shi, Y.Q., Su, W.: Steganalysis based on markov model of thresholded prediction-error image. In: IEEE International Conference on Multimedia and Expo, Toronto, Canada (2006)
15. Galloway, M.M.: Texture analysis using gray level run lengths. Comput. Graphics Image Process. 4, 172–179 (1975)
16. Tang, X.: Texture information in run-length matrices. IEEE Transactions on Image Processing 7(11) (November 1998)
17. Poynton, C.: Frequently asked questions about color, http://www.poynton.com/Poynton-color.html
18. Hsu, Y.-F., Chang, S.-F.: Detecting image splicing using geometry invariants and camera characteristics consistency. In: International Conference on Multimedia and Expo, Toronto, Canada (July 2006)
19. DVMM Laboratory of Columbia University: Columbia Image Splicing Detection Evaluation Dataset, http://www.ee.columbia.edu/ln/dvmm/downloads/AuthSplicedDataSet/photographers.htm
20. Chang, C.C., Lin, C.j.: LIBSVM: a library for support vector machines (2001), http://www.csie.ntu.edu.tw/~cjlin/libsvm

Discriminating Computer Graphics Images and Natural Images Using Hidden Markov Tree Model

Feng Pan and Jiwu Huang

Guangdong Key Lab of Information Security Technology, Guangzhou, 510275, China
School of Information Science and Technology, Sun Yat-Sen University, Guangzhou, 510275, China

Abstract. People can make highly photorealistic images using rendering technology of computer graphics. It is difficult to human eye to distinguish these images from real photo images. If an image is photorealistic graphics, it is highly possible that the content of the image was made up by human and the reliability of it becomes low. This research field belongs to passive-blind image authentication. Identifying computer graphics images is an important problem in image classification, too. In this paper, we propose using HMT(hidden Markov tree) to classifying natural images and computer graphics images. A set of features are derived from HMT model parameters and its effect is verified by experiment. The average accuracy is up to 84.6%.

1 Introduction

Advanced rendering techniques of computer graphics are able to generate all kinds of photorealistic images,called as CG (computer graphics) image in this paper. At current, there are many kinds of software, such as 3DMAX and Photoshop can generate CG images. According to current computer graphic technique, it is difficult to distinguish these photorealistic images from real photo image by human eye. Aliasis company focuses on the 3D computer graphics technology [10]. On its website(http://www.fakeorfoto.com), there is a challenge question for visitors, which requires vistors to make a judgement between CG images and photo images. For human eye, it is a difficult task due to the highly photorealistic. With the improvement of computer graphics technique, software will become more and more powerful and will be able to produce more realistic images.

According to [13], natural images are the photographic images of scenes which human eye can see (different from satellite, or microscopic images). Classifying images into CG and natural images is a new research area. Vassilis Athitsos [1] designed a system, which classified the images downloaded from Internet into natural images and CG images by image content . This system proposed some features, such as color transition model from one pixel to another, the probability on which color present in images, the number of color in an image and image size, and then used binary decision tree to determinate natural or CG

H.-J. Kim, Y. Shi, and M. Barni (Eds.): IWDW 2010, LNCS 6526, pp. 23–28, 2011.

image. Ng et al. [12] gave an image model based on geometry, and implemented a online system [11] for the discrimination of CG and natural images. Ng et al. also proposed the difference between CG and natural images in three aspects: object model, light model and acquisition device. A set of 108 features based on fractal geometry, differential geometry and Beltrami flow were extracted to measure these differences. The average accuracy of the algorithm achieved 83.5%. Dehnie et al. [5] proposed that, although different cameras had different types of pattern noise, natural images taken by different cameras showed that their pattern noises had common property compared with CG images. Farid, et al. [6] gave an algorithm based on wavelet coefficient statistics. The test image is decomposed in horizontal, vertical and diag directions on the three color channels by QMF(quadrature mirror filter). Predictive errors are attained. At last, the statistics on coefficient and predictive errors: average, deviation, skewness and kurtosis, are used as features. Fisher classifier is used for classification, and correct rate of discrimination of natural images is 98.7%, CG images is 35.4%. Lyu[9] and Farid[7] improved this result by SVM classifier. Ianeva[8] detects cartoon videos, and the values of histogram of gradient intensity and direction are chosen as features. In this paper, we propose an effective features vector to classify CG images and natural images. Our main contribution is that a set of features are extracted from HMT model parameters and experiment prove the effectiveness.

The paper is organized as follows. In section 2, we give an introduction about HMT. The experiment results are then given in section 3, and finally section 4 makes an analysis and draw a conclusion.

2 Wavelet-Domain HMT Model

Because of its energy compaction and decorrelative properties, wavelet transform is a very useful tool for image compression, image coding, noise removal, and image classification. Modeling wavelet coefficient is a interesting research field and a lot of research results were published. However, research shows that, neither of wavelet coefficient is independent, nor ideally Gaussian. Independent Gaussian model can not completely capture the wavelet coefficient properties of natural images[14], because of:

- Heavy-tailed marginal histograms: Wavelet coefficient histogram has a strong zero peak and a heavy tail.
- Persistence across scales: Wavelet coefficients always are large when coefficients of neighbor scales are large, and vice versa.

A number of models have been developed for modeling marginal and joint statistics of wavelet coefficient, which mostly focus on non-Gaussian,mixture of Gaussian, generalized Gaussian distribution, Bessel function and hidden Markove tree model[2].The HMT captures the statistical properties of the coefficients of the wavelet transform.

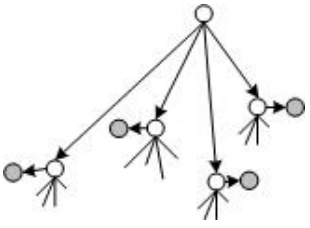


Fig. 1. Quad-tree structure of the HMT model: each white node denotes a discrete state and the adjacent gray node corresponds to the wavelet coefficient

In this work, we use the hidden Markov tree, which is first introduced by Crouse [4]. Fig.1 is a 2-D wavelet hidden Markov tree model for one decomposition. Each white node denotes as a state and each gray node denotes a wavelet coefficient. Wavelet coefficient is modeling by a Gaussian mixture controlled by a hidden state. A Markov chain is made up of hidden states of different scale in vertical direction.

HMT models both the marginal distribution of the wavelet coefficients and the persistence of coefficients across scale.

The marginal distribution of each coefficient is then modeled by a M-state Gaussian mixture. Due to the compression property of wavelet transform, a large number of coefficients are small and a small number of coefficients are large. This Gaussian mixture closely matches the nonGaussian marginal statistics in natural images.

The pdf(probability density function) $f(w_i)$ of each wavelet coefficient is approximated by M-sate Gaussian mixture model.

$$f(w_i) = \sum_{m=1}^{M} p_s(m) f_{w|s}(w \,|s = m) \tag{1}$$

$$\sum_{m=1}^{M} p_s(m) = 1 \tag{2}$$

where s denotes discrete state variable taking values $s \in 1, 2, ..., M$, $f_{w|s}(w \,|s = m)$ is conditionally Gaussian probability density function.

Because a little number of wavelet coefficients are large value and a large number of coefficients are small, two-state can model this phenomenon. One state stands for wavelet coefficient of large value, another for small value.

The HMT captures the persistence of coefficients across scale, which is implemented by using Markov chain of the hidden states across scale in the tree structure. The dependency relationship is showed in 2. Each parent wavelet coefficient corresponds to four children in finer scale. When the parent coefficient is small, it is possible that the children are small too. A hidden state is attached to a wavelet coefficient, which indicate whether this coefficient is more likely large

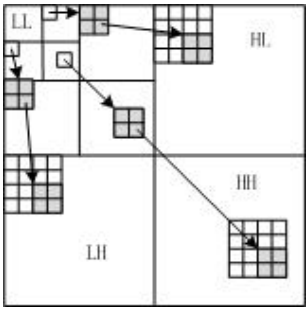


Fig. 2. Wavelet coefficient dependency across scale

or small. All the hidden states across scale compose of a Markov chain, which capture the dependency among wavelet coefficient.

For each child-parent pair of hidden states $\{s_{\rho(i)}, s_i\}$ in the Markov chain, the state transition probability $\varepsilon_{i,m'}^{\rho(i),m}$ represents the probability that the child coefficient is in state of m' when the parent $\rho(i)$ is in m state. As to two-state model, we have the following state transition probability matrix.

$$\begin{bmatrix} \varepsilon_{i,1}^{\rho(i),1}, \varepsilon_{i,2}^{\rho(i),1} \\ \varepsilon_{i,1}^{\rho(i),2}, \varepsilon_{i,2}^{\rho(i),2} \end{bmatrix} = \begin{bmatrix} \varepsilon_{i,1}^{\rho(i),1}, 1-\varepsilon_{i,1}^{\rho(i),1} \\ 1-\varepsilon_{i,2}^{\rho(i),2}, \varepsilon_{i,2}^{\rho(i),2} \end{bmatrix} \tag{3}$$

By [4], the HMT model is made up of the following parameters:

1. $p_{s1}(m)$, the probability that the root node is in state m.
2. $\varepsilon_{i,m'}^{\rho(i),m}$, the transition probability that i is in state m' when $\rho(i)$ in state m.
3. $\mu_{i,m}, \sigma_{i,m}^2$, the mean and variance of $f_{w|s}(w \mid s = m)$.

These parameters compose of the model parameter vector θ. These parameters can be calculated using EM algorithm[4]. Classification features are extracted from θ.

3 Experiments

In our experiment, we test the proposed algorithm on an image set including 3000 natural images and 3000 CG images. 1500 natural images and 1500 CG images are used as training images, and the rest are used as test images. The photorealistic CG images were downloaded from some 3D graphics websites, www.3dshop.com, www.raph.com, etc. Their contents are diverse, including figures, buildings, animals, etc. The natural images come from two source, two thousands are from Washington image databases (http://www. cs. washington. edu /research /imagedatabase /groundtruth/), the rest one thousand are taken by ourself. The content and style of natural images are plentiful as far as possible.

In order to accelerate the computing, all images are cropped into size 256×256. Daubechie wavelet is adopted to construct the HMT. In the HMT model

(a) CG images

(b) natural images

Fig. 3. Some experiment samples

parameter vector θ, several parameter are redundant as classification, which can be removed.

1. As to Eq.(2), when $M = 2$, $p_s(1) = 1 - p_s(2)$. So, $p_s(1)$ and $p_s(2)$ have the same ability to describe the hidden state probability. Only $p_s(1)$ is used as one classification feature and the other is neglected.
2. By Eq.(3), transition probability $(\varepsilon_{i,1}^{\rho(i),1}, \varepsilon_{i,2}^{\rho(i),2})$ are chosen as classification feature.
3. We assume that the $f_{w|s}(w\,|s = m)$ is zero mean Gussian pdf, so $\mu_{i,m}$ is neglected. $\delta_{i,m}^2$ is taken into account.

Two-state Gaussian mixture model is chosen, that means $M = 2$. So on each wavelet coefficient sub-band of certain scale, there are 5 parameters($p_s(1)$, $\varepsilon_{i,1}^{\rho(i),1}$, $\varepsilon_{i,2}^{\rho(i),2}$, $\delta_{i,1}^2$, $\delta_{i,2}^2$) are adopted as classification features. Three sub-bands(HH, HL, LH) on each scale have 15 features in all. Three levels wavelet decomposition are carried out and a gray image can produce 45 features. 135 features can be got from a color image.

The classification experiment is based on the Support Vector Machine (SVM) classifier of the LIBSVM [3]. The Radial Basis Function (RBF) kernel is used for the SVMand optimal parameters are selected by grid [3] in the joint parameter space.

Experiments are done on HSV color space and RGB color space, respectively.When The method is applied on RGB color space, classification accuracy reaches 79.6%. On HSV color space, the accurate rate is 84.6%. The classifying results on color channels are showed by Table 1. Each color channel is taken as a gray image.

Table 1. Detection comparison

	hue	saturation	value
accuracy	82.4%	82.85%	81.93%

4 Conclusions

In this paper, HMT is used to classified CG images and natural images. Experiment prove that HMT parameters are effective classification features.

Experiments are done on HSV color space and RGB color space, respectively. It is clear that the performance is better on HSV color space. Color space distribution in CG images may differ from natural images, which is caused by reducing the computation complexity of rendering algorithm. HSV is appropriate color space to present the difference of color between natural images and CG images.

References

1. Athitsos, V., Swain, M.J., Franke, C.: Distinguishing photographs and graphics on the world wide web. Tech. rep., Department of Computer Science, The University of Chicago (1997)
2. Azimifar, Z., Fieguth, P., Jernigan, E.: Towards random field modeling of wavelet statistics. In: Proceedings of the 9th ICIP, vol. 50, p. 106 (2002)
3. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines (2001)
4. Crouse, M.S., Nowak, R.D., Baraniuk, R.G.: Wavelet-based statistical signal-processing using hidden markov-models. IEEE Trans. Signal Processing 46(4), 886–902 (1998)
5. Dehnie, S., Sencar, T., Memon, N.: Digital image forensics for identifying computer generated and digital camera images. In: International Conference on Image Processing, pp. 2313–2316 (2006)
6. Farid, H., Lyu, S.: Higher-order wavelet statistics and their application to digital forensics (2003)
7. Farid, H.: Creating and detecting doctored and virtual images: Implications to the child pornography prevention act. Tech. Rep. TR2004-518, Dartmouth College, Computer Science, Hanover, NH (September 2004)
8. Ianeva, T., de Vries, A., Rohrig, H.: Detecting cartoons: a case study in automatic video-genre classification. In: IEEE International Conference on Multimedia and Expo., pp. I-449–I-452 (2003)
9. Lyu, S., Farid, H.: How realistic is photorealistic? IEEE Transactions on Signal Processing 53(2-2), 845–850 (2005)
10. Ng, T.T., Chang, S.F.: Classifying photographic and photorealistic computer graphic images using natural image statistics. Tech. rep., Columbia University (October 2004)
11. Ng, T.-T., Chang, S.-F.: An online system for classifying computer graphics images from natural photographs. In: SPIE Electronic Imaging, San Jose, CA (January 2006)
12. Ng, T.-T., Chang, S.-F., Hsu, J., Xie, L., Tsui, M.-P.: Physics-motivated features for distinguishing photographic images and computer graphics. In: Proc. of ACM Multimedia, pp. 239–248 (2005)
13. Schaaf, V.D.: Natural image statistics and visual processing. Ph.D. thesis, Rijksuniversiteit Groningen, The Netherlands (1998)
14. Simoncelli, E.: Modeling the joint statistics of images in the wavelet domain. In: Proc. SPIE., Citeseer, vol. 3813, pp. 188–195 (1999)

A New Scrambling Evaluation Scheme Based on Spatial Distribution Entropy and Centroid Difference of Bit-Plane

Liang Zhao[1,2,⋆], Avishek Adhikari[3], and Kouichi Sakurai[1]

[1] Graduate School of Information Science and Electrical Engineering
Kyushu University, Fukuoka, Japan 819-0395
zhaoliang@itslab.csce.kyushu-u.ac.jp, sakurai@csce.kyushu-u.ac.jp
[2] College of Computer Science
Chongqing University, Chongqing, China 400044
zhaoliangjapan@gmail.com
[3] Department of Pure Mathematics
University of Calcutta, Kolkata, India 700019
avishek.adh@gmail.com

Abstract. Watermarking is one of the most effective techniques for copyright protection and information hiding. It can be applied in many fields of our society. Nowadays, some image scrambling schemes are used as one part of the watermarking algorithm to enhance the security. Therefore, how to select an image scrambling scheme and what kind of the image scrambling scheme may be used for watermarking are the key problems. Evaluation method of the image scrambling schemes can be seen as a useful test tool for showing the property or flaw of the image scrambling method. In this paper, a new scrambling evaluation system based on spatial distribution entropy and centroid difference of bit-plane is presented to obtain the scrambling degree of image scrambling schemes. Our scheme is illustrated and justified through computer simulations. The experimental results show (in Figs. 6 and 7) that for the general gray-scale image, the evaluation degree of the corresponding cipher image for the first 4 significant bit-planes selection is nearly the same as that for the 8 bit-planes selection. That is why, instead of taking 8 bit-planes of a gray-scale image, it is sufficient to take only the first 4 significant bit-planes for the experiment to find the scrambling degree. This 50% reduction in the computational cost makes our scheme efficient.

Keywords: Gray-scale image, Generalized Arnold cat map, Generalized Gray code.

1 Introduction

Due to the fast development of computer science and network technique, the protection of digital information has been brought into focus on a large scale.

⋆ The first author of this research Liang Zhao is supported by the governmental scholarship from China Scholarship Council.

H.-J. Kim, Y. Shi, and M. Barni (Eds.): IWDW 2010, LNCS 6526, pp. 29–44, 2011.

Watermarking can be applied to implement the data hiding, copyright protection, information transmission and so on. Image scrambling, which is suitable for practice applications on information protection [1,2], is one kind of the most prevailing encryption methods, and it, recently, is used for the data hiding and digital watermarking [3,4,5]. In [3], the scrambling based on chaotic cellular automata is used to scramble digital image as a pretreatment for the watermarking process. [4] presents one kind of novel image scrambling scheme for digital watermarking, and in [5], the pixel scrambling method is adopted by the information hiding. Based on the wide application of this technique, specially in watermarking, the performance of it is very significant and needs to be evaluated.

1.1 Previous Works

For the image scrambling schemes, Arnold cat map [6], Fibonacci transformation [7], Baker map, sub-affine transformation, etc. can be seen as the wide usage. Some image scrambling evaluation methods are proposed [8,9] for testing these transformations. Specifically, Li [8] presented a measure for the image scrambling degree, which takes advantage of the gray level difference and information entropy. Yu et al. [9] used the correlation of pixels to analyze and evaluate the image scrambling.

1.2 Present Challenges

For the degree evaluation methods of the image scrambling, how to acquire the perfect result about scrambling schemes and how to analyze the 'weakness' about them in practice are very crucial. Although some image scrambling degree evaluation methods have been presented, the following four serious challenges about the evaluation system still should be considered:

- Due to the fact that when a plain-text is scrambled, not only the pixel positions are changed, but also the relationship of the pixels with adjacent pixels are completely disordered. That is why, both of the pixel values and pixel positions should be considered at the same time.
- The evaluation degree can reflect the relationship between a cipher image and the used scrambling scheme effectively, such as the relationship between a cipher image and the used time of iteration for a scrambling scheme.
- The evaluation of the scrambling degree should be independent of the plain image. That is to say, the scrambling degree of a cipher image is an objective value, not a "relative" value comparing with the plain image.
- As a digital image has large volumes of data, whether the proposed evaluation scheme can obtain the approximate evaluation with less data of an image comparing with the result using all of the image data.

1.3 Our Contribution

According to the analysis and the summarization of Subsection 1.2, our priority focus is to design an efficient and effective evaluation method that can measure

the scrambling degree of a cipher-image and can find the existing weakness in the image scrambling scheme. In this paper, a novel evaluation scheme based on the bit-plane has been proposed. We choose the 256-gray scale image as the test image, and the bit-plane theory as the core of our evaluation scheme. In the evaluation steps, the spatial distribution entropy and centroid difference for bit-planes are used for measuring scrambling degree of every bit-plane. Finally, the value of the scrambling degree is obtained according to the steps in Section 3.3. Note that for a general gray-scale image like 'Lena', as the relation between the original image and most significant bit-plane to least significant bit-plane reduces gradually, we can set a level decreasing-based weight for every bit-plane. In particular, as the last 4 least significant bit-planes have less relationship with the original image, instead of using the whole original image data, we can use the first 4 most significant bit-planes for the evaluation scheme. This will reduce 50% of the computation cost. The experimental results show (in Figs. 6 and 7) that the evaluation degree of a cipher image for the 4 significant bit-planes selection is nearly the same as that for the 8 bit-planes. Specially, from the experimental analysis (comparing with Fig. 6 and Fig. 7), the dividing size 32×32 in the evaluation scheme can produce a better evaluation degree than the size 16×16. As a result, the proposed scheme can evaluate the performance of the corresponding scrambling algorithm accurately and efficiently.

The rest of this paper is organized as follows: In Section 2, the corresponding knowledge of the bit-plane and the principle of this evaluation scheme are introduced. Section 3 presents an evaluation scheme based on the spatial distribution entropy and centroid difference of bit-planes. Simulation experiments and analysis about this evaluation method are provided in Section 4. Future works and conclusions are drawn in the last section.

2 Bit-Plane of a Digital Image and Its Application in Scrambling Evaluation

2.1 Bit-Plane of Gray Scale Image

Suppose that one pixel is located at the position (x, y) of a gray scale image. Let us denote the corresponding value of it by $P(x, y)$ which is the brightness intensity of that pixel. As the computer can only display discrete numbers, according to its representation precision, for a gray scale image, the brightness intensity of a pixel is divided into 256 parts and the intensity can take any value in the interval [0, 255]. As the computer only deals with the binary number, every pixel value is represented by an 8-bit binary stream, such as 127='01111111', namely, $127=0\times2^7+1\times2^6+1\times2^5+\cdots+1\times2^0$. The example presents the fact that for a general gray-scale image, which has some large texture areas, it can be represented as 8 bit-planes from the most significant bit-plane (MSB-P) to the least significant bit-plane (LSB-P). This is shown in Fig. 1 ('Lena' of size 128×128).

From Fig. 1, it can be seen that there is a different contribution to such a gray-scale image for each bit-plane. The impact can increase when the bit-plane is from LSB-P to MSB-P. The higher the bit-plane is, the stronger the

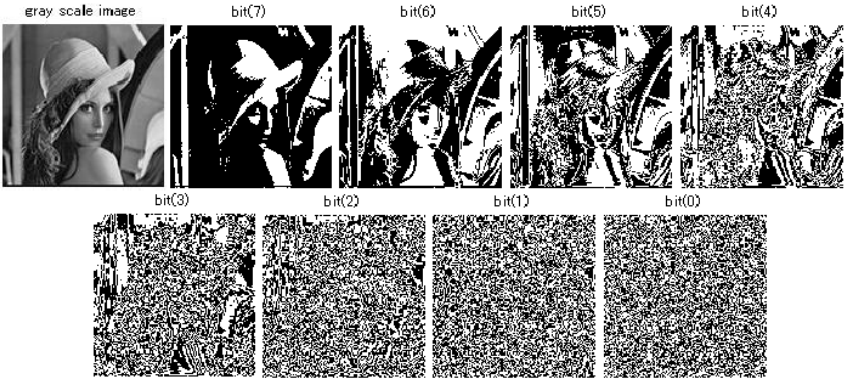


Fig. 1. 8 bit-planes of the gray scale 'Lena': from MSB-P to LSB-P

correlation between the bit-plane and the original gray scale image is. Especially, for this kind of general gray-scale image, we can see that the significant bit-planes portray the outline of an image which reflect the information of the original image. However, the less significant bit-planes look like the random noise.

Another important fact is that for the general gray-scale image which has some large texture areas, the relationship between two adjacent bit-planes does also increase for the higher bit-planes. For testing the correlation value between any two bit-planes, Theorem 1 is used, and the result is shown in Fig. 2.

Theorem 1. *Let, for an $M \times N$ gray-scale image, bit_i and $bit_i(x,y)$ denote respectively the ith bit-plane and the pixel value at position (x,y) in the ith bit-plane, $i = 0, 1, \ldots, 7$. Further let X and Y denote respectively the random variables corresponding to bit_i and bit_j, where $i \neq j \in \{0, 1, \ldots, 7\}$. The correlation coefficient $r(X, Y)$ can be expressed as:*

$$|r(X, Y)| = \left| \frac{p(X = 1, Y = 1) - p(X = 1)p(Y = 1)}{\sqrt{p(X = 1)p(X = 0)p(Y = 1)p(Y = 0)}} \right|. \tag{1}$$

where $p(\cdot)$ stands for the probability.

Proof: Note that X and Y can be described as follows:

$$X = \begin{cases} 1, & bit_i(x, y) = 1 \\ 0, & bit_i(x, y) = 0 \end{cases}, \qquad Y = \begin{cases} 1, & bit_j(x, y) = 1 \\ 0, & bit_j(x, y) = 0 \end{cases},$$

Let $E(X)$, $E(Y)$ and $E(XY)$ denote the expectations of X, Y and the joint distribution of X and Y, respectively. Further let, $D(X)$ and $D(Y)$ are the variances for X and Y, respectively. Note that

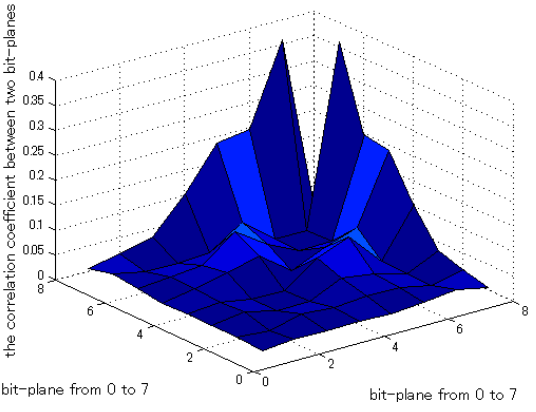


Fig. 2. Correlation coefficients between any two bit-planes of the gray-scale 'Lena'

$$E(X) = \frac{1}{M\times N}\sum_{k=0}^{M\times N} X_k = p(X=1); \quad E(Y) = \frac{1}{M\times N}\sum_{k=0}^{M\times N} Y_k = p(Y=1);$$

$$E(XY) = \frac{1}{M\times N}\sum_{k=0}^{M\times N} X_k Y_k = p(X=1,\, Y=1);$$

$$D(X) = \frac{1}{M\times N}\sum_{k=0}^{M\times N} [X_k - E(X)]^2$$
$$= p(X=0)[p(X=1)]^2 + p(X=1)[p(X=0)]^2 = p(X=1)p(X=0);$$

$$D(Y) = \frac{1}{M\times N}\sum_{k=0}^{M\times N} [Y_k - E(Y)]^2$$
$$= p(Y=0)[p(Y=1)]^2 + p(Y=1)[p(Y=0)]^2 = p(Y=1)p(Y=0);$$

$$\text{Consequently, } |r(X,\, Y)| = \left|\frac{E(XY)-E(X)E(Y)}{\sqrt{D(X)D(Y)}}\right| = \left|\frac{p(X=1,\, Y=1)-p(X=1)p(Y=1)}{\sqrt{p(X=1)p(X=0)p(Y=1)p(Y=0)}}\right|.$$

Note that using Eq. (1), the correlation coefficient of any two bit-planes can be calculated. Fig. 2 demonstrates that there is a relationship between any two bit-planes (for the convenience, the autocorrelation coefficient is set to 0). Particularly, for a general gray-scale image ('Lena'), the correlation coefficient of the 7^{th} bit-plane and the 8^{th} bit-plane is much higher than others. Moreover, we also can find that the correlation between the 8^{th} bit-plane and the 6^{th} or the 5^{th} bit-plane is also high. However, for the other pairs of bit-planes, the correlation coefficients are comparatively small and part of them are nearly equal to 0.

Meanwhile, for obtaining the relationship between such a gray-scale image and the corresponding bit-planes, the following test is introduced (Fig. 3).

In Fig. 3, it is shown that the contribution of each bit-plane for the original image is different. This means that if the bit-plane is the MSB-P, it has a strong relationship with the original image, and if the bit-plane is the LSB-P, the correlation is extremely small.

According to the above analysis on the bit-plane and the relationship between the original image and bit-plane, Definition 1 is used to present a correlation

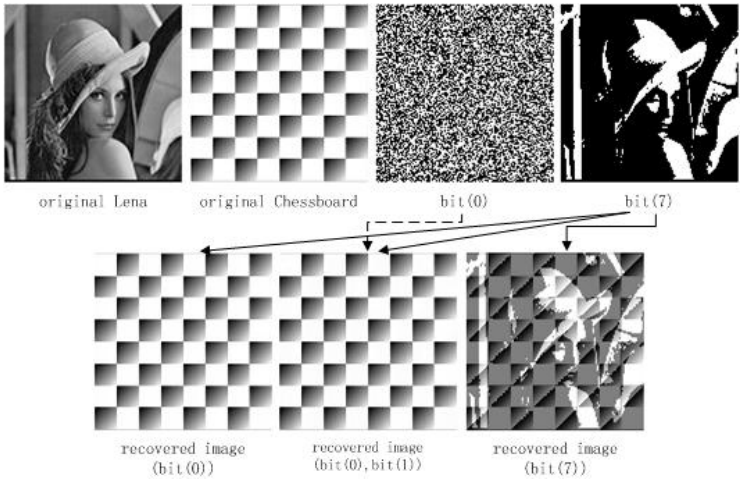


Fig. 3. Effect of the bit-plane for the original image

strength (CS) between the original gray-scale image and each bit-plane. It can be seen as a quantified relationship between them.

Definition 1. *For an $M \times N$ gray-scale image P, the correlation strength between P and bit(i), $i = 0, 1, \ldots, 7$ is expressed as: $CS(i)=Ibit(i)/255$, $Ibit(i) \in \{2^7, 2^6, 2^5, \ldots, 2^0\}$, where $CS(i)$ is the correlation strength, $Ibit(i)$ is the impact of each bit-plane (bit(i)) to the original gray-scale image. It satisfies that $\sum_{i=0}^{i=7} Ibit(i)=255$.*

In particular, the conclusion of Definition 1 is based on such a precondition that the contribution of each bit-plane to the original gray-scale image ($Ibit(i)$) is largely dependent on the plane coefficient 2^i, such as $Ibit(7)=1\times2^7+0\times2^6+0\times2^5 +\cdots+0\times2^0=128$. This precondition is accord with the test in Fig. 3.

2.2 Reasons for Using Bit-Plane Theory in Scrambling Evaluation

With the analysis and deduction discussed in Section 2.1, it is evident that the following two important and reasonable explanations may be used for demonstrating the reason why the bit-plane can be applied in the scrambling evaluation:

– Each bit-plane has an effect on the original gray-scale image. When the gray-scale image is divided into 8 bit-planes, if the operation acts on the 8 bit-planes, it can be seen as the processing for the original gray-scale image. Especially, according to Figs. 1 and 2, the observation and analysis can verify that for some general gray-scale images, the 8^{th} bit-plane and 7^{th} bit-plane are largely related to the original image, while the last 4 bit-planes have less relationship with the original image. Moreover, from Fig. 2, it can be concluded that the 6^{th} and 5^{th} bit-planes also have great relationship

with the 8^{th} bit-plane. Therefore, for some scrambled gray-scale images, the evaluation system can be constructed according to these 4 significant bit-planes, which are 50% data of an image, instead of all the 8 bit-planes.
– For every bit-plane, there is only 0 or 1 in each position of pixel. It can be seen as a binary image which is simple and easily analyzed by a scrambling evaluation. However, for a gray-scale image, as the value range of pixel is [0, 255], the calculation is large and the design of a scrambling evaluation method is hard, which may impact the final measurement result.

The above discussions show that the bit-plane division can be applied in the image evaluation and produce a good effect for measuring a scrambled image. Nevertheless, for each bit-plane, the evaluation should consider the relationship among pixels, deeply. Therefore, the following section introduces the spatial distribution entropy and centroid difference of each bit-plane for this purpose.

3 Details of Scrambling Evaluation Based on Bit-Plane

3.1 Spatial Distribution Entropy of Digital Image

In [10], Sun et al. proposed the spatial distribution entropy. For the image U, on the assumption that there are N kinds of pixel values, namely, B_1, B_2, B_3,..., B_N, let $\mathrm{S}_q=\{(x,\ y)|(x,\ y)\in U,\ P(x,\ y)\ =\ B_q,\ q\in[1,\ N]\}$. Then, the centroid of S_q is calculated, and some ring cycles are ensured according to the centroid and radius which are produced by the segmentation with an equal distance or unequal distance. Finally, for the pixel value B_q, the spatial distribution entropy can be expressed as:

$$E_q^s = -\sum_{j=1}^{k} p_{qj} \log_2(p_{qj}), \tag{2}$$

where s denotes that this entropy is the spatial distribution entropy, k is the number of ring cycles, here,$p_{qj}=|\mathrm{R}_{qj}|/|\mathrm{R}_i|$ is the probability density of B_q in ring cycle j, R_i is the number of B_q in S_q, and R_{qj} is the number of B_q in the ring cycle j. The details about spatial distribution entropy are available in [10].

For a binary image, as there are only two kinds of pixel values, namely, 0 and 1, the uncertainty of the pixel value is not quite important in our work. On the contrary, we are interested in the distribution of 0 and 1 in the bit-plane image. Based on this fact, the spatial distribution entropy is used for evaluating the scrambling distribution in our scheme. However, since the pixel distribution of a scrambled image can be regarded as the uniform distribution, and for the convenience of the calculation and post-processing, we can make use of the average partitioning which cuts the bit-plane into some rectangles (or square) with the same size ($m\times n$) instead of ring cycles for dividing the bit-plane. If the column (row) cannot be divided evenly by any other integer except 1, the image can add one or several row(s) (column(s)) with the pixels of the last row (column).

After the average partitioning, the spatial distribution entropy of each small block is calculated by Eq. (2). It should be noticed that k is the number of block,

p_{qj} is the probability density of q ($q \in \{0, 1\}$) in every block. Finally, we can obtain two spatial distribution entropys-${E_0}^s$ and ${E_1}^s$ for each bit-plane. For measuring the scrambling degree of each bit-plane, we take advantage of the first moment of the spatial distribution entropy (Eq. (3)) as one part of the scrambling degree, and find that the larger the first moment is, the better the effect of a scrambled bit-plane is.

$$\mu_g = \frac{1}{2}\sum_{q=0}^{1} E_q^s, \qquad g \in \{0, 1, 2, 3, 4, 5, 6, 7\}. \tag{3}$$

where μ_g is the corresponding first moment of the g^{th} bit-plane.

3.2 Centroid Difference of Bit-Plane

Centroid is a mathematics tool which is used in engineering application field. It can be seen as the average location of a system of a particles distribution, namely, the center of the quality for an object. In general, the centroid of a finite set of points $M_1(1,1)$, $M_2(1,2)$, $M_3(1,3)$,..., $M_k(x,y)$ in R^2 is: $C_X = \sum_{i=1}^{k} M_i X_i / \sum_{i=1}^{k} M_i$, $C_Y = \sum_{i=1}^{k} M_i Y_i / \sum_{i=1}^{k} M_i$, where $(C_X,\ C_Y)$ is the corresponding centroid. M_i is the quality in $M_i(x,\ y)$, and $(X_i,\ Y_i)$ is the location. If the quality of this finite set is uniform and the geometry is regular, the centroid is the geometric center.

For the bit-plane (generally speaking, the bit-plane is regular), we can suppose that each pixel can take the value 0 or 1. We refer the value of the pixel as the 'quality' of the pixel. After a digital image is scrambled, the location of '1' pixel and '0' pixel of every bit-plane can be changed and the distribution of them is disordered. According to this information, every bit-plane of an original image can be seen as having 'quality' with many '1' pixels, and the centroid is not located in the geometric center. For a scrambled image, the distribution of '1' pixels in every bit-plane is relatively uniform, which means that if the centroid of each bit-plane is calculated, it should be near to the geometric center in theory. For achieving the accurate coordinate of each bit-plane, in our scheme, the centroids of blocks using average partitioning (the same as the partitioning of the spatial distribution entropy) is calculated firstly, and then, they are applied to obtain the final centroid of each bit-plane.

For every bit block, the location of a centroid can be found according to the following formulas: $C_X^{rg} = \sum_{i=1}^{h} X_i / \sum_{i=1}^{h} n_i$, $C_Y^{rg} = \sum_{i=1}^{h} Y_i / \sum_{i=1}^{h} n_i$, where $(C_X^{rg},\ C_Y^{rg})$ is the location of the centroid in each block, n_i=1, h is the number of '1' pixels in the block, r means that this is the r^{th} block, $g \in \{0, 1, 2, 3, 4, 5, 6, 7\}$ demonstrates that which bit-plane the centroid belongs to.

For the computer, as the location $(x,\ y)$ of a pixel in each bit-plane is a discrete integer, and for making sure that the coordinate of a calculated centroid is not a decimal, the final centroid should be the nearest integral location, namely, $(\mathrm{C}_X^{rg})' = round(\mathrm{C}_X^{rg})$, $(\mathrm{C}_Y^{rg})' = round(\mathrm{C}_Y^{rg})$, where $round(\cdot)$ is a function to get the nearest integer.

For the final centroid of each bit-plane, the centroids of blocks are used according to Eq. (4). Especially, all of centroids of blocks have 'quality' which are equal to the amount of '1' pixels in one block.

$$C_X^g = \sum_{r=1}^{a}(\sum_{i=1}^{h} n_i(C_X^{rg})')_r \Big/ \sum_{r=1}^{a}(\sum_{i=1}^{h} n_i)_r; C_Y^g = \sum_{r=1}^{a}(\sum_{i=1}^{h} n_i(C_Y^{rg})')_r \Big/ \sum_{r=1}^{a}(\sum_{i=1}^{h} n_i)_r. \quad (4)$$

where a is the number of blocks in a bit-plane. (C_X^g, C_Y^g) is the centroid of the g^{th} bit-plane.

Based on the above preliminaries, the centroid difference of one bit-plane can be obtained with Eq. (5), which is another part of the scrambling degree.

$$diffva^g = \sqrt{(C_X^g - X_c^g)^2 + (C_Y^g - Y_c^g)^2}, \quad (5)$$

where $(X^g{}_c,\ Y^g{}_c)$ is the geometric center. $diffva^g$ is the centroid difference of the g^{th} bit-plane. The smaller the value of $diffva^g$ is, the better the effect of a scrambled bit-plane is.

3.3 Basic Steps of Scrambling Evaluation Scheme Based on Bit-Plane

According to the character of the bit-plane, as each pixel only takes two types values, namely, 0 and 1, the traditional scrambling evaluation scheme for analyzing the gray scale image based on the gray-scale value is not appropriate. In fact, we should focus on the location distribution of each pixel which can represent the scrambling degree of a bit-plane. The spatial distribution entropy and centroid difference can acquire this purpose effectively. They all can reflect the distribution condition of '0' pixel and '1' pixel in a bit-plane. Specially, since there is only 0 or 1 for every pixel, the calculation is not very large, which means that it is very suitable for practice applications. The details of the proposed scramble evaluation criterion are as follows:

- **Step 1:** Divide the scrambled gray scale image into 8 bit-planes, and take each bit-plane into the evaluation system. Particularly, taking performance into account, for the general image, only the first 4 bit-planes are sufficient for our proposed evaluation system. These bit-planes are the 8^{th}, 7^{th}, 6^{th} and 5^{th} bit-plane.
- **Step 2:** Calculate the spatial distribution entropy and centroid difference of each bit-plane according to the methods of Sections 3.1 and 3.2. For each bit-plane, μ_g and $diffva^g$ can be obtained respectively using Eqs. (3) and (5).
- **Step 3:** Evaluate the bit-plane with the scrambling degree. As the spatial distribution entropy is in the direct proportion to the scrambling degree, and the centroid difference is the opposite, the value of the scrambling degree of every bit-plane is determined by following Eq. (6) which also considers the normalization.

$$scraval^g = \frac{\mu_g \cdot \sqrt{(X_c^g)^2 + (Y_c^g)^2}}{diffva^g \cdot \log 2(2a)}; \qquad g \in \{0, 1, 2, 3, 4, 5, 6, 7\}, \quad (6)$$

where $scraval^g$ is the scrambling degree of one bit-plane. a is the number of blocks, $(X^g{}_c,\ Y^g{}_c)$ is corresponding geometric center.

- **Step 4:** Acquire the final value of the scrambling degree for a gray-scale image. Since there may be a different impact from each bit-plane to the original gray-scale image, the scrambling degree (*scraderee*) of an image should be the sum of all the 8 or 4 bit-planes with corresponding weights of the bit-planes, namely, $scraderee = \sum_{g=0/4}^{7} w(g) \cdot scraval^g$, $g \in \{0, 1, 2, 3, 4, 5, 6, 7\}$.
- **Step 5:** Finally, to remove the impact of the size of the image, we divide the value of the scrambled degree *scraderee* by the size of the image to get the final result: $Fscraderee \doteq scraderee/(M \times N)$.

The above **Step 1** to **Step 5** are the process of the proposed scrambling evaluation scheme. From these steps, it can be found that this method solves the present challenges of Section 1.2. Note that the weight ($w(g)$) in **Step 4** is significant for the final evaluation result. Consequently, two kinds of weights for 8 and 4 bit-planes are used in our proposal:

- As the most scrambling schemes carry out the encryption based on the gray-scale pixel, there is an effect from each bit-plane. If 8 bit-planes are used in this evaluation, the weight ($w(g)$) of each bit-plane ($\mathrm{bit}(i)$) is the corresponding $\mathrm{CS}(i)$ which is defined in Section 2.1.
- If only 4 bit-planes are applied in the proposed evaluation scheme, this means that the first 4 MSB-Ps have stronger relationship than the last 4 LSB-Ps to the original gray-scale image. The corresponding weights of the first 4 MSB-Ps is self-adaptive, which is decided by correlation coefficient $|r(X,Y)|$ in Section 2.1. The details are described in Eq. (7).

$$\begin{cases} w(7) + w(6) + w(5) + w(4) = 1 \\ w(6) = |r(6,7)| \times w(7) \\ w(5) = |r(5,7)| \times w(7) \\ w(4) = |r(4,7)| \times w(7) \end{cases}, \tag{7}$$

  where $w(g)$, $g \in \{7,6,5,4\}$ is the weight in **Step 4**, which is also the correlation measurement between the bit-plane and original gray-scale image.

4 Simulation Experiments and Analysis

4.1 Scrambling Strategy

Many digital image scrambling methods are referred in Section 1.1. Considering the convenience and fairness, Arnold cat map [6] and generalized Gray code [11], which are simple and the most useful, are applied to test the proposed scrambling evaluation method. As the pixels of a digital image are the discrete number, the generalized versions of two transformations are introduced. In particular, the binary-scale generalized Gray Code here is used for encrypting the coordinates of pixels. The corresponding matrix of this generalized Gray code comes from [11], which is an 8×8 matrix. For the consistency, the tests in our paper use the same control parameters for every image when scrambled by the above two maps. That is to say, for the generalized Arnold cat map, the corresponding

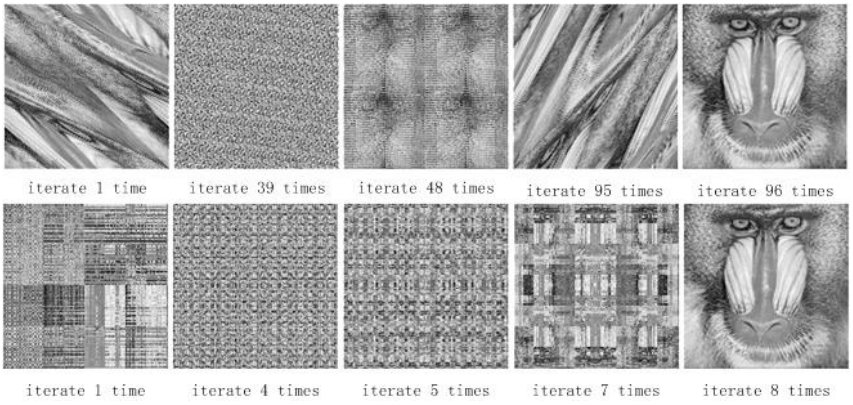


Fig. 4. Scrambled gray scale 'Baboon' using two kinds of transformations in different times of iterations: the results of the first row is from generalized Arnold cat map; the results of the second row is from generalized Gray code

transformation matrix is the same as the matrix A of Eq. (5) in [6]. Fig. 4 is some scrambling results of the above two scrambling transformations with the different number of times of the iterations (128×128 standard gray-scale 'Baboon').

From the above Fig. 4, some important information can be acquired. 96 and 8 are the periods of the generalized Arnold cat map and generalized Gray code (size: 128×128), respectively. When an image is encrypted by iterated many times, the visual effect of the scrambled image may not be 'good'. For the transformation with a large period, when it is iterated half times of the period, there is a "half-period phenomenon" in the corresponding cipher-image, which may leak the information of the original image. However, for the transformation with a short period, it seems that this phenomenon does not happen.

4.2 Scrambling Measuring

In order to test the effectiveness of the proposed scrambling evaluation method, two standard gray-scale images ('Baboon' and 'Boat') and two general images (we call them 'Internetgirl' and 'Landscape') of size 128×128 are chosen for

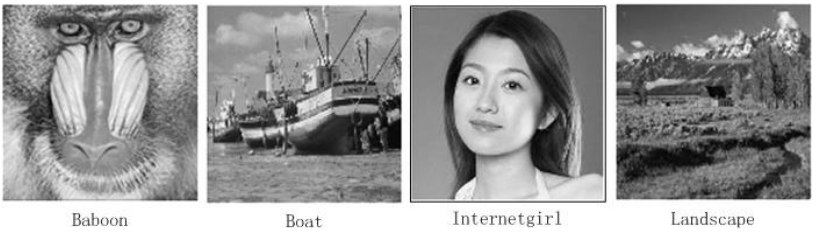


Fig. 5. Original gray-scale images ('Baboon', 'Boat', 'Internetgirl', 'Landscape')

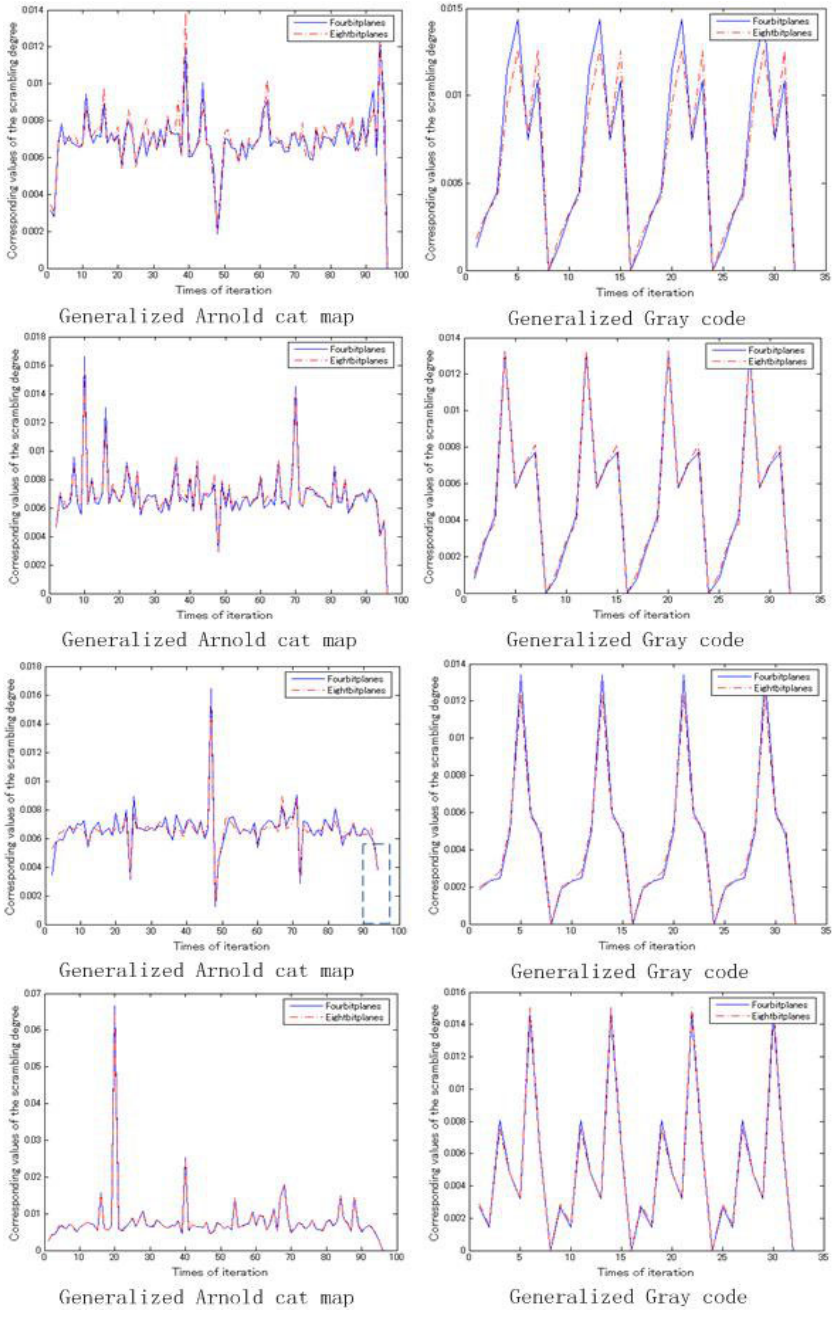


Fig. 6. Scrambling degree values for the used two transformations (The size of the block is 16×16): the first row is the corresponding results of 'Baboon'; the second row is the corresponding results of 'Boat'; the third row is the corresponding results of 'Internetgirl'; the fourth row is the corresponding results of 'Landscape'. (blue: the 4 bit-planes selection; red: the 8 bit-planes selection).

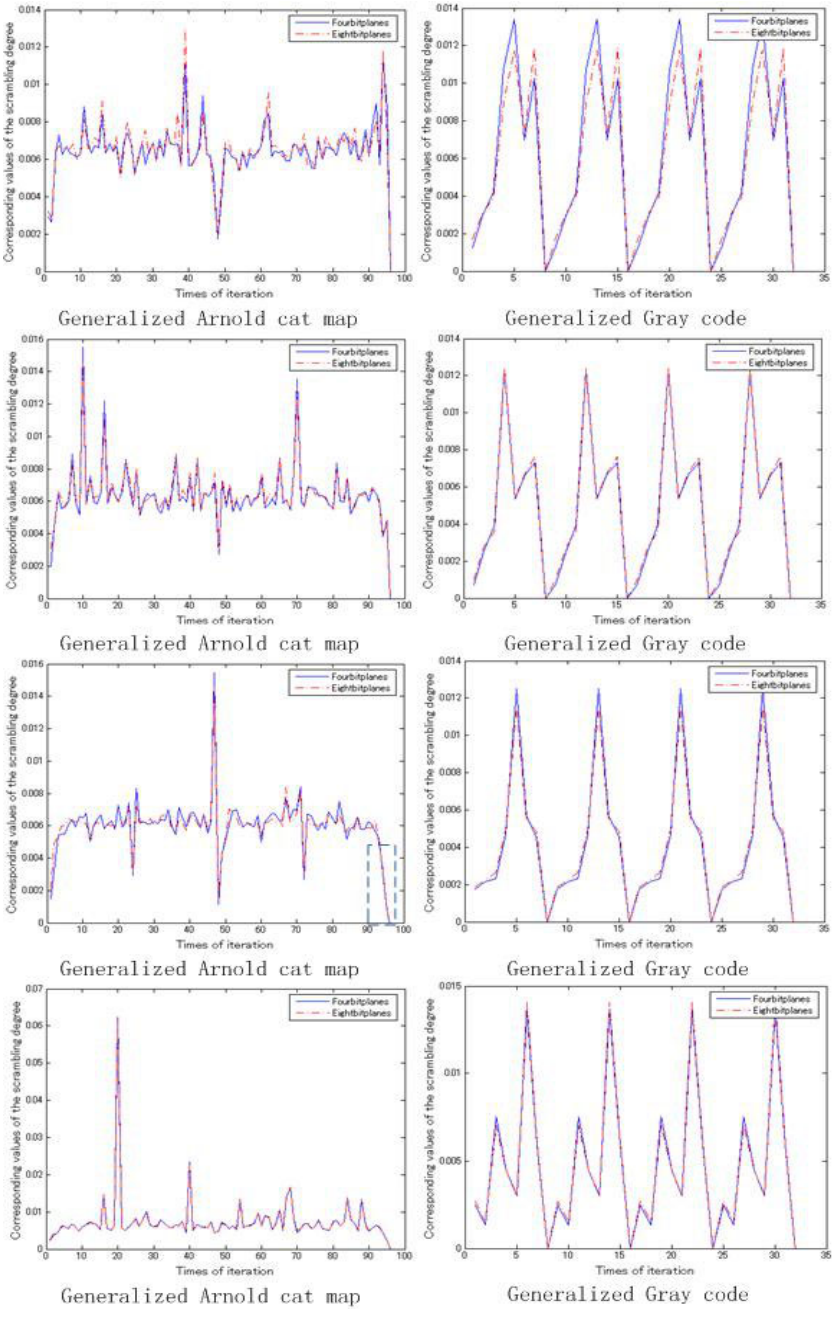


Fig. 7. Scrambling degree values for the used two transformations (The size of the block is 32×32): the first row is the corresponding results of 'Baboon'; the second row is the corresponding results of 'Boat'; the third row is the corresponding results of 'Internetgirl'; the fourth row is the corresponding results of 'Landscape'. (blue: the 4 bit-planes selection; red: the 8 bit-planes selection).

scrambling by the above generalized transformations (Fig. 5). For making a comparison about the size of the block dividing every bit-plane, 16×16 and 32×32 are set as this size in our tests. The weights for 8 bit-planes and 4 bit-planes, in these tests, are produced by the methods in Section 3.3.

Figs.6 and 7 are used for showing the scrambling degree values of two standard images and two general images encrypted by the above two transformations within 1 period and 4 periods, respectively (Fig. 6: the size of the block is 16×16; Fig. 7: the size of the block is 32×32). From these figures, the following facts come out: it is easily found that the proposed scrambling evaluation method basically reflects the effect of the scrambled images, which can be also used for revealing the performance of the corresponding scrambling method. Particularly, considering the transformation with a long period, the iteration time of the obvious "half-period phenomenon" can be appeared according to this proposed scheme. However, the generalized Gray code, which is a short period map, is not suitable for this "rule", which is also illustrated by the proposed scheme. From the above facts, it is testified that comparing with the results of evaluation methods in [8,9], our proposal can reflect the scrambling effect more precise.

Meanwhile, it can also be concluded that for these general gray-scale images, the result from the 4 bit-planes selection is similar to that of the 8 bit-planes selection. The reason is that the last 4 LSB-Ps have a less impact than the first 4 MSB-Ps on the original image. Therefore, if a gray-scale image like an image in Fig. 5 is encrypted by a scrambling method, we can make use of the first 4 MSB-Ps, instead of all the 8 bit-planes, to evaluate the degree of this scrambling method. In particular, from the comparison between Fig. 6 and Fig. 7, if the size of the block is 32×32, the evaluation results are intacter and better than those of the size 16×16. In Fig. 6, some evaluation results are set to 0 as when the size 16×16 is used, one or more block(s) in some bit-planes may be full of 0 or 1. However, this may not reflect the practical situation that the scrambled image has a 'bad' scrambling effect instead of no scrambling effect. Therefore, the size 32×32 is seen to be more suitable for the proposed evaluation scheme.

Moreover, according to Figs. 6 and 7, another merit of the proposed method can be obtained: the highest scrambling degree and the lowest scrambling degree can be calculated and shown obviously by using the proposed evaluation scheme, which is helpful for analyzing the optimal and 'weak' iteration time for encryption and watermarking. Specially, if the security of a combination cryptosystem/watermarking system is considered, the proposed evaluation algorithm is suitable for analyzing the first step of this cryptosystem/watermarking system: the scrambling step. Meanwhile, from the scrambling results shown in Figs. 6 and 7, these highest and lowest scrambling degrees, which can be also analyzed from Fig. 4, are in accordance with the observation of the visual from human.

5 Conclusions and Future Works

In this paper, for measuring and revealing the weakness of the scrambling method effectively, a novel scrambling evaluation method and the corresponding scheme

is proposed. Based on our design, the bit-plane dividing for the gray-scale image can be used as the first step of the proposed evaluation scheme. The spatial distribution entropy and centroid difference are introduced for measuring the scrambling degree of every bit-plane. The final result of the scrambling evaluation degree is the weighted sum of the 8 bit-planes or the first 4 bit-planes. The final experiment results demonstrate that for some gray-scale images, when the proposed scrambling evaluation system is used, not only the 8 bit-planes, but also the first 4 bit-planes can be applied to reflect the merit and disadvantage of the corresponding scrambling scheme effectively.

Moreover, If a scrambling scheme is used for encrypting RGB color image, the corresponding scrambling effect can also be evaluated effectively. The RGB color image is usually regarded as the integration of three primary color-components, namely, Red, Green and Blue. Since there are 256 possible values or intensities based on 8 bits for each color-component, the proposal for the gray-scale image can be used to evaluate each color-component of the RGB color image. Particularly, as the three color-components have a different correlation with the color image, the final scrambling degree of the color image should be the linear combination of the three color-components. It can be defined as follow:

$$Fscraderee_{color} = \\ 0.301 \times Fscraderee_R + 0.586 \times Fscraderee_G + 0.113 \times Fscraderee_B,$$

where 0.301, 0.586 and 0.113 are the corresponding weights from [12], $Fscraderee_{color}$ is the scrambling degree of the color image, $Fscraderee_R$, $Fscraderee_G$ and $Fscraderee_B$ are the scrambling degree of Red, Green and Blue components, respectively. The more details about the scrambling degree of the color image will be discussed in the future.

Nevertheless, two important factors which have an effect on our evaluation method should be considered in the future:

- The size of the average partitioning in the centroid difference: the size of the block in the average cutting is crucial and can make an impact on the final degree test. In our experiments, the size 16×16 and 32×32 are discussed. However, more analysis on how to choose the size reasonably for every bit-plane should be explored. The final selection can be fit for any kind of image.
- The weight $w(g)$ of the proposed evaluation system: for the proposed scrambling evaluation based on 8 bit-planes, CS(i) is considered as the weight of each bit-plane. However, if a bit based scrambling scheme is used to encrypt a gray-scale image, the same weight for each bit-plane, such as $w(0)$=$w(1)$=$w(2)$=$w(3)$=$w(4)$=$w(5)$=$w(6)$=$w(7)$=1/8, can also be used in the proposed evaluation scheme. The analysis needs to be developed for this kind of weight, and the comparison should do with the CS(i).

Acknowledgements

This research including the visit of Dr. Avishek Adhikari to Kyushu University is supported by Strategic Japanese-Indian Cooperative Programme on

Multidisciplinary Research Field, which combines Information and Communications Technology with Other Fields Supported by Japan Science and Technology Agency and Department of Science and Technology of the Government of India. The authors would like to thank Dr. Di Xiao, who is an associate professor of Chongqing University, China, for his helpful comments and suggestions.

References

1. Viile, V.D., Philips, W., Walle, V.D., Lemahieu, I.: Image scrambling without bandwidth expansion. IEEE Trans. Circuits Syst. Video Technol. 14, 892–897 (2004)
2. He, J.K., Han, F.L.: A pixel-based scrambling scheme for digital medical images protection. J. Network Comput. Appl. 32, 788–794 (2009)
3. Ye, R.S., Li, H.L.: A Novel Image Scrambling and Watermarking Scheme Based on Cellular Automata. In: 1st IEEE International Symposium on Electronic Commerce and Security, pp. 938–941. IEEE Press, Guangzhou (2008)
4. Zhu, L.H., Li, W.Z., Liao, L.J., Li, H.: A Novel Image Scrambling Algorithm for Digital Watermarking Based on Chaotic Sequences. Int. J. Comput. Sci. Network Secur. 6, 125–130 (2006)
5. Lin, K.T.: Information hiding based on binary encoding methods and pixel scrambling techniques. Appl. Opt. 49, 220–228 (2010)
6. Qi, D.X., Zou, J.C., Han, X.Y.: A new class of scrambling transformation and its application in the image information covering. Sci. Chin. E (China) 43, 304–312 (2000)
7. Zou, J.C., Ward, R.K.: Qi. D.X.: A new digital image scrambling method based on Fibonacci numbers. In: 16th IEEE International Symposium on Circuits and Systems, pp. 965–968. IEEE Press, Vancouver (2004)
8. Li, X.J.: A new measure of image scrambling degree based on grey level difference and information entropy. In: 3rd IEEE International Conference on Computational Intelligence and Security, pp. 350–354. IEEE Press, Suzhou (2008)
9. Yu, X.Y., Zhang, J., Ren, H.E., Li, S., Zhang, X.D.: A new measure method of image encryption. J. Phys.: Conf. Ser. 48, 408–411 (2006)
10. Sun, J.D., Ding, Z.G., Zhou, L.H.: Image retrieval based on image entropy and spatial distribution entropy. J. Infrared Millim. Waves. 24, 135–139 (2005) (in Chinese)
11. Zou, J.C., Li, G.F., Qi, D.X.: Generalized Gray code and its application in the scrambling technology of digital images. Appl. Math. J. Chinese Univ. Ser. A. 17, 363–370 (2002) (in Chinese)
12. Jiao, H.L., Chen, G.: A Color Image Fractal Compression Coding Method. Journal of Software 14, 864–868 (2003) (in Chinese)

Cryptanalysis on an Image Scrambling Encryption Scheme Based on Pixel Bit

Liang Zhao[1,2,⋆], Avishek Adhikari[3], Di Xiao[2], and Kouichi Sakurai[1]

[1] Graduate School of Information Science and Electrical Engineering
Kyushu University, Fukuoka, Japan 819-0395
zhaoliang@itslab.csce.kyushu-u.ac.jp, sakurai@csce.kyushu-u.ac.jp
[2] College of Computer Science
Chongqing University, Chongqing, China 400044
zhaoliangjapan@gmail.com, dixiao@cqu.edu.cn
[3] Department of Pure Mathematics
University of Calcutta, Kolkata, India 700019
avishek.adh@gmail.com

Abstract. Recently, an image scrambling encryption algorithm which makes use of one-dimensional chaos system for shuffling the pixel bits was proposed in [G.-D. Ye, Pattern Recognition Lett. 31(2010) 347-354]. Through the scrambling algorithm, the pixel locations and values can be encrypted at the same time. This scheme can be thought of as a typical binary image scrambling encryption considering the bit-plain of size $M\times 8N$. In [Li C.Q., Lo K. T., http://arxiv.org/PS_cache/arxiv/pdf/0912/0912.1918v2.pdf], Li et al. proposed an attack using more than $\lceil log_2(8MN\text{-}1)\rceil$ many known-plaintext images to recover the original plain image with the noise of size $M \times N$. The same principle is also suitable for the chosen-plaintext attack which can obtain the exact plain image. In the current paper, a simple attack on the original scheme is presented by applying chosen-plaintext images. Using our attack, the encryption vectors TM and TN and the decryption vectors TM' and TN' can be recovered completely. The experimental simulations on two standard images of size 128×128 and 256×256 justify our analysis. It is shown that the recovered images are identical with the corresponding original images. For both the original images, the number of chosen-plaintext images required in our scheme is 9, where as to do the same using the scheme proposed in Li et al.' attack, at least 17 and 19 chosen-plaintext images there will be required respectively. Moreover, the some method can be also used for chosen-ciphertext attack which reveals the decryption vectors TM' and TN' directly. Note that our attacks are also successful under iteration system which is remarked in the conclusions.

Keywords: Chosen-plaintext attack, Chosen-ciphertext attack, Logistic chaos map.

⋆ The first author of this research Liang Zhao is supported by the governmental scholarship from China Scholarship Council.

H.-J. Kim, Y. Shi, and M. Barni (Eds.): IWDW 2010, LNCS 6526, pp. 45–59, 2011.

1 Introduction

1.1 Background Presentation

Security protection techniques of digital image, such as image authentication [1-3] or image hash [4-6], are always significant facet of multimedia data study with the popularization and development of computer science and internet technology. Note that the inherent characteristics that include massive volumes of data, high correlations among adjacent pixels and so on make the digital image different from other kind of data, such as text. Nowadays, many image encryption methods [7-11] have been proposed. For example, Chung et al. [7] presented an image encryption scheme for encrypting the binary image based on the modified SCAN language. Chen et al. [8] proposed a generalized three-dimensional Arnold cat map, and used it in a symmetrical image encryption algorithm based on chaos. Guan et al. [9] introduced a fast image encryption design according to the character of hyper-chaos. Tong et al. [10] designed a new compound two-dimensional chaotic function, and a novel image encryption scheme is introduced, which is based on this new compound chaos. Along with the constructions of the image encryption schemes, the security analysis or cryptanalysis on the image encryption methods have also been developed. Unfortunately, according to the analysis, many image encryption schemes are not enough secure and cannot be used in practice [12-17]. Particularly, some classical attacks, such as chosen-plaintext attack and know-plaintext attack, are very effective to analyze the image encryption systems [14,15,18]. The more discussions about security of digital image that readers can get are based on some recent surveys [19-21]. In [22], several useful rules about how to evaluate the security of cryptosystems based on chaos are presented, and in [23], a quantitative cryptanalysis on the performance of permutation-only multimedia ciphers against plaintext-based attacks is performed.

1.2 Our Contribution

Recently, G.-D. Ye [24] proposed one type of image encryption scheme for shuffling all the bits of pixels using one-dimensional chaos system. The proposed encryption method mainly possesses two characters which are different with some traditional schemes, such as [8-10]: The first one is that scrambling of pixels is not considered from gray level [0, 255] but from bit-plane level $\{0, 1\}$, which can bring into a good pixel-value encryption. The second one is about encryption process that only row and column exchange are used for encrypting the pixels' values and locations at the same time. Meanwhile, the proposed structure makes the encryption process be similar to the decryption process, which is convenient for the practice application.

In [25], Li et al. have given a kind of the known-plaintext attack and chosen-plaintext attack to the original scheme [24]. The plain image which has some noise points can be recovered if more than $\lceil log_2(8MN) \rceil$ (M and N are the size of an image) known-plaintext images are used in the first attack. Meanwhile, for the

chosen-plaintext attack, the requisite number of chosen-plaintext images, which are used for recovering the exact plain image, is also at least $3+\lceil log_2(MN) \rceil$. The ideas of both attacks are nearly the same, which involve constructing a multi branch tree for recovering the permutation matrix W. This paper studies the security of the original scheme and reports that this encryption method is not secure for the practice application, firstly. Then, according to the corresponding analysis, a particular type of chosen-plaintext attack/chosen-ciphertext attack is applied for completely breaking the original scrambling algorithm proposed by Ye. Concretely speaking, for the chosen-plaintext attack, two kinds of plain images are taken advantage of revealing the encryption vectors TM and TN, respectively. After TM and TN are revealed, the original plain image can be acquired easily. Particularly, the number of chosen-plaintext image is not decided by the size of M and N, but determined by the comparison between M and N, which is different with the attack method in [25]. That is to say, a cipher image with the size of 128×128 and one with the size of 256×256 can be recovered with the same number of chosen-plaintext image. For the chosen-ciphertext attack, the same way is suitable for achieving the decryption vectors TM' and TN', which are the inverse vectors of TM and TN. For these attacks, the recovered images are all the same as the original image which is not go through the encryption.

The outline of this paper is as follow. In Section 2, the brief introduction of the original scrambling algorithm is presented and some typical attacks are listed. Section 3 indicates the drawbacks of the original scheme and demonstrates how to implement the chosen-plaintext attack/chosen-ciphertext attack. In Section 4, some simulation examples are used to illustrate the success of our attacks. The concluding remarks are drawn in the last section.

2 Description of Original Scrambling Encryption Scheme and Four Classical Attacks

2.1 Steps of Original Scrambling Encryption Scheme

The details of this image scrambling scheme [24] on a gray scale image of size $M \times N$ is described as follows:

- The original gray-scale image of size $M \times N$ can be considered as an $M \times N$ matrix, say P, with entries, denoting the pixel values, chosen from the set $\{0, 1, 2, \ldots, 255\}$.
- Every pixel value in the matrix P should be transformed to an 8-bit sequence. Then, these bit sequences are connected and integrated into a $M\times8N$ binary matrix P^t. Especially, the gray pixels can be transformed into bit pixels according to the following Eq.(1) [24]:

$$p^t(i,j) = \begin{cases} 1 & if\ (P(i,j)/2^t) \bmod 2 = 1 \\ 0 & others \end{cases} \tag{1}$$

  where $P(i,j)$ denotes the gray-scale pixel in the ith row and jth column of the original image and $p^t(i,j)$ denotes 0/1 after transforming $P(i,j)$ with 2^t, $t = 0, 1, 2, \ldots, 7$.

- The one-dimensional Logistic chaos map $x_{n+1}=\mu x_n(1-x_n)$ (μ is the system parameter, x_n is the iteration value) is used for producing the scrambling vectors: TM and TN, where two iteration value sequences $\{x_{m+1}, x_{m+2}, x_{m+3},\ldots, x_{m+M}\}$ and $\{x_{n+1}, x_{n+2}, x_{n+3},\ldots, x_{n+N}\}$ are generated firstly, and then sorted for marking down the transform position $TM=(t_1, t_2,\ldots, t_{M)}$ and $TN=(t'_1, t'_2,\ldots, t'_N)$.
- The binary matrix P^t is encrypted by the scrambling vectors TM and TN which shuffle every row and column of P^t. The corresponding cipher matrix $D=[TM^{TR}\times P^t\times TN^{TR}]_{M,8N}$, ($TM^{TR}$ and TN^{TR} are the corresponding transformation matrices of TM and TN) is obtained after this process.
- The decimal matrix C is acquired when the matrix D is recovered by Eq.(2):

$$C(i,j)=\sum_{t=0}^{7}2^t\times d^t(i,j), \tag{2}$$

  where $d^t(i,j)$ is the bit pixel in the ith row and jth column of the cipher matrix D. $C(i,j)$ is the corresponding gray-scale pixel.
- This scrambling encryption process is completed, and the decimal matrix $C=[C(i, j)]_{M,N}$ is the corresponding cipher image.

For the decryption scheme, as the cipher matrix D can be transformed back to the binary matrix P^t by only exchanging every row and column with the inverse vectors (TM' and TN') of the scrambling vectors TM and TN, the whole process is almost the same as the encryption method except for the use of the scrambling vectors. Therefore, this image encryption algorithm can be seen as a "half-symmetry" encryption scheme, which is in favor of the hardware application. Fig. 1 is a simulation result about the original scheme (gray-scale image "Baboon" of size 128×128, the secret keys and parameters are x_0=0.2009, μ=3.98, m=20, n=51), and the corresponding histogram about the cipher image is also listed. For obtaining more details, the readers can refer to [24].

2.2 Introduction of Four Classical Attacks

Based on *Kerckhoffs' principle* [26], when cryptanalyzing a cryptosystem, a general assumption is that cryptanalyst can acquire the information on the design and working of the studied cryptosystem. That is to say, for any researcher,

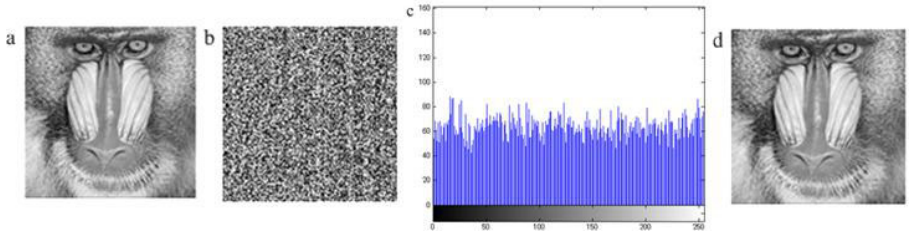


Fig. 1. Encryption and decryption effect on the original scrambling encryption scheme

he/she can know everything about the cryptosystem except the secret keys for encryption and decryption. This criterion is a basic standard for any encryption system in nowadays' secure communications networks. Therefore, based on the above principle, four typical attacks [27] are listed as follows, and the difficulty level is from the hardest attack to the easiest one:

- Ciphertext Only Attack: the opponent only possesses a string of ciphertext.
- Known-plaintext Attack: the opponent can possesses a string of plaintext M, and the corresponding ciphertext string C.
- Chosen-Plaintext Attack: the opponent has obtained temporary access to the encryption machinery. Hence he/she can choose a plaintext string M, and construct the corresponding ciphertext string C.
- Chosen-Ciphertext Attack: the opponent has obtained temporary access to the decryption machinery. Hence he/she can choose a ciphertext string C, and construct the corresponding plaintext string M.

For the above four attacks, the main purpose of opponent is to recover the secret keys which are used in a cryptosystem. In this paper, two kinds of attacks, namely, the chosen-plaintext attack and chosen-ciphertext attack, are proposed to reveal the "true" keys used in [24].

3 Drawbacks of Original Scrambling Encryption Scheme under Study and Corresponding Attacks

3.1 Drawbacks of Original Scrambling Encryption Scheme

The original encryption scheme is a novel image scrambling algorithm if considered from the bit-plane based encryption. It can carry out the scrambling for the pixels' values and locations simultaneously. As the symmetric of the design structure, the encryption and decryption is nearly the same process except the encryption and decryption vectors. Consequently, this scheme, comparing with the reversible encryption scheme, is more suitable for the hardware application. However, three potential drawbacks consist in the original scheme, and have a significant effect on the security of the cipher image:

- According to the description of the original image scrambling scheme, for every bit in the binary matrix P^t, it only implements the row and column exchange respectively (Fig. 2), which means that location $\{(x,\ y)|x{=}i,\ i{\in}[0,\ M];\ y{=}k,\ k{\in}[0,\ 8N]\}$ of every original-bit in each column maps to the location $\{(x^*,\ y)|x^*{=}j,\ j{\in}[0,\ M];\ y{=}k,\ k{\in}[0,\ 8N]\}$ firstly, and then to the location $\{(x^*,\ y^*)|x^*{=}j,\ j{\in}[0,\ M];\ y^*{=}e,\ e{\in}[0,\ 8N]\}$. Therefore, the transformation range of a bit is confined into a narrow space which consists of one row and one column. From another aspect, for every gray-scale pixel, the original scrambling scheme is also not a perfect one, as the row transformation (TM) only complies the column permutation of the pixel location, and the column transformation (TN) can be seen as an encryption for the gray value in one row.

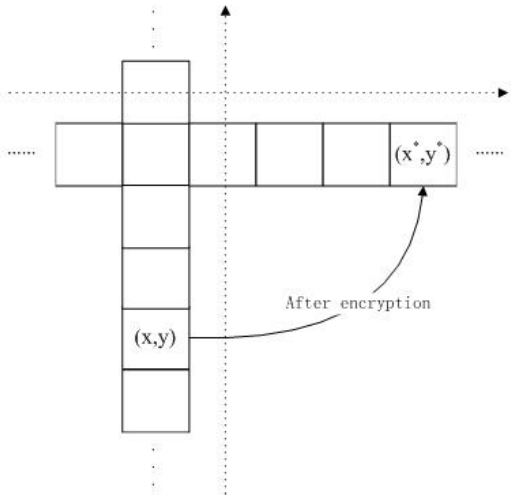


Fig. 2. Transformation of a bit pixel in one binary matrix

- The second obvious problem is that the position exchange of every bit only depends on the produced secret keys, but is not related to other information of an image, such as the pixel value. From the steps of the original scheme, the process of the bit pixel encryption is only decided by the scrambling vectors (TM and TN): C=E$_{Ye}(I, TM, TN)$, which demonstrates that if TM and TN have been found, the rule for scrambling the image (E$_{Ye}(\cdot)$) is also been presented, and the plain image I is easily obtained.
- In the original scheme, for achieving the fast encryption and decreasing the computation complexity, only two scrambling vectors (TM and TN) are used for encrypting each row and column of the corresponding binary matrix P^t of size M×8N. According to the encryption process, at first, P^t is encrypted by TM : $TM \times P^t(i, 8N)_M \rightarrow P^t(i, 8N)'_M$. For every bit in one row i of $P^t(i, 8N)'_M$, the exchange rule is the same. Then $(P^t)'$ is encrypted by TN : $(P^t)'(M, j)_{8N} \times TN \rightarrow \mathcal{P}(M, j)_{8N}$. For every bit in one column j of $\mathcal{P}(M, j)_{8N}$, the exchange rule is also the same. On one hand, this method can enhance the speed assuredly, as only two vectors are used in whole process. On the other hand, this can also let down the secure of the scrambling encryption. The reason is that when the rule of one row (column) is cracked, the exchange method of the whole columns (rows), which can be applied by opponents, are found.

3.2 Corresponding Attacks to Original Scrambling Encryption Scheme under Study

From the analysis of the drawbacks, especially, from the second drawback, as described in Section 3.1, it is found that if the scrambling vectors TM and TN,

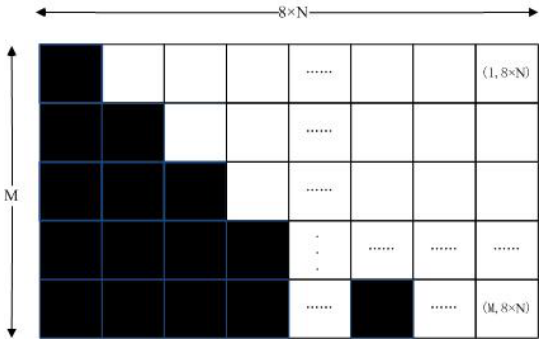


Fig. 3. Example of the plain image for recovering the row scrambling vector TM

which are considered as the "true" secret keys, can be revealed, the cipher image C can be decrypted apparently. That is to say, when we attack the original scheme, the used one-dimensional chaos system can be isolated, and we should directly find a way to obtain TM and TN. Meanwhile, the first and third drawbacks present the fact that since the x-coordinate and y-coordinate of each pixel bit are encrypted independently, and the same encryption rule is carried out, there may exist some especial images for searching these "true" secret keys in the used image cryptosystem. Therefore, our purpose is to find some particular and simple plain images for revealing TM and TN. In cryptanalysis, it is called chosen-plaintext attack. Meanwhile, as there are two scrambling vectors: TM and TN which are used for scrambling the row and column, for each one, there should be at least one special plain image for acquiring it. Fortunately, such plain images do exist, and can perform the attack very well, such as Fig. 3 (for recovering the row scrambling vector TM). The process of the chosen-plaintext attack is as follow:

– On the assumption that the used encryption machinery has been acquired provisionally. According to the size of the cipher image C ($M\times N$), we construct the chosen-plaintext images (RCM, RCN_{s+1} (s$\in${0,1,2,3,4,5,6,7,. . ., m}, $s\geq 7$)) for the chosen-plaintext attack with the same size. The details of the constitution principle are as follows:

  - $M\leq 8N$:
    The constitution method for revealing the row scrambling vector TM is described as "Algorithm 1", and the constitution method for revealing the column scrambling vector TM is described as "Algorithm 2".
    When $N<M\leq 8N$, the need of different plain images for revealing the vector TN is decided by the size of image, which demonstrates that the number of RCN_{s+1} only need to meet $RCN_{s+1}>2$. However, we set that the number of RCN_{s+1} must meet $RCN_{s+1}\geq 8$.

Algorithm 1. For revealing the row scrambling vector TM

```
Step1
for i = 1 to M do
    for j = 1 to N do
        RCM(i, j) = 255;
    end for
end for
Step2
for i = 1 to M do
    for j = 1 to N do
        for g = 8(j − 1) + 1 to 8j do
            RCM'(i, g) = 1 ⇐ [RCM(i, j) ← Eq.(1)];
        end for
    end for
end for
Step3
for p = 1 to M do
    {e(k)|e(k) ⊆ {1, 2, 3, . . . , 8N}} ←Choose any k many y-coordinate(s) in
    {1, 2, 3, . . . , 8N};
    for u = 1 to k do
        RCM'(p, e(u)) ← 0;
    end for
end for
Step4
for i = 1 to M do
    for j = 1 to N do
        for g = 8(j − 1) + 1 to 8j do
            t ⇐ [RCM'(i, g) ← Eq.(2)];
        end for
        RCM(i, j) ← t;
    end for
end for
```

- M>$8N$:
  The conformation method of the used plain images is the same as the condition of M≤$8N$. However, the size should be exchanged at first. That is to say, $M^{'}$=$8N$ and $(8N)^{'}$=M are used for setting the plain images $RCM^{'}$ and $RCN^{'}{}_{t}$ (t∈ {0,1,. . .,m}, t≥1) respectively. This can be seen in Fig. 4.

– Attack on TM:
The chosen-plaintext image RCM is used in this temporary encryption machinery (M≤$8N$):

$$RCMM = E_{Ye}(RCM, TM, TN) = EQTWO([TM^{TR} \times EQONE(RCM) \times TN^{TR}]_{M,8N})_{M,N}, \quad (3)$$

where $EQONE(\cdot)$ means Eq. (1), and $EQTWO(\cdot)$ indicates Eq. (2) in the above expression.
Eq. (1) is utilized for the cipher image $RCMM$ to get $RCMM^{'}$ of size M×$8N$:

$$RCMM' \Leftarrow [RCMM \leftarrow Eq.(1)], \quad (4)$$

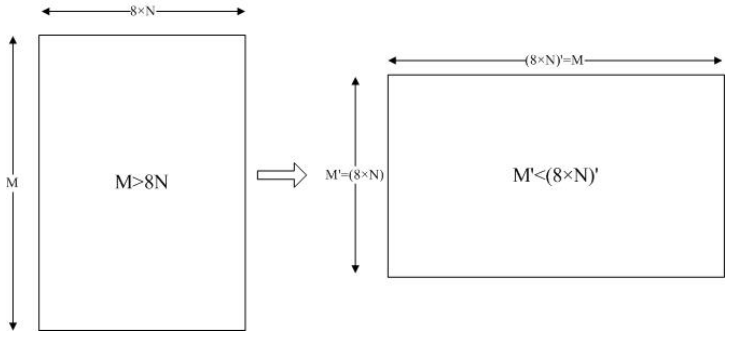


Fig. 4. Size exchanging before the chosen-plaintext attack when $M>8N$

As the number of "0" in every row is not changed during the encryption process, the number of "0" in each row of a cipher image can be counted:

$$TM(x) = \sum_{y=0}^{8N-1} ZL(RCMM'(y)), \tag{5}$$

where $ZL(\cdot)$ is a function for counting the number of "0", $x \in [0, M]$.
The result is the encryption vector sequence TM, and for the decryption, the inversion of TM is the real key.

– Attack on TN:
The attack method on TN is similar to the way on TM. However, there are some differences for revealing TN:
  - As the number of RCN_{s+1} is more than 8, the attack process should be repeated at least 8 times for obtaining TN.
  - We should decide that whether this column has "0" pixel: if this column has, we should count the number of "0" pixel. Otherwise, this column can be ignored. For the column which has "0" pixel, the number of "0" pixel (r) is counted in this column, and the corresponding value in the vector TN is $(r+s\times N)$.

When $M>8N$, the attack process on TM and TN is an exchange as $M\leq 8N$: RCM is used for revealing TN, and RCN_{s+1} is used for revealing TM. As this decryption process is the same as the encryption procedure except the "keys" for the decryption, which are the inversion vectors of the encryption vectors TM and TN, the chosen-ciphertext attack can be also applied for revealing the "true" decryption keys TM' and TN', not considering the used one-dimensional chaos system. The crack procedure and the used chosen-ciphertext images (RCM' and RCN'_{s+1}) are identical with the chosen-plaintext attack. That is to say, the chosen-plaintext images RCM and RCN_{s+1} are considered as chosen-ciphertext images RCM' and RCN'_{s+1} in the process of decryption analysis for acquiring the vectors TM' and TN'. If the decryption vectors TM' and TN' are found,

Algorithm 2. For revealing the column scrambling vector TN

for $L = 1$ to $s + 1$ **do**
  // $s \in \{0, 1, \ldots, 6, 7, \ldots, m\}$, $s \geq 7$, $m \in \mathbb{Z}$
  Step1
  for $i = 1$ to M **do**
    for $j = 1$ to N **do**
      $RCN_L(i, j) = 255$;
    end for
  end for
  Step2
  for $i = 1$ to M **do**
    for $j = 1$ to N **do**
      for $g = 8(j - 1) + 1$ to $8j$ **do**
        $RCN'_L(i, g) = 1 \Leftarrow [RCN_L(i, j) \leftarrow Eq.(1)]$;
      end for
    end for
  end for
  Step3
  $Q \leftarrow min(M, N)$
  if $(8N - L \times M) < min(M, N)$ **then**
    $D \leftarrow (8N - L \times M)$;
  else
    $D \leftarrow min(N, M)$
  end if
  for $k = (L - 1) \times Q + 1$ to $(L - 1) \times Q + D$ **do**
    $\{e(\omega)|e(\omega) \subseteq \{1, 2, 3, \ldots, M\}\} \leftarrow$ Choose any ω many x-coordinate(s) in $\{1, 2, 3, \ldots, M\}$ // $\omega \in \{1, 2, \ldots, D\}$;
    for $u = 1$ to ω **do**
      $RCN'_L(e(u), k) \leftarrow 0$;
    end for
  end for
  Step4
  for $i = 1$ to M **do**
    for $j = 1$ to N **do**
      for $g = 8(j - 1) + 1$ to $8j$ **do**
        $t \Leftarrow [RCN'_L(i, g) \leftarrow Eq.(2)]$;
      end for
      $RCN_L(i, j) \leftarrow t$;
    end for
  end for
end for

they can be used for recovering the needful cipher image directly. The detailed steps of the chosen-ciphertext attack are similar to the above chosen-plaintext attack.

4 Simulation Experiments on Our Proposed Attacks

In this section, some experiments are listed for illustrating the proposed chosen-plaintext attack/chosen-ciphertext attack. For demonstrating the proposed attacks sufficiently, the standard images "Lena" and "Cameraman" of size 128×128 and 256×256 (Fig. 5(a) and (d)) are used in the original scrambling scheme. The whole simulations are performed under the MATLAB program running on AMD Turion(tm) Neo X2 Dual Core Processor L625 1.60GHz with 2 GB RAM. The

Fig. 5. Test images for the proposed attacks: (a-c) plain image, cipher image and decrypted image of "Lena";(d-f) plain image, cipher image and decrypted image of "Cameraman"

secrets keys and corresponding parameters are chosen from the original example in [25]: x_0=0.2009, μ=3.98, m=20, n=51.

For the proposed chosen-plaintext attack, the attacker can choose 9 images, namely, RCM and RCN_{s+1} ($s\in\{0,1,2,3,4,5,6,7,\ldots, m\}$, $s\geq 7$; m=8 in these examples), as the used plain images which are applied for revealing the encryption vectors TM and TN, respectively. Particularly, RCM is used for obtaining the vector sequence TM, and RCN_{s+1} are utilized for getting the vector sequence TN. The attack results are as follows:

For the attack process, it must note that the used plain images a(d) and b(e) are used for revealing TM and TN, respectively. In particular, as an example, for the chosen-plaintext images a and d, the black pixel values belong to {254, 252, 248, 240, 224, 192, 128, 0}, and the white pixels only signify 255, such as $(n\times n)=(19\times 19)$:

$$RCM = \begin{bmatrix} 254 & 255 & 255 & \cdots & 255 & 255 \\ 252 & 255 & 255 & \cdots & 255 & 255 \\ 248 & 255 & 255 & \cdots & 255 & 255 \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & 252 & \cdots & 255 & 255 \\ 0 & 0 & 248 & \cdots & 255 & 255 \end{bmatrix}.$$

In the whole attack, only 9 chosen-plaintext images are applied for revealing TM and TN. This number is less than the number of the used plain images in Li et al.' attack [25] when the size M and N are very large:

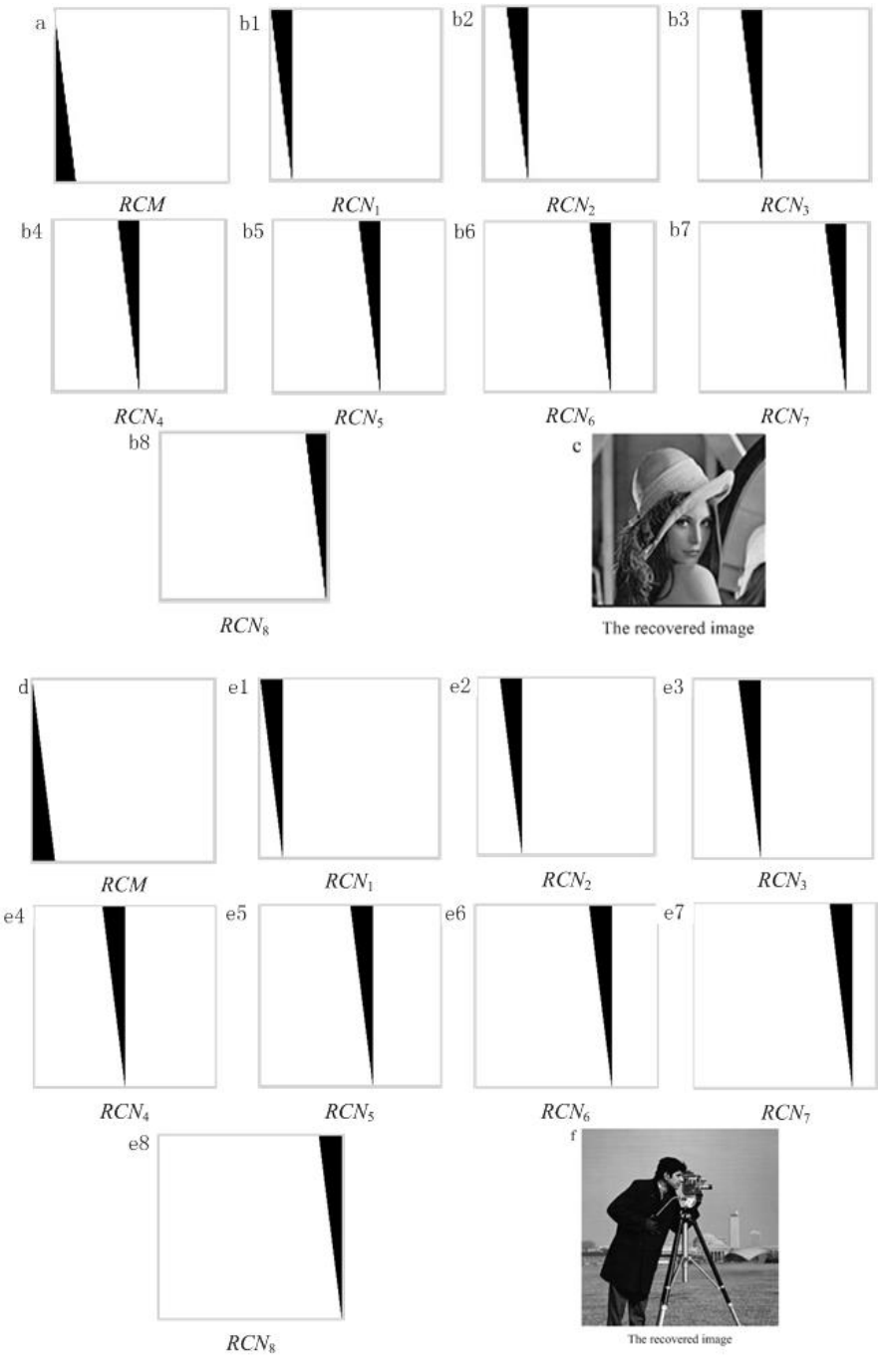


Fig. 6. Chosen-plaintext attack to the cipher images b and e in Fig. 5: (a, d) plain images for revealing *TM*; (b, e) plain images for revealing *TN*; (c, f) recovered images of "Lena" and "Cameraman"

$(TM, TN) = \sum ZL(EQONE((E_{Ye}(RCM, TM, TN), E_{Ye}((RCN_1, RCN_2, RCN_3, RCN_4, RCN_5, RCN_6, RCN_7, RCN_8), TM, TN))))$.

After that, the decryption vector sequences TM' and TN' are the inversion of TM and TN:

$$(TM', TN') = INV(TM, TN), \tag{6}$$

in which $INV(\cdot)$ is a function of the inversion operation. Then, the original plain image can be acquired successfully.

For the chosen-ciphertext attack, the process is similar to the chosen-plaintext attack. The chosen-ciphertext images are the same as the chosen-plaintext images, which means that RCM and RCN_{s+1} can be seen as the used cipher images (RCM' and RCN'_{s+1}). Meanwhile, the revealed vector sequences are the decryption keys which can be used for decrypting the needful cipher image immediately.

5 Conclusions

In the present paper, the plaintext-based attack was presented to break a recent image scrambling scheme based on chaos. The analysis reveals that the proposed scrambling scheme in [24] is insecure against the chosen-plaintext attack/chosen-ciphertext attack. Particularly, in our test, 9 plain images/cipher images are used for revealing the necessarily "true" secret vectors TM and TN if the image size M and N are the same. The experiment results demonstrate that our attacks can successfully acquire the secret keys (TM, TN) and (TM', TN') for decrypting the cipher image.

Let us point out some comments on our proposed attacks.

- The chaos system used in the scrambling algorithm of [24]: Logistic map is a simple chaos map which is used for producing the random sequence in the original scheme. However, in [28], it is shown that Logistic map cannot be considered as a good random number generator. In our opinion, whether the chaos system is the Logistic map or not is not important for the whole encryption process. Since the transformation vector sequences TM and TN which are produced by a chaos system can be revealed by our attacks, no matter which chaos system is applied.
- The iteration encryption in the scrambling algorithm of [24]: The original scrambling scheme can be iterated many times for ensuring security. This character is presented in Fig. 5 of [24]. However, the iteration encryption of this scheme is the same as the multiplication of many matrixes, as follows:

$$\left(\begin{bmatrix}0&1&0\\0&0&1\\1&0&0\end{bmatrix}\times\cdots\times\begin{bmatrix}0&0&1\\0&1&0\\1&0&0\end{bmatrix}\right)\times\begin{bmatrix}127&46&2\\6&254&78\\89&48&24\end{bmatrix}\times\left(\begin{bmatrix}1&0&0\\0&0&1\\0&1&0\end{bmatrix}\times\cdots\times\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}\right). \tag{7}$$

  Our attacks can effectively obtain the final product of many matrixes on the left (right) side of the original image, which makes the iteration encryption ineffective. The products of the matrixes can also be seen as the "secret keys" for recovering the original plain image.

– The used image in our attack: The chosen-plaintext image/chosen-ciphertext images RCM and RCN_{s+1} are not unique. e.g., if the matrix A as defined bellow is a chosen-plaintext image, then its complement image A^c can also play the role of a chosen-plaintext image, where A and A^c are defined bellow:

$$A = \begin{bmatrix} 254\ 255\ 255 \\ 252\ 255\ 255 \\ 248\ 255\ 255 \end{bmatrix} \Rightarrow A^c = \begin{bmatrix} 1\ 0\ 0 \\ 3\ 0\ 0 \\ 7\ 0\ 0 \end{bmatrix}. \tag{8}$$

Finally, as a conclusion we can say from the security analysis that the original scrambling scheme [24] proposed by Ye should be further improved before it is used in practice.

Acknowledgements

This research including the visit of Dr. Avishek Adhikari to Kyushu University is supported by Strategic Japanese-Indian Cooperative Programme on Multidisciplinary Research Field, which combines Information and Communications Technology with Other Fields Supported by Japan Science and Technology Agency and Department of Science and Technology of the Government of India. Meanwhile, this research is partially supported by the Postgraduate Technology Innovation Foundation of Chongqing University, China (Grant No.200903A1B0010303), the National Nature Science Foundation of China (Grant No. 61003246), and the Program for New Century Excellent Talents in University of China (Grant No. NCET-08-0603).

References

1. Paquet, A.H., Ward, R.K., Pitas, L.: Wavelet packets-based digital watermarking for image verification and authentication. Signal Process. 83, 2117–2132 (2003)
2. Tsai, H.M., Chang, L.W.: Secure reversible visible image watermarking with authentication. Signal Process.: Image Commun. 25, 10–17 (2010)
3. Ahmed, F., Siyal, M.Y., Abbas, V.U.: A secure and robust hash-based scheme for image authentication. Signal Process. 90, 1456–1470 (2010)
4. Vishal, M., Banerjee, A., Evans, B.L.: A clustering based approach to perceptual image hashing. IEEE Trans. Information Forensics Security 1, 68–79 (2006)
5. Wu, D., Zhou, X.B., Niu, X.M.: A novel image hash algorithm resistant to print-scan. Signal Process. 89, 2415–2424 (2009)
6. Swaminathan, A., Mao, Y.N., Wu, M.: Robust and secure image hashing. IEEE Trans. Information Forensics Security 1, 215–230 (2006)
7. Chung, K.L., Chang, L.C.: Large encrypting binary images with higher security. Pattern Recognition Lett. 19, 461–468 (1998)
8. Chen, G.R., Mao, Y.B., Chui, C.K.: A symmetric image encryption scheme based on 3D chaotic cat maps. Chaos, Solitons Fract. 21, 749–761 (2004)
9. Guan, Z.H., Huang, F.J., Guan, W.J.: Chaos-based image encryption algorithm. Phys. Lett. A. 346, 153–157 (2005)

10. Tong, X.J., Cui, M.G.: Image encryption scheme based on 3D baker with dynamical compound chaotic sequence cipher generator. Signal Process. 89, 480–491 (2009)
11. Gao, T.G., Chen, Z.Q.: A new image encryption algorithm based on hyper-chaos. Phys. Lett. A. 372, 394–400 (2008)
12. Chang, C.C., Yu, T.X.: Cryptanalysis of an encryption scheme for binary images. Pattern Recognition Lett. 23, 1847–1852 (2002)
13. Wang, K., Pei, Z.L.H., Song, A.G., He, Z.Y.: On the security of 3D Cat map based symmetric image encryption scheme. Phys. Lett. A 343, 432–439 (2005)
14. Cokal, C., Solak, E.: Cryptanalysis of a chaos-based image encryption algorithm. Phys. Lett. A. 373, 1357–1360 (2009)
15. Xiao, D., Liao, X.F., Wei, P.C.: Analysis and improvement of a chaos-based image encryption algorithm. Chaos, Solitons Fract. 40, 2191–2199 (2009)
16. Solak, E.: Cryptanalysis of image encryption with compound chaotic sequence. In: 6th International Multi-Conference on Systems, Signals and Devices, pp. 1–5. IEEE Press, Djerba (2009)
17. Solak, E., Rhouma, R., Belghith, S.: Cryptanalysis of a multi-chaotic systems based image cryptosystem. Opt. Commun. 283, 232–236 (2010)
18. Solak, E.: On the security of a class of discrete-time chaotic cryptosystems. Phys. Lett. A. 320, 389–395 (2004)
19. Li, S.J., Chen, G.R., Zheng, X.: Chaos-based encryption for digital images and videos. CRC Press, Boca Raton (2004)
20. Furht, B., Socek, D., Eskicioglu, A.M.: Fundamentals of multimedia encryption techniques. CRC Press, Boca Raton (2004)
21. Uhl, A., Pommer, A.: Image and Video Encryption: From Digital Rights Management to Secured Personal Communication. Springer Science/Business Media Inc., Boston (2005)
22. Álvarez, G., Li, S.: Some basic cryptographic requirements for chaos-based cryptosystems. Int. J. Bifurcat Chaos. 16, 2129–2151 (2006)
23. Li, S.J., Li, C.Q., Chen, G.R., Bourbakis, N.G., Lo, K.T.: A general quantitative cryptanalysis of permutation-only multimedia ciphers against plaintext attacks. Signal Processing: Image Communication 23, 212–223 (2008)
24. Ye, G.D.: Image scrambling encryption algorithm of pixel bit based on chaos map. Pattern Recognition Lett. 31, 347–354 (2010)
25. Li, C.Q., Lo, K.T.: Security analysis of a binary image permutation scheme based on Logistic map. Cornell University Library, e-prints in computer science (2009), http://arxiv.org/PS_cache/arxiv/pdf/0912/0912.1918v2.pdf
26. Stinson, D.R.: Cryptography: theory and practice. CRC Press, Boca Raton (1995)
27. Rhouma, R., Safya, B.: Cryptanalysis of a spatiotemporal chaotic cryptosystem. Chaos, Solitons Fract. 41, 1718–1722 (2009)
28. Li, C.Q., Li, S.J., Álvarez, G., Chen, G.R., Lo, K.T.: Cryptanalysis of two chaotic encryption schemes based on circular bit shift and xor operations. Phys. Lett. A. 369, 23–30 (2007)

Plane Transform Visual Cryptography

Jonathan Weir and WeiQi Yan

Queen's University Belfast, Belfast, BT7 1NN, UK

Abstract. Plane transformation visual cryptography takes a unique approach to some of the current shortcomings of current visual cryptography techniques. Typically, the direction and placement of the encrypted shares is critical when attempting to recover the secret. Many schemes are highly dependant on this stacking order. Within this paper, the scheme presented illustrates a technique whereby this restriction is loosened such that the number of acceptable alignment points is increased by performing a simple plane transform on one of the shares. This results in the same secret being recovered when the shares correctly aligned. The technique has also been extended to encompass multiple secrets, each of which can be recovered depending on the type of transformation performed on the shares.

1 Introduction

Visual cryptography provides a very powerful means by which one secret can be distributed into two or more pieces known as shares. When the shares are superimposed exactly together, the original secret can be discovered without requiring any computation. The decryption is performed by the human visual system (HVS) [7].

Many schemes within visual cryptography suffer from alignment issues and are dependant on how the shares are stacked together [6] [15]. Loosening or removing this restriction would be a very desirable advance, as it enables an end user to recover the secret without having to work out how he must stack the shares.

Figure 1 provides an example of this stacking and alignment problem. It can be observed that successful recovery is achieved when the shares are superimposed correctly. However, if the second share is transformed about its center point in the x-axis direction, then the secret cannot be recovered. Removing this limitation would improve the end users experience when it comes to recovering the hidden secret.

As the name suggests, plane transformation visual cryptography deals with a plane figure that is flipped or reflected across a line, creating a mirror image of the original figure, or part mirror image. By making use of these types of transformations, and analyzing how the shares are manipulated under these transformations, designing suitable shares which can recover secrets from any alignment point would be very useful in terms of efficient secret recovery.

H.-J. Kim, Y. Shi, and M. Barni (Eds.): IWDW 2010, LNCS 6526, pp. 60–74, 2011.

(a) Share one. (b) Share two. (c) Secret recovered by superimposing share two on share one. (d) Attempted secret recovery after flipping share two vertically and superimposing it on share one.

Fig. 1. Traditional visual cryptography decryption process

Creating shares in such a way that allows for secret recovery when the shares are superimposed after having been transformed was a valid line of research as it removes these specific types of restrictions which are demonstrated in Figure 1.

The main idea is that one share is printed onto a normal white page, but the second is printed onto a transparency. This transparency is then transformed as previously mentioned. Figure 2 illustrates each of the transformations that each share undergoes in order to recover each of the secrets. Share one is marked with an A, share two is marked with a G. The arrow denotes superimposing the shares in their specific configurations. After each of the transformation, the same or unique secrets can be recovered.

Two different types of secret recovery is possible with the proposed scheme. Each type uses the same methods, but the results are quite different. The first type recovers the same secret when the shares are transformed. The second type recovers a unique secret.

Simply superimposing the shares in the normal way reveals the secret, transforming the upper share along the horizontal or vertical plane also reveals the same secret while combining a horizontal transform with a vertical transform also reveals a secret.

This transform can also be thought of as a mirror image or reflection about a point. As previously mentioned, the space is Euclidean, therefore a reflection is a type of Euclidean plane isometry. Euclidean plane isometry refers to a way of transforming the plane such that the transformation preserves geometrical properties such as length. This makes it very suitable for use in visual cryptography.

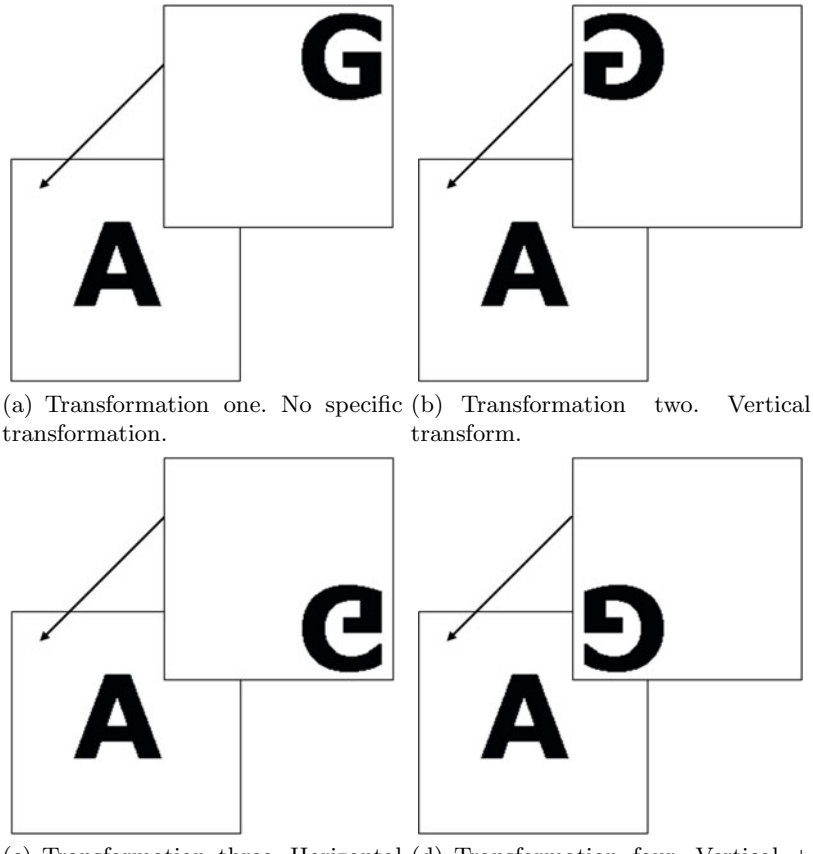
Formally, isometry of the Euclidean plane is a distance-preserving transformation of the plane, that is, a map:

$$M : R^2 \rightarrow R^2 \tag{1}$$

such that for the points p and q in the plane,

$$d(p, q) = d(M(p), M(q)), \tag{2}$$

where $d(p, q)$ is the Euclidean distance between p and q.



(a) Transformation one. No specific transformation.

(b) Transformation two. Vertical transform.

(c) Transformation three. Horizontal transform.

(d) Transformation four. Vertical + horizontal transform.

Fig. 2. Share configurations under specific transformations

This reflection (mirror isometries) about a point can be represented accordingly, where $\mathrm{Ref}(\theta)$ corresponds to the reflection:

$$\mathrm{Ref}(\theta) = \begin{bmatrix} \cos 2\theta & \sin 2\theta \\ \sin 2\theta & -\cos 2\theta \end{bmatrix}. \tag{3}$$

If this mirror/reflection/flip(F) is denoted as $F_{\boldsymbol{C},\boldsymbol{V}}$ where $\boldsymbol{C}$ is a point in the plane, $\boldsymbol{V}$ is a unit vector in R^2. This has the effect of reflecting a point $\boldsymbol{P}$ in the line L that is perpendicular to $\boldsymbol{V}$ and that passes through $\boldsymbol{C}$. The line L is known as the reflection axis (the mirror).

A formula for $F_{\boldsymbol{C},\boldsymbol{V}}$ can be deduced using the dot product to find the component t of $\boldsymbol{P} - \boldsymbol{C}$ in the $\boldsymbol{V}$ direction,

$$t = (\boldsymbol{P} - \boldsymbol{C}) \cdot \boldsymbol{V} = (\boldsymbol{P}_x - \boldsymbol{C}_x)\boldsymbol{V}_x + (\boldsymbol{P}_y - \boldsymbol{C}_y)\boldsymbol{V}_y, \tag{4}$$

and then we obtain the reflection of p by subtraction,

$$F_{\boldsymbol{C},\boldsymbol{V}} = \boldsymbol{P} - 2t\boldsymbol{V}. \tag{5}$$

These reflections about a line through the origin is obtained with all orthogonal matrices forming an orthogonal group O(2). When the determinant is -1 the operation is a reflection in the x-axis followed by a rotation by an angle of θ, where θ in this case is zero:

$$R_{0,\theta}(\boldsymbol{P}) = \begin{pmatrix} \cos\theta & \sin\theta \\ \sin\theta & -\cos\theta \end{pmatrix} \begin{bmatrix} \boldsymbol{P}_x \\ \boldsymbol{P}_y \end{bmatrix}. \tag{6}$$

This theory forms the basis of this paper and more specifically, how exactly the visual cryptography shares are manipulated in order to recover each of the secrets.

In this paper, methods are proposed whereby the same, or unique secrets can be shared in this way. The scheme was designed with recovering the same secret in mind. This removes the specific need to stack the shares in an unequivocal way. Additionally, the scheme also supports four unique secrets, which are recovered in the way previously mentioned.

The remainder of this paper is set out as follows, Section 2 presents the related work, Section 3 outlines contributions and methods for this new scheme while Section 4 illustrates the results. The conclusion is drawn in Section 5.

2 Related Work

Much of the research involving multiple secret sharing using visual cryptography primarily takes rotation of the shares into consideration. Rotating four-sided polygons or quadrilaterals is not very intuitive. Transforming quadrilaterals about a plane is much more intuitive and easier to do.

It is also more practical. The reason for this is that a user may not know whether the share he is superimposing is in the correct position [12]. This is one of the major shortcomings of many visual cryptography schemes which employ random shares to recover the secrets, in both single and multi-secret sharing configurations. For the simplest case, sharing a single secret, how does the person who is using the shares know whether the shares are correctly positioned?

This is where the proposed scheme comes in. Removing this ambiguity results in instant secret recovery by simply stacking the shares as per usual. A single secret will be available to the user from many unique stacking positions. This assists the user immensely from the point of view of correctly stacking the shares.

The following work presented within this section gives an overview of previous multiple secret sharing techniques, from the original idea to how the schemes progressed.

The multiple secret sharing problem was initially examined by Wu and Chen [13]. They concealed two secrets within two sets of shares S_1 and S_2. The first secret is revealed when S_1 and S_2 are superimposed. The second becomes available when S_1 is rotated anti-clockwise 90° and superimposed on S_2. Due to the

nature of the angles required for revealing the secrets (90°, 180° or 270°) and the fact that this scheme can only share, at most, two secrets, it becomes apparent that it is quite limited in its use.

It is also worth noting that another extended form of secret sharing was proposed [5] that is quite similar to the one discussed which involves stacking the transparencies to reveal a different secret each time a new layer is stacked. An improvement on this extended scheme is achieved by reducing the number of subpixels required [16].

Multiple secret sharing was developed further [14] by designing circular shares so that the limitations of the angle ($\theta = 90°, 180°, 270°$) would no longer be an issue. The secrets can be revealed when S_1 is superimposed on S_2 and rotated clockwise by a certain angle between 0° and 360°. However, this scheme suffers from the same alignment issues as the previous schemes in that supplementary lines must be added to the shares before secret recovery can take place. Accurately aligning the circular shares is not possible otherwise.

A further extension of multiple secret sharing was implemented [4] which defines another scheme to hide two secret images in two shares with arbitrary rotating angles. This scheme rolls the share images into rings to allow easy rotation of the shares and thus does away with the angle limitation of Wu and Chen's scheme. The recovered secrets are also of better quality when compared to [14], this is due to larger difference between the black and white stacked blocks.

More recently a novel secret sharing scheme was proposed [9] that encodes a set of $x \geq 2$ secrets into two circle shares where x is the number of secrets to be shared. This is one of the first set of results presented that is capable of sharing more than two secrets using traditional visual cryptography methods. The algorithms presented can also be extended to work with grayscale images by using halftone techniques. Colour images could also be employed by using colour decomposition [3] or colour composition [8].

The scheme presented within [10] takes yet another approach which requires shifting the shares around within a small area in order to recover the secrets. A maximum of four secrets can be hidden within the presented scheme, however, as with the previously discussed schemes, alignment issues can become problematic when recovering the remainder of the secrets.

A rather unique way of looking at the multiple secret sharing problem in a practical way is detailed within [11]. The authors take the idea of sharing multiple secrets and apply it to Google's Street View implementation used within Google Maps. The secrets that are recovered are used as part of the watermarking system which can be used at varying resolutions.

The scheme in [11] uses a recursive secret sharing scheme which caters for the multiple zoom levels which are supported by Google Maps. After the recursive shares have been embedded, superimposing the corresponding share reveals the secret. If the map itself is zoomed it may be impossible to identify the secret. This is why the recursive scheme works so effectively. As the levels are zoomed,

the same, or unique secret can still be successfully recovered at the various levels of zooming.

The nature of this resolution variant visual cryptography is one of a digital nature. This removes the need for specific alignment and share placement, as a digital system can be pre-configured with the correct alignment and share location. However, having this same advantage with VC schemes that require physical stacking has not yet be explored. An attempt to rectify this problem and improve current multiple secret sharing schemes are detailed herein.

The work presented within this paper on plane transform visual cryptography develops the work in [10] and provides a more useful and practical approach for secret recovery while taking into consideration the multiple secret sharing problem.

The majority of these schemes require some type of rotation in order to recover the secret, this is where the proposed scheme in this paper differs. As of yet, the authors have yet to find another piece of work on multiple secret sharing which encompasses the use of simple plane transformations in order to recover the secrets. In the following section, details are posed which highlight the differences and improvements over existing techniques.

3 Our Contribution

The term "plane" used within this paper refers to a flat two-dimensional surface. We used this term when describing the shares in order to illustrate the type of movement that they undergo using geometric expressions. Therefore the whole space is used when working in a two-dimensional Euclidean space.

When compared to the plethora of visual cryptography schemes currently in use today, this scheme attempts to improve upon them by allowing the shares to be stacked in a variety of ways, after having been transformed about the horizontal, vertical, and a combination of both axes. This is a much more intuitive way to manipulate a quadrilateral in order to recover each of the secrets. Especially when dealing with two shares.

Removing the specific stacking order required by the majority of the previous schemes is a great advantage, as it allows for easier secret recovery. Illustrated within this section is one main idea which accomplishes two goals, the same secret recovery based on different transforms and unique secret recovery based on the same set of transformations. Ideally, the same secret is used, this means that no matter how the shares are stacked, the same results are obtained. The unique secrets are illustrated to prove that it is possible for unique secrets to be shares as well.

The steps involved in order to create the resulting two shares can be examined in Figure 3. Figure 3 provides a flowchart of the proposed system which details each of the corresponding actions required. Each of these steps are detailed below.

It can be observed from Figure 3 that a secret is input and four sets of shares are generated accordingly, $Sec_1S_1 \rightarrow Sec_4S_1$ for the set of secrets belonging to

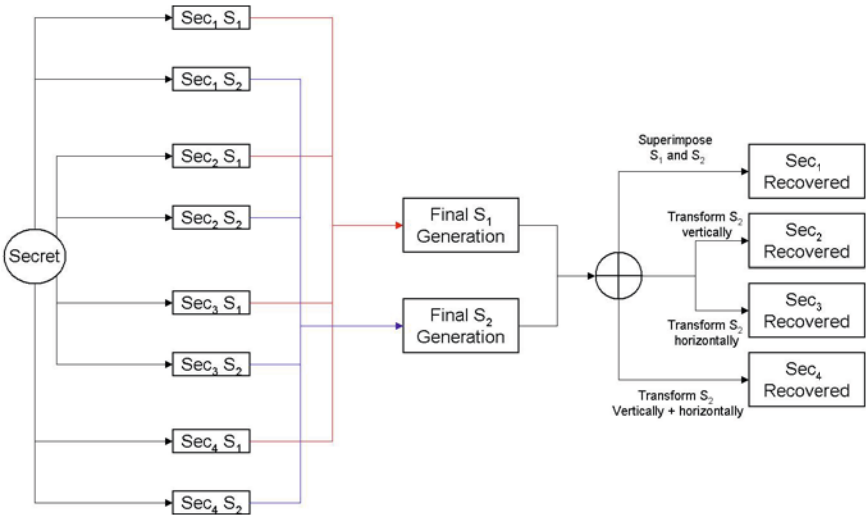


Fig. 3. Plane transform visual cryptography flowchart

share one and $Sec_1S_2 \rightarrow Sec_4S_2$ for the set of secrets belong to share two. Where Sec_1S_1 refers to share one from secret and Sec_1S_2 refers to share two from the corresponding set of secrets.

Whether one secret is input (recovering the same secret for each transform: $Sec_1 = Sec_2 = Sec_3 = Sec_4$) or four secrets (unique secret recovery for each transform), four sets of shares are generated. Based on these sets of shares, the final set of two shares are generated which allow for the recovery. When Final S_1 and Final S_2 are superimposed, Sec_1 is recovered. When Final S_2 is transformed vertically about its center point on the x-axis, Sec_2 can be recovered. Sec_3 can be observed when Final S_2 is transformed about its center point along the y-axis in a horizontal direction. Finally, Sec_4 is revealed after Final S_2 is transformed about both center points about each axis.

The algorithm required is presented within Algorithm 1, which provides a pseudocode implementation of the plane transformation visual cryptography process. Further details are presented in the following sections. They provide more insight into what happens during each of the steps. This algorithm simply provides a computational reference as to how the process is executed.

The generateShare(·) and expandShare(·) functions are discussed in more detail in Sections 3.1 and 3.2 respectively. The processShare(·) function is also detailed in Section 3.2.

This transformation requires a lot of thought when creating a suitable scheme that can recover black and white pixels accordingly. Some pixel configurations may be representing white pixels, while, after a vertical transformation the pixel representation required is black.

This paper primarily extends the work presented within [10] which greatly improves on these previous techniques and makes the resulting scheme much

Algorithm 1. Pseudocode for the algorithm which generates two shares which are suitable for plane transformation visual cryptography.

Input: One secret four times or four unique secrets $Sec_1, \ldots, Sec_4$.
Output: Final set of shares Final S_1 and Final S_2.
begin
  for $i \in range(4)$ **do**
    Sec_iS_1, Sec_iS_2 = generateShares(Sec_i);
  return $Sec_1S_1 \rightarrow Sec_4S_1$;
  return $Sec_1S_2 \rightarrow Sec_4S_2$;
  for $i \in range(4)$ **do**
    expandShare(Sec_iS_1);
    expandShare(Sec_iS_2);
  for $i \in range(4)$ **do**
    processShare(Sec_iS_1);
    processShare(Sec_iS_2);
  return Final S_1;
  return Final S_2;
end

Table 1. A comparison of multiple secret sharing schemes against the authors proposed scheme

Authors	Format	Pixel Expansion	Number of Secrets
Authors proposed scheme	Binary	4	4
Wu and Chen [13]	Binary	4	2
Hsu et al. [4]	Binary	4	2
Wu and Chang [14]	Binary	4	2
Shyu et al. [9]	Binary	$2n$	$n \geq 2$
Chen et al. [1]	Binary	1	2
Fang [2]	Binary	1	2
Weir and Yan [10]	Binary	4	n

more practical in terms of use. Table 1 presents a comparison with other multiple secret sharing schemes.

From the table, the majority of the schemes compared involved so form of rotation or shifting of the shares in order to recover the secrets. Each share must be correctly positioned before one of the secrets can be recovered. The proposed scheme presented within the following sections not only removes this alignment issue, but also improves the capacity of the previous schemes in terms of number of secrets hidden when compared against pixel expansion.

3.1 Share Generation

The shares are generated using a combination of processes. A size invariant scheme is used initially and then using these size invariant shares, are then expanded into a more traditional scheme where one pixel from the invariant

shares are represented by a 2×2 block. This is the general process used to create the final share. Each of the invariant shares patterns are used to create a new suitable pattern capable of recovering each of the secrets.

The structure of this scheme is described by a Boolean n-vector $\mathbf{V} = [v_0, v_1]^T$, where v_i represents the colour of the pixel in the i-th shared image. If $v_i = 1$ then the pixel is black, otherwise, if $v_i = 0$ then the pixel is white. To reconstruct the secret, traditional ORing is applied to the pixels in $\mathbf{V}$. The recovered secret can be viewed as the difference of probabilities with which a black pixel in the reconstructed image is generated from a white and black pixel in the secret image. As with traditional visual cryptography, $n \times m$ sets of matrices need to be defined for the scheme (in this case 2×2):

$C_0 = \{\text{all the matrices obtained by permuting the columns of } \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}\}$

$C_1 = \{\text{all the matrices obtained by permuting the columns of } \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\}$

Because this scheme uses no pixel expansion, m is always equal to one and n is based on the type of scheme being used, for example a $(2, 2)$ scheme, $n = 2$.

Using the defined sets of matrices C_0 and C_1, $n \times m$ Boolean matrices S^0 and S^1 are chosen at random from C_0 and C_1, respectively:

$$S_0 = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}, S_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{7}$$

To share a white pixel, one of the columns in S_0 is chosen and to share a black pixel, one of the columns in S_1 is chosen. This chosen column vector $\mathbf{V} = [v_0, v_1]^T$ defines the colour of each pixel in the corresponding shared image. Each v_i is interpreted as black if $v_i = 1$ and as white if $v_i = 0$. Sharing a black pixel for example, one column is chosen at random in S^1, resulting in the following vector:

$$\mathbf{V} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{8}$$

Therefore, the i-th element determines the colour of the pixels in the i-th shared image, thus in this (2,2) example, v_1 is white in the first shared image, v_2 is black in the second shared image.

3.2 Share Expansion

After the shares for each identical or unique secret have been generated, each set of shares for each secret are expanded into a 2×2 block and inserted into the final set of shares by the processShare$(\cdot)$ function from Algorithm 1. The following steps are involved when processShare$(\cdot)$ is executing. This function generates the final set of shares required in order to successfully recover the secrets.

Generating Final S_1 is a relatively simple procedure where each of the corresponding expanding shares are placed into the following coordinates on the share:

- Sec_1S_1 no change, leave its current pixel locations intact.
- Sec_2S_1 shift its pixel locations one pixel to the right, in order to fill in the space to the right of Sec_1S_1's pixels.
- Sec_3S_1 shift its pixel locations down one pixel, this fills in the space beneath Sec_1S_1's pixels.
- Sec_4S_1 shift its pixel locations down one and right one, this fills in the final space remaining on the final share.

Generating Final S_2 is more challenging. The reason being is that the transformations that this share undergoes need to be taken into consideration so that the correct black and white pixels can be represented. Accurate reconstruction is very difficult because four different situations arise due to the transforms.

Final S_2 can be generated according to the following scheme:

- Sec_1S_2 no change, leave its current pixel locations intact.
- Sec_2S_2 place its pixels in the same locations as those which belong to Sec_2S_1, but its vertical inverse must be placed at those locations.
- Sec_3S_2 place its pixels in the same locations as those which belong to Sec_3S_1, but its horizontal inverse must be placed at those locations.
- Sec_4S_2 place its corresponding vertical and horizontal inverse pixels at the same coordinates as those of Sec_4S_1.

No change is made to the placement of the first set of secret shares, this corresponds to simply superimposing each of the shares in the traditional way. The inverse of the pixel locations are required in order to reconstruct each of the secrets after a specific transformation occurs. Determining the inverse pixel patterns required for each of the specific transformed patterns proved to be rather difficult in terms of alignment.

After a transform on a pixel block was performed, simply supplying the inverse at a pixels transformed location was not possible. This is down to the fact that other pixels may be required at that location in order to provide a white pixel representation at one instance, but a black pixel at another.

This resulted in a compromise between full secret recovery a probabilistic secret recovery which would be closer to a "best effort" style of recovery. This best effort is mostly a tradeoff between visual representation and resulting contrast.

The results from this process are good when the same secret is to be recovered after each transformation. The recovered quality would be similar in terms of contrast the extended visual cryptography schemes which employ halftoning [17]. The contrast ratio is typically around $\frac{1}{4}$. The contrast suffers, when different secrets are added. The recovered secrets remain readable, but a much lower contrast is available. This is due to the nature of the scheme, completely new patterns have to be generated which must represent a unique letter each time. Using the same letter as the secret, the same patterns can be selected, therefore giving a higher contrast. This is not possible when using unique secrets.

Another important aspect of the scheme that must be mentioned and analysed is the security. Traditional VC schemes exhibit good security due to the nature of the patterns that are chosen to represent pixels from the original. If a white

pixel is to be represented then each pattern used to represent the white pixel is placed in each share. Similarly, corresponding patterns are placed in each share when a black pixel is to be represented. This results in a potential attacker (who has obtained one of the shares) having to make a 50/50 choice for each pixel pattern in order to guess the correct corresponding share. It can be observed that this is not feasible at all.

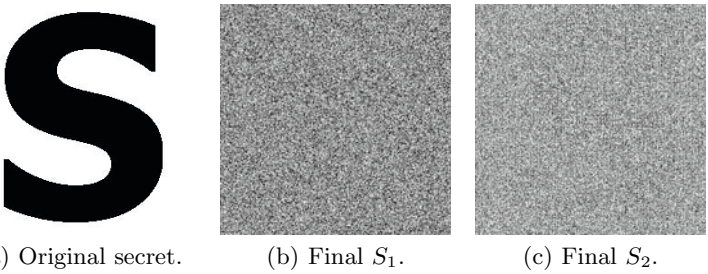
The same is true for the proposed scheme in this paper. Based on each of the individual shares that are created for each of the secrets, a new pattern is created which is capable of revealing the secret while being transform invariant. These new patterns work in the same way as the traditional patterns. An attacker would have to generate identical or corresponding patterns for each of the pixel representations. Correctly guessing those patterns to reveal one of the secrets is extremely unlikely, guessing the correct patterns so that four secrets are revealed is even more unlikely again. The probabilities drop even further when four unique secrets are examined.

Randomness of the generated shares can also be examined in a security context. Visually, the shares do not leak any information about the secrets that are hidden within. On further inspection of the shares, the distribution of the pixels is uniform. This makes it much harder for an attacker to concentrate on a specific area on the share in order to force information to leak out regarding the secret.

4 Results

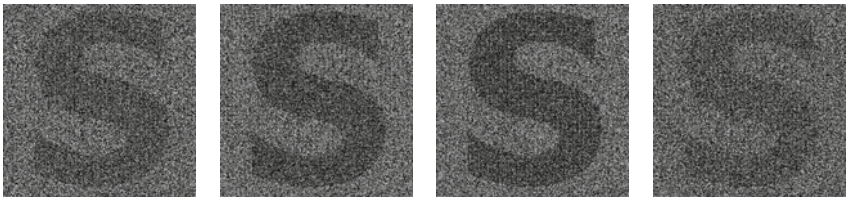
A number of results are presented within this section which show the capability of the scheme discussed. The two shares that are generated using this scheme are depicted in Figure 4. These shares look like normal visual cryptography shares and do not give away any information about the secret or secrets concealed within.

When superimposed, these shares can recover the secret "S". Figure 5 provides the results of each of the transformations which the share can be made to go through in order to recover the same secret. Figure 5(a) is simply share one superimposed upon share two. Figure 5(b) shows the secret recovery after share



(a) Original secret. (b) Final S_1. (c) Final S_2.

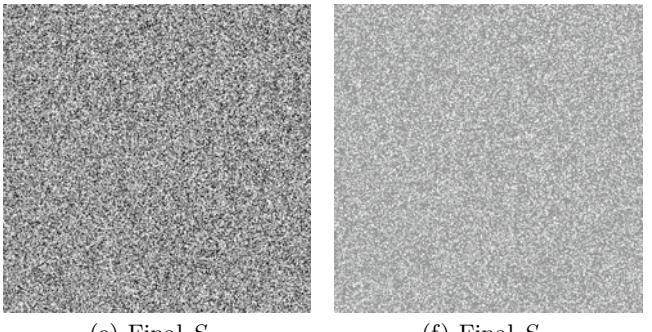
Fig. 4. The corresponding shares where all the secrets are identical



(a) Share two no transformation. (b) Share two transformed about the horizontal axis. (c) Share two transformed about the vertical axis. (d) Share two transformed about the horizontal and vertical axis.

Fig. 5. The same secret recovered after different plane transformations

(a) Original secret one. (b) Original secret two. (c) Original secret three. (d) Original secret four.



(e) Final S_1. (f) Final S_2.

Fig. 6. The corresponding shares when all the secrets are unique

two has been transformed about the horizontal axis. Figure 5(c) highlights the secret recovery after share two has been transformed about the vertical axis and Figure 5(d) provides the final secret recovery after share two has been transformed in both the horizontal and vertical axis.

In the following results, multiple, unique secrets have been embedded within a set of shares. Using the same technique as previously described, each of the secrets can be recovered.

Figure 6 provides each of the secrets along with their corresponding shares. Each secret has its own set of decryption blocks embedded within the shares so

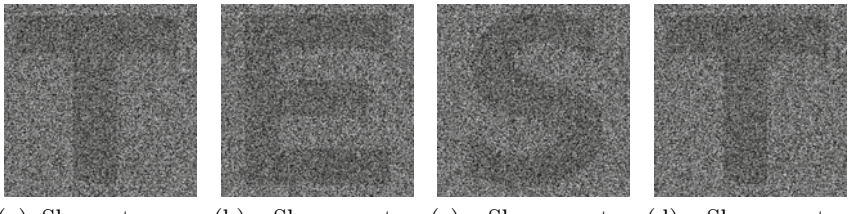
that as each of the secrets are recovered, no information leaks out with regard to the other secrets. This is vital in any multi-secret sharing visual cryptography scheme.

The recovered results are presented within Figure 7. Figure 7(a) shows the first "T" from the secret "TEST". Figure 7(b) to 7(d) provide the remaining results after specific transformations have been performed on the second share as it is superimposed.

Using a simple transform, accurate and effective secret recovery can be achieved. No rotation is required, all that is needed is a simple geometric transformation. This helps users recover secrets almost immediately without having to determine the correct angle and stacking order of the shares.

Testing these shares can be done very easily and quickly using a the very simple Microsoft Paint program. Final S_1 can be loaded into the application, Final S_2 can be pasted on top and set to have a transparent background. Using the Flip/Rotate option, Final S_2 can be manipulated vertically, horizontally and both in order to test the validity of the results.

From these results it is clear that contrast improvements can be made in particular when transforming share two twice, in both axial directions. The secret is still readable but the contrast does suffer.



(a) Share two no transformation. (b) Share two transformed about the horizontal axis. (c) Share two transformed about the vertical axis. (d) Share two transformed about the horizontal and vertical axis.

Fig. 7. The same secret recovered at different plane transformations

5 Conclusion

From the results and discussion presented, it is easy to see the advantages a scheme like this has over existing schemes. Reducing the alignment problem to a simple transform while being able to recover four identical or unique secrets is a great advantage to the end user. This scheme removes the onus on the user when aligning and recovering the secrets.

This type of invariant placement of shares should be considered in the future when new cutting-edge VC schemes are being proposed. Making secret recovery

easy for the end user is highly valuable and may help to push VC and into the mainstream. A key improvement to be considered in the future is contrast improvement. The current secret recovery is adequate, but a revision of the current techniques would be required in order to achieve this progression. Currently, the probabilistic style of recovery is acceptable but improvements would definitely be welcomed.

Future work would include a combination of the techniques presented along with rotation, which would result in a true position invariant visual cryptography scheme that would allow secret recovery no matter how the shares are stacked together. By removing these restrictions which plague current schemes, visual cryptography may be better suited to more practical applications in which recovering a secret piece of information should be fast and efficient and not dependant on how the shares are physically stacked.

References

1. Chen, T.-H., Tsao, K.-H., Wei, K.-C.: Multiple-image encryption by rotating random grids. In: International Conference on Intelligent Systems Design and Applications, vol. 3, pp. 252–256 (2008)
2. Fang, W.-P.: Non-expansion visual secret sharing in reversible style. International Journal of Computer Science and Network Security 9(2), 204–208 (2009)
3. Hou, Y.-C.: Visual cryptography for color images. Pattern Recognition 36, 1619–1629 (2003)
4. Hsu, H.-C., Chen, T.-S., Lin, Y.-H.: The ringed shadow image technology of visual cryptography by applying diverse rotating angles to hide the secret sharing. Networking, Sensing and Control 2, 996–1001 (2004)
5. Katoh, T., Imai, H.: An extended construction method for visual secret sharing schemes. IEICE Transactions J79-A(8), 1344–1351 (1996)
6. Liu, F., Wu, C.K., Lin, X.J.: The alignment problem of visual cryptography schemes. Designs, Codes and Cryptography 50(2), 215–227 (2009)
7. Naor, M., Shamir, A.: Visual cryptography. In: De Santis, A. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 1–12. Springer, Heidelberg (1995)
8. Shyu, S.J.: Efficient visual secret sharing scheme for color images. Pattern Recognition 39(5), 866–880 (2006)
9. Shyu, S.J., Huang, S.-Y., Lee, Y.-K., Wang, R.-Z., Chen, K.: Sharing multiple secrets in visual cryptography. Pattern Recognition 40(12), 3633–3651 (2007)
10. Weir, J., Yan, W.: Sharing multiple secrets using visual cryptography. In: IEEE International Symposium on Circuits and Systems, ISCAS 2009, pp. 509–512 (2009)
11. 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)
12. Weir, J., Yan, W.: A comprehensive study of visual cryptography. Transactions on Data Hiding and Multimedia Security 5 (2010)
13. Wu, C., Chen, L.: A study on visual cryptography. Master's thesis, Institute of Computer and Information Science, National Chiao Tung University, Taiwan, R.O.C. (1998)

14. Wu, H.-C., Chang, C.-C.: Sharing visual multi-secrets using circle shares. Computer Standards & Interfaces 28, 123–135 (2005)
15. Yan, W., Duo, J., Kankanhalli, M.S.: Visual cryptography for print and scan applications. In: Proceedings of IEEE International Symposium on Circuits and Systems 2004, Vancouver, Canada, pp. 572 – 575 (May 2004)
16. Yang, C.-N., Chen, T.-S.: Extended visual secret sharing schemes: Improving the shadow image quality. IJPRAI 21(5), 879–898 (2007)
17. Zhou, Z., Arce, G.R., Crescenzo, G.D.: Halftone visual cryptography. IEEE Transactions on Image Processing 15(8), 2441–2453 (2006)

A Statistical Model for Quantized AC Block DCT Coefficients in JPEG Compression and Its Application to Detecting Potential Compression History in Bitmap Images

Gopal Narayanan and Yun Qing Shi

Department of Electrical and Computer Engineering,
New Jersey Institute of Technology,
Newark, New Jersey, USA 07102
{gt25,shi}@njit.edu

Abstract. We first develop a probability mass function (PMF) for quantized block discrete cosine transform (DCT) coefficients in JPEG compression using statistical analysis of quantization, with a Generalized Gaussian model being considered as the PDF for non-quantized block DCT coefficients. We subsequently propose a novel method to detect potential JPEG compression history in bitmap images using the PMF that has been developed. We show that this method outperforms a classical approach to compression history detection in terms of effectiveness. We also show that it detects history with both independent JPEG group (IJG) and custom quantization tables.

Keywords: JPEG, Generalized Gaussian, Image Forensics, Compression History, PMF.

1 Introduction

JPEG encoding is a popular technique for effective compression of two-dimensional raster images. The compression method involves dividing the image into equal-size blocks and performing a 2D Discrete Cosine Transform (DCT) on them. The transform coefficients hence obtained are then integer divided by a prefixed *quantization* matrix which is determined from the chosen JPEG *Q-factor*, where the Q-factor is an integer which quantifies the extent of JPEG compression. The integral transform coefficients are then zigzag scanned, entropy coded and written into a file.

JPEG encoding is an inherently lossy process due to the integer division that is performed on the DCT coefficients. That is, when the JPEG image is decoded into an uncompressed format such as bitmap, the magnitude of the obtained pixel values do not match exactly with those of the original, uncompressed image. This lossy form of compression can lead to visible artifacts in the decompressed image if the quantization is excessively aggressive. In most cases, however, the artifacts are not visibly evident, but can be detected using various kinds of approaches. Detection of such artifacts, or more broadly, the detection of historical JPEG compression is an important image processing and forensic application. For example, as indicated in [1], such

H.-J. Kim, Y. Shi, and M. Barni (Eds.): IWDW 2010, LNCS 6526, pp. 75–89, 2011.

history detection is important for JPEG artifact removal or for obviating additional distortion when the decompressed image is re-encoded in a lossy manner. Such knowledge may also be useful in covert messaging and for authentication [2].

Historical JPEG compression detection, or JPEG compression history detection as it shall be known henceforth, has been explored to a significant extent in the past decade. Approaches to detection of JPEG compression history have been statistical, such as those proposed in [1] [2], based on first digit distribution [3] or based on image textures [4]. In this paper, we propose a new statistical approach to detecting JPEG compression history and estimating the quantization matrix thereof. We develop a theoretical framework for the statistical properties of quantized 2D DCT coefficients used during the process of JPEG encoding, and employ it to successfully detect historical JPEG compression. We show empirically that the success rate of quantization matrix estimation is higher than that of the approach proposed in [1], and that even in the case of erroneous detection, the error in Q-factor detection is generally no higher than a value of 5.

The rest of this paper is structured as follows. Section 2 summarizes the probability density functions associated with 2D block DCTs in JPEG encoding. Section 3 recalls the impact of uniform quantization on a random variable with a general, unbiased probability distribution function. A new probability mass function (PMF) for quantized 2D block DCT coefficients is derived in Section 4. This derived PMF is used to detect historical JPEG compression in bitmap images in Section 5. Section 5 also compares our detection results with those of the method proposed by Fan and Queiroz [1]. Section 6 summarizes the paper.

2 2D Block DCT Probability Density Functions (PDFs)

This section briefly summarizes the type-II 2D DCT used in JPEG encoding. It introduces the term *mode*, and presents two distinct PDFs proposed in literature for 2D block DCTs.

The type-II DCT is one of seven distinct forms of a real, unitary transform applied to symmetric signals [5]. In two dimensions, the type-II DCT, $G_x(l,m)$ is defined as below [6].

$$G_x(l,m) = \frac{\sqrt{2}}{\sqrt{M}}\frac{\sqrt{2}}{\sqrt{N}}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1} X(i,j)\cos\frac{(2i+1)l\pi}{2N}\cos\frac{(2j+1)m\pi}{2N}. \quad (1)$$

where (M, N) is the size of the 2D block and $X(i,j)$ is the input signal matrix. The type-II 2D DCT will be known simply as 2D *block* DCT henceforth.

The term *mode* is used in this paper to denote a single coefficient in a 2D DCT coefficient block. The location of the coefficient is indicated via an ordered pair, with coordinates relative to the top-left corner of the block, which is denoted as (0, 0). The following figure locates the coefficient at (1, 1), or the (1, 1) mode, in an 8x8 block.

The coefficient at each mode is modeled as a distinct random variable, considered across multiple blocks in an image. The coefficient at mode (0, 0) is conventionally known as *DC* coefficient, while all others are known as *AC* coefficients.

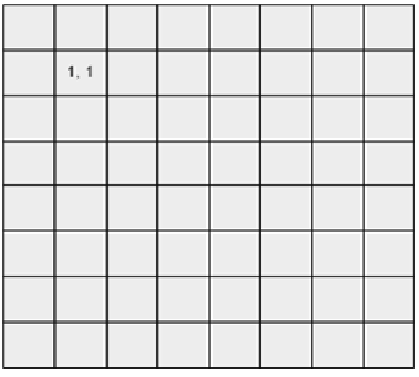


Fig. 1. Mode (1,1) [AC] is shown in an 8x8 image DCT grid

DC coefficients are random variables which are generally considered Gaussian distributed [7]. This conclusion is statistically intuitive considering that the DC coefficient is simply the average of multiple independent and possibly identically distributed image pixel values [8]. This paper does not explore DC coefficients in detail.

PDFs of AC coefficients have been shown [9] to be more leptokurtic than Gaussian distributions. Therefore, the Laplacian distribution [7] and the Generalized Gaussian [10] have been proposed as potential models for the distributions of AC coefficients. The Generalized Gaussian PDF $f(x;\sigma,\vartheta,\mu)$ is given as [11],

$$f(x;\sigma,\vartheta,\mu) = \frac{\vartheta\alpha(\vartheta)}{2\sigma\Gamma(1/\vartheta)}\exp\left\{-\left[\alpha(\vartheta)\left|\frac{x-\mu}{\sigma}\right|\right]^{\vartheta}\right\}. \quad (2)$$

where σ and ϑ are model parameters related to the standard deviation and kurtosis of the PDF, respectively, while μ is the mean of the PDF. $\Gamma(\ldots)$ is the standard Gamma function.

It is easily seen that the Laplacian distribution is a specific case of the Generalized Gaussian PDF, when $\vartheta = 1$, and therefore we shall consider the latter to be a suitable model for AC coefficient probability distributions.

3 Statistics of Quantized Random Variables

Quantization is a non-linear operation, which may be defined for a general random variable x as follows.

$$l = Q(x) \;\mid\; \left(l - \frac{q}{2}\right) \le x < \left(l + \frac{q}{2}\right).$$

The half-open interval, $\left(-\frac{q}{2},\frac{q}{2}\right]$ is known as the *quantization interval*, where q is the *quantization step* and l is an integer approximation to x in the specific quantization interval.

A statistical analysis of quantization is presented in [12], where the effect of uniform quantization on the PDF of a random variable is explored. The analysis is detailed below.

The process of quantization is viewed as adding uniform noise to the random variable, as shown in the following figure.

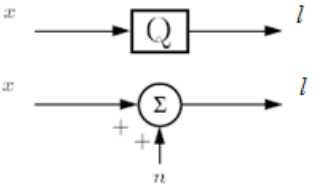


Fig. 2. Quantization is modeled systemically, as a random noise component being added to the random variable undergoing quantization (source: [12])

It is known from probability theory that when two independent random variables are added, the probability density of the resultant random variable is a convolution of the probability densities of the two operand random variables. Therefore,

$$f_l(t) = f_x(t) * f_n(t).$$

In the case of uniform quantization in $\left(-\frac{q}{2}, \frac{q}{2}\right]$, $f_n(t)$ is,

$$f_n(t) = \begin{cases} \frac{1}{q}, & -\frac{q}{2} < t \leq \frac{q}{2} \\ 0, & otherwise \end{cases}.$$

Therefore, with *r* being the independent variable,

$$f_l(t) = \frac{1}{q} \int_{\left(t-\frac{q}{2}\right)}^{\left(t+\frac{q}{2}\right)} f_x(r) dr.$$

Quantization results in random variables with discrete magnitude values at integral multiples of the quantization step. This implies a probability *mass* function (PMF) for the quantized random variable, with the mass function having non-zero values at integral multiples of the quantization step. The PDF is converted to a PMF by sampling it using an impulse train $c(t)$ at integral multiples of q. Therefore,

$$f_l[k] = f_l(t).c(t) = \frac{1}{q}\left[\int_{\left(t-\frac{q}{2}\right)}^{\left(t+\frac{q}{2}\right)} f_x(r) dr\right].c(t), \tag{3}$$

where $c(t) = \sum_{k=-\infty}^{\infty} [q\delta(t - kq)]$, with $\delta(\ldots)$ being the delta function.

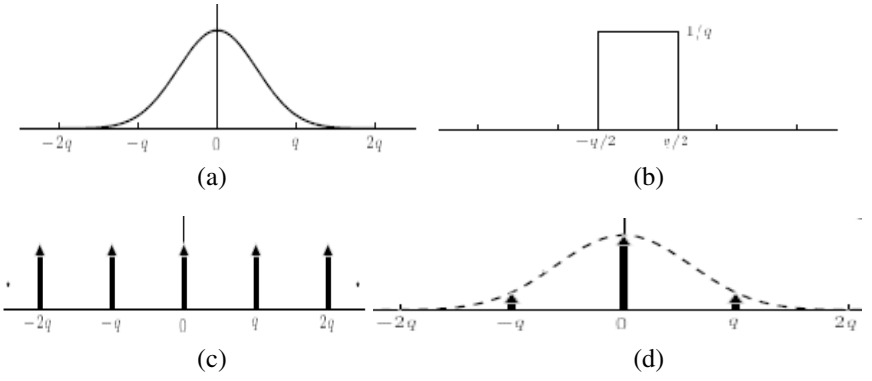


Fig. 3. Plots corresponding to a general bell-shaped PDF, a Uniform distribution in [-q/2. q/2], an impulse train and the corresponding PMF (source: [12])

The PMF of the quantized random variable l is thus derived, starting with the PDF of the pre-quantized random variable x. The process is graphically depicted in Figure 3, with (a) showing a general bell-shaped PDF $f_x(t)$, (b) showing the Uniform distribution $f_n(t)$, (c) showing the impulse train $c(t)$ and (d) showing $f_l[k]$.

It is known that JPEG DCT quantization noise is Uniform distributed for quantization noise values centered on non-zero values [13]. Therefore, the analysis above is completely valid for quantization in JPEG encoding.

4 A New Model for the PMF of Quantized AC 2D Block DCTs

A model for quantized AC 2D block DCT coefficients is derived by substituting (2) in (3). Therefore,

$$f_l[k] = f_l(t).c(t) = \frac{1}{q}\left[\int_{\left(t-\frac{q}{2}\right)}^{\left(t+\frac{q}{2}\right)} \left(\frac{\vartheta\alpha(\vartheta)}{2\sigma\Gamma(1/\vartheta)} \exp\left\{-\left[\alpha(\vartheta)\left|\frac{r-\mu}{\sigma}\right|\right]^{\vartheta}\right\}\right) dr\right].c(t),$$

$$f_l[k] = f_l(t).c(t) = \frac{\vartheta\alpha(\vartheta)}{2q\sigma\Gamma(1/\vartheta)}\left[\int_{\left(t-\frac{q}{2}\right)}^{\left(t+\frac{q}{2}\right)} \left(\exp\left\{-\left[\alpha(\vartheta)\left|\frac{r-\mu}{\sigma}\right|\right]^{\vartheta}\right\}\right) dr\right].c(t).$$

The integral above is evaluated using an approximation. For simplicity, the Simpson's 1/3 rule [14] stated below, is used.

$$\int_{x_0}^{x_2} f(x)\,dx \cong \frac{h}{3}[f(x_0) + 4f(x_1) + f(x_2)],$$

where $h = \frac{x_2 - x_0}{2}$ and $x_1 = x_0 + h = x_0 + \frac{x_2 - x_0}{2} = \frac{x_2 + x_0}{2}$.

Thus,

$$f_l[k] = \frac{\vartheta\alpha(\vartheta)}{12q\sigma\Gamma(1/\vartheta)}\left[\exp\left\{-\left[\alpha(\vartheta)\left|\frac{t+q/2}{\sigma}\right|\right]^{\vartheta}\right\} + 4\exp\left\{-\left[\alpha(\vartheta)\left|\frac{t}{\sigma}\right|\right]^{\vartheta}\right\} + \exp\left\{-\left[\alpha(\vartheta)\left|\frac{t-q/2}{\sigma}\right|\right]^{\vartheta}\right\}\right].c(t). \quad (4)$$

The interval, $t-\frac{q}{2} < 0 \leq t+\frac{q}{2}$ is not infinitely differentiable [15] in this context. The interval is therefore split into two infinitely differentiable half-intervals given as, $\left[t-\frac{q}{2}, 0^{-}\right]$ and $\left[0^{+}, t+\frac{q}{2}\right]$. Evaluating in each half-interval and combining,

$$f_l[k] = \frac{\vartheta\alpha(\vartheta)}{12q\sigma\Gamma(1/\vartheta)}\left(|t|+\frac{q}{2}\right)\left[\exp\left\{-\left[\alpha(\vartheta)\left(\frac{|t|+q/2}{\sigma}\right)\right]^{\vartheta}\right\} + 4\exp\left\{-\left[\alpha(\vartheta)\left(\frac{|t|+q/2}{2\sigma}\right)\right]^{\vartheta}\right\} + 1\right].c(t). \quad (5)$$

(4) and (5) may be summarized as,

$$f_l[kq] = \begin{cases} \frac{q\vartheta\alpha(\vartheta)}{12\sigma\Gamma(1/\vartheta)}\left[\exp\left\{-\left[\alpha(\vartheta)\left(\frac{q}{2\sigma}\right)\right]^{\vartheta}\right\} + 4\exp\left\{-\left[\alpha(\vartheta)\left(\frac{q}{4\sigma}\right)\right]^{\vartheta}\right\} + 1\right] & k=0. \\ \frac{q\vartheta\alpha(\vartheta)}{12\sigma\Gamma(1/\vartheta)}\left[\exp\left\{-\left[\alpha(\vartheta)\left|\frac{kq+q/2}{\sigma}\right|\right]^{\vartheta}\right\} + 4\exp\left\{-\left[\alpha(\vartheta)\left|\frac{kq}{\sigma}\right|\right]^{\vartheta}\right\} + \exp\left\{-\left[\alpha(\vartheta)\left|\frac{kq-q/2}{\sigma}\right|\right]^{\vartheta}\right\}\right] & k\neq 0, k\in Z. \end{cases} \quad (6)$$

Kolmogorov-Smirnov (KS) test [16] statistics are generated to measure the goodness-of-fit between the PMF in (6) and samples of quantized AC block DCT coefficients. We consider that a significance level of 0.3 (i.e., 30 % critical value) is sufficient for null hypothesis rejection. Table (1) lists KS test statistics for the Lena image [17], for varying JPEG Q-factors and for three different modes (modes (0,1), (1,0) and (1,1)) of quantized AC DCT.

It is evident from the table that for most Q-factors, the magnitude of the KS test statistic is below the significance level. The match between (6) and samples of quantized AC block DCT coefficient histograms is thus confirmed.

Similar results are seen in the cases of the Peppers and Boat images from [20], as shown in the following tables.

Table 1. Kolmogorov-Smirnov (KS) test statistics for varying Q-factors used to JPEG compress the Lena image

	Mode		
Q-factor	(0, 1)	(1, 0)	(1, 1)
100	0.287129	0.198020	0.188119
90	0.168317	0.148515	0.099010
80	0.128713	0.099010	0.148515
70	0.099010	0.207921	0.425743
60	0.118812	0.306931	0.524752
50	0.188119	0.405941	0.603960

Table 2. Kolmogorov-Smirnov (KS) test statistics for varying Q-factors used to JPEG compress the Peppers image

	Mode		
Q-factor	(0, 1)	(1, 0)	(1, 1)
100	0.277228	0.217822	0.217822
90	0.188119	0.188119	0.108911
80	0.118812	0.128713	0.227723
70	0.108911	0.099010	0.465347
60	0.128713	0.207921	0.554455
50	0.158416	0.297030	0.623762

Table 3. Kolmogorov-Smirnov (KS) test statistics for varying Q-factors used to JPEG compress the Boat image

	Mode		
Q-factor	(0, 1)	(1, 0)	(1, 1)
100	0.217822	0.227723	0.099010
90	0.188119	0.158416	0.108911
80	0.118812	0.089109	0.217822
70	0.118812	0.108911	0.465347
60	0.138614	0.148515	0.554455
50	0.217822	0.277228	0.643564

5 A Novel Method to Estimate Historical JPEG Q-Factor in Previously Compressed Bitmap Images

As indicated in Section 1, JPEG encoding quantifies the extent of compression with an integral quantity known as Q-factor. Q-factor generally ranges from 1 to 100, and each value of Q-factor maps to a specific quantization table. A Q-factor of 100 traditionally indicates nearly no compression, while a Q-factor of 1 indicates the highest amount of compression. The actual value of the quantization step is generally implementation specific, but tables suggested by the independent JPEG group (IJG) in Annex K of [18] are commonly used. Since the first set of results in this paper is from detecting historical IJG Q-factor in bitmaps, we use the terms Q-factor and IJG Q-factor interchangeably. The approach suggested in this section can however be easily adapted to detect non-IJG Q-factors as well, since the algorithm is independent of the quantization table employed. For example, Adobe Photoshop optionally offers a set of quality levels ranging from 0 to 12, with 12 representing the highest quality. The quantization tables used for these quality levels are non-IJG, and in Section 5.3, we show that the proposed algorithm successfully detects the quality factor used in Photoshop's JPEG compression.

5.1 Algorithm

The PMF developed in (6) is used to estimate historical JPEG quality factor in bitmap images that are known to have been JPEG encoded and decoded in the past. A candidate bitmap image is considered for Q-factor detection, and a 2D DCT is performed on it. A specific low-frequency mode (i.e., (0,1), (1,0) or (1,1)) is extracted and a normalized histogram is performed on it. From this histogram, parameters for the PMF in (6) are derived. Samples of the PMF are then generated for Q-factors ranging from 10 to 100 in discrete steps, and are compared against the normalized histogram previously extracted, using a Chi-squared goodness-of-fit test [19]. The Q-factor corresponding to the lowest value of the generated Chi-squared statistic is the Q-factor of the most recent historical JPEG compression. The Chi-squared

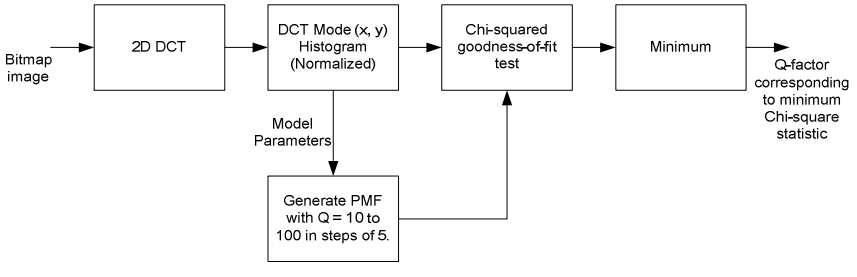


Fig. 4. Historical JPEG compression Q-factor is detected by comparing the generated quantized 2D DCT PMF against samples of the normalized histogram of a specific mode of the 2D DCT of the bitmap image

goodness-of-fit test is preferred over the Kolmogorov-Smirnov goodness-of-fit test because the latter has a relatively small range of [0,1], thereby causing increased difficulty in locating the required minimum.

In the experiment carried out, the (1,1) mode is used to estimate PMF parameters. The performance of this method in terms of success rate of detection is shown in Figure 5 for compression Q-factors ranging from 10 to 100 in steps for 5. The tests have been run over forty well-known images from the USC-SIPI database [20]. It is seen that the performance peaks for Q-factors ranging from 30 to 95, and tails off at the extremes. The reasoning behind the reduced performance at values close to 100 is that the detected Q-factor ranges between 95 and 100. Indeed, it has been found in these ranges that the error in detection is generally no more than ±5. At Q-factor values close to 0, the PMF derived in (6) is inaccurate enough to affect the detection. Similar to Q-factor values close to 100, the error in detection is generally no greater than ±5.

The choice of a single mode of the quantized DCT is found to lead to fairly high success rates of detection. While this paper does not explore the use of more than one mode for JPEG Q-factor detection, it may possibly improve detection performance at the cost of increased computation. It must be noted though, that the modes used for detection are invariably low-frequency modes. This is because at high levels of quantization, i.e., Q-factors of 60 and below, higher frequency modes, say, modes (0, 5) or (1, 6) tend to have an integer value of 0. This restricts the modes chosen for detection to the ones corresponding to lower frequencies.

Experiments have shown that in the case of images with more than a single compression history, i.e., double or multiple compressed images, this procedure most often detects the most dominant JPEG compression. That is, if an image is first compressed with a Q-factor of 30, and is then followed up with a compression Q-factor of 90, the detected Q-factor will be 30. This is attributed to the irreversible, lossy quantization caused by the very low Q-factor used in the first compression. In the case when the first compression is performed with a Q-factor of 90, and is followed up by a compression with Q-factor of 30, the detected Q-factor will still be 30, owing to the same reason as the previous case. Results are tabulated below for a few arbitrary Q-factors in sequence.

Table 4. Detected Q-factors on four double-compressed images, where Q1 and Q2 have arbitrary values

Image	Q1	Q2	Detected Q-factor
Lena	80	70	71
Baboon	40	70	42
Peppers	55	20	22
Boat	70	35	37
Bridge	45	90	50

Table 5. Detected Q-factors on five double-compressed images, where Q1 and Q2 are separated by a value of 5

Image	Q1	Q2	Detected Q-factor
Lena	80	85	84
Lena	85	80	80
Baboon	70	75	76
Baboon	75	70	71
Peppers	40	45	50
Peppers	45	40	42
Boat	30	35	37
Boat	35	30	32
Bridge	20	25	27
Bridge	25	20	22

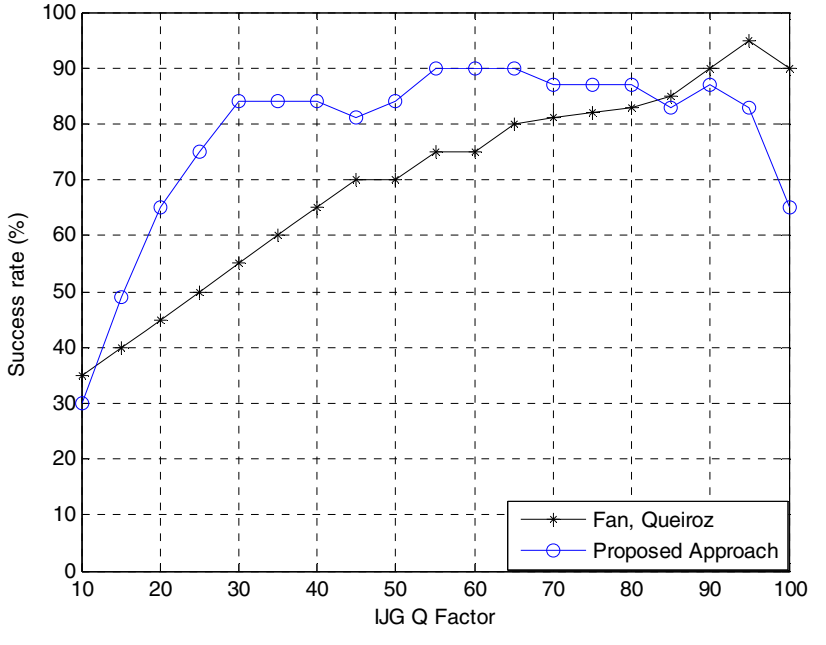


Fig. 5. Success rate of IJG Q-factor detection for the proposed approach and the approached proposed by Fan and Queiroz [1] are compared. It is seen that for a large number of Q-factors, the detection rate is better in the proposed approach.

In the case of comparably close Q-factors, such as 80 followed by 85, or vice versa, the histogram of the DCT mode shows periodic artifacts as indicated in [21]. In this case, the detected Q-factor is generally closer to that of the most recent compression, implying that the order of compression is important. The following table lists the detected Q-factor when an image is compressed with two closely spaced Q-factors.

Thus, in most cases above, the most recent compression Q-factor is reported as the detected quality factor. It may be possible that the periodicity in the DCT Q-factor histogram as detailed in [21] is exploited to detect the presence of double-compression when the Q-factors of compression are comparable in magnitude. Detection of double-compression is however beyond the scope of this paper.

5.2 Comparison with a Classical Approach

The graph in Figure 5 compares the success rate of the proposed approach against that of the approach in [1]. It is evident that the performance of the proposed approach is generally better than that of [1] for Q-factors in the range of 20 to 80. As detailed in the previous subsection, even when the success rate is not high, the error of detection in the proposed approach is no greater than about ±5.

5.3 Non-IJG Quality Factor Detection

Adobe Photoshop uses a JPEG quality factor which ranges from 0 to 12, with 0 representing the lowest quality and 12 the highest [22]. The quantization tables used are different from the IJG tables, and are listed below for each quality factor ranging from 0 to 12 in a sequential manner. The tables were obtained by analyzing JPEG files compressed by Photoshop using JPEGsnoop [23].

$$\begin{bmatrix} 32 & 33 & 51 & 81 & 66 & 39 & 34 & 17 \\ 33 & 36 & 48 & 47 & 28 & 23 & 12 & 12 \\ 51 & 48 & 47 & 28 & 23 & 12 & 12 & 12 \\ 81 & 47 & 28 & 23 & 12 & 12 & 12 & 12 \\ 66 & 28 & 23 & 12 & 12 & 12 & 12 & 12 \\ 39 & 23 & 12 & 12 & 12 & 12 & 12 & 12 \\ 34 & 12 & 12 & 12 & 12 & 12 & 12 & 12 \\ 17 & 12 & 12 & 12 & 12 & 12 & 12 & 12 \end{bmatrix} \quad \begin{bmatrix} 27 & 26 & 41 & 65 & 66 & 39 & 34 & 17 \\ 26 & 29 & 38 & 47 & 28 & 23 & 12 & 12 \\ 41 & 38 & 47 & 28 & 23 & 12 & 12 & 12 \\ 65 & 47 & 28 & 23 & 12 & 12 & 12 & 12 \\ 66 & 28 & 23 & 12 & 12 & 12 & 12 & 12 \\ 39 & 23 & 12 & 12 & 12 & 12 & 12 & 12 \\ 34 & 12 & 12 & 12 & 12 & 12 & 12 & 12 \\ 17 & 12 & 12 & 12 & 12 & 12 & 12 & 12 \end{bmatrix}$$

$$\begin{bmatrix} 20 & 17 & 26 & 41 & 51 & 39 & 34 & 17 \\ 17 & 18 & 24 & 39 & 28 & 23 & 12 & 12 \\ 26 & 24 & 32 & 28 & 23 & 12 & 12 & 12 \\ 41 & 39 & 28 & 23 & 12 & 12 & 12 & 12 \\ 51 & 28 & 23 & 12 & 12 & 12 & 12 & 12 \\ 39 & 23 & 12 & 12 & 12 & 12 & 12 & 12 \\ 34 & 12 & 12 & 12 & 12 & 12 & 12 & 12 \\ 17 & 12 & 12 & 12 & 12 & 12 & 12 & 12 \end{bmatrix} \quad \begin{bmatrix} 18 & 14 & 14 & 21 & 30 & 35 & 34 & 17 \\ 14 & 16 & 16 & 19 & 26 & 23 & 12 & 12 \\ 14 & 16 & 17 & 21 & 23 & 12 & 12 & 12 \\ 21 & 19 & 21 & 23 & 12 & 12 & 12 & 12 \\ 30 & 26 & 23 & 12 & 12 & 12 & 12 & 12 \\ 35 & 23 & 12 & 12 & 12 & 12 & 12 & 12 \\ 34 & 12 & 12 & 12 & 12 & 12 & 12 & 12 \\ 17 & 12 & 12 & 12 & 12 & 12 & 12 & 12 \end{bmatrix}$$

$$\begin{bmatrix} 16 & 11 & 11 & 16 & 23 & 27 & 31 & 17 \\ 11 & 12 & 12 & 15 & 20 & 23 & 12 & 12 \\ 11 & 12 & 13 & 16 & 23 & 12 & 12 & 12 \\ 16 & 15 & 16 & 23 & 12 & 12 & 12 & 12 \\ 23 & 20 & 23 & 12 & 12 & 12 & 12 & 12 \\ 27 & 23 & 12 & 12 & 12 & 12 & 12 & 12 \\ 31 & 12 & 12 & 12 & 12 & 12 & 12 & 12 \\ 17 & 12 & 12 & 12 & 12 & 12 & 12 & 12 \end{bmatrix} \begin{bmatrix} 12 & 8 & 8 & 12 & 17 & 21 & 24 & 17 \\ 8 & 9 & 9 & 11 & 15 & 19 & 12 & 12 \\ 8 & 9 & 10 & 12 & 19 & 12 & 12 & 12 \\ 12 & 11 & 12 & 21 & 12 & 12 & 12 & 12 \\ 17 & 15 & 19 & 12 & 12 & 12 & 12 & 12 \\ 21 & 19 & 12 & 12 & 12 & 12 & 12 & 12 \\ 24 & 12 & 12 & 12 & 12 & 12 & 12 & 12 \\ 17 & 12 & 12 & 12 & 12 & 12 & 12 & 12 \end{bmatrix}$$

$$\begin{bmatrix} 8 & 6 & 6 & 8 & 12 & 14 & 16 & 17 \\ 6 & 6 & 6 & 8 & 10 & 13 & 12 & 12 \\ 6 & 6 & 7 & 8 & 13 & 12 & 12 & 12 \\ 8 & 8 & 8 & 14 & 12 & 12 & 12 & 12 \\ 12 & 10 & 13 & 12 & 12 & 12 & 12 & 12 \\ 14 & 13 & 12 & 12 & 12 & 12 & 12 & 12 \\ 16 & 12 & 12 & 12 & 12 & 12 & 12 & 12 \\ 17 & 12 & 12 & 12 & 12 & 12 & 12 & 12 \end{bmatrix} \begin{bmatrix} 10 & 7 & 7 & 10 & 15 & 18 & 20 & 17 \\ 7 & 8 & 8 & 10 & 13 & 16 & 12 & 12 \\ 7 & 8 & 8 & 10 & 16 & 12 & 12 & 12 \\ 10 & 10 & 10 & 18 & 12 & 12 & 12 & 12 \\ 15 & 13 & 16 & 12 & 12 & 12 & 12 & 12 \\ 18 & 16 & 12 & 12 & 12 & 12 & 12 & 12 \\ 20 & 12 & 12 & 12 & 12 & 12 & 12 & 12 \\ 17 & 12 & 12 & 12 & 12 & 12 & 12 & 12 \end{bmatrix}$$

$$\begin{bmatrix} 6 & 4 & 4 & 6 & 9 & 11 & 12 & 16 \\ 4 & 5 & 5 & 6 & 8 & 10 & 12 & 12 \\ 4 & 5 & 5 & 6 & 10 & 12 & 12 & 12 \\ 6 & 6 & 6 & 11 & 12 & 12 & 12 & 12 \\ 9 & 8 & 10 & 12 & 12 & 12 & 12 & 12 \\ 11 & 10 & 12 & 12 & 12 & 12 & 12 & 12 \\ 12 & 12 & 12 & 12 & 12 & 12 & 12 & 12 \\ 16 & 12 & 12 & 12 & 12 & 12 & 12 & 12 \end{bmatrix} \begin{bmatrix} 4 & 3 & 3 & 4 & 6 & 7 & 8 & 10 \\ 3 & 3 & 3 & 4 & 5 & 6 & 8 & 10 \\ 3 & 3 & 3 & 4 & 6 & 9 & 12 & 12 \\ 4 & 4 & 4 & 7 & 9 & 12 & 12 & 12 \\ 6 & 5 & 6 & 9 & 12 & 12 & 12 & 12 \\ 7 & 6 & 9 & 12 & 12 & 12 & 12 & 12 \\ 8 & 8 & 12 & 12 & 12 & 12 & 12 & 12 \\ 10 & 10 & 12 & 12 & 12 & 12 & 12 & 12 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 2 & 2 & 2 & 3 & 4 & 5 & 6 \\ 2 & 2 & 2 & 2 & 3 & 4 & 5 & 6 \\ 2 & 2 & 2 & 2 & 4 & 5 & 7 & 9 \\ 2 & 2 & 2 & 4 & 5 & 7 & 9 & 12 \\ 3 & 3 & 4 & 5 & 8 & 10 & 12 & 12 \\ 4 & 4 & 5 & 7 & 10 & 12 & 12 & 12 \\ 5 & 5 & 7 & 9 & 12 & 12 & 12 & 12 \\ 6 & 6 & 9 & 12 & 12 & 12 & 12 & 12 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 & 2 & 2 & 2 & 3 \\ 1 & 1 & 1 & 1 & 1 & 2 & 2 & 3 \\ 1 & 1 & 1 & 1 & 2 & 3 & 4 & 5 \\ 1 & 1 & 1 & 2 & 3 & 4 & 5 & 7 \\ 2 & 1 & 2 & 3 & 4 & 5 & 7 & 8 \\ 2 & 2 & 3 & 4 & 5 & 7 & 8 & 8 \\ 2 & 2 & 4 & 5 & 7 & 8 & 8 & 8 \\ 3 & 3 & 5 & 7 & 8 & 8 & 8 & 8 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 2 \\ 1 & 1 & 1 & 1 & 1 & 1 & 2 & 2 \\ 1 & 1 & 1 & 1 & 1 & 2 & 2 & 3 \\ 1 & 1 & 1 & 1 & 2 & 2 & 3 & 3 \\ 1 & 1 & 1 & 2 & 2 & 3 & 3 & 3 \\ 1 & 1 & 2 & 2 & 3 & 3 & 3 & 3 \end{bmatrix}$$

The following table lists the success rate of detection for all quality factors in the Photoshop JPEG compression, for eight images from the USC-SIPI miscellaneous image database [20].

Table 6. Detection success rates for eight images from the USC-SIPI miscellaneous database, when compressed with Photoshop quantization tables

Photoshop quality factor	Detection success rate (%)
12	50
11	50
10	100
9	100
8	100
7	100
6	100
5	100
4	100
3	100
2	100
1	100
0	100

It is of note that the detection success rates at quality factors of 11 and 12 are 50 %, since the algorithm cannot differentiate between the two quality factors due to the similarity of the quantization tables.

5.4 JPEG Compression Presence Detection

The algorithm presented in Section 5.1 may be adapted to detecting historical JPEG compression in bitmap images. Indeed, if a Q-factor of lower than 100 is detected, the image has a very large likelihood of having compression history. However, if the image had been historically JPEG compressed with a Q-factor of 100, the algorithm fails to detect compression history. However, since a Q-factor of 100 represents practically no compression, the proposed algorithm may represent an acceptable JPEG compression presence detection technique.

6 Conclusions

This paper presents a new model for the probability mass function (PMF) of quantized AC block DCT coefficients, and uses it to detect potential historical JPEG Q-factors in bitmap images. To summarize,

1) A novel model for the probability mass function (PMF) of quantized AC block DCT coefficients has been derived starting from a well-known probability density function for non-quantized AC block DCT coefficients and the statistical analysis of quantization.
2) The PMF thus derived is shown to fit empirical data well for Q-factors as low as 50, using Kolmogorov-Smirnov test statistics.
3) A new approach to detecting potential historical JPEG compression in bitmap images using the derived PMF is proposed. The approach is shown to have a compression Q-factor detection success rate of over 81 % in the range [30, 90] for IJG compression, and a success rate of 100 % in the range [0, 10] for non-IJG (Photoshop) compression.
4) The proposed approach to detecting historical Q-factor is found to outperform the approach proposed in [1], for a large range of Q-factors.
5) In the case of a bitmap image having been historically JPEG compressed more than once, the proposed approach detects either the most recent historical compression or the most dominant compression, depending on how close the quality factors of historical compression are. Here, 'dominant compression' indicates the lowest quality factor, implying therefore the highest amount of compression. When the magnitudes of the quality factors of historical compression are close to each other, i.e., with the difference being no greater than about 5 (for IJG Q-factors), the detected quality factor is that of the most recent compression. When the quality factors are sufficiently different in value, the detected quality factor is that of the most dominant compression.

References

1. Fan, Z., de Queiroz, R.: Identification of bitmap compression history: JPEG detection and quantizer estimation. IEEE Transactions on Image Processing 12, 230–235 (2003)
2. Neelamani, R., de Queiroz, R., Fan, Z., Baraniuk, R.G.: JPEG compression history estimation for color images. In: International Conference on Image Processing, vol. (3) (2003)
3. Fu, D., Shi, Y.Q., Su, W.: A generalized Benford's law for JPEG coefficients and its applications in image forensics. In: Proceedings of SPIE, Security, Steganography and Watermarking of Multimedia Contents IX, San Jose, USA (January 2007)
4. Lin, Z., He, J., Tang, X., Tang, C.: Fast, automatic and fine-grained tampered JPEG image detection via DCT coefficient analysis. Pattern Recognition 42(11), 2492–2501 (2009)
5. Püschel, M., Moura, J.M.: The Algebraic Approach to the Discrete Cosine and Sine Transforms and Their Fast Algorithms. SIAM Journal on Computing 32(5), 1280–1316 (2003)
6. Jain, A.K.: Fundamentals of Digital Image Processing. Prentice Hall, Englewood Cliffs (1989)
7. Reininger, R., Gibson, J.: Distribution of the two-dimensional DCT coefficients for images. IEEE Transactions on Communications 31(6) (1983)
8. Kay, S.M.: Intuitive probability and random processes using MATLAB, 1st edn. Springer Science and Business Media, New York (2005)
9. Lam, E., Goodman, J.A.: A Mathematical Analysis of the DCT Coefficient Distributions for Images. IEEE Transactions on Image Processing 9(10), 1661–1666 (2000)

10. Muller, F.: Distribution Shape of Two-Dimensional DCT Coefficients of Natural Images. Electronics Letters 29, 1935–1936 (1993)
11. Nadarajah, S.: A generalized normal distribution. Journal of Applied Statistics 32(7), 685–694 (2005)
12. Widrow, B., Kollár, I., Liu, M.C.: Statistical theory of quantization. IEEE Transactions on Instrumentation and Measurement 45(2), 353–361 (1996)
13. Robertson, M., Stevenson, R.: DCT quantization noise in compressed images. IEEE Transactions on Circuits and Systems for Video Technology 15, 27–38 (2005)
14. Gerald, C.F., Wheatley, P.O.: Applied Numerical Analysis, 7th edn. Addison-Wesley, Reading (2003)
15. Kreyszig, E.: Differential Geometry (Paperback), 1st edn. Dover Publications, New York (1991)
16. Weisstein, E.W.: Kolmogorov-Smirnov Test. From MathWorld–A Wolfram Web Resource, http://mathworld.wolfram.com/Kolmogorov-SmirnovTest.html (retrieved March 22, 2010)
17. http://sipi.usc.edu/database/misc/4.2.04.tiff
18. ISO/IEC 10918-1:1994, Information technology – Digital compression and coding of continuous-tone still images: Requirements and guidelines, February 15 (1994)
19. Weisstein, E.W.: Chi-Squared Test. From MathWorld–A Wolfram Web Resource, http://mathworld.wolfram.com/Chi-SquaredTest.html (retrieved March 30, 2010)
20. http://sipi.usc.edu/database/database.cgi?volume=misc
21. Popescu, A.C., Farid, H.: Statistical Tools for Digital Forensics. In: Proceedings of the Sixth International Workshop on Information Hiding, pp. 128–147 (2004)
22. http://livedocs.adobe.com/en_US/Photoshop/10.0/help.html?content=WSfd1234e1c4b69f3e0a53e41001031ab64-7426.html
23. http://www.impulseadventure.com/photo/jpeg-snoop.html

A Smart Phone Image Database for Single Image Recapture Detection

Xinting Gao[1,⋆], Bo Qiu[1], JingJing Shen[2], Tian-Tsong Ng[1], and Yun Qing Shi[3]

[1] Institute for Infocomm Research, A*STAR, Singapore
{xgao,qiubo,ttng}@i2r.a-star.eud.sg
[2] Department of Electrical and Computer Engineering
National University of Singapore, Singapore
shenjingjing89@gmail.com
[3] Department of Electrical and Computer Engineering
New Jersey Institute of Technology, USA
shi@njit.edu

Abstract. Image recapture detection (IRD) is to distinguish real-scene images from the recaptured ones. Being able to detect recaptured images, a single image based counter-measure for rebroadcast attack on a face authentication system becomes feasible. Being able to detect recaptured images, general object recognition can differentiate the objects on a poster from the real ones, so that robot vision is more intelligent. Being able to detect recaptured images, composite image can be detected when recapture is used as a tool to cover the composite clues. As more and more methods have been proposed for IRD, an open database is indispensable to provide a common platform to compare the performance of different methods and to expedite further research and collaboration in the field of IRD.

This paper describes a recaptured image database captured by smart phone cameras. The cameras of smart phones represent the middle to low-end market of consumer cameras. The database includes real-scene images and the corresponding recaptured ones, which targets to evaluate the performance of image recapture detection classifiers as well as provide a reliable data source for modeling the physical process to obtain the recaptured images. There are three main contributions in this work. Firstly, we construct a challenging database of recaptured images, which is the only publicly open database up to date. Secondly, the database is constructed by the smart phone cameras, which will promote the research of algorithms suitable for consumer electronic applications. Thirdly, the contents of the real-scene images and the recaptured images are in pair, which makes the modeling of the recaptured process possible.

Keywords: image database, image recapture detection, image forensics.

⋆ The authors would like to thank Dao Wei Lim, Alvin Neo, Te Ye Yang, Quan Yang Yeo, Boon Siong Tan, Jun Ting Lee, Kan Xian Yang, and Jun Xuan Ng for their effort on the data collection. The authors would also like to thank Yan Gao, Xinqi Chu and Patchara Sutthiwan for their valuable discussions. This work is done when JingJing Shen is working in I^2R for her internship. The work is supported by A*STAR Mobile Media Thematic Strategic Research Program of Singapore.

H.-J. Kim, Y. Shi, and M. Barni (Eds.): IWDW 2010, LNCS 6526, pp. 90–104, 2011.

1 Introduction

It is necessary to compare different algorithms in a fair way. Through a fair comparison, researchers can get insight of the performance, pros and cons of each method quickly and understand the research problem deeply. For the research in the field of image processing, a common database is one of the basic factors to provide such comparison platform in nearly every research problem. The design and implementation of the database itself also show the specific properties of the problem, which expedites further research and collaboration. Based on the above motivations, we construct a smart phone recaptured database for single image recapture detection problem in this work.

IRD is to distinguish images of real scenes from the recaptured images, i.e., images of media that display real-scene images such as printed pictures or LCD display. One of the important applications of IRD is in face authentication system. Due to the convenience of face authentication, some companies begin to produce face recognition based access control system, ATM face verification system and face recognition based PC/smart phone security system [1,2] etc. However, faked identity through recapturing of a faked print face photo has become a great security concern for such systems. To protect the user from such kind of rebroadcast attack, IRD is an effective way. IRD is also useful for general object recognition to differentiate the objects on a poster from the real ones, which improves the intelligence of robot vision. Another important application for IRD is in composite image detection . One way to cover composition in an composite image is to recapture it. With IRD, such composite method can be detected. As more and more research has been done to detect recaptured images from real ones, it is necessary to have a common database to compare these methods. In this work, we present a smart phone image database for IRD, which is intended to fulfill such necessity and further promote the research and collaboration in the IRD field.

The constructed smart phone IRD database consists of two classes of images, i.e., real-scene images and the corresponding recaptured images taken by five types of smart phone cameras. Fig. 1 shows some examples from the database. The first row of Fig. 1 are the real-scene images, and the second row of Fig. 1 are the corresponding recaptured ones. The examples demonstrate that it is even

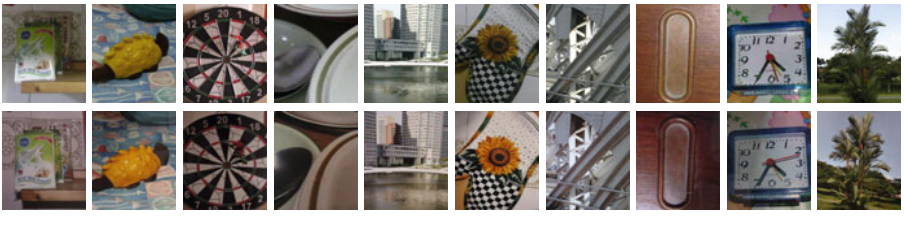


Fig. 1. Example images from the database. The first row of images are the real-scene images, while the second row of the images are the corresponding recaptured ones.

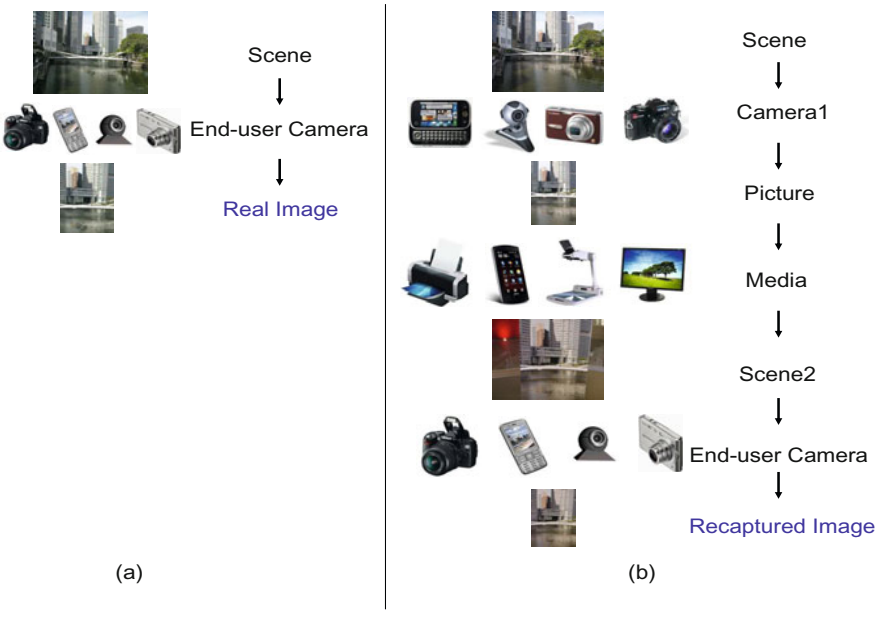


Fig. 2. The processes to produce (a) the real-scene images and (b) the recaptured images

difficult for human visual system to detect the recaptured images from the real ones for most of the cases.

The processes to produce the real-scene images and the recaptured images are shown in Fig. 2 (a) and Fig. 2(b) respectively. The real-scene images can be obtained through any type of cameras, e.g. Acer M900 back-facing camera. To get the corresponding recaptured images, reproduction process is needed. In the reproduction process, the real scene is first captured by any type of cameras, e.g. a DSLR (digital single-lens reflex) camera in our work, then reproduced using different types of display or printing, such as displayed on an LCD screen of a PC, or printed on an office A4 paper using a color laser printer etc. The recaptured images are obtained through the corresponding cameras utilized in the process to produce the real-scene images.

There are three main contributions of the work. Firstly, the recaptured database is the first publicly accessible database for IRD, which provides a common platform to compare different algorithms for image recapture detection problem. Secondly, both the recaptured dataset and the corresponding real-scene image dataset are constructed by the same end-user devices, i.e., the smart phone cameras, which will promote the research of algorithms suitable for consumer electronic applications. Thirdly, the contents of the real-scene images and the recaptured ones are in pair, which makes the modeling of the recaptured process possible. In this work, we perform some experiments and analysis on the presented database to show the high-quality of the database.

The outline of the paper is as follows. In Sec. 2, we will review the existing IRD methods and the existing database related to the IRD problem, which will manifest the reason why the presented database is needed. Sec. 3 proposes the design criteria of the database and Sec. 4 describes the implementation process of the database. The post processing to improve the quality of the database is presented in Sec. 5. Sec. 6 describes the structure of the database. The experiments on the database are discussed in Sec. 7 followed by the conclusion in Sec. 8.

2 Prior Work

2.1 Scanned Image Detection

In [5], Farid and Lyu performed an experiment on classifying photographic images and scanned images using wavelet statistical features which capture the deviation from the normal image statistics resulted from the image printing and scanning process. In their work, the real images are printed using a laser printer and scanned with a flat-bed scanner at 72dpi, which results in the rebroadcast images. 1000 real images and 200 rebroadcast ones are used in the experiment. The method could classify 99.5% of the real images and 99.8% of the rebroadcast ones. This method is statistical in nature therefore it does not admit to physical intuitions that can lead to further ideas on improving the method. In their work, the recaptured image is obtained by a scanner, the illumination of which is fully controlled and fixed.

2.2 Recaptured Photorealistic Computer Graphic Image Detection

In [10], Ng et al. devised a set of physics-motivated geometry features to classify photographic images and photorealistic computer graphic images. Local fractal dimension and local patches are used as the features at the finest scale of an images and surface gradient, and the second fundamental form and the Beltrami flow vectors are used to characterize the properties of images at an intermediate scale. They showed that the geometry features have the capability of distinguishing computer graphics recaptured from an LCD display when the recaptured images have a resolution high enough to resolve the grid structure of the display. The authors construct a database consisting 1600 real images, 800 photorealistic computer graphic images and the corresponding 800 recaptured computer graphic images displayed on an LCD screen. The recaptured photorealistic computer graphic images are different from the recaptured real-scene ones from both the imaging-process authenticity and the scene authenticity. For imaging-process authenticity, recaptured content of a recaptured image is a result of real imaging process, while the recaptured content of a recaptured computer graphic image is a result of simpler model of the physical light transport.

2.3 Image Recapture Detection

In [11], Yu et al. proposed a way to extract the micro-texture on the A4-size printing paper from the specularity component of a recaptured image. Later

in [3], Bai et al. assessed the method on a dataset consisting of 65 human face images where the method showed 2.2% False Acceptance Rate and 13% False Rejection Rate of the recaptured images. The method proposed by Yu et al. requires the input images to have a high enough resolution (generally high-resolution DSLR images) in order to resolve the fine micro-texture of printing papers and its performance may be unreliable on images other than face images with limited amount of fine texture.

In [7], Gao et al. proposed a general physical model for the recaptured process. A set of physical features is inspired by the model such as the contextual background information, the spatial distribution of specularity that is related to the surface geometry, the image gradient that captures the non-linearity in the recaptured image rendering process, the color information and contrast that is related to the quality of reproduction rendering, and a blurriness measure that is related to the recapturing process. The physical features are used to classify the recaptured images from the real ones. A recaptured database of smart phone images is used in the performance evaluation and the physical method demonstrates promising result. The database presented in this paper is an improvement version of the database presented in [7], in both quantity and quality aspects of view.

2.4 Steganalysis

In [4], Chen and Shi presented a statistical method for steganalysis. Both the intrablock and interblock correlations among JPEG coefficients are modeled by an Markov Process. Transition Probability matrix are computed and the elements of the matrix are used as the features. In their work, 7560 JPEG images with quality factors ranging from 70 to 90 are used in the experiments. The classifier demonstrates very good performance on attacking several existing steganographic methods. If we consider the steganalysis and recapture detection as recognition problems, both of them have one same class, i.e., the real-scene images. Steganography changes the statistics of the original real-scene images, so does the recaptured process. As the steganalysis method measures the statistics deviation from the real-scene images, we believe that it is also suitable for recaptured detection.

In this work, we will test the database using the above two statistical methods presented in [4] and [5], and two physical methods presented in [7] and [10].

2.5 Existing Databases

In the above publications, they use different databases for experiment in their work and the only publically available database is Ng et al.'s database [9]. However, the database is not suitable for image recapture detection as the content of the image should be real-scene objects instead of computer graphics objects in IRD problem. There are some existing face database [8] or object database [6]. For example, Yale face database is well-know in face recognition field. However, these databases contain only real face/object images. There are no recaptured

datasets correspondingly. To our best knowledge, there is no recaptured database with real-scene contents publically available for image recapture detection. According to our review, it is also found that the image resolution affects the performance of the algorithms [3]. In this work, we present a recapture image database taken by smart phone cameras, whose resolution is mainly set to VGA (640×480 pixels) and the lens is of low quality compared to DSLR cameras.

3 Design Criteria

Image recapture detection is to classify the real-scene images from the recaptured images, whose content will be recognized by the automatic system as the real-scene object otherwise. For example, in a face authentication system, the system may recognize a lifelike printed face photo as the subject himself/herself without recapture detection. Hence, the content of the recaptured image should be vivid and similar as the one of the real-scene image to deceive the automatic recognition system. Generally speaking, an image recapture detection method is evaluated in the following aspects.

a) The accuracy of the image recapture detection.
b) The robustness of the detector to the diverse illumination.
c) The robustness of the detector to the diverse environment/background.
d) The robustness of the detector to the diverse image content. In another word, the detector should be independent to the image content.
e) The robustness of the detector to the diverse reproduction process.

To fulfill the above evaluation, we use three brands of smart phones with five cameras in total as the end-user devices to construct the database. In the recaptured dataset construction, three DSLR cameras are used to capture the real scene. Then three types of printing and two types of displays are used as the reproduction devices. Finally, the same end-user device is used to obtain the recaptured image. There are twelve people involved in the database construction, which increases the diversity in terms of image contents and the photographing styles. We abide by the following criteria when constructing the database.

a) The image contents are roughly in pair for the real-scene image and the recaptured one taken by the same end-user camera.
b) The image contents are as diverse as possible for each end-user camera. The images are captured at different locations, such as offices, residents, campus, downtown and park etc. Both close objects and far away scenes are intentionally captured.
c) The image is taken in indoor and outdoor and at different time during the day to guarantee the diversity of the illumination and environment.

4 Procedure to Produce the Database

It is straightforward to construct the real-scene image dataset according to the criteria stated in Sec. 3. To improve efficiency and achieve the corresponding

recaptured images, we follow the following three steps to produce the real-scene images and recaptured ones.

a) A real-scene is captured with a smart phone camera and a high-end DSLR camera. The resolution is VGA for smart phone cameras and greater than 3000×2000 pixels for DSLR cameras.
b) The DSLR image is displayed on an LCD screen or printed on paper, which is called reproduction process.
c) Using the same smart phone camera, the reproduced image is captured again (capture of the printed/displayed real-scene image), which results in a corresponding recaptured image.

The details of the procedure are described in the following subsections.

4.1 Real Image Dataset

As shown in Fig. 2 (a), the real images can be obtained by any end-user camera which is determined by the application. In this work, we focus on the smart phone cameras which is common in daily life. The resolution is set to VGA in general, which is widely used in applications such as in video surveillance. Three popular brands of smart phones are used, which are Acer M900, Nokia N95, and HP iPAQ hw6960. Acer M900 and Nokia N95 has both front-facing camera and back-facing camera, while HP iPAQ hw6960 has only the back-facing camera. The cameras are set to auto mode whenever possible.

4.2 Recaptured Image Dataset

To get the corresponding recaptured dataset, the same end-user camera should be used for each image pair. As shown in Fig. 2 (b), there exists different reproduction processes to display the real-scene image captured by any camera. In this work, we use a high-end DSLR camera in the first capture. The DSLR camera has a high resolution and high quality, which will produce a vivid real-scene image. The DSLR cameras used in the database construction are Nikon D90, Canon EOS 450D, and Olympus E-520. The resolution is set to be greater than 3000×2000 pixels and the image is saved in JPEG format. The cameras are set to auto mode whenever possible. We use three types of reproduction processes in constructing the recaptured dataset. For the first stage of data collection, we print or display the whole DSLR images to get the recaptured images. The recaptured images contain the white border of the paper. In the second stage, the white border of the printing paper is cut away through a paper cutter, so that the recaptured content is well immerged into the real environment in the recaptured image. As the real-scene environment is included into the recaptured image, the scale of the object in the recaptured image is smaller than the corresponding real-scene image in general for the images obtained in the first two stages. Imaging we are taking a photo of a people's face using a DSLR camera, the face will occupy a portion of the whole image view. The face part is around 1/9 to 1/3 of the whole image, depending on the imaging settings. If the whole

image is printed on an A4 office paper or displayed on an LCD screen, the actual resolution is decreased when we recaptured the face part. Therefore, the face will be cut off from the whole image and printed or displayed as the recaptured content in the recaptured process of face images. To simulate such scenario, the center part of the DSLR image is cropped automatically using Matlab and kept as the scene object for recapturing in the third stage. Generally speaking, the scale for the recaptured images produced in the third stage is larger than the corresponding real-scene ones.

In the third step, the DSLR image displayed on an LCD screen or printed on a paper is recaptured by the same end-user smart phone camera correspondingly used in the capture of the real-scene image. In our work, we adopt diverse reproduction process. For LCD screen display, we use Dell 2007FP LCD screen (1600×1200 pixels), Acer M900 smart phone screen, and iPhone 3GS smart phone screen to display the DSLR image. The image is printed on an A4-size office paper using HP CP3505dn laser printer and Xerox Phaser 8400 ink printer. The images are also send to a specific photo printing shop and are printed into 4R glossy and matte photos, which is the most challenging printing attack in terms of its high quality and vivid color.

5 Post Processing for the Database

After the capturing procedure, we get the raw database. Imaging in an attacking scenario, the attacker would show an attack image that is as vivid as the real object to deceive the system. Therefore, some post processing is needed to make sure that the database is qualified for evaluating the recapture detection methods. The raw database is post processed as follows.

5.1 Post Processing for Recaptured Images

For the recaptured images, three categories of recaptured datasets are obtained due to the post processing, which are recaptured images with real environment background, recaptured images without real environment background, and the image pairs through homography transform and cropping.

Fig. 3. Examples of recaptured images. The first row shows the examples of recaptured images with the real environment as the background. The second row shows the corresponding cropped images of the first row.

a) Recaptured Dataset A - Recaptured Images with Real Environment Background.
In the dataset, there are two types of recaptured printing images with the real environment background. One type is with white border and the other type is without the white border (the white border is cut away to make the recaptured content look more immersive into the real environment background). Some examples of the recaptured images with real environment background are shown in the first row of Fig. 3.
The recaptured images are screened according to its quality and content diversity. An image is removed if its quality is not good enough, for example, the image is blurred due to the shaky hand during the recapturing (see the first two examples in the first row of Fig. 4), or the image is too dark (see the intermediate two examples in the first row of Fig. 4) or too bright (see the last two examples in the first row of Fig. 4) due to the unsuitable illumination or exposure time. To keep the recaptured contents of the images are as diverse as possible, the images with similar background or color are removed, for example, the images of different objects with the same yellow mat as the background, or the images of different types of plants but the majority contents of the images are in green color (see the examples in the second row of Fig. 4). Furthermore, the real environment background should be diverse as well. Some examples of the rejected recaptured images due to the similar real environment background are shown in the third row of Fig. 4.

b) Recaptured Dataset B - Recaptured Images without Real Environment Background.
To compare the properties of the real-scene images with just the recaptured images, we removed the real environment background in the recaptured

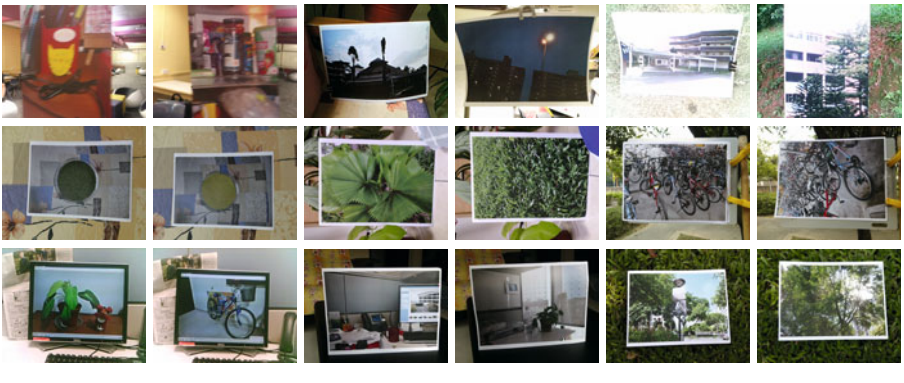


Fig. 4. Examples of rejected recaptured images. The first row shows the examples of the rejected recaptured images with low quality, i.e., the first two images are blur, the intermediate two are too dark and the last two are too bright. The second row shows the examples of the rejected recaptured images with similar contents or color. The third row shows the examples of the rejected recaptured images with similar real-scene environment background.

images. Matlab function, 'imcrop', is used to manually crop the recaptured content of the recaptured images as shown in the second row of Fig. 3. The cropped images are also screened according to its quality and content diversity of the images as we did for Dataset A.

c) Recaptured Dataset C - Image Pairs through Homography Transform and Cropping.
As stated in previous sections, the scales and view angles of the real-scene images are roughly aligned with the first capture of the recaptured ones. For each pair, a homography transform between the real-scene image and the recaptured one is computed. After transformation, the images are cropped to have the same central parts. For such pairs, the geometry distortion is removed and the photometric distortion is remained between the real-scene images and the recaptured ones. Some examples are shown in Fig. 1.

5.2 Post Processing for Real-Scene Images

Corresponding to the post processing applied to the recaptured images, the real-scene images are post processed as follows.

a) Real Dataset A - Real Images Corresponding to the Recaptured Images with Real Environment Background.
The real-scene images are screened according to its quality and content diversity as we stated in previous subsection.

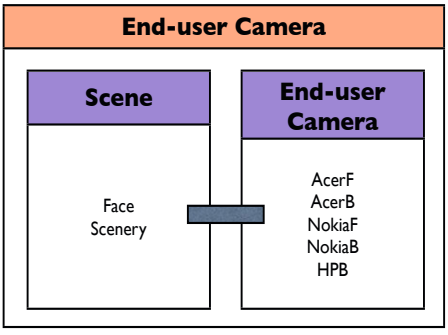
b) Real Dataset B - Real Images Corresponding to the Recaptured Images without Real Environment Background.
The images in Real Dataset A are cropped using Matlab. The sizes of the images are randomly chosen from the sizes of the images in the corresponding Recaptured Dataset B.

c) Real Dataset C - Image Pairs through Homography Transform and Cropping.
As the homography transform and cropping may introduce some artifacts into the final images, we applied forward processing (from real to recaptured ones) and backward processing (from recaptured to real ones) alternatively to the image pairs.

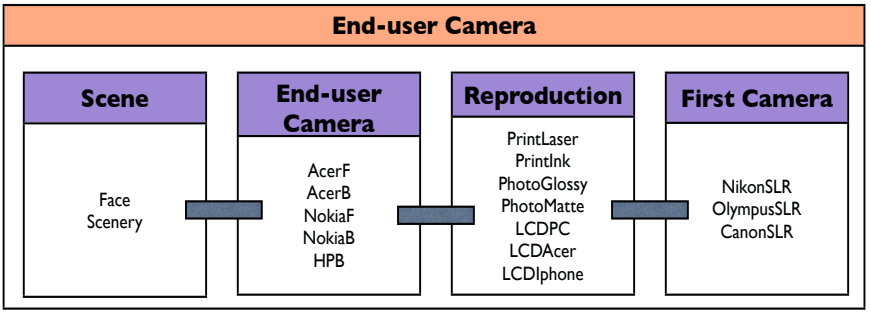
6 Structure of the Database

6.1 Nomenclature

For each of the three datasets, the directory is named according to the content and the devices used to produce the images (refer to Fig. 2). Fig. 5 illustrates the nomenclature for the name of the directory. For example, The folder, "Scenery-AcerB-PhotoGlossy-NikonSLR", contains all the recaptured images, whose first capture is obtained using Nikon SLR camera and then printed in a specific photo shop on glossy photo paper, finally the recaptured is obtained using the back-facing camera of Acer smart phone.



(a) The two-level directory structure and directory naming convention for real-scene images. For example, Face-AcerF images are kept in Face-AcerF directory, a subdirectory of AcerF directory.



(b) The two-level directory structure and directory naming convention for recaptured images. For example, Face-AcerF-LCDPC-NikonSLR images are kept in a directory with the same name which is under AcerF directory.

Fig. 5. Nomenclature illustration

Table 1. The number of real-scene images captured and screened

Types	Captured	Rejected	Rejected(%)	Selected	Selected(%)
AcerB	752	345	45.88%	407	54.12%
HPB	647	278	42.97%	369	57.03%
NokiaB	841	523	62.19%	318	37.81%
Total	2240	1146	51.16%	1094	48.84%

6.2 Dataset A - Recaptured Images with Real Environment Background

a) Real-scene Images.
So far, we finished the capturing and screening of the real-scene images for the three back-facing smart phone cameras. Table 1 lists the numbers of captured real-scene images and the selected ones through screening. In summary, we collected 2240 real-scene images using the three smart phone

Table 2. The number of recaptured images with real environment background captured and screened

Types	Captured	Rejected	Rejected(%)	Selected	Selected(%)
(a) Scenery-AcerB					
LCDPC-NikonSLR	419	410	97.85%	9	2.15%
PhotoGlossy-NikonSLR	73	53	72.60%	20	27.40%
PhotoGlossy-OlympusSLR	50	20	40.00%	30	60.00%
PhotoMatte-NikonSLR	145	115	79.31%	30	20.69%
PhotoMatte-OlympusSLR	51	14	27.45%	37	72.55%
PrintInk-OlympusSLR	63	13	20.63%	50	79.37%
PrintLaser-NikonSLR	575	482	83.83%	93	16.17%
Scenery-AcerB-in-Total	1376	1107	80.45%	269	19.55%
(b) Scenery-HPB					
LCDPC-NikonSLR	420	414	98.57%	6	1.43%
PhotoGlossy-CanonSLR	214	138	64.49%	76	35.51%
PhotoMatte-CanonSLR	254	181	71.26%	73	28.74%
PrintInk-CanonSLR	95	30	31.58%	65	68.42%
PrintLaser-CanonSLR	228	98	42.98%	130	57.02%
PrintLaser-NikonSLR	411	363	88.32%	48	11.68%
Scenery-HPB-in-Total	1622	1224	75.46%	398	24.54%
(c) Scenery-NokiaB					
PhotoGlossy-OlympusSLR	48	2	4.17%	46	95.83%
PhotoMatte-OlympusSLR	109	29	26.61%	80	73.39%
PrintInk-OlympusSLR	230	61	26.52%	169	73.48%
PrintLaser-NikonSLR	434	399	91.94%	35	8.06%
PrintLaser-OlympusSLR	320	180	56.25%	140	43.75%
Scenery-NokiaB-in-Total	1776	1306	73.54%	470	26.46%
(d) Scenery-in-Total					
Scenery-in-Total	4774	3637	76.18%	1137	23.82%

cameras. Among them, 1146 images are rejected due to the considerations of quality and content diversity, and 1094 images are selected.

b) Recaptured Images with Real Environment Background.
For the recaptured images with real environment background, Table 2 lists the numbers of captured images and selected images through screening. In summary, we collected 4774 recaptured images using the three smart phone cameras. Among them, 3637 images are rejected due to the considerations of quality, content diversity and real environment background diversity, and 1137 images are selected.

6.3 Dataset B - Recaptured Images without Real Environment Background

a) Real-scene Images.
As we stated in section 5, the real-scene images for Dataset B is obtained through Matlab cropping on the real-scene images of Dataset A. The numbers are the same as shown in Table 1.

Table 3. The number of recaptured images without real environment background captured and screened

Types	Captured	Rejected	Rejected(%)	Selected	Selected(%)
(a) Scenery-AcerB					
LCDPC-NikonSLR	419	330	78.76%	89	21.24%
PhotoGlossy-NikonSLR	73	14	19.18%	59	80.82%
PhotoGlossy-OlympusSLR	50	21	42.00%	29	58.00%
PhotoMatte-NikonSLR	145	105	72.41%	40	27.59%
PhotoMatte-OlympusSLR	51	13	25.49%	38	74.51%
PrintInk-OlympusSLR	63	12	19.05%	51	80.95%
PrintLaser-NikonSLR	575	408	70.96%	167	29.04%
Scenery-AcerB-in-Total	1376	903	65.63%	473	34.37%
(b) Scenery-HPB					
LCDPC-NikonSLR	420	224	53.33%	196	46.67%
PhotoGlossy-CanonSLR	214	122	57.01%	92	42.99%
PhotoMatte-CanonSLR	254	86	33.86%	168	66.14%
PrintInk-CanonSLR	95	30	31.58%	65	68.42%
PrintLaser-CanonSLR	228	92	40.35%	136	59.65%
PrintLaser-NikonSLR	411	278	67.64%	133	32.36%
Scenery-HPB-in-Total	1622	832	51.29%	790	48.71%
(c) Scenery-NokiaB					
PhotoGlossy-OlympusSLR	48	2	4.17%	46	95.83%
PhotoMatte-OlympusSLR	109	29	26.61%	80	73.39%
PrintInk-OlympusSLR	230	60	26.09%	170	73.91%
PrintLaser-NikonSLR	434	374	86.18%	60	13.82%
PrintLaser-OlympusSLR	320	174	54.38%	146	45.62%
Scenery-NokiaB-in-Total	1776	1274	71.73%	502	28.27%
(d) Scenery-Crop-in-Total					
Scenery-in-Total	4774	3009	63.03%	1765	36.97%

b) Recaptured Images without Real Environment Background.
For the recaptured images without real environment background, Table 3 lists the numbers of captured images and selected images through screening. In summary, we collected 4774 recaptured images using the three smart phone cameras. Among them, 3009 images are rejected due to the considerations of quality and content diversity, and 1765 images are selected.

6.4 Dataset C - Image Pairs through Homography Transform and Cropping

Table 4 lists the numbers of pairs obtained through transforming and cropping. In summary, we obtained 587 pairs of real and recaptured images with same scene contents.

Table 4. The number of pairs obtained through transforming and cropping

Types	Transformed and Cropped Pairs
Scenery-AcerB-PhotoGlossy-NikonSLR	24
Scenery-AcerB-PhotoGlossy-OlympusSLR	35
Scenery-AcerB-PhotoMatte-NikonSLR	30
Scenery-AcerB-PhotoMatte-OlympusSLR	26
Scenery-AcerB-PrintInk-OlympusSLR	46
Scenery-AcerB-PrintLaser-NikonSLR	48
Scenery-HPB-PhotoGlossy-CanonSLR	60
Scenery-HPB-PhotoMatte-CanonSLR	53
Scenery-HPB-PrintInk-CanonSLR	46
Scenery-NokiaB-PhotoGlossy-OlympusSLR	29
Scenery-NokiaB-PhotoMatte-OlympusSLR	61
Scenery-NokiaB-PrintInk-OlympusSLR	129
Total	587

7 Experiments and Discussions

We group all the real-scene images and recaptured ones for Dataset A, B and C. We compare the performance of the two sets of statistical features, i.e., the wavelet statistical features [5] and the Markov statistics [4] and the two sets of physical features, i.e., the geometry based features [10] and the physical features presented in [7] through SVM classifications of the two image classes (real-scene images and the recaptured ones within each dataset). A 3×3 block structure is used for modeling the background contextual information when extracting the physics-based features [7]. The dimensions of the four sets of features are shown in Table 5. The results shown in Table 5 are the average accuracy for ten independent iterations of SVM classification with random data partition and five-fold cross-validation. In general, the physical features outperform the statistical features although physical features have lower feature dimensions.

Table 5. Dimensions of features and performance on datasets

Features	Dimensions	**Accuracy on**			
		Dataset A	Dataset B	Dataset C	Average
Wavelets statistics [5]	216	80.76%	80.93%	73.78%	78.49%
Markov statisticS [4]	486	81.35%	77.30%	65.85%	74.84%
Geometry based [10]	192	86.31%	89.33%	80.12%	85.26%
Physics [7]	166	91.30%	86.66%	74.88%	84.28%

Without the contextual information, geometry features [10] outperform physical based method [7]. Dataset C is the most challenging one for all IRD methods. The overall average accuracy is also shown in Table 5. The results demonstrate the improved quality of the database comparing to the results shown in [7].

8 Conclusions

In this paper, we described a smart phone image database for image recapture detection. The database includes real-scene images and the corresponding recaptured images taken by smart phone cameras. The database will be available to the research community from http://www1.i2r.a-star.edu.sg/∼ttng/Home.html.

References

1. Korea identification inc., http://www.korea-id.co.kr/eng/index.html
2. XID technologies, http://www.xidtech.com/
3. Bai, J., Ng, T.-T., Gao, X., Shi, Y.-Q.: Is physics-based liveness detection truly possible with a single image? In: IEEE International Symposium on Circuits and Systems, ISCAS (2010)
4. Chen, C., Shi, Y.: Jpeg image steganalysis utilizing both intrablock and interblock correlations. In: IEEE International Symposium on Circuits and Systems (ISCAS), pp. 3029–3032 (2008)
5. Farid, H., Lyu, S.: Higher-order wavelet statistics and their application to digital forensics. In: IEEE Workshop on Statistical Analysis in Computer Vision (2003)
6. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In: IEEE. CVPR 2004, Workshop on Generative-Model Based Vision (2004)
7. Gao, X., Ng, T.-T., Qiu, B., Chang, S.-F.: Single-view recaptured image detection based on physics-based features. In: IEEE International Conference on Multimedia & Expo, ICME (2010)
8. Georghiades, A., Belhumeur, P., Kriegman, D.: From few to many: Illumination cone models for face recognition under variable lighting and pose. IEEE Transactions on Pattern Analysis and Machine Intelligence 23(6), 643–660 (2001)
9. Ng, T.-T., Chang, S.-F., Hsu, Y.-F., Pepeljugoski, M.: Columbia photographic images and photorealistic computer graphics dataset. ADVENT Technical Report 205-2004-5 Columbia University (February 2005)
10. Ng, T.-T., Chang, S.-F., Hsu, Y.-F., Xie, L., Tsui, M.-P.: Physics-motivated features for distinguishing photographic images and computer graphics. In: ACM Multimedia (2005)
11. Yu, H., Ng, T.-T., Sun, Q.: Recaptured photo detection using specularity distribution. In: IEEE International Conference on Image Processing, ICIP (2008)

Detection of Tampering Inconsistencies on Mobile Photos

Hong Cao and Alex C. Kot

School of Electrical and Electronic Engineering, Nanyang Technological University
639798 Jurong West, Singapore
{hcao,eackot}@ntu.edu.sg

Abstract. Fast proliferation of mobile cameras and the deteriorating trust on digital images have created needs in determining the integrity of photos captured by mobile devices. As tampering often creates some inconsistencies, we propose in this paper a novel framework to statistically detect the image tampering inconsistency using accurately detected demosaicing weights features. By first cropping four non-overlapping blocks, each from one of the four quadrants in the mobile photo, we extract a set of demosaicing weights features from each block based on a partial derivative correlation model. Through regularizing the eigenspectrum of the within-photo covariance matrix and performing eigenfeature transformation, we further derive a compact set of eigen demosaicing weights features, which are sensitive to image signal mixing from different photo sources. A metric is then proposed to quantify the inconsistency based on the eigen weights features among the blocks cropped from different regions of the mobile photo. Through comparison, we show our eigen weights features perform better than the eigen features extracted from several other conventional sets of statistical forensics features in detecting the presence of tampering. Experimentally, our method shows a good confidence in tampering detection especially when one of the four cropped blocks is from a different camera model or brand with different demosaicing process.

Keywords: CFA, demosaicing, forensics, image inconsistency, regularity, source identification, tampering detection.

1 Introduction

Mobile cameras are typically low-end cameras attached on handheld devices, e.g. cellular phones and personal digital assistants. Fast proliferation of these mobile devices and the growing popularity of attaching cameras into these devices have made photos captured by these low-end cameras become a main stream [1]. Similar to the high-quality photos acquired by commercial digital still cameras (DSC), these mobile photos can be used as evidences of real happenings in a wide array of applications ranging from citizen journalism [2], police investigation, legal service, insurance claims and consumer photography such as photo blogs and online sharing. As the mobile photos are also subject to the easy fakery by state-of-the-arts image editing tools, their credibility cannot be taken for granted especially in occasions where security is needed. Forensics analyses that can tell their origin and integrity are in urgent needs.

H.-J. Kim, Y. Shi, and M. Barni (Eds.): IWDW 2010, LNCS 6526, pp. 105–119, 2011.

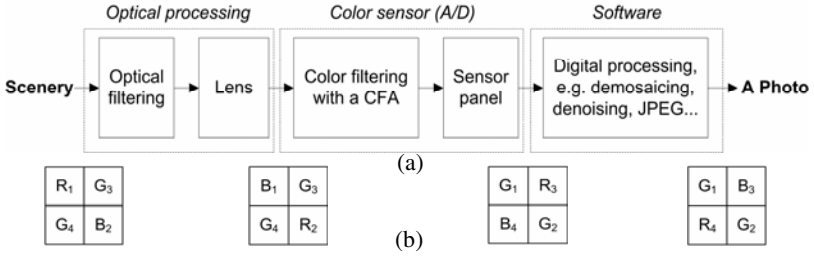


Fig. 1. Mobile camera model in (a) and four Bayer color filter array (CFA) patterns in (b)

In recent years, we have seen a large number of passive image forensics works [4-27], which identify image sources, expose forgeries or both through detecting some intrinsic image regularities or some common tampering anomalies. Many different forms of image regularities and tampering anomalies have been used in the prior works, where one can refer to [20] for the different forensics methodologies. One can also refer to [27] for a comprehensive bibliography. Among the existing tampering forensics works, Swaminathan *et al.* [6] extended their early work [4] on estimation of the underlying color filter array (CFA) pattern and color interpolation coefficients (CIC) for non-intrusive component analysis. By formulating the extrinsic image manipulation as a linear time invariant filtering process, the filter parameters are estimated by a recursive deconvolution technique for detecting different forms of image manipulations. Chen *et al.* [11] improved the pixel response non-uniformity (PRNU) sensor noise model, the preprocessing techniques as well as the PRNU estimation framework to identify individual cameras and to detect integrity tampering. Avcibas *et al.* [16] described a method to quantify image distortions in a way that is less content dependent. These distortions can be used as features for detecting some specific image manipulations. Bayram *et al.* [17] combined three types of forensics features including image quality metrics (IQM), binary similarity (BS) and multi-scale wavelet statistics (MSWS) for detecting some common image manipulations. Ng *et al.* [18] and Chen *et al.* [23] proposed several statistical features to detect the presence of sharp image discontinuities caused by the splicing or photomontage forgery. In view that it is difficult to match the lighting in creating an image forgery, Johnson *et al.* [19] developed a tool to estimate the lighting direction from a point light source. The inconsistent lighting directions can be used as an evidence of tampering. He *et al.* [21] proposed to recompress a JPEG photo with a high quality factor and identify the blocks that do not exhibit double-quantization effect as doctored blocks. Through analyzing the discrete cosine transform (DCT) coefficients, the probability for each block being doctored is estimated and the probability map helps a forensics analyst to visually identify the tampered image region. Fu *et al.* [22] discovered that the first digits of JPEG DCT coefficients closely follow a generalized Benford's law but not for the double JPEG compressed images. The violation of the generalized Benford's law is detected as an evidence of possible image tampering. Luo *et al.* [24] and Barni *et al.* [25] measured the JPEG blocking artifacts for differentiating single-compressed and double-compressed image blocks.

Though each of these works has demonstrated some convincing results, their limitations shall not be overlooked in a practical scenario. Some methods either require knowledge of tampering operations [16] or report less satisfactory results in a blind context [17]. Other methods require the source information, e.g. CIC for [6] or PRNU for [11], or they require a good number of genuine training images from the same source [11, 17]. The splicing detection methods [18, 23] would face difficulties if the splicing boundary is deliberately rendered smooth. For [19], estimation of the lighting directions on noisy image data by itself can be very challenging and error prone. The methods [21, 22, 24, 25] based on detecting double JPEG quantization generally do not work for non-JPEG compressed images and some heavily JPEG compressed images. Moreover, the efficacies of these tampering forensics works are commonly evaluated on the high-quality DSC photos, but not on the low-end mobile photos. Though a mobile camera shares a good similarity with a DSC camera in the skeleton in Fig. 1(a), it is worth to note that a mobile camera is typically ten times cheaper, ten times smaller in physical size and consume ten times less power [1]. The large amount of optical distortions and sensor noises due to the cheap lens and the small sensor size would require denoising technique being implemented in the software unit. The low power consumption would require simple digital processing algorithms, where low complexity is one primary concern. The small storage space also requires smaller image file size, hence a higher default JPEG compression rate. Since these low-end digital processes could render some high-frequency forensics features undetectable, not all tampering forensics techniques for DSC photos can be readily extended to mobile photos. The existing forensics works for mobile cameras mainly focus on image source identification. These includes: McKay *et al.* [5] computed both the CIC features in [4] and some noise statistics features (NStats) feature to identify different types of image acquisition devices including DSCs, mobile cameras, scanners and computer graphics. By extending Lucas *et al.* [10], Alles *et al.* [12] estimated PRNU sensor noise patterns to identify individual mobile cameras and webcams. Tsai *et al.* [13] combined several sets of statistical image features including color features, IQM and wavelet statistical features to identify both DSCs and mobile cameras. Celiktutan *et al.* [15] used various fusion methods to combine 3 sets of forensics features including IQM, MSWS and BS features to distinguish various cell-phone models. Our previous work in [9] proposed combining three sets of accurately detected demosaicing features, including weights, error cumulants and normalized group sizes [8]. Together with an eigenfeature extraction technique [28], our compact set of features show better source identification performance than other forensics features as suggested in [5, 15]. Encouraged by the good source identification results, we extend our demosaicing detection model and weights features to address the tampering forensics challenge on mobile photos.

We propose to measure the statistical inconsistency from image blocks sampled at different photo locations. Fig. 2 shows the flow graph of the proposed method. By first cropping some image blocks from a mobile photo at different regions, we estimate the underlying demosaicing formulas for different demosaiced sample categories in terms of a set of derivative correlation weights. By computing the within-photo covariance matrix from a representative set of training mobile photos and through regularizing its eigen spectrum, we learn an eigenfeature transformation to derive a compact set of eigen weights features. These features are the most sensitive to the

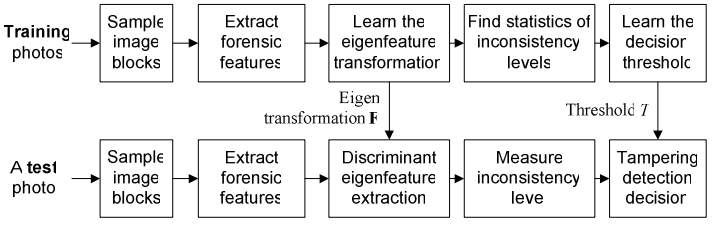


Fig. 2. Flow graph of the proposed method

tampering inconsistencies as the eigen subspace transformation is learned based minimizing the within-photo covariance and maximizing the between-photo covariance. Knowing that local image tampering would generally have uneven impacts on the cropped blocks at different locations, we propose a metric to quantify this inconsistency by measuring the total variances of our derived eigen weights associated with the cropped blocks. A tampering detection decision is then made by comparing the measured inconsistency with a decision threshold, which can be predetermined based on some training statistics.

Our proposed framework is novel, which offers several contributions. First, our method requires no prior information about the specific image source or the specific tampering type for the test mobile photo to be known in advance. The eigen transformation can be learned offline based on a representative mobile photo dataset from multiple sources, where the test source may not be available. Second, in learning the eigenfeature transformation, our concept of minimizing the "within-photo" covariance is novel. The framework also allows fusion of other statistical forensics features in the literature to achieve better performance. Third, a performance comparison shows that our eigen weights features tend to perform better than the eigen features extracted from several other types of mobile-photo forensics features in detecting tampering inconsistencies. Fourth, we propose a novel metric to quantify the amount of inconsistency between cropped image blocks from different photo locations. Experimentally, we show that this metric works well in detecting integrity tampering especially when image signals are mixed from different source mobile camera models or brands.

The remainder of this paper is organized as follows. Section 2 details the proposed method. Section 3 experimentally shows the effectiveness of our proposed method in detecting the tampering inconsistency. Section 4 concludes this paper and discusses several possible future extensions.

2 Proposed Method

Our method in Fig. 2 considers detection of tampering inconsistencies using accurately estimated demosaicing weight features. The aim is to answer one most frequently asked forensics question, i.e. "Is a given digital photo the original output of a camera device or has its integrity been tampered?". Our idea is based on the fact that an intact mobile photo often possesses good statistical harmony since different regions of the photo has gone through the same image creation pipeline as illustrated in Fig. 1(a). While tampering on another hand often involves mixing image signals from

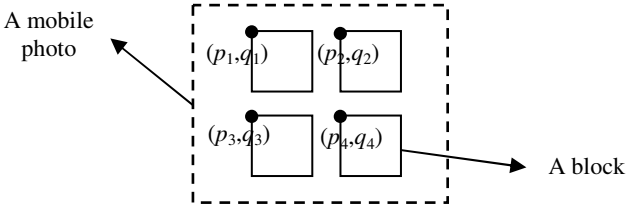


Fig. 3. Cropping blocks at different regions of a mobile photo, where (p_i, q_i) for $1 \le i \le 4$ are set to be odd numbers to avoid shifts of the underlying CFA

multiple sources, applying location-dependant tampering processing, or both, these would inevitably introduce some statistical inconsistencies between the intact image region and the tampered region. As the primary concern of an image forger is to create visually flawless forgeries for deceiving others, neither is it important nor do they have good tools nowadays to cover up the various forms tampering inconsistencies, which are invisible to human eyes. Below, we elaborate on our main steps in Fig. 2.

2.1 Block Cropping

There are many ways to crop some image blocks in different regions of a test mobile photo. For instance, one can choose on the number of blocks needed, the block size as well as the cropping locations. Several important assumptions in the cropping are: 1) the blocks should be of sufficiently large sizes so that sufficient color samples are present to ensure good estimation of the underlying forensics statistics; 2) The cropped blocks shall cover the region of interests where local tampering more likely occurs; 3) Blocks cropped at different regions shall not overlap in a large percentage of the total block area; 4) Preferably, the cropping locations shall be selected so that image tampering would more likely affect some blocks but not others. Since designing the cropping scheme by itself is not the primary concern of this paper, we have adopted a simple scheme as illustrated in Fig. 3. In this scheme, four non-overlapping blocks of about 512×512 are cropped at fixed locations, which are close to image center. For a typical 3 to 5 mega-pixel mobile camera, these four blocks would be able cover the central photo area, which is often the region of interest. The photo coordinates (p_i, q_i) of the top-left corner of a cropping box are set to be odd numbers. Since we are using demosaicing weights features in this paper, this is to ensure that we do not artificially create an unnecessary shift of the underlying CFA patterns for the cropped blocks at different locations. Based on our previous research [8], our demosaicing weights estimated based on the four different shifted versions of Bayer CFAs in Fig. 1(b) are usually distinctively different.

2.2 Demosaicing Weights Estimation

The demosaicing weights for a cropped image block **P** are estimated based on our earlier work in [8]. Below we briefly summarize the main concept and procedures.

Image demosaicing in Fig. 1(a) has been intensively researched in the past few decades to reconstruct the missing color samples (due to the color filtering) with good fidelity and less visible distortions. Many demosaicing algorithms have been

developed and state-of-the-arts algorithms often utilize the information of sensor samples from all 3 color channels and adaptively applied different interpolation formulas on different edge types [3].

To estimate the set of interpolation formulas used, i.e. the demosaicing weights, we first separate the interpolated samples from sensor samples in **P**. As Bayer CFAs in Fig. 1(b) are dominantly used commercially [3], we initially assume the first Bayer CFA is underlying CFA and write

$$\mathbf{P}=\begin{pmatrix} \{r,G,B\}_{11} & \{R,g,B\}_{12} & \cdots \\ \{R,g,B\}_{21} & \{R,G,b\}_{22} & \\ \vdots & & \ddots \end{pmatrix},\quad \mathbf{A}=\begin{pmatrix} a_{11}=r_{11} & a_{12}=g_{12} & \cdots \\ a_{21}=g_{21} & a_{22}=b_{22} & \\ \vdots & & \ddots \end{pmatrix} \tag{1}$$

where **A** denotes sensor samples and capital *R*, *G*, *B* represent the interpolated samples. With a reverse classification technique [8], we classify all the interpolated samples into 16 categories with known demosaicing directions so that each category contains the interpolated samples by the same formula. For the k^{th} interpolated sample X_{jk} of the j^{th} category $\{D_{jk}\}$, we write a demosaicing equation below based on a partial second-order derivative correlation model.

$$e_{jk}=D''_{jk}-\mathbf{a}''^{T}_{jk}\mathbf{w}_j \tag{2}$$

Here, e_{jk} is the prediction error, D''_{jk} is the second-order derivative of D_{jk} computed on **P** along the demosaicing direction, $1\le j\le 16$, $1\le k\le K$ with *K* denoting the size of the j^{th} category and $\mathbf{a}''_{jk}\in\mathbb{R}^{c\times 1}$ denotes the support partial derivatives computed from the down-sampled **A**. In the case that the j^{th} category is on the red or blue color channels, $\mathbf{a}''_{jk}$ also include the support derivatives computed from the green channel. $\mathbf{w}_j\in\mathbb{R}^{c\times 1}$ is the vector of derivative weights, representing the applied demosaicing formula for the j^{th} category and *c* is the number of weights chosen. The derivative correlation model is based on an observation that interpolating a sample along one direction is equivalent to estimating its partial second-order derivative on a 2-by-2 periodical CFA lattice. For demosaicing detection, this model has the desirable property of enabling detecting both intra-color channel and cross-channel demosaicing correlation with reduced detection variations caused by different image contents. Since for each sample in the j^{th} category, we can write a similar equation to (2), by organizing all *K* equations into a matrix form, we have

$$\mathbf{e}_j=\mathbf{d}_j-\mathbf{Q}_j\mathbf{w}_j \tag{3}$$

where

$$\mathbf{e}_j=\begin{bmatrix} e_{j1} \\ \vdots \\ e_{jK} \end{bmatrix},\quad \mathbf{d}_j=\begin{bmatrix} D''_{j1} \\ \vdots \\ D''_{jK} \end{bmatrix},\quad \mathbf{Q}_j=\begin{bmatrix} \mathbf{a}''^{T}_{j1} \\ \vdots \\ \mathbf{a}''^{T}_{jK} \end{bmatrix},\quad \mathbf{w}_j=\begin{bmatrix} w_{j1} \\ \vdots \\ w_{jm} \end{bmatrix}$$

Since $K \gg m$, the weights $\mathbf{w}_j$ is solved as a regularized least square solution below,

$$\min\left(\left\|\mathbf{e}_j\right\|^2+\eta\left\|\mathbf{w}_j\right\|^2\right)\Rightarrow \mathbf{w}_j=\left(\mathbf{Q}_j{}^{\mathrm{T}}\mathbf{Q}_j+\eta\mathbf{I}\right)^{-1}\mathbf{Q}_j\mathbf{d}_j \tag{4}$$

where η is a small regularization constant and $\mathbf{I}\in\mathbb{R}^{m\times m}$ denotes an identity matrix. By solving the weights separately for the 16 categories, we obtain a total of 312 weight features. Also based on three other Bayer CFAs in Fig. 1(b), we repeat the weights estimation for three more times. This makes our weights features more comprehensive [8] though the feature dimension increases 4-fold to 1248. Not only representing the underlying set of demosaicing formulas used, our weights features also carry the distortion fingerprints caused by the post-demosaicing processing. By fixing the same Hamilton's demosaicing algorithm, we demonstrated in [8] that our demosaicing features can successfully identify 7 different post-demosaicing camera processes. Therefore, our weights features characterize both the demosaicing process and the post-demosaicing camera processing.

2.3 Eigen Weights Extraction

The high feature dimensionality incurs high computational cost and makes our weights features difficult to be used directly for image forensics analysis. In this section, we learn an eigen feature transformation for deriving a compact set of eigen weights. This is achieved by minimizing the within-photo covariance and maximizing the between-photo covariance, where the within-photo eigen spectrum is regularized using [28] for increased reliability. Given a set of M non-tampered training mobile photos, each with N=4 cropped blocks, we let $\{\mathbf{x}_{mn}\}$, where $1\le m\le M$ and $1\le n\le N$, denote the normalized training feature vector. Here, a linear normalization has been performed on each feature so that it has zero mean and unity variance. $\mathbf{x}_{mn}\in\mathbb{R}^{L\times 1}$ and L=1248 is our feature dimension. The learning of eigen feature transformation is explained in the following steps:

1. Compute the within-photo covariance matrix $\mathbf{S}^{(\mathrm{w})}$ using

$$\mathbf{S}^{(\mathrm{w})}=\frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}\left(\mathbf{x}_{mn}-\bar{\mathbf{x}}_m\right)\left(\mathbf{x}_{mn}-\bar{\mathbf{x}}_m\right)^T \tag{5}$$

   where $\bar{\mathbf{x}}_m=\frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_{mn}$ is the mean vector of the m^{th} photo.

2. Perform eigen decomposition using

$$\mathbf{\Lambda}^{(\mathrm{w})}=\mathbf{\Phi}^{(\mathrm{w})T}\mathbf{S}^{(\mathrm{w})}\mathbf{\Phi}^{(\mathrm{w})} \tag{6}$$

   where $\mathbf{\Phi}^{(\mathrm{w})}=[\boldsymbol{\varphi}_1^{(\mathrm{w})},\ldots,\boldsymbol{\varphi}_L^{(\mathrm{w})}]$ is the eigenvector matrix of $\mathbf{S}^{(\mathrm{w})}$ and $\mathbf{\Lambda}^{(\mathrm{w})}$ is the diagonal matrix with the corresponding eigen values $\lambda_1^{(\mathrm{w})}\ge\lambda_2^{(\mathrm{w})}\ge\ldots\ge\lambda_L^{(\mathrm{w})}$ plotted as the training eigen spectrum in Fig. 4. Note in Fig. 4 that we can observe the widening gap in the log-scale plot between the training and the testing spectrums. This suggests that the eigen values learned in the training becomes less reliable and more susceptible to estimation errors. As disciminant analysis often requires computing the inverse of these eigen values for feature scaling, the unreliable eigen spectrum need to be regularized;

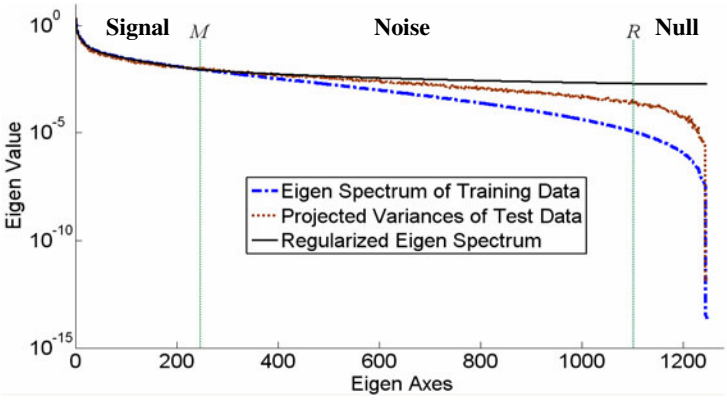


Fig. 4. Within-photo eigen spectrum obtained on a training mobile photo dataset, projected variances of the corresponding test mobile photo set and the regularized eigen spectrum

3. Following the model in [28], we separate the eigen spectrum into 3 regions, "signal", "noise" and "null", and fit a regularized spectrum as plotted in Fig. 4. By choosing several model parameters M, R, α and β according to [28], the regularized eigen value corresponding to the ℓ^{th} eigenvector is written as [28]

$$\tilde{\lambda}_{\ell}^{(\mathrm{w})} = \begin{cases} \lambda_{\ell}^{(\mathrm{w})}, & \ell < M \\ \alpha/(\ell+\beta), & M \le \ell \le R \\ \alpha/(1+R+\beta), & R < \ell \le L \end{cases} \tag{7}$$

4. Perform the whitening feature transformation using

$$\mathbf{y}_{mn} = \tilde{\mathbf{\Psi}}_L^{(\mathrm{w})T} \mathbf{x}_{mn} \tag{8}$$

where $\tilde{\mathbf{\Psi}}_L^{(\mathrm{w})} = \left[\boldsymbol{\varphi}_1^{(\mathrm{w})} \Big/ \sqrt{\tilde{\lambda}_1^{(\mathrm{w})}}, \ldots, \boldsymbol{\varphi}_L^{(\mathrm{w})} \Big/ \sqrt{\tilde{\lambda}_L^{(\mathrm{w})}} \right]$.

5. Based on the $\{\mathbf{y}_{mn}\}$, compute the total covariance matrix using

$$\mathbf{S}^{(\mathrm{t})} = \frac{1}{MN} \sum_{m=1}^{M} \sum_{n=1}^{N} (\mathbf{y}_{mn} - \bar{\mathbf{y}})(\mathbf{y}_{mn} - \bar{\mathbf{y}})^T \tag{9}$$

where $\bar{\mathbf{y}} = \frac{1}{MN} \sum_{m=1}^{M} \sum_{n=1}^{N} \mathbf{y}_{mn}$ is the global mean vector.

6. Perform eigen decomposition on $\mathbf{S}^{(\mathrm{t})}$ and construct $\tilde{\mathbf{\Psi}}_E^{(\mathrm{t})} = [\boldsymbol{\varphi}_1^{(\mathrm{t})}, \ldots, \boldsymbol{\varphi}_E^{(\mathrm{t})}]$ for feature reduction using principal component analysis (PCA), where typically $E \ll L$. $\boldsymbol{\varphi}_1^{(\mathrm{t})}, \ldots, \boldsymbol{\varphi}_E^{(\mathrm{t})}$ are the leading eigenvectors of $\mathbf{S}^{(\mathrm{t})}$ corresponding to the E largest eigenvalues. The compact eigen weights feature vector

$$\mathbf{z}_{mn} = \tilde{\mathbf{\Psi}}_E^{(\mathrm{t})T} \mathbf{y}_{mn} = \mathbf{F}\mathbf{x}_{mn} \tag{10}$$

where $\mathbf{z}_{mn} \in \mathbb{R}^{E\times 1}$ and our learned eigen transformation $\mathbf{F} = \left(\tilde{\mathbf{\Psi}}_L^{(\mathrm{w})} \tilde{\mathbf{\Psi}}_E^{(\mathrm{t})}\right)^T$.

2.4 Measuring the Inconsistency Level

Let $\mathbf{z}_1,\ldots,\mathbf{z}_n,\ldots,\mathbf{z}_N$, where $\mathbf{z}_n \in \mathbb{R}^{E\times 1}$, be the eigen feature vectors corresponding to the different cropped blocks from a given test mobile photo. We compute the covariance matrix

$$\mathbf{Z} = \frac{1}{N}\sum\nolimits_{n=1}^{N} \left(\mathbf{z}_n - \overline{\mathbf{z}}\right)\left(\mathbf{z}_n - \overline{\mathbf{z}}\right)^T \tag{11}$$

where $\overline{\mathbf{z}} = \frac{1}{N}\sum\nolimits_{n=1}^{N} \mathbf{z}_n$ is the mean vector. We propose to measure the inconsistency by

$$J = \sqrt{\mathrm{Trace}\left(\mathbf{Z}\right)} \tag{12}$$

This quantifies the total variance of the eigen feature vectors computed from the different cropped blocks.

Since our weights features model both the demosaicing and the post-processing, ideally our features estimated from different regions of an intact photo are identical. However, inevitably the different image contents would affect our estimated weights and contribute to an increased within-photo covariance. Through regularizing the within-photo eigen spectrum, performing whitening transformation and PCA feature reduction, our derived eigen weights features are expected to be insensitive to within-photo image content variations and highly sensitive to the image signal mixing from multiple photo sources. Our proposed metric *J*, hence, translates to a small value for a non-tampered mobile photo and a large value for a tampered photo, where some cropped blocks are transplanted from other photo sources. The decision threshold for using *J* to detect the presence of tampering can be determined from some training statistics of a representative mobile photo dataset.

3 Experimental Results

We have set up a mobile photo set containing 1000 photos from a total of 10 mobile cameras from 6 brands in Table 1, where cameras of identical or close models are present. These photos are collected from a number of contributors by their cellular phones and all photos are the direct camera output stored in the default JPEG format. These photos cover a large variety of common indoor and outdoor scenes captured under different lighting conditions. Several representative photo samples from different mobile cameras are shown in Fig. 5. We randomly select 500 photos, with 50 from each camera, to learn the eigen feature transformation **F** and the remaining photos are reserved for testing.

Table 1. Mobile camera used with 100 photos from each camera

ID	Brand	Model	Photo Dimension
N1	Nokia	5300	1280×960
N2		5300	1280×960
N3		N73	2048×1536
N4		N73	2048×1536
N5		N73	2048×1536
SE6	Sony Ericsson	K750c	1632×1224
M7	Motorola	L6	640×480
D8	Dopod	P3600i	1600×1200
O9	O2	XDA	1200×1600
S10	Sumsung	SGH i780	1600×1200

Fig. 5. Sample mobile photos

As tabulated in Table 2, we also generate a set of tampered photos by modifying the intact test photos from 4 source cameras of different models. Each tampered photo is created from an intact test Photo-A by replacing one of its 4 cropped blocks randomly with a block from a different Photo-B. For each of the 4 Photo-A sources, we create 7-8 times of tampered test photos by considering different source cameras for Photo B.

3.1 Determining the Number of Eigen Features

With the learned the eigen feature transformation, we vary the number of eigen features E and compute the statistics of our measured inconsistencies separately for the different sources in Table 2. For a given E, let $\mu_g(E)$ and $v_g(E)$ denote the mean and variance, respectively, of the inconsistencies for the intact photos. $\mu_t(E)$ and $v_t(E)$ are for the tampered photos. We then compute the Fisher's ratio using [29]

$$f(E) = \frac{\left(\mu_g(E) - \mu_t(E)\right)^2}{v_g(E) + v_t(E)} \tag{13}$$

to measure the discriminant power of our proposed metric. Fig. 6 shows Fisher's ratio versus number of eigen weights E for the 4 different sources. From the plots, we can see that the Fisher's ratio increase quickly initially to the maximal value and starts decreasing gradually when E is increased from 1 to 100. Based on these results, we choose E=14 for our eigen weights features in the following experiments.

Table 2. Dataset of the tampered photos, where one of the four cropped blocks from Photo A is replaced with a randomly cropped block from a different Photo B. Refer to Table 1 for the source cameras.

Source camera (Photo A)	Source camera (Photo B)							
N1	N1	N2	N3	SE6	M7	D8	O9	S10
N3	N3	N4	N1	SE6	M7	D8	O9	S10
SE6	SE6	N1	N3	M7	D8	O9	S10	-
M7	M7	N1	N3	SE6	D8	O9	S10	-

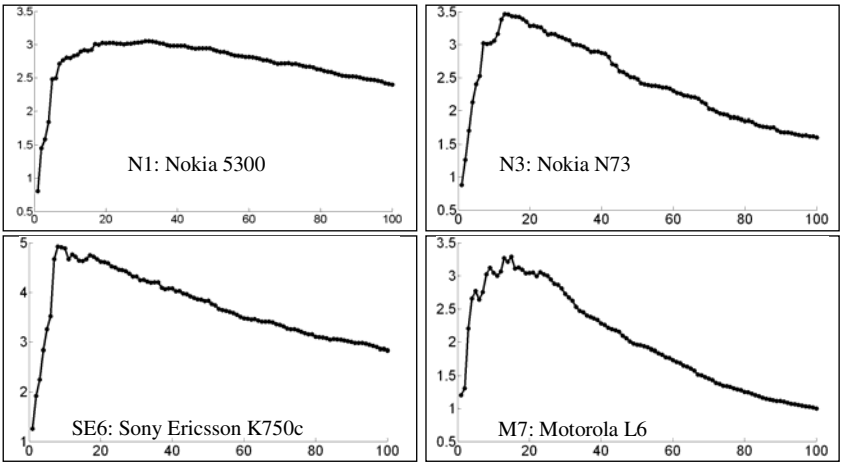


Fig. 6. Fisher's ratio [29] versus number of eigen weights features for four different sources

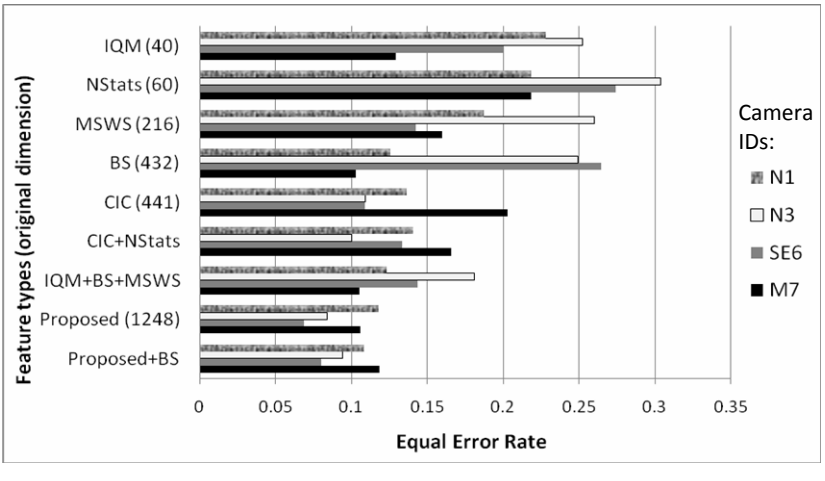


Fig. 7. Feature comparison in term of equal error rate (%) for tampering detection on different source mobile cameras. Fourteen eigen features are used.

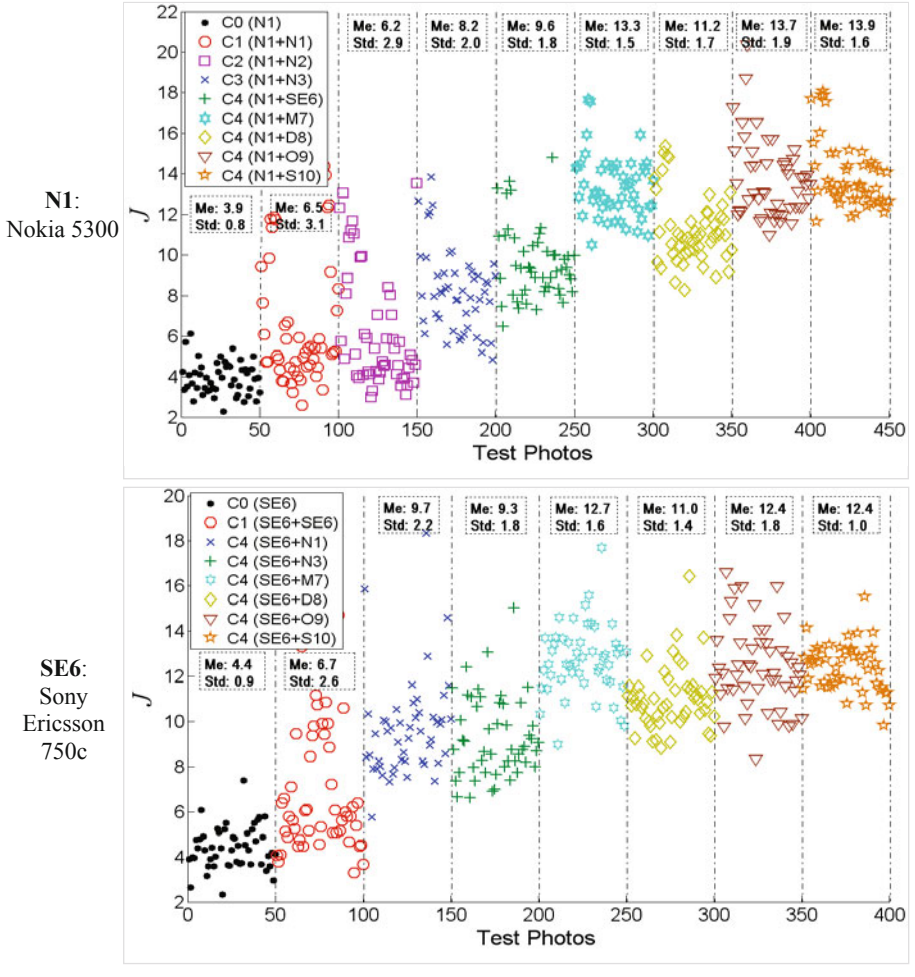


Fig. 8. Scatter plots of measured inconsistencies for intact and tampered photos from two different camera sources

3.2 Tampering Detection

In this section, we compare our eigen weights features with the eigen features extracted from several other sets of forensics features in terms of equal error rates (EER) for tampering detection. These other feature sets include multi-scale wavelet statistics (MSWS) [15, 17], binary similarity measures (BS) [14, 15, 17], image quality metrics (IQM) [13, 15-17], color interpolation coefficients (CIC) [4, 5], noise statistics (NStats) [5], the combination of MSWS, BS and IQM [15] and the combination of CIC and NStats [5]. For the BS features, we compute a total of 432 features based on the description in [15]. Though this number is still less than the 480 BS features used in [15], our BS feature set still covers majority of the BS features. We conducted the

similar experiments as in Fig. 6 to determine the number of eigen features required for other feature types. For IQM, NStats and MSWS, we use 5, 3 and 6 eigen features, repectively. For the remaining feature types, we use 12 to 14 eigen features. The eigen feature size is likely related with three factors including the discriminative power of the features, the number of camera sources and the number of cropped blocks. Based on this, the EERs are compared in Fig. 7 in tampering detection for different Photo-A sources. For source camera N1, N3 and SE6, our eigen weights features consistently give the best EERs. For M7, whose photos are of relatively low quality, our proposed features perform no better than BS. Though our framework allows combining different types of forensics features before learning the eigen transformation, we find the fusion of our features with other features may not necessarily lead to a better overall performance. By combining BS and our weights features, the performance improves on N1 but degrades on N3, SE6 and M7. The average EER rate for the four sources degrades slightly from 9.4% to 10%.

We also consider differentiating five classes of photos including C0 (non-tampered photos), C1 (tampered with Photo-A and B from the same camera), C2 (different cameras of the same model), C3 (different models of the same brand) and C4 (different brands). As shown in the scatter plots in Fig. 8, we observe much higher mean and standard deviations for Class C1 and C2's statistics than that for the intact C0 class. Tampered classes C3 or C4 are visually easy to be separated from Class C0.

In practice, it is desirable to detect image tampering in a blind context that information of the source cameras is not required. We put the non-tampered and the tampered statistics from 4 different Photo-A sources together to study the source-independent tampering detection rate. Fig. 9 shows our ROC curves for identifying different tampering classes. As expected, C1 and C2 tampering are generally harder to

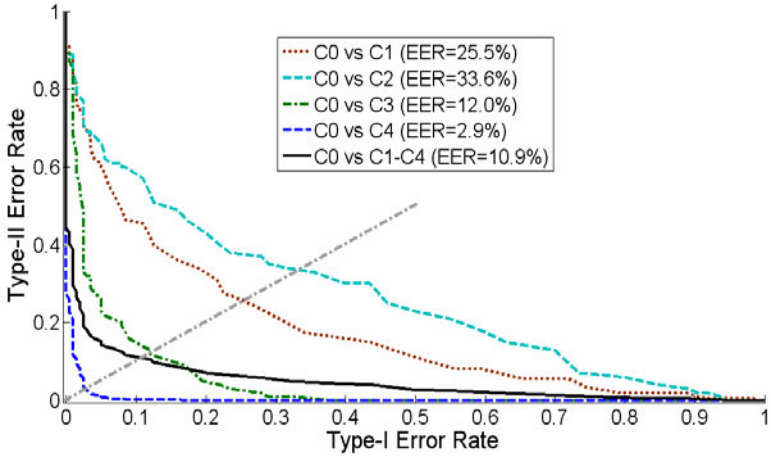


Fig. 9. Receiver operation characteristic (ROC) curves for discriminating intact photos and different classes of tampered photos. Type-I error rate is the percentage of non-tampered photos being misclassified as tampered. Type-II error rate is the percentage of tampered photos being misclassified as non-tampered.

detect due to a very similar camera processing. For Class C3 and C4, our detection is reliable. Especially we achieve a low EER of 2.9% for C4. The overall EER of 10.9% also suggests a satisfactory performance for detecting the group of C1 to C4 photo tampering based on a common decision threshold.

4 Conclusions

In this paper, we propose a novel framework to quantify the tampering inconsistency on a mobile photo by measuring the variations of our eigen weights features from cropped blocks at different photo locations. By regularizing and minimizing the within-photo covariance, comparison results show that our eigen weights perform better than the same number of eigen features extracted from several conventional sets of forensics features in detecting the presence of tampering. In a blind context, where information of the source mobile camera is not required, our methods show a good reliability in detecting the C3 and C4 types of tampered photos. Especially, we achieve a low EER of 2.9% for detecting C4 tampering. These suggest that our proposed method has a good confidence to detect the presence of tampering when one of the four cropped block is from a different mobile camera model or brand.

Our work can be further improved in several directions. First, our tampering detection can be applied to digital images acquired by other devices, such as DSCs. Second, localization of the blocks affected by tampering can be a good additional feature to be included. Third, it is interesting to investigate on other forms of tampering inconsistencies besides mixing image signals. Fourth, more sophisticated cropping schemes with smaller block sizes shall be investigated, where the potential trade-off between block size and detection accuracy will be analyzed.

References

1. Mosleh, F. (kodak): Cameras in Handsets Evolving from Novelty to DSC Performance, Despite Constraints. In: Embedded.com (2008)
2. Lewis, J.: Don't Just Stand and Stare, Shoot it, Too. In: The Singapore Straits Times (April 28, 2007)
3. Li, X., Gunturk, B., Zhang, L.: Image Demosaicing: a Systematic Survey. In: Proc. of SPIE, vol. 6822 (2008)
4. Swaminathan, A., Wu, M., Liu, K.J.R.: Nonintrusive Component Forensics of Visual Sensors Using Output Images. IEEE Trans.on Information Forensics and Security 2(1), 91–106 (2007)
5. McKay, C., Swaminathan, A., Gou, H., Wu, M.: Image Acquisition Forensics: Forensics Analysis to Identify Imaging Source. In: Proc. of ICASSP, pp. 1657–1660 (2008)
6. Swaminathan, A., Wu, M., Liu, K.J.R.: Digital Image Forensics via Intrinsic Fingerprints. IEEE Trans. on Information Forensics and Security 3(1), 101–117 (2008)
7. Cao, H., Kot, A.C.: A Generalized Model for Detection of Demosaicing Characteristics. In: Proc. of ICME, pp. 1513–1516 (2008)
8. Cao, H., Kot, A.C.: Accurate Detection of Demosaicing Regularity for Digital Image Forensics. IEEE Trans. on Information Forensics and Security 4(4), 899–910 (2009)
9. Cao, H., Kot, A.C.: Mobile Camera Identification Using Demosaicing Features. In: Proc. of ISCAS, pp. 1683–1686 (2010)

10. Lucas, J., Fridrich, J., Goljan, M.: Digital Camera Identification from Sensor Pattern Noise. IEEE Trans. Information Forensics and Security 1(2), 205–214 (2006)
11. Chen, M., Fridrich, J., Goljan, M., Lucas, J.: Determining Image Origin and Integrity Using Sensor Noise. IEEE Trans. on Information Forensics and Security 3(1), 74–89 (2008)
12. Alles, E.J., Geradts, Z.J.M.H., Veenman, C.J.: Source Camera Identification for Low Resolution Heavily Compressed Images. In: Proc. of ICCSA, pp. 557–567 (2008)
13. Tsai, M.-J., Lai, C.-L., Liu, J.: Camera/Mobile Phone Source Identification for Digital Forensics. In: Proc. of ICASSP, vol. 2, pp. 221–224 (2007)
14. Avcibas, I., Kharrazi, M., Memon, N., Sankur, B.: Image Steganalysis with Binary Similarity Measures. EUROSIP Journal of Applied Signal Processing 17, 2749–2757 (2005)
15. Celiktutan, O., Sankur, B., Avcibas, I.: Blind Identification of Source Cell-Phone Model. IEEE Trans. on Information Forensics and Security 3(3), 553–566 (2008)
16. Avcibas, I., Bayram, S., Memon, N., Ramkumar, M., Sankur, B.: A Classifier Design for Detecting Image Manipulations. In: Proc. of Int. Conf. on Image Processing, vol. 4, pp. 2645–2648 (2004)
17. Bayram, S., Avcibas, I., Sankur, B., Memon, N.: Image Manipulation Detection. Journal of Electronic Imaging 15, 41102 (2006)
18. Ng, T.-T., Chang, S.-F., Sun, Q.: Blind Detection of Photomontage Using High Order Statistics. In: Proc. of ISCAS, vol. 5, pp. 688–691 (2004)
19. Johnson, M.K., Farid, H.: Exposing Digital Forgeries by Detecting Inconsistencies in Lighting. In: Proc. of ACM Multimedia Security Workshop, pp. 1–10 (2005)
20. Farid, H.: A Survey of Image Forgery Detection. IEEE Signal Processing Magazine 26(2), 16–25 (2009)
21. He, J., Lin, Z., Wang, L., Tang, X.: Detecting Doctored JPEG Images via DCT Coefficient Analysis. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3953, pp. 423–435. Springer, Heidelberg (2006)
22. Fu, D., Shi, Y.Q., Su, W.: A Generalized Benford's Law for JPEG Coefficients and its Applications in Image Forensics. In: Proc. of SPIE, vol. 6505, p. 65051L (2007)
23. Chen, W., Shi, Y.Q.: Image Splicing Detection Using 2-D Phase Congruency and Statistical Moments of Characteristic Function. In: Proc. of SPIE, vol. 6505, p. 65050R (2007)
24. Luo, W., Qu, Z., Huang, J., Qiu, G.: A Novel Method for Detecting Cropped and Recompressed Image Blocks. In: Proc. of ICASSP 2007, vol. 2, pp. 217–220 (2007)
25. Barni, M., Costanzo, L., Sabatini, L.: Identification of Cut & Paste Tampering by Means of Double-JPEG Detection and Image Segmentation. In: Proc. of ISCAS 2010, pp. 1687–1690 (2010)
26. Li, C.-T.: Source Camera Identification Using Enhanced Sensor Pattern Noise. IEEE Trans. on Information Forensics and Security 5(2), 280–287 (2010)
27. Mahdian, B., Saic, S.: A Bibliography on Blind Methods for Identifying Image Forgery. Signal Processing: Image Communication 25(6), 389–399 (2010)
28. Jiang, X., Mandal, B., Kot, A.C.: Eigenfeature Regularization and Extraction in Face Recognition. IEEE Trans. on Pattern Analysis and Machine Intelligence 30(3), 383–394 (2008)
29. Duda, O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. John Wiley & Sons, New York (2001)

Tampered Region Localization of Digital Color Images Based on JPEG Compression Noise

Wei Wang, Jing Dong, and Tieniu Tan

National Laboratory of Pattern Recognition,
Institute of Automation, Chinese Academy of Sciences,
P.O. Box 2728, Beijing, P.R. China, 100190
{wwang,jdong,tnt}@nlpr.ia.ac.cn

Abstract. With the availability of various digital image edit tools, seeing is no longer believing. In this paper, we focus on tampered region localization for image forensics. We propose an algorithm which can locate tampered region(s) in a lossless compressed tampered image when its unchanged region is output of JPEG decompressor. We find the tampered region and the unchanged region have different responses for JPEG compression. The tampered region has stronger high frequency quantization noise than the unchanged region. We employ PCA to separate different spatial frequencies quantization noises, i.e. low, medium and high frequency quantization noise, and extract high frequency quantization noise for tampered region localization. Post-processing is involved to get final localization result. The experimental results prove the effectiveness of our proposed method.

Keywords: Image forensics, Tampered region localization, JPEG compression noise, PCA.

1 Introduction

Along with the rapid development of image editing software (e.g. Adobe Photoshop), digital images can be easily manipulated and tampered images can hardly be detected by human eyes. Seeing is no longer believing. It is necessary to develop authentication techniques to verify the integrity of a digital image.

Generally speaking, there are two types of approaches for image authentication: active [3, 4] and passive [13, 19] approaches. Active approaches often require pre-processing (e.g. watermark embedding or signature generating), and they are not desired for practical use in daily life since the image capture devices are not usually all integrated with watermarking embedding module. Passive approaches, which gather evidence of tampering from images themselves, however, have more potential for practical use and gains more attention among researches in image forensics.

We focus on passive approaches and try to locate the tampered region in a tampered image. Tampered region(s) localization in tampered image is more meaningful and convincible than simple detection of existence of tampered image for image forensics. Tampered image detection can only tell us whether an

H.-J. Kim, Y. Shi, and M. Barni (Eds.): IWDW 2010, LNCS 6526, pp. 120–133, 2011.

image is tampered or not. However, we do not know whether it is the tampering operation or other operations (e.g. JPEG compression) that affect information for tampered images. Whereas, tampered region localization can directly imply where the tampering operation occurs. In this paper, we will propose an algorithm which can locate tampered region(s) in a lossless compressed tampered image when its unchanged region is output of JPEG decompressor.

For such a tampered image, we find the tampered region and the unchanged region have different responses for JPEG compression. The unchanged region has weaker high frequency quantization noise than the tampered region. We then employ principle component analysis (PCA) to separate different spatial frequencies quantization noises, i.e. low, medium and high frequency quantization noise, and extract high frequency noise for tampered region localization.

The rest of this paper is organized as follows. Some related works are introduced in Section 2. Section 3 mainly introduces our proposed algorithm for tampered region localization. The experimental results and analysis are given in Section 4. Conclusions are drawn in Section 5.

2 Related Works

In recent years, many researchers focus on digital image tampering detection and have proposed a number of techniques. There are several methods for passive image tampering detection proposed in the recent literature [2, 5, 6, 7, 8, 9, 10, 11, 15, 16, 17, 18, 20, 21].

Farid et al. have done pioneering work in this area. In [6] and [7], *Johnson* and *Farid* developed a technique of tampering detection by analyzing the inconsistency of lighting in image. But it may fail when source images used for tampering are taken under similar lighting conditions. Besides, it needs to manually select the points near the boundary of suspicious object. *Popescu* and *Farid* [17] argued that color interpolation (demosaicing) introduced specific correlations between neighboring pixels of a color image, while image tampering might destroy or alter them and based on this they proposed an image tampering detection algorithm to check the periodicity of these correlations. Actually, they did not try their method on real tampered examples. Besides, in [2], *Dirik* and *Memon* utilized artifacts created by Color Filter Array (CFA) to detect image tampering. They proposed two features for tampering detection. One is based on CFA pattern estimation and the other is based on the fact that sensor noise power in CFA interpolated pixels should be significantly lower than non-interpolated pixels due to the low pass nature of CFA demosaicing. Actually, CFA artifacts are hardly detected for many images with heavy JPEG compression. In [11] and [16], authors assumed that image tampering would involve resampling. They proposed approaches to detect periodicity of correlations introduced by resampling. However, they did not give enough real examples for tampered region localization. *Lukáš et al.* [10] proposed a digital image tampering detection method to detect camera pattern noise which is considered as a unique stochastic characteristic of imaging sensor. The tampered region is determined when image region is

detected as lacking of the pattern noise. However, this method is only applicable when the tampered image is claimed to have been taken by a known camera or at least we have images taken by the camera before. In [8], *Krawetz* proposed a suit of tools to analyze images and do forensics. He did a series of experiments rather than deep analysis. *Shi et al.* [18] proposed a splicing detection method using effective features extracted from image Markov transfer matrices. Experiments were carried on Columbia image splicing detection evaluation dataset [14] and the results were satisfying. Aiming at color image tampering detection, we proposed an effective color image tampering detection approach based on image chroma [20, 21]. We found that the analysis on chroma of color image was more reasonable for tampered image detection than on illuminance because chroma could reflect more information left by tampering which human eyes might not observe. If we use the proposed methods in [10, 11, 16, 17, 18, 20, 21] to find the tampered region by sliding window within an image, we should carefully choose the window size. Too small will not have enough statistical information while too big will not locate accurately.

There are also some methods for JPEG image forensics since JPEG is the most widely used image format. Double JPEG compression can be a cue for image tampering, but detecting double JPEG compression [12, 15] does not necessarily prove malicious tampering. *He et al.* [5, 9] proposed a workable method by using the double JPEG quantization effect hidden among the DCT coefficients to automatically detect the tampered regions of images. They agreed that the unchanged region in a tampered JPEG image undergoing double JPEG compression while the tampered region undergoing only once. They tried to use the inconsistency to locate the tampered region. However, the average detection rate both in image level and region level are below 65%; and their method are sensitive to the estimation of the period.

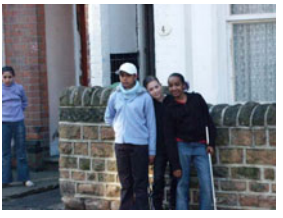
3 Proposed Approach

Basically, all image manipulations can be roughly classified into local changing and global changing operations. In this paper, we focus on local changing operation. We want to locate the local changing, i.e. tampered region. We define a tampered image as in [5, 9]. *Lin et al.* regards an image as tampered one when part of its content has been altered. In other words, that an image is tampered implies that it must contain two parts: the unchanged region and the tampered region.

Since JPEG is the most widely used image format, we mainly focus on locating tampered regions in a lossless compressed image when the unchanged region of the image is output of JPEG decompressor. We will utilize properties of JPEG compression to locate the tampered region.

3.1 JPEG Compression Noise

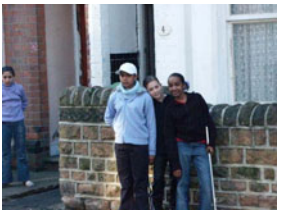
JPEG compression noise can be simply calculated by subtracting a given image from its JPEG compressed version. Different responses can be get if we



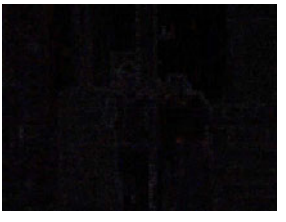
(a) Original image in TIFF format

(b) JPEG compression noise of (a) with Q=80

(c) JPEG compression noise of (a) with Q=50



(d) JPEG format of (a) with Q=75



(e) JPEG compression noise of (d) with Q=80

(f) JPEG compression noise of (d) with Q=50

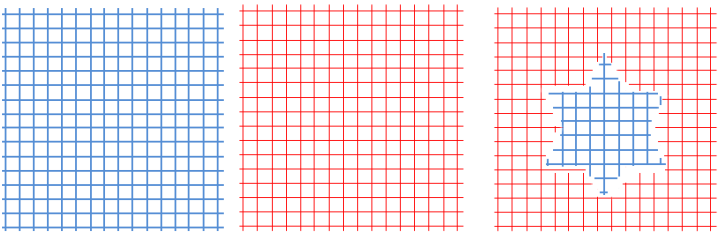
Fig. 1. JPEG compression noises

compress image originally stored in lossless compressed format and the same one in JPEG compressed format respectively. Fig. 1 shows noise images of JPEG compressions of TIFF image and its JPEG version. All the JPEG compressions use the standard JPEG quantization tables recommended by Independent JPEG Group (IJG). From Fig. 1, we find that (b) and (c) have different kinds of noise from (e) and (f). Hence, we can draw a conclusion that with the same quality Q, JPEG compression noise of an original lossless compressed image is quite different from that of its JPEG compressed version no matter the compression quality of the JPEG compressed version is smaller or bigger than Q. However, we prefer bigger one since there are much more difference between Fig.1(b) and (e) than that between (c) and (f).

As long as the unchanged region of a tampered image has been compressed by JPEG previously, the JPEG compression noise of unchanged region is different from that of the tampered region if we compress the whole tampered image with high quality. There are several reasons [9]:

1. If the tampered region comes from the a BMP image or other lossless compressed format image, the tampered region will have different noise as we see in Fig. 1.
2. If the tampered region comes from other JPEG image and its JPEG grid[1] is mismatched with that of the unchanged region, we can consider it as without undergoing JPEG compression before (see Fig. 2). Fig. 2 illustrates

[1] A JPEG grid is the horizontal lines and the vertical lines that partition an image into 8×8 blocks during JPEG compression.



(a) JPEG grids of two JPEG image (b) The tampered image of (a)

Fig. 2. Illustration of tampering two JPEG image. The JPEG grids of the blue region and the red region in (b) are mismatched.

the tampering operation of two JPEG images. If we compress Fig. 2(b), the tampered region (blue grid) can be considered as only undergoing once JPEG compression while the unchanged region (red grid) undergoing twice. Besides, the tampered region may undergo pre-processing (e.g. resizing or (and) rotation) which makes it like never JPEG compressed before. Hence, the JPEG compression noise should be different between these two regions.

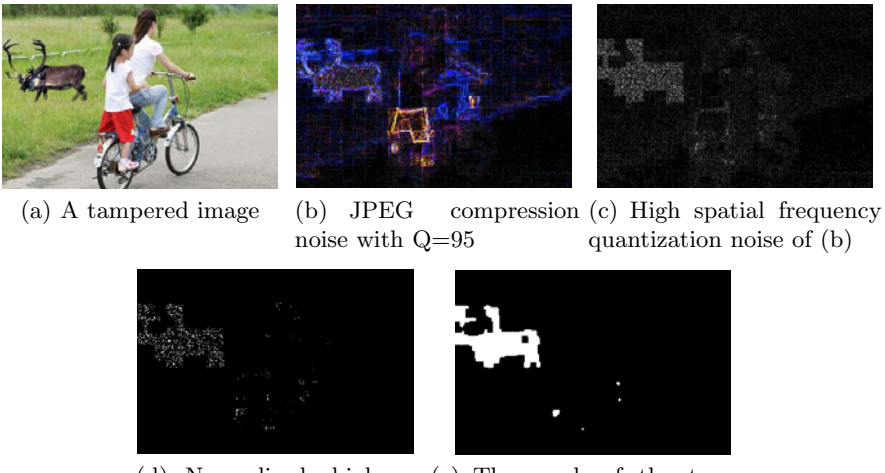
3. Even if the JPEG grids of the tampered region and the unchanged region are matched, the 8×8 blocks along the boundary of the tampered region will consist of pixels in the tampered region and also pixels in the unchanged region. These blocks have different noise from others.

When a tampered image is compressed, the unchanged region actually undergoes double JPEG compression and the tampered region can be considered as being compressed only once. If the compression is with high quality (compressed slightly), for the unchanged region, most high frequencies are erased by previous JPEG compression, hence, for the second JPEG compression, the noise almost comes from quantization of low and medium frequencies. High frequency DCT coefficients are already quantized to zeros by previous JPEG compression. However, for the tampered region, which only undergoes once JPEG compression, its compression noise contains low, medium and high frequency quantization noise. Therefore, the JPEG compression noise of the tampered image consists of two different regions. This is the basic idea of our approach for locating tampered region. It motivates us to use JPEG compression to compress a suspicious image and check whether its noise contains two different regions, just like in Fig. 3. The tampered image in Fig. 3 is generated by two different JPEG images. The animal in the right-bottom of the image is copied from another JPEG image. From Fig. 3 we can see that the unchanged region and the tampered region have different noises for JPEG compression with $Q = 95$.

The idea in this section is enlightened by [8]. *Krawetz* calls this phenomenon error level analysis. He intentionally resaves a given image at a known JPEG compression quality and calculates the difference between these two images. The tampered region will be found by just watching the difference.

(a) A tampered image (b) JPEG compression noise with Q=95

Fig. 3. A real example of a tampered image. The animal in the right-bottom of the image is the tampered region. The tampered image is generated by two different JPEG images.



(a) A tampered image (b) JPEG compression noise with Q=95 (c) High spatial frequency quantization noise of (b)

(d) Normalized highspatial frequency quantization noise (e) The mask of the tampered region location of (a)

Fig. 4. Tampered region localization of a tampered image. The animal in the middle left of the image is the tampered region.

3.2 Principal Component Analysis

Actually, only using the above idea is not enough for locating the tampered region. We are hardly able to tell the tampered region from the unchanged one sometimes just by human visual perception of JPEG compression noise. Fig. 4 shows an example of tampered image and its JPEG compression noise with quality of 95. The animal in the middle left of the image is the tampered region. However, someone may think the red shorts of the girl is tampered after seeing Fig 4(b). We need deeper analysis. As we mentioned above, JPEG compression noise is related to the quantization step. It can be roughly divided into three components: low, medium and high spatial frequency quantization noise. Low spatial frequency quantization noise comes from quantizing low frequencies DCT

coefficients while high frequency noise comes from quantizing high frequencies DCT coefficients. The biggest difference between the noises of two regions of a tampered image is high spatial frequency quantization noise. However, the noises in the above figures appear in RGB color space. In other words, they are composed of red, green and blue spectrum compression noises. Hence, how to extract high spatial frequency quantization noise from the apparent spectrum noise of JPEG compression should be a key point of tampered region localization.

For each pixel of an image, each component value (R, G, B) can be expressed as weighted combination of 8×8 DCT coefficients, as shown in equation (1). Each pixel can be considered as contains 64 spatial frequencies information. Hence, we can extract spatial frequency information from RGB values.

$$f(x,y) = \sum_{\mu=0}^{8}\sum_{\nu=0}^{8} \alpha(\mu)\alpha(\nu)C(\mu,\nu)\cos[\frac{\pi(2x+1)\mu}{16}]\cos[\frac{\pi(2y+1)\nu}{16}], \quad (1)$$

where $f(x,y)$ is component value at location (x,y) in spatial domain and $C(\mu,\nu)$ is DCT coefficient. $\alpha(\mu)$ is defined as

$$\alpha(\mu) = \begin{cases} \sqrt{1/8} & \text{for } \mu = 0 \\ \sqrt{2/8} & \text{for } \mu \neq 0 \end{cases},$$

It is well known that DCT is used in JPEG compression since it can decorrelate image data to achieve better compression. Different frequencies of DCT coefficients are nearly uncorrelated which can be justify by Fig 5. Fig 5 shows correlation coefficients distribution of 8×8 block DCT frequencies coefficients of an authentic image. We can find most of correlation coefficients are below 0.15 which means different frequencies coefficients are uncorrelated.

Therefore, different spatial frequencies quantization noise of JPEG compression should be uncorrelated. We employ PCA to extract them from RGB values, because PCA involves a mathematical procedure that transforms possibly correlated variables into uncorrelated variables (components). The first principal component accounts for as much of the variability in the data as possible, and each succeeding component accounts for as much of the remaining variability

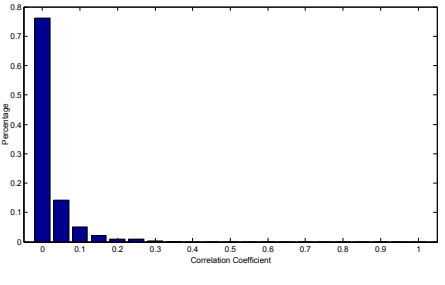


Fig. 5. Correlation coefficients distribution of 8×8 block DCT frequencies coefficients of an authentic image

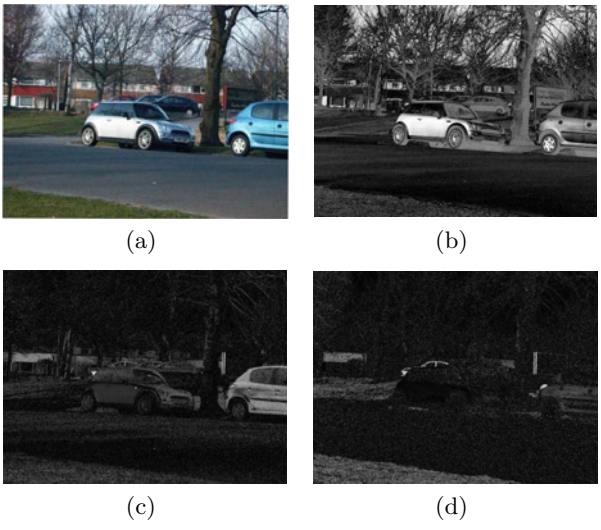


Fig. 6. PCA of an uncompressed image. (a)an uncompressed color image. (b) first PCA component of (a). (c) second PCA component of (a). (d) third PCA component of (a).

as possible [22]. PCA is theoretically the optimum transform for given data in least square terms. Fig 6 shows an example of PCA of an uncompressed color image, from which we can find that the third component of PCA is actually high frequency information (noise).

Hence, for JPEG compression noise of a given image, we take each pixel as an observation and its RGB value, i.e. red, green and blue spectrum compression noise as variables. In this way, we can get the original data set X. X is an $N \times 3$ matrix where N is the number of pixels of the given image. Our goal is to get another $N \times 3$ matrix by a linear transformation P, as shown in equation (2), so that components of re-expressed data are de-correlated.

$$Y = XP\ . \tag{2}$$

Of the re-expressed JPEG compression noise, high spatial frequency quantization noise should be the smallest variance component. As stated above, JPEG compression noise of the tampered region has stronger high spatial frequency quantization noise than that of the unchanged region. Hence, we extract high spatial frequency quantization noise to locate the tampered region.

3.3 Post-Processing

In this section, we introduce post-processing operations on the high frequency quantization noise to try to locate the tampered region. The high frequency quantization noise of a tampered image should have obviously two parts: concentrated high values region (probably the tampered region) and low values

region (probably the unchanged region). In its high values region, there may be low noise values scattered because not everywhere in the tampered region is high frequency information. We want to find these high noise values and use morphology operation to locate the tampered region.

We first employ sigmoid function (3) to normalize the high frequency noise value t to $P(t)$ within range of $[0, 1]$.

$$P(t) = \frac{1}{1 + e^{-a(t-b)}}, \tag{3}$$

where a controls the shape of function and b is determined by the high frequency noise. Fig. 4(d) shows normalized high frequency quantization noise for which $a = 3$ and b equals mean of high frequency noise plus three times of its variance.

Beyond normalizing, we also explore some morphology operations since the high value noise is not very close to each other but they are concentrated. Fig 4(e) shows the tampered region locating result of Fig 4(a).

3.4 Algorithm Overview

To summarize, our proposed algorithm for tampered region localization of digital color image are shown as follows:

Algorithm 1. Our tampered region localization algorithm

Input:
a suspicious image I

Output:
a mask of the tampered region localization result M;

1: Resave image I to JPEG image I' with quality Q;
2: $Noise = I - I'$;
3: $[repreNoise] = PCA(Noise)$;
4: Extract high frequency noise $High\_Noise = repreNoise(:, :, 3)$;
5: Normalize the high frequency noise using sigmoid function. $High\_Noise\_Norm = sigmoid(High\_Noise, a, b)$;
6: Post-processing normalized high frequency noise using morphology operations $M = imopen(imclose(High\_Noise\_Norm))$;
7: return M;

4 Experiments

We used our public color image dataset CASIA TIDE v2.0 [1] in our experiments. It consists of 7,491 authentic and 5,123 sophisticatedly tampered color images of different sizes, varying from 240×160 to 900×600. We randomly chose some tampered image in TIFF format to show here to check the effectiveness of our proposed approach. In our experiments, we set JPEG compression quality to 95 to get compression noise. We also let $a = 3$ and b equals mean of high frequency noise plus three times of its variance for sigmoid function (3). For

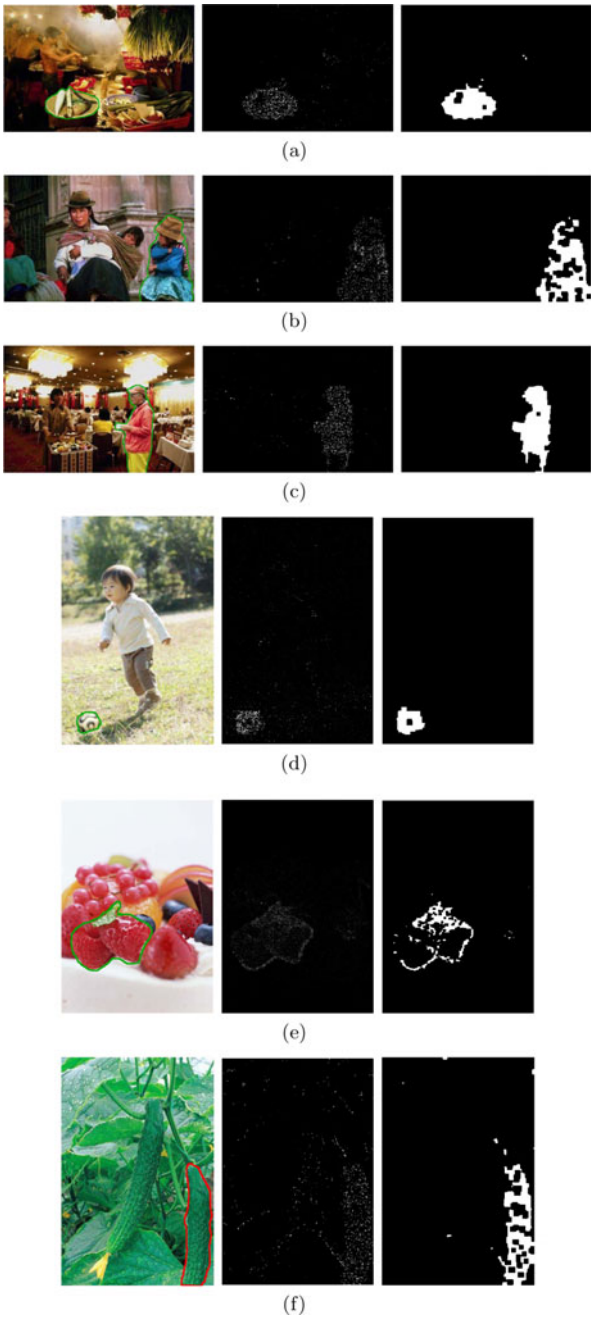
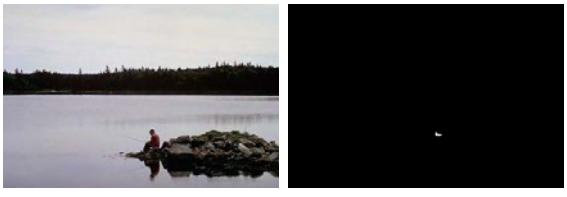


Fig. 7. Experimental results of tampered images. First column shows tampered images with ground truth marked by red or green contour. Second column is the normalized highspatial frequency quantization noises. Last column shows the masks of the tampered region locations given by our proposed algorithm.

(a)



(b)

(c)

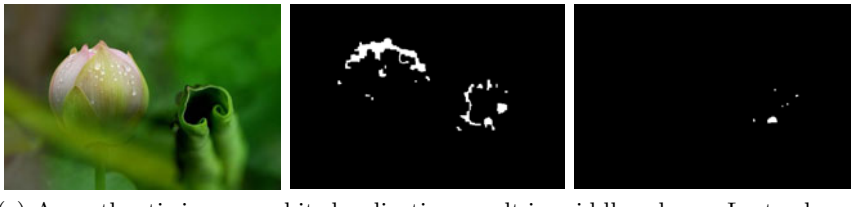
Fig. 8. Experimental results of authentic images. First column shows authentic images. Second column shows the masks of the tampered region locations given by our proposed algorithm. No tampered regions were found except some false alarm points.

morphology operations, we used matlab morphology function *imclose* with 8×8 square structure which followed by *imopen* operation with 3×3 square structure to get locations of tampered regions. All these parameters were set empirically. Fig. 7 shows the localization results of some tampered images in CASIA TIDE v2.0. First column shows tampered images with ground truth marked by red or green contour. Second column is their normalized highspatial frequency quantization noises. Last column shows the masks of the tampered region locations given by our proposed algorithm. Fig. 8 shows the localization results of some authentic images in CASIA TIDE v2.0. First column shows authentic images. Second column shows localizations results with no tampered regions being found except some false alarm points. Since not all tampered regions have enough high frequency information, the localization results of the tampered region, i.e. the white regions in masks are not always connected.

Fig. 9 shows some unsuccessful cases in our experiments, in which (a) and (b) are tampered images and its localization results, while (c) is an authentic

(a) A tampered image with background (sky) being substituted and its localization result.

(b) A tampered image of adding a flower bud in right bottom of the image and its localization result in middle column. Last column shows its localization results with JPEG compression quality $Q = 100$.



(c) An authentic image and its localization result in middle column. Last column shows its localization results with JPEG compression quality $Q = 100$.

Fig. 9. Some unsuccessful cases. (b) is a tampered image based on (c). (c) is original saved in JPEG format with quality Q=100.

image with its localization results. The image in (a) is a tampered image with background (sky) being substituted. Since the sky is almost low frequency information, we cannot use its high frequency JPEG compression noise to locate it. However, our algorithm successfully located the boundary of the tampered region. The boundary consists of pixels from the unchanged region and the tampered region. It can be considered as never JPEG compressed region. That is why our algorithm can locate it. Nonetheless, we cannot tell which part of the image is tampered from the located boundary. Hence, when the tampered region has little high frequency information, our method may fail. The image in (b) is a tampered image of (c) by adding a flower bud in the bottom right of it. Since the unchanged region of the image in (b), i.e. parts of the image in (c) is JPEG compressed with quality $Q = 100$ which can be considered as with no lossy compression, if we use $Q = 95$ to compress the image to get its JPEG compression noise, both the unchanged region and the tampered region will have strong high frequency noise. The localization result will not be correct, like middle column shows in (b). When we use $Q = 100$ to compress the image, the localization

result (last column in (b)) are correct. For the authentic image in (c), we will get false tampered region(s) localization result by compressing it with $Q = 95$, while using $Q = 100$ we can get the correct localization result. Fig. 9 (b) and (c) failed because of our underlying assumption that all JPEG format images used for tampering are saved with a quality below a reasonable value (in our experiments, we assume this value is 95). We will focus more on the estimation of Q factor in our future work.

5 Conclusions

In this paper, we have proposed an algorithm which can locate the tampered region in a lossless compressed tampered image when its unchanged region is output of JPEG decompressor. We have utilized different responses for JPEG compression of the tampered region and the unchanged region as the cue for tampered region localization. The tampered region always has some high frequency information while that of the unchanged region is almost erased by previously JPEG compression. The experimental results have proved the effectiveness of our proposed algorithm. However, if the tampered region of a tampered image has little high frequency information or the source image of its the unchanged region saved in JPEG format with higher quality than the quality we used in our experiments, our algorithm may fail. The unsuccessful cases in later situation alert us that we should estimate the JPEG compression history of a given image first and then use the reasonable quantization matrices to compress the image to do further analysis in our future work. Beside, making use of double JPEG effect like *He et al.* [5, 9] proposed approach to improve our proposed approach should also be considered in our future work.

Acknowledgments. This work is funded by National Basic Research Program (Grant No. 2004CB318100), National Natural Science Foundation of China (Grant No. 60736018, 60702024, 60723005), and National Hi-Tech R&D Program (Grant No. 2006AA01Z193, 2007AA01Z162).

References

1. CASIA Tampered Image Detection Evaluation Database (2010), `http://forensics.idealtest.org`
2. Dirik, A., Memon, N.: Image tamper detection based on demosaicing artifacts. In: IEEE International Conference onImage Processing (ICIP), pp. 1497–1500 (2009)
3. Feng, W., Liu, Z.-Q.: Region-level image authentication using bayesian structural content abstraction. IEEE Transactions on Image Processing 17(12), 2413–2424 (2008)
4. Haouzia, A., Noumeir, R.: Methods for image authentication: a survey. Multimedia Tools and Applications 39(1), 1–46 (2008)
5. He, J., Lin, Z., Wang, L., Tang, X.: Detecting doctored JPEG images via DCT coefficient analysis. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3953, pp. 423–435. Springer, Heidelberg (2006)

6. Johnson, M.K., Farid, H.: Exposing digital forgeries by detecting inconsistencies in lighting. In: ACM Multimedia and Security Workshop, pp. 1–10 (2005)
7. Johnson, M.K., Farid, H.: Exposing digital forgeries in complex lighting environments. IEEE Transactions on Information Forensics and Security 2(3), 450–461 (2007)
8. Krawetz, N.: A picture's worth: Digital image analysis and forensics (August 2007), http://www.hackerfactor.com/
9. Lin, Z., He, J., Tang, X., Tang, C.K.: Fast, automatic and fine-grained tampered jpeg image detection via dct coefficient analysis. Pattern Recognition 42(11), 2492 (2009)
10. Lukáš, J., Fridrich, J., Goljan, M.: Detecting digital image forgeries using sensor pattern noise. In: Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, vol. 6072, pp. 362–372 (February 2006)
11. Mahdian, B., Saic, S.: Blind authentication using periodic properties of interpolation. IEEE Transactions on Information Forensics and Security 3(3), 529–538 (2008)
12. Mahdian, B., Saic, S.: Detecting double compressed jpeg images. In: IET Seminar Digests 2009, vol. (2), p. P12 (2009)
13. Ng, T., Chang, S., Lin, C., Sun, Q.: Passive-blind image forensics. In: Multimedia Security Technologies for Digital Rights, ch. 6. Elsevier, Amsterdam (2006)
14. Ng, T., Chang, S., Sun, Q.: A data set of authentic and spliced image blocks. Tech. rep., DVMM, Columbia University (2004), http://www.ee.columbia.edu/ln/dvmm/downloads/AuthSplicedDataSet/photographers.htm
15. 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)
16. Popescu, A.C., Farid, H.: Exposing digital forgeries by detecting traces of resampling. IEEE Transactions on Signal Processing 53(2), 758–767 (2005)
17. Popescu, A.C., Farid, H.: Exposing digital forgeries in color filter array interpolated images. IEEE Transactions on Signal Processing 53(10), 3948–3959 (2005)
18. Shi, Y., Chen, C., Xuan, G.: 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)
19. Wang, W., Dong, J., Tan, T.: A survey of passive image tampering detection. In: Ho, A.T.S., Shi, Y.Q., Kim, H.J., Barni, M. (eds.) IWDW 2009. LNCS, vol. 5703, pp. 308–322. Springer, Heidelberg (2009)
20. Wang, W., Dong, J., Tan, T.: Effective image splicing detection based on image chroma. In: IEEE International Conference on Image Processing, pp. 1257–1260 (2009) (accepted)
21. Wang, W., Dong, J., Tan, T.: Image tampering detection based on stationary distribution of markov chain. In: IEEE International Conference on Image Processing (2010) (accepted)
22. Wikipedia: Principal component analysis — wikipedia, the free encyclopedia (2010), http://en.wikipedia.org/w/index.php?title=Principal_component_analysis\&oldid=366194078 (accessed June 12, 2010)

Robust Audio Watermarking by Using Low-Frequency Histogram

Shijun Xiang*

Department of Electronic Engineering,
School of Information Science and Technology,
Jinan University, Guangzhou 510632, China
xiangshijun@gmail.com

Abstract. In continuation to earlier work where the problem of time-scale modification (TSM) has been studied [1] by modifying the shape of audio time domain histogram, here we consider the additional ingredient of resisting additive noise-like operations, such as Gaussian noise, lossy compression and low-pass filtering. In other words, we study the problem of the watermark against both TSM and additive noises. To this end, in this paper we extract the histogram from a Gaussian-filtered low-frequency component for audio watermarking. The watermark is inserted by shaping the histogram in a way that the use of two consecutive bins as a group is exploited for hiding a bit by reassigning their population. The watermarked signals are perceptibly similar to the original one. Comparing with the previous time-domain watermarking scheme [1], the proposed watermarking method is more robust against additive noise, MP3 compression, low-pass filtering, etc.

1 Introduction

With the development of the Internet, illegal copying of digital audio have become more widespread. As a traditional data protection method, encryption cannot be applied in that the content must be played back in the original style. There is a potential solution to the problem, that is to mark the audio signal with an imperceptible and robust watermark [2,3,4].

In the past 10 years, attacks against audio watermarking are becoming more and more complicated with the development of watermarking technique. According to International Federation of the Phonographic Industry (IFPI) [5], in a desired audio watermarking system, the watermark should be robust to content-preserving attacks including desynchronization attacks and additive noise-like audio processing operations. From the audio watermarking point of view, desynchronizaiton attacks (such as cropping and time-scale modification)

* This work was supported in part by NSFC (60903177), in part supported by Ph.D. Programs Foundation of Ministry of Education of China (200805581048), the Fundamental Research Funds for the Central Universities (21609412), the research funding of Jinan University (51208050) and the Project-sponsored by SRF for ROCS, SEM ([2008]890).

H.-J. Kim, Y. Shi, and M. Barni (Eds.): IWDW 2010, LNCS 6526, pp. 134–147, 2011.

mainly introduce synchronization problems between encoder and decoder. The watermark is still present, but the detector is no longer able to extract it. Different from desynchronization attacks, additive noise-like processing operations (including requantization, Gaussian noise, MP3 lossy compression, and low-pass filtering) do not cause synchronization problems, but will reduce the watermark energy.

The problem of audio watermarking against noise-like operations can be solved by embedding a mark in frequency domain instead of in time domain. The time domain based solutions (such as LSB schemes [6] and echo hiding [7] usually have a low computational cost but somewhat sensitive to additional noises, while the frequency domain watermarking methods provide a satisfactory resistance to additive noise-like operations by watermarking low-frequency component of audio signal. There are three dominant frequency domain watermarking methods: Discrete Fourier Transform (DFT) based [8,9], Discrete Wavelet Transform (DWT) based [10,11] and Discrete Cosine Transform (DCT) based [12]. They have shown satisfactory robustness performance to MP3 lossy compression, additional noise and low-pass filtering operations.

In the literature, there are a few algorithms aiming at solving desynchronization attacks. For cropping (such as editing, signal interruption in wireless transmission and data packet loss in IP network), researchers repeatedly embedded a template into different regions of the signal, such as synchronization code based self-synchronization methods [10,11,12,13] and the use of multiple redundant watermarks[14,15]. However, the template-based watermarking can not cope with time-scale modification (TSM) operations. In the audio watermarking community, there exists some TSM-resilient watermarking strategies, such as peak points based [16,17] and recently reported histogram based [1]. In [16], a bit can be hidden by quantizing the length of each two adjacent peak points. In [17], the watermark was repeatedly embedded into the edges of an audio signal by viewing pitch-invariant TSM as a special form of random cropping, removing and adding some portions of the audio signal while preserving the pitch. The two dominant peak points-based watermarking method is resistant to TSM because the peaks is still able to be detected after a TSM operation. The histogram-based methods [1] are robust to TSM operations because the shape of histogram of an audio signal is provably invariant to temporal linear scaling. In addition, the histogram is independent of the samples in position.

In this work, we consider the problem of a watermark's resistance to both desynchronization and additive noise-like operations. In this direction, an existing work [18] has modeled the effect of jitter and additive noise on a watermark. In practice, it still lacks of a reliable technical solution for such a problem. Towards this end, in our earlier watermarking scheme [1], the watermark for synchronization attacks has been better solved by using time domain histogram feature by shaping the histogram for watermarking. In this paper, the watermark's resistance to TSM and additive noise-like is simultaneously achieved by computing the histogram from low-frequency component of a Gaussian-filtered audio file. Experimental results show that the watermark in the low-frequency domain

is more robust to additive noise-like operations such as Gaussian noise, MP3 compression and low-pass filtering while keeping satisfactory robustness to TSM.

In the next section, a detailed description of our proposed watermarking method is introduced. Then, we analyze underlying robustness principle of the watermark to TSM and additive noises. This is followed by a discussion on watermark performance. Furthermore, we examine imperceptibility and robustness of the watermark. Finally, we draw the conclusions.

2 Proposed Watermarking Algorithm

Histogram shape of an audio signal is a time-scale invariant and cropping resistant feature, which has been fully proved in our previous work [1]. Due to the watermark in [1] embedded into time domain, the watermarking system cannot provide a satisfactory robustness performance to additive noise-like audio processing manipulations (such as additive Gaussian noise, low-pass filtering, MP3 lossy compression, etc). To enhance the robustness to additive noises while keeping the resistance to TSM and cropping attacks, in this paper we design a robust audio watermarking algorithm by extracting the watermark features (histogram and absolute mean) from the Gaussian filtered low-frequency component of audio files.

2.1 Watermark Insertion

As illustrated in Fig. 1, the watermark insertion consists of two broad steps: Gaussian filtering and histogram-based embedding.

Gaussian Filtering. The input image (F) is filtered with a Gaussian kernel low-pass filter for removing the high-frequency information F_{High}. The design of the Gaussian filter will be discussed in detail in Section 3.2.

The use of Gaussian filter-based preprocessing step for the extraction of the robust feature is not new. In [19,20], the authors have used the two-dimensional

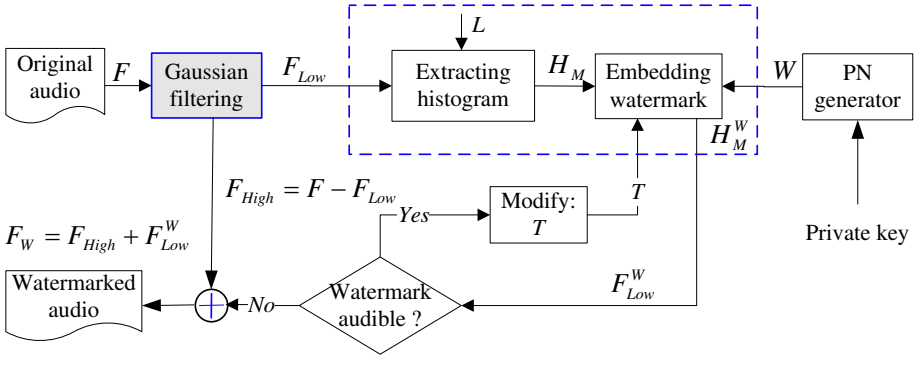


Fig. 1. Watermark embedding framework

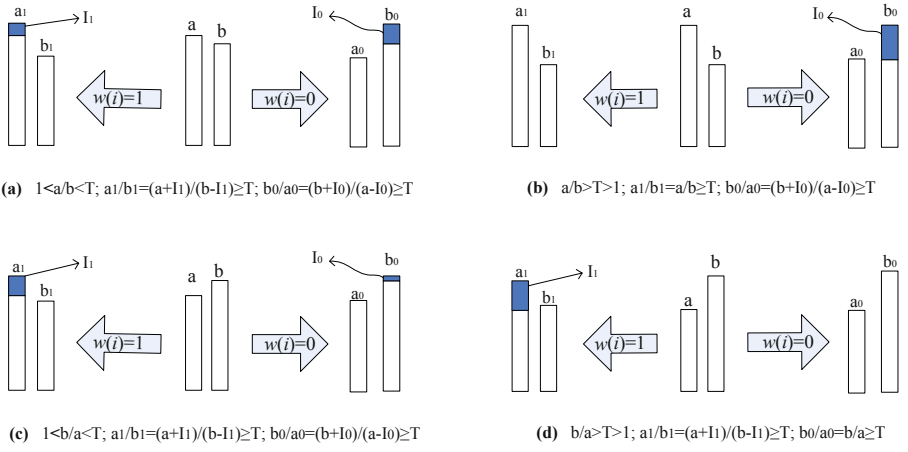


Fig. 2. Illustration of embedding one bit of watermark information. There are four cases in total: (a) $a > b$ and $a/b < T$, (b) $a > b$ and $a/b \geq T$, (c) $b > a$ and $b/a < T$, and (d) $b < a$ and $b/a \geq T$. I_1 and I_0 are the numbers of the least modified samples according to the watermark embedding rule.

(2-D) filter to seek image meshes and extract the low-frequency component for watermarking. Differently, in this work we exploit the one-dimensional (1-D) Gaussian filter to extract low-frequency component for audio watermarking.

Histogram-Based Embedding. The histogram (H_M) is extracted from the filtered image (F_{Low}) by referring to its absolute mean value in such a way that the watermark is immune to volume change. Divide the bins into many groups, each two neighboring bins as a group. Based on the resistance of the ratios of sample count among different histogram bins to synchronization attacks addressed in [1], we design the following rule to embed one bit of message by reassigning the number of samples in two neighboring bins, as illustrated in Fig. 2.

In this work, the watermark, denoted by $W = \{w_i \mid i = 1, \cdots, L_w\}$, is a key-based PN sequence. The private key is shared with the detector during the decision-making for presence of the watermark as used in [21]. The average value of the low-frequency component F_{Low} of a digital image F is calculated as $\bar{A}$. By referring to $\bar{A}$, an embedding range denoted by $B = [(1-\lambda)\bar{A}, (1+\lambda)\bar{A}]$ is designed for the extraction of the histogram $H = \{h_M(i) \mid i = 1, \cdots, L\}$, where L should not be less than $2L_w$ in order to embed all bits.

After extracting the histogram, let Bin_1 and Bin_2 be two consecutive bins including the number of samples denoted by a and b. The watermark embedding rule is formulated as

$$\begin{cases} a/b \geq T & \text{if} \quad w(i) = 1 \\ b/a \geq T & \text{if} \quad w(i) = 0 \end{cases} \tag{1}$$

where T is a threshold reflecting the number of modified samples. Imperceptibility of the watermark expressed with the SNR value is ensured over 20 dB by adaptively controlling the T value. The SNR is computed in the low-frequency component (refer to Section 4.1).

Consider the case that $w(i)$ is bit value $'1'$. If $a/b \geq T$, no operation is needed. Otherwise, the number of samples in two neighboring bins, a and b, will be reassigned until satisfying the condition $a_1/b_1 \geq T$. In case of embedding a bit value of $'0'$, the procedure is similar. The rules to reassign the samples are formulated as shown in Equations (2) and (3).

If $w(i)$ is $'1'$ and $a/b < T$, some randomly selected samples from Bin_2, in the number denoted by I_1, will be modified to fall into Bin_1, achieving $a_1/b_1 \geq T$. If $w(i)$ is $'0'$ and $b/a < T$, some randomly selected samples from Bin_1 are put into Bin_2 , satisfying the condition $b_0/a_0 \geq T$. The rule for reassigning the samples is shown in Equation (2).

$$\begin{cases} f_1'(i) = f_1(i) + M, & 1 \leq i \leq I_0 \\ f_2'(j) = f_2(j) - M, & 1 \leq j \leq I_1 \end{cases} \tag{2}$$

where M is the bin width, $f_1(i)$ is the i^{th} modified sample in Bin_1, and $f_2(j)$ denotes the j^{th} modified sample in Bin_2. The modified samples $f_1'(i)$ will fall into Bin_2 while the sample $f_2'(i)$ goes to Bin_1. I_0 and I_1 can be computed by using the following mathematical expressions,

$$\begin{cases} I_0 = (T * b - a)/(1 + T) & making \quad a_1/b_1 \geq T \quad from \quad a/b \leq T \\ I_1 = (T * a - b)/(1 + T) & making \quad b_0/a_0 \geq T \quad from \quad b/a \leq T, \end{cases} \tag{3}$$

where $a_1 = a + I_1$, $b_1 = b - I_1$, $a_0 = a - I_0$, and $b_0 = b + I_0$.

This process is repeated until all bits in the PN sequence are embedded. The watermarked histogram is denoted by H_M^W. The marked version of F_{Low} is denoted by F_{Low}^W. Finally, the marked image (F_W) is generated by combining F_{Low}^W and F_{High}.

2.2 Watermark Recovery

Denote the attacked image F_W', which has undergone some content-preserving attacks. The mean value $\bar{A}'$ is computed from the filtered version of F_W' with the Gaussian filter of standard deviation σ as the embedding process. The goal is to get an estimate of the embedded PN sequence. We denote the estimate by $W' = \{w'(i) \mid i = 1, \cdots, L_w\}$. A low bit error rate between W and W' will indicate the presence of the watermark.

Referring to Fig. 3, the watermark detection includes two crucial processes:

i) *The mean value-based search*: In the extraction procedure, a searching space $SM = [\bar{A}'(1-\triangle_1), \bar{A}'(1+\triangle_2)]$ is designed according to the error ratio of the mean induced by various content-preserving attacks as shown in Fig. 5 in such a manner that exhaustive search can be avoided. The searching factors

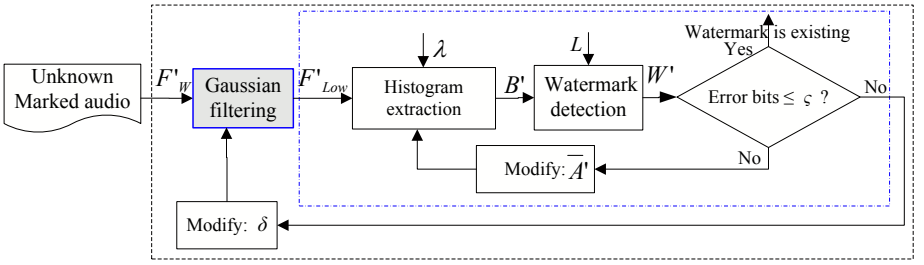


Fig. 3. Watermark extraction framework

$\triangle_1$ and $\triangle_2$ should not be less than 7% for tolerating the effect of various attacks on the mean.

ii) *The σ-based search*: The Gaussian kernel filter is additive-noise resistant, but is not invariant to time-scale modification. Considering the effect of TSM, we design the σ-based matching process. In the detection, the σ is assigned from the set of five elements $nSigma = [\sigma, \frac{19\sigma}{20}, \frac{21\sigma}{20}, \frac{9\sigma}{10}, \frac{11\sigma}{10}, \frac{4\sigma}{5}, \frac{6\sigma}{5}]$ in order. The fist element is the standard deviation used in the embedding process for additive noises without involving scaling. The other elements are respectively for processing those possible time scale modifications with factors $[95\%, 105\%, 90\%, 110\%, 80\%, 120\%]$[1].

The detailed search process is as follows:

a) First, we use σ (the same as in the embedding phase) for the Gaussian filtering to start the mean-based search process:
   i) Compute the histogram of F'_W from the range $B = [(1-\lambda)\bar{A}', (1+\lambda)\bar{A}']$ with L equal-sized bins as in the process of watermark embedding. $\bar{A}'$ is the mean of F'_W.
   ii) Divide the histogram bins as groups, two neighboring bins as a group. Suppose that the numbers of samples in two consecutive bins, are a' and b', respectively. By comparing their population relation, one hidden bit is able to be extracted by the following equation

$$w'(i) = \begin{cases} 1 & \text{if} \quad a'/b' \geq 1 \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

   The process is repeated until all bits are extracted.
   iii) If the detected sequence is matched with W, the mean-based searching process is completed. Otherwise, keep the best matching sequence as W' and let $\bar{A}' = \bar{A}' + S$ if $mod(O, 2) = 0$ or $\bar{A}' = \bar{A}' - S$ if $mod(O, 2) = 1$ where O are the searching times and S is the step size. The searching times and the search step will be discussed in Section 4.3.
   iv) Repeat step i) for the new search until the mean-based search is over.

[1] That is, $\frac{19\sigma}{20}$ is for the scaling of 95%, $\frac{21\sigma}{20}$ is for the scaling of 105%, $\frac{9\sigma}{10}$ is for the scaling of 90%, $\frac{11\sigma}{10}$ is for the scaling of 110%, $\frac{4\sigma}{5}$ is for the scaling of 80%, $\frac{6\sigma}{5}$ is for the scaling of 120%.

b) If the bit error rate (BER) between W' and W is not greater than a threshold $\frac{\zeta}{length(W)}$ (ζ is the number of the error bits, defined as 13 for a 60-bit PN sequence), the audio is claimed as the *marked* copy. The sequence W' is taken as the estimation of the watermark. Otherwise, it means that the watermarked audio may suffer from the time-scale operation. Select the next element from the set $nSigma$ for Gaussian filtering and repeat *step i)* for the new search.

In the watermark recovery, the PN sequence can be recovered with the private key, and the embedding parameters, F_w, λ, and σ are known beforehand. Thus, the watermark can be extracted blindly by Equation (4) without knowledge of the original audio files. Corresponding to the effect of synchronization attacks and the mean's alteration under various additive noise-like attacks, in this work we search for the watermark by referring to the mean and the standard deviation, so that the watermark robustness can achieve the expected robustness against various synchronization attacks and audio processing operations.

3 Underlying Robustness Principles

In this section, we will address the watermark's robustness principles by introducing the property of the time domain histogram in shape and 1-D Gaussian low-pass filtering operation.

3.1 Time Domain Histogram

Pitch-invariant TSM is a special time-scale algorithm with the consideration of human auditory system. Under the TSM, the duration of an audio file is compressed or expanded but the pitch preserves. Since the human ears are not sensitive to the TSM, it is taken as a challenging desynchronization attack in the audio watermarking community. In the previous work [1], we have showed that the histogram in shape and the absolute mean computed from the time domain are two robust features to the TSM. The resistance of the histogram in shape to cropping attacks (including local cropping and jittering) is also fully addressed in [1]. About the theoretical proof and experimental testing of the robustness, please refer to our previous work [1].

In this section, we will report that the property of the histogram shape [1] can be applied in the low-frequency component of a Gaussian-filtered audio signal, so that the low-frequency histogram in shape is able to be applied as an audio watermark feature for the TSM and additive noise-like operations.

3.2 Gaussian Low-Pass Filter

In the literature (such as [19,20]), 2-D low-pass filtering has been shown to be an effective preprocessing step for enhancing robustness of the hash and the watermark to image processing operations, such as common compression and

filtering operation. The 1-D low-pass filtering operation can be represented as the convolution of the Gaussian function $G(x, \sigma)$ and an audio file $F(x)$:

$$F_{Low}(x) = \psi(x, \sigma, \mu) * F(x). \tag{5}$$

The 1-D Gaussian filter has an impulse response given by:

$$\psi(x, \sigma, \mu) = \frac{1}{\sqrt{2\pi}\sigma} exp^{-\frac{(x-\mu)^2}{2\sigma^2}}, \tag{6}$$

where σ is the standard deviation of the distribution. As shown in Fig. 4, the red line is the standard normal distribution with the parameters of $\sigma^2 = 1$ and $\mu = 0$.

Since digital audio clip is stored as a collection of discrete samples we need to produce a discrete approximation to the Gaussian function before we can perform the convolution. Let the Gaussian function be represented with at least k times of standard deviation, which is described as a 1-D filter of length $2 * k * \sigma + 1$. In theory, the Gaussian distribution is non-zero everywhere, which would require an infinitely large convolution mask, but in practice *three* standard deviations from the mean can effectively represent 99.7% the energy of the Gaussian distribution, and so we can truncate the mask at point of $k = 3$.

How to choose σ is a crucial step since it plays an important role on the kernel size of the filter. Too large a kernel will excessively smooth out the dynamic content of the audio files, and too small a kernel will impact the robustness due to the use of some high-frequency information for watermarking. When the

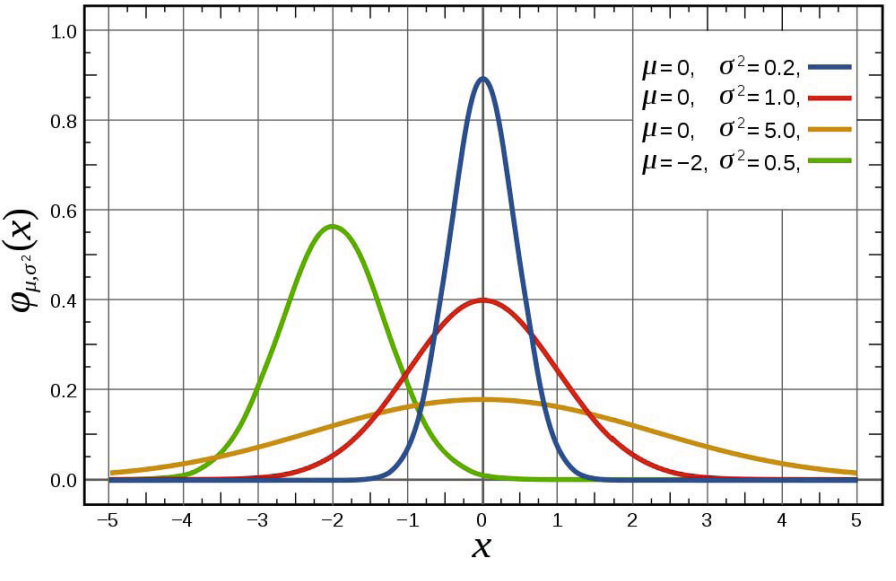


Fig. 4. Distribution of 1-D Gaussian filter

standard deviation σ is 10 and the mean μ is ZERO[2], we can gain a satisfactory result according to our observations.

Resistance of the Mean. Take a SQAM [22] test clip *Bass.wav* as example file. The absolute mean of this file in the low-frequency domain is 1523.8641. Fig. 5 illustrates the proportional deviation of the absolute mean in the Gaussian filtered low-frequency component under the pitch-invariant TSM with scaling factors of 80%-120% (see Fig. 5(a)) and additive noises including MP3 lossy compression with bit rates of 32-128 kbps (see Fig. 5(b)), additive Gaussian noise with power strength of 30-45 dB (see Fig. 5(c)) and low-pass filter with cutoff frequency of 5-8 kHz (see Fig. 5(d)). The vertical axes is the proportional deviation of the absolute mean, averaging over the 10 tested clips. We can see from the figure 5 that the absolute mean is a robust feature to the TSM and additive noises. And, one can observe that the deviation is increasing as the scaling factor or the strength of additive noises increases. Given the worst case of the TSM of 80%, the absolute maximum of the deviation is less than -5%. For additive noises, the worst case is the MP3 compression operation of 32 kbps, which will cause the proportional deviation of -7%. We refer to the maximum deviation caused by the TSM operations and additive noises as $\triangle$. The value of $\triangle$ plays an important role in the watermark detection phase as addressed in Section 4.3.

4 Performance Analysis

In this section, we evaluate the performance of the proposed watermarking algorithm in terms of SNR computation, the embedding capacity (payload), and computational cost in the extraction.

4.1 SNR Computation

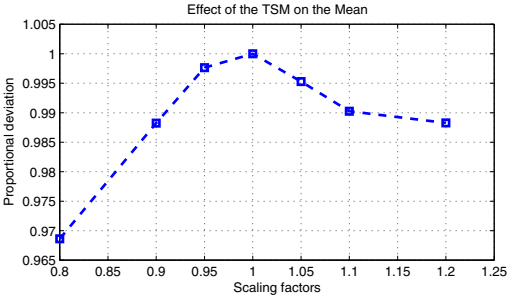
Since we use the low-frequency component for watermarking and the embedding process keeps the high-frequency information unchanged, the SNR value can be computed by using the low-frequency component:

$$\begin{aligned} SNR &= -10\log_{10}\left(\frac{||F - F^w||^2}{||F||^2}\right) \\ &= -10\log_{10}\left(\frac{||F_{Low} - F^w_{Low}||^2}{||F_{Low}||^2}\right), \end{aligned} \tag{7}$$

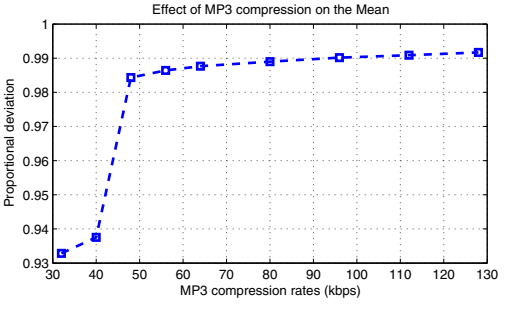
where F and F^w denote the time-domain signals before and after watermarking, F_{Low} and F^w_{Low} are their corresponding low-frequency component, and $||F||$ denotes Euclidean norm of the signal F[3]. The above conclusion is very useful for

[2] That is, the Gaussian filter kernel in length is 61 ($2 * k * \sigma + 1 = 2 * 3 * 10 + 1 = 61$).

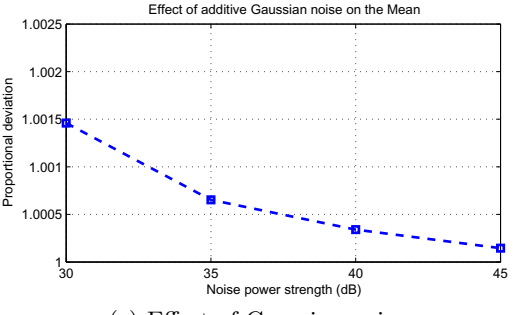
[3] The audio file F is an 1-D signal, which can be considered as 1-D vector for the computation of Euclidean norm.



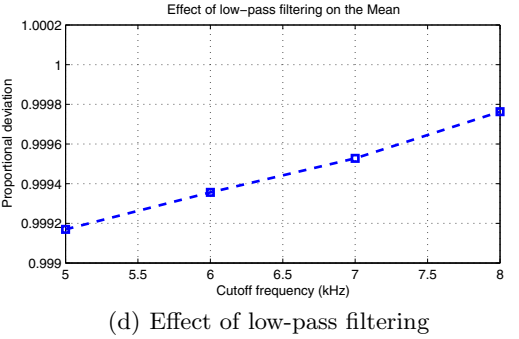
(a) Effect of the TSM



(b) Effect of MP3 compression



(c) Effect of Gaussian noise



(d) Effect of low-pass filtering

Fig. 5. Effect of the TSM and additive noises (including MP3 compression, Gaussian noise and low-pass filter operation) on the absolute mean

reducing the computational cost in the watermark embedding phase since we can use T to control the embedding distortion in the low-frequency domain instead of in the spatial domain. In such a way, the time-consuming Gaussian filtering processing operation in the embedding is not necessary to repeatedly run.

4.2 Embedding Capacity

Suppose that the mean of audio is $\bar{A}$ and the parameter λ is applied to compute the embedded region. The embedding capacity P of the proposed algorithm can be expressed as

$$P = 2\lambda \cdot \bar{A}/(M \cdot G) \tag{8}$$

where M denotes the size of the bins, and G is the number of the bins designed to embed one bit. In this paper, G is set as 2 (two bins for a watermark bit).

4.3 Computational Cost in the Detection

In the extraction, an ideal searching step is related to λ, and designed as $S = 1/\lambda$ so that the selected amplitude range B is added or reduced with a unit at a time. The maximum searching times is estimated as

$$\begin{aligned} O &= \bar{A} \cdot (\triangle_1 + \triangle_2)/S \\ &= \lambda \cdot \bar{A} \cdot (\triangle_1 + \triangle_2) \end{aligned} \tag{9}$$

Equation (9) shows that for a suspected audio file, the computational cost in decision-making for presence of the watermark is related to its absolute mean $\bar{A}$ and the maximal deviation of the mean possibly caused by an acceptable level of content-preserving operations $\triangle_1$ and $\triangle_2$. In this paper, $\triangle_1$ and $\triangle_2$ are given as 7% for most content-preserving attacks (including TSM and additive noises).

Notably, the watermarking scheme proposed in this paper has avoided exhaustive search. Due to the watermark being embedded into the histogram, the computational cost is low.

5 Experimental Results

The parameter values are given as follows: $\lambda = 2.45$ and $T = 1.4$. 120 bins extracted from a 20s light music entitled *danube.wav* is watermarked with 60 bits of PN sequence. Referring to the error ratio of the mean caused by the TSM and additive noises in Fig. 5, we assign $\triangle_1 = \triangle_2 = 7\%$ for the watermark detection. The audio editing and attacking tools adopted in our experiments are CoolEditPro v2.1, and the objective difference grade (ODG) is estimated by EAQUAL 0.1.3 [23] alpha which considers the Homan Auditory System.

Imperceptibility of the watermark is implemented by the SNR and ODG standards. In the low-frequency component, the SNR value is computed as 40.85 dB with the ODG score of -2.34. In time domain, the SNR and ODG values are 40.97 dB and -2.33, respectively. It shows the watermark distortion in the

Table 1. Watermark Robustness Against Common Processing Operations

In [1]	BER(%)	Proposed	BER(%)
Resampling 44.1-16-44.1 (kHz)	0	Resampling 44.1-16-44.1 (kHz)	0
Re-quantization 16-32-16 (bit)	0	Re-quantization 16-32-16 (bit)	0
Volume (±20%)	0	Volume (±20%)	0
Low-pass filtering (8 kHz)	10 ($=\frac{6}{60}$)	Low-pass filtering (8 kHz)	5 ($=\frac{3}{60}$)
Low-pass filtering (7 kHz)	-	Low-pass filtering (7 kHz)	6.7 ($=\frac{4}{60}$)
Low-pass filtering (6 kHz)	-	Low-pass filtering (6 kHz)	10 ($=\frac{6}{60}$)
Gaussian noise (35 dB)	0	Gaussian noise (35 dB)	0
Gaussian noise (40 dB)	10 ($=\frac{6}{60}$)	Gaussian noise (40 dB)	0
Gaussian noise (45 dB)	-	Gaussian noise (45 dB)	1.7 ($=\frac{1}{60}$)

Table 2. Watermark Robustness Against MP3 Lossy Compression

In [1]	BER(%)	Proposed	BER(%)
MP3 (128 kbps)	10 ($=\frac{6}{60}$)	MP3 (128 kbps)	0
MP3 (112 kbps)	-	MP3 (112 kbps)	0
MP3 (96 kbps)	-	MP3 (96 kbps)	0
MP3 (80 kbps)	-	MP3 (80 kbps)	0
MP3 (64 kbps)	-	MP3 (64 kbps)	0
MP3 (56 kbps)	-	MP3 (56 kbps)	3.3 ($=\frac{2}{60}$)

time domain and low-frequency component are almost equal. Subjective listening testing also shows that the watermark is inaudible.

The robustness experimental results of the watermark in the low-frequency component and in the time domain against the TSM and additive noise operations (such as Gaussian white noise, low-pass filtering, re-quantization, and re-sampling) are listed in Table 1 with given BER (Bit Error Rate). Suppose that the BER should be less than 15% for decision-making presence of the watermark. When the BER between the inserted PN sequence and the sequence detected from a clip is less than 15%, the clip can be claimed as a $'watermarked'$ version. When the BER is higher than 15%, we consider that a marked file is missed, (denoted by the symbol $'-'$).

Comparing with the watermarking scheme in the time domain [1], it is noting from Tables 1-4) that the proposed watermarking scheme in the low-frequency component indeed improves the watermark performance to additive noise attacks (such as MP3 compression and low-pass filtering) while it is still robust to the pitch-invariant TSM of ±20%, local cropping (such as deleting one part of the signal in length 2s), and jittering (such as, deleting one from each 400 samples randomly). In [1], the time domain watermark is robust to MP3 compression of 128 kbps, low-pass filtering of 8 kHz and additive Gaussian noise of 35 dB. In the proposed method, the low-frequency domain watermark is robust to MP3 compression of 56 kbps, low-pass filtering of 6 kHz and additive Gaussian noise of 45 dB.

Table 3. Watermark Robustness Against Local Cropping and Jitter Attacks

In [1]	**BER(%)**	**Proposed**	**BER(%)**
Cropping (5%)	0	Cropping (5%)	0
Cropping (10%)	1.7 ($=\frac{1}{60}$)	Cropping (10%)	3.3 ($=\frac{2}{60}$)
Cropping (15%)	10 ($=\frac{6}{60}$)	Cropping (15%)	11.7 ($=\frac{11}{60}$)
Jitter (1/400)	0	Jitter (1/400)	0
Jitter (1/200)	0	Jitter (1/200)	0
Jitter (1/100)	0	Jitter (1/100)	0

Table 4. Watermark Robustness Against Pitch-invariant TSM

In [1]	**BER(%)**	**Proposed**	**BER(%)**
pitch-invariant TSM (20%)	0	pitch-invariant TSM (20%)	0
pitch-invariant TSM (15%)	0	pitch-invariant TSM (15%)	0
pitch-invariant TSM (10%)	0	pitch-invariant TSM (10%)	0
pitch-invariant TSM (5%)	0	pitch-invariant TSM (5%)	0
pitch-invariant TSM (−5%)	0	pitch-invariant TSM (−5%)	0
pitch-invariant TSM (−10%)	0	pitch-invariant TSM (−10%)	0
pitch-invariant TSM (−15%)	0	pitch-invariant TSM (−15%)	0
pitch-invariant TSM (−20%)	0	pitch-invariant TSM (−20%)	0

6 Conclusions and Future Works

In this paper, we show the invariance of the histogram and mean to TSM in the time domain [1] can be extended to the low frequency domain by applying Gaussian filter. As a bonus, the watermark can be embedded by modifying the histogram shape of the low-frequency component and achieve the goal robust against both TSM and additive noises. In future researches, one consideration is to embed an inaudible watermark into local histogram by using the Human Auditory System instead of the SNR standard.

References

1. Xiang, S., Huang, J.: Histogram-Based Audio Watermarking against Time-Scale Modification and Cropping Attacks. IEEE Transactions on Multimedia 9(7), 11357–11372 (2007)
2. Arnold, M.: Audio Watermarking: Features, Applications and Algorithms. In: Proc. IEEE International Conference on Multimedia and Expo., New York, USA, vol. 2, pp. 1013–1016 (2000)
3. Swanson, M.D., Zhu, B., Tewfik, A.H.: Robust Audio Watermarking Using Perceptual Masking. Signal Processing 66(3), 337–355 (1998)
4. Swanson, M.D., Zhu, B., Tewfik, A.H.: Current State of the Art, Challenges and Future Directions for Audio Watermarking. In: Proc. of IEEE Int. Conf. on Multimedia Computing and Systems, vol. 1, pp. 19–24 (1999)

5. Katzenbeisser, S., Petitcolas, F.A.P. (eds.): Information Hiding Techniques for Steganography and Digital Watermarking. Artech House, Inc., Norwood (2000)
6. Gerzon, M.A., Graven, P.G.: A High-Rate Buried-Data Channel for Audio Cd. Journal of the Audio Engineering Society 43, 3–22 (1995)
7. Gruhl, D., Lu, A., Bender, W.: Echo Hiding. In: Anderson, R. (ed.) IH 1996. LNCS, vol. 1174, pp. 295–315. Springer, Heidelberg (1996)
8. Bender, W., Gruhl, D., Morimoto, N.: Techniques for Data Hiding. IBM Systems Journal 35, 313–336 (1996)
9. Lee, S.K., Ho, Y.S.: Digital Audio Watermarking in the Cepstrum Domain. IEEE Transactions on Consumer Electronics 46, 744–750 (2000)
10. Kim, H.O., Lee, B.K., Lee, N.Y.: Wavelet-Based Audio Watermarking Techniques: Robustness and Fast Synchronization, http://amath.kaist.ac.kr/research/paper/01-11.pdf
11. Wu, S., Huang, J., Huang, D.R., Shi, Y.Q.: Efficiently Self-Synchronized Audio Watermarking for Assured Audio Data Transmission. IEEE Transactions on Broadcasting 51(1), 69–76 (2005)
12. Huang, J.W., Wang, Y., Shi, Y.Q.: A Blind Audio Watermarking Algorithm with Self-Synchronization. In: Proc. of IEEE Int. Sym. on Circuits and Systems, vol. 3, pp. 627–630 (2002)
13. Lie, W.N., Chang, L.C.: Robust and High-Quality Time-Domain Audio Watermarking Based on Low-Frequency Amplitude Modification. IEEE Transactions on Multimedia 8(1), 46–59 (2006)
14. Bassia, P., Pitas, I., Nikolaidis, N.: Robust Audio Watermarking in the Time Domain. IEEE Transactions on Multimedia 3(2), 232–241 (2001)
15. Kirovski, D., Malvar, H.: Spread-Spectrum Watermarking of Audio Signals. IEEE Transactions on Sginal Processing 51(4), 354–368 (2003)
16. Mansour, M., Tewfik, A.: Data Embedding in Audio Using Time-Scale Modification. IEEE Transactions on Speech and Audio Processing 13(3), 432–440 (2005)
17. Li, W., Xue, X.: Content Based Localized Robust Audio Watermarking Robust against Time Scale Modification. IEEE Transaction On Multimedia 8(1), 60–69 (2006)
18. Zaidi, A., Boyer, R., Duhamel, P.: Audio Watermarking Under Desynchronization and Additive Noise Attacks. IEEE Transactions On Signal Processing 54(2), 570–584 (2006)
19. Lu, C.S., Sun, S.W., Hsu, C.Y., Chang, P.C.: Media Hash-dependent Image Watermarking Resilient Against Both Geometric Attacks and Estimation Attacks Based on False Positive-Oriented Detection. IEEE Transactions on Multimedia 8(4), 668–685 (2006)
20. Xiang, S., Kim, H.J., Huang, J.: Invariant Image Watermarking Based on Statistical Features in the Low-Frequency Domain. IEEE Transactions on Circuits and Systems for Video Technology 18(6), 777–790 (2008)
21. Zheng, D., Zhao, J., Saddik, A.: Rst-Invariant Digital Image Watermarking Based on Log-Polar Mapping and Phase Correlation. IEEE Transactions on Circuits and Systems for Video Technology 13(8), 753–765 (2003)
22. [Online]. Available, http://sound.media.mit.edu/mpeg4/audio/sqam/
23. [Online]. Available, http://www.mp3-tech.org/programmer/sources/eaqual.tgz

Robust Blind Watermarking Scheme Using Wave Atoms

H.Y. Leung and L.M. Cheng

Department of Electronic Engineering
City University of Hong Kong
hyleung@cityu.edu.hk, itlcheng@cityu.edu.hk

Abstract. In this paper, a robust blind watermarking scheme using Multiple Descriptions (MD) is proposed. The watermark is embedded in the Wave Atom Transform domain by modifying one of the scale bands. The detection and extraction procedure do not need the original host image. We tested the proposed algorithm against nine types of attacks like JPEG compression, Gaussian Noise addition, Median Filtering, Salt and Pepper noise, etc. They are carried out using Matlab Software. The experimental results demonstrate that the proposed algorithms have great robustness against various imaging attacks.

Keywords: digital watermarking, wave-atoms, security.

1 Introduction

The widespread use of digital recording, storage devices, Internet, and the promise of higher bandwidth for both wired and wireless networks has made it possible to create, replicate, transmit, and distribute digital content in an effortless way [1]. However, unrestricted copying and convenient digital media manipulation cause considerable financial loss and show up an issue of intellectual property rights (IPR) [1, 2]. Due to the concern about protection of digital content, many digital data hiding techniques are developed including digital watermarking [3]. Digital watermarking is a technology that embeds information within the content of a digital media file [4]. By extracting this secret digital data, it can protect the copyright of digital media and provide authentication to digital media.

A digital watermark should have two main properties, which are robustness and imperceptibility. Robustness means that the watermarked data can withstand different image processing attacks and imperceptibility means that the watermark should not introduce any perceptible artifacts [1].

In 1999, Candes and Donoho [5] introduced a new multiscale transform called the Curvelet transform. The transform can represent edges and other singularities along curves much more efficiently than traditional transforms [6]. Several watermarking schemes based on curvelet domain were also proposed recently [7, 8].

H.-J. Kim, Y. Shi, and M. Barni (Eds.): IWDW 2010, LNCS 6526, pp. 148–158, 2011.

In 2006, Demanet [9] introduced a generalization of curvelets, named wave atoms. It can be used to effectively represent warped oscillatory functions [10]. Oriented textures have a significantly sparser expansion in wave atoms than in other fixed standard representations like Gabor filters, wavelets, and curvelets. Many existing applications of wave atom transform show its great potential for image de-noising [11, 12]. However, there are no researches to explore the applications of wave atom transform to the field of digital watermarking until now. It would be very interesting to find out whether wave atom transform is suitable for digital watermarking. Since sensitivity of human eye to noise in textured area is less and it is more near the edges according to the HVS characteristics [13], little modification of textures area are usually imperceptible by human eyes. And the wave atom can provide significantly sparser expansion for the oscillatory functions or oriented textures [10]. Thus, modifying significant wave atom coefficients may result in little image quality degradation.

According to whether the original image is needed or not during the detection, watermarking methods can be sorted as non-blind, semi-blind or blind [14]. Non-blind technique requires the original image; semi-blind technique only requires the watermark; blind technique requires neither the original image nor the watermark. In this paper, we proposed a blind watermarking method which makes use of the Multiple Description Coding (MDC) idea [15]. The detail of MDC is described in section 3. And the robustness tests for the proposed method are also described.

This paper is organized as follows: In Section 2, Wave atom Transform is presented. Multiple Description Coding (MDC) is described in section 3. The detail of embedding and extracting approaches are given in section 4. The experimental results are described in section 5. Finally, section 6 provides the conclusion.

2 Wave Atom Transform

Demanet and Ying [9] introduced wave atoms, that can be seen as a variant of 2-D wavelet packets and obey the parabolic scaling law, i.e. wavelength $\approx$ $(diameter)^2$. They prove that oscillatory functions or oriented textures (e.g., fingerprint, seismic profile, engineering surfaces) have a significantly sparser expansion in wave atoms than in other fixed standard representations like Gabor filters, wavelets, and curvelets.

Wave atoms have the ability to adapt to arbitrary local directions of a pattern, and to sparsely represent anisotropic patterns aligned with the axes. The elements of a frame of wave packets $\{\phi_u(x)\}, x \in \mathbb{R}^2$, are called Wave Atoms (WAs) when there is a constant C_M such that

$$|\hat{\phi_u}| \leq C_M 2^{-j}(1+2^{-j}|\omega-\omega_u|)^{-M} + C_M 2^{-j}(1+2^{-j}|\omega+\omega_u|)^{-M}. \quad (1)$$

And $|\phi_u| \leq C_M 2^j (1+2^j|x-x_u|)^{-M}$, with M=1, 2, $\cdots$. The hat denotes Fourier transformation and the subscript $u=(j,m_1,m_2,n_1,n_2)$ of integer-valued quantities index a point (x_u, w_u) in phase space as

$$x_u = (x_1, x_2)_\mu = 2^{-j}(n_1, n_2), \omega_u = (\omega_1, \omega_2)_\mu = \pi 2^j (m_1, m_2). \quad (2)$$

where $C_A 2^j \leq max_{k=1,2} |m_k| \leq C_B 2^j$, with C_A and C_B positive constants whose values depend on the numerical implementation. Hence, the position vector x_μ is the center of $\phi_u(x)$, and the wave vector ω_u denotes the centers of both bumps of $\hat{\phi_u}$

The parabolic scaling is encoded in the localization conditions as follows: at scale 2^{-2j}, the essential frequency support is of size $\approx 2^{-j}$. The subscript j denotes different dyadic coronae and the subscripts (m_1, m_2) label the different wave number ω_u within each dyadic corona.

In fact, WAs are constructed from tensor products of 1D wavelet packets. The family of real-valued 1D wave packets is described by $\psi^j_{m_1,n_1}(x_1)$ functions, where $j \geq 0, m_1 \geq 0$, and $\psi^j_{m_1,n_1}(x_1) = 2^{\frac{j}{2}} \psi^0_{m_1}(2^j x_1 - n_1)$ with

$$\hat{\psi}^0_{m_1}(w_1) = e^{-i\omega/2} \left\{ e^{-i\alpha_{m_1}} g\left[\epsilon_{m1}\left(\omega_1 - \pi m_1 - \pi/2\right)\right] + e^{-i\alpha_{m_1}} g\left[\epsilon_{m1+1}\left(\omega_1 + \pi m_1 + \pi/2\right)\right] \right\}. \tag{3}$$

where $\epsilon_{m1} = (-1)^{m_1}$, $\alpha_{m1} = (2m_1 + 1)\pi/4$. The function g is real valued and is selected to obtain an orthonormal basis $\{\psi_{m_1}(t - n_1)\}$ of $L^2(R)$. The 2D extension is formed by the products

$$\phi^+_u(x_1, x_2) = \psi^j_{m_1}\left(x_1 - 2^{-j} n_1\right) \psi^j_{m_2}\left(x_2 - 2^{-j} n_2\right). \tag{4}$$

$$\phi^-_u(x_1, x_2) = H\psi^j_{m_1}\left(x_1 - 2^{-j} n_1\right) H\psi^j_{m_2}\left(x_2 - 2^{-j} n_2\right). \tag{5}$$

where H is the Hilbert transform and $u = (j, m_1, m_2, n_1, n_2)$. The recombinations $\phi^{(1)}_u = (\phi^+_u + \phi^-_u)/2$ and $\phi^{(2)}_u = (\phi^+_u - \phi^-_u)/2$ form the WA frame.

3 Multiple Description Coding

The concept of Multiple Description Coding can be adapted to digital image watermarking as suggested by Chandramouli et al. [15]. It has been applied into DCT and Contourlet Transform domain [15, 16]. The idea behind using multiple descriptions of a source is to partition the source information into various descriptions such that by using one or more of these source descriptions a receiver will be able to reconstruct the original source within some prescribed distortion constraints [15]. For example, a host image is decomposed into two descriptions, i.e. odd and even pixel intensities (two descriptions) of host image. These descriptions are chosen in such a way that some correlation exists between them. One description can be used for watermark insertion and the other description can be used as a reference for watermark extraction. After watermark embedding, two descriptions are combined to get the watermarked image.

4 Proposed Method

In [15, 16], Chandramouli and Mohan proposed blind watermarking schemes using multiple description coding (MDC). By using MDC, the watermark can be

extracted by comparing to several sub-images instead of an original image. The Multiple Description Coding is therefore incorporated in our proposed method. Suppose that I denotes the host image of size N × N. The host image is decomposed into four descriptions as follows:

$$\begin{aligned} I_1(p,q) &= I(p,2q-1), \qquad I_2(p,q) = I(p,2q), \\ I_3(p,q) &= I\left(\tfrac{N}{2}+p,2q-1\right), I_4(p,q) = I\left(\tfrac{N}{2}+p,2q\right). \end{aligned} \tag{6}$$

where $p = 1,2,...,N/2,\ q = 1,2,...,N/2$. I_1 ,I_2, I_3 *and* I_4 denote four descriptions, namely upper odd description, upper even description, lower odd description, lower even description.

From equation 6, we can see that descriptions I_1 and I_3 are very similar to I_2 and I_4. After applying the wave-atom transform to four descriptions, we assume that the wave-atom coefficients between the odd and even descriptions are approximately equal or similar. To embed the watermark in the wave-atom domain, the means of some wave-atom wedges are changed by coefficient modification. In extracting the watermark, the means of wave-atom wedges corresponding to the odd and even descriptions are compared. The details of our proposed method are shown below:

4.1 The Embedding Procedure

The proposed watermark embedding scheme is shown in Fig. 1. The embedding process is described as follows:

1. Divide the original image I of size N × N to form four descriptions, I_1, I_2, I_3 and I_4, by MDC using equation 6.

2. Wave-atom Transform is then applied to the four descriptions. Four coefficient sets, S_1, S_2, S_3 and S_4 are obtained with respect to the upper odd description, upper even description, lower odd description, lower even description. Hence, these descriptions are decomposed into five bands in our case. The fourth scale band is selected to embed the watermark w with length m bits.

3. From the sets S_1, S_2, S_3 and S_4, select the coefficients C_u whose absolute values smaller than r to modify, where $u = (j, m_1, m_2, n_1, n_2)$ of integer-valued quantities index a point (x_u, ω_u) in phase space.

4. S_1 and S_2 are used to embed the first half of watermark bits w containing m/2 bits, while S_3 and S_4 are used to embed the second half of the watermark w. One wave-atom wedge from the odd description and one wedge from the even description are both used to embed one bit. They are modified as follows:

For all non-empty wedges in S_1 and S_2,
    IF w_i =1
        In S_1 ,
            IF abs(C_u)≥ δ
                C_u= C_u × α_a
            ELSE
                C_u= C_u × α_b
        In S_2 ,

$$
\begin{aligned}
&\text{IF abs}(C_u) \geq \delta \\
&\quad C_u = C_u \times \alpha_c \\
&\text{ELSE} \\
&\quad C_u = C_u \times \alpha_d \\
&\text{ELSE} \\
&\quad \text{In } S_2 \text{ ,} \\
&\quad\quad \text{IF abs}(C_u) \geq \delta \\
&\quad\quad\quad C_u = C_u \times \alpha_a \\
&\quad\quad \text{ELSE} \\
&\quad\quad\quad C_u = C_u \times \alpha_b \\
&\quad \text{In } S_1 \text{ ,} \\
&\quad\quad \text{IF abs}(C_u) \geq \delta \\
&\quad\quad\quad C_u = C_u \times \alpha_c \\
&\quad\quad \text{ELSE} \\
&\quad\quad\quad C_u = C_u \times \alpha_d \text{ .}
\end{aligned} \tag{7}
$$

where $i = 1, 2, m/2$, $abs(.)$ is the absolute value of $(.)$, and $u = (j, m_1, m_2, n_1, n_2)$ of integer-valued quantities, and α_a , α_b, α_c and α_d are strength factors which can be used to control robustness and perceptual quality, and δ is the embedding threshold.

Similarly, coefficients within S_3 and S_4 are modified as same as those within S_1 and S_2. For S_3 and S_4 case, S_1 and S_2 are replaced by S_3 and S_4 respectively in equation 7. Thus, we get the altered wave-atom coefficients sets $S_1^{'}$, $S_2^{'}$, $S_3^{'}$ and $S_4^{'}$.

5. Apply the inverse wave-atom transform to the modified coefficients sets $S_1^{'}$, $S_2^{'}$, $S_3^{'}$ and $S_4^{'}$.

6. Obtain the output watermarked image $I^{'}$ by collecting 4 modified descriptions.

4.2 The Extracting Procedure

In extracting the watermarks, the original image I is not required in our algorithm. The proposed watermark extraction scheme is shown in Fig. 3. Suppose $I^{'}$ is the watermarked image for watermark detection. The extracting process is described as follows:

1. Divide $I^{'}$ to four descriptions, $I_1^{'}$, $I_2^{'}$, $I_3^{'}$ *and* $I_4^{'}$, by MDC using equation 6.

2. Wave-atom Transform is then applied to these descriptions $I_1^{'}$, $I_2^{'}$, $I_3^{'}$ *and* $I_4^{'}$ to obtain four coefficients sets, $S_1^{'}$, $S_2^{'}$, $S_3^{'}$ *and* $S_4^{'}$.

3. The extraction band is as same as the embedding band so as to extract correct watermark bits. Through comparing the mean of the wave atom wedges between the odd and even descriptions, we extract the watermark sequence w. First, we denote the wedge of $S_1^{'}$ (odd description) and $S_2^{'}$ (even descriptions) be $W_1(j, m, n)$ and $W_2(j, m, n)$ respectively. It is described as follows:

For all non-empty wedges in $S_1^{'}$ and $S_2^{'}$,

$$
w_m = \begin{cases} 1, if\ mean(abs(W_1(j,m,n))) > mean(abs(W_2(j,m,n))) \\ 0, if\ mean(abs(W_1(j,m,n))) \leq mean(abs(W_2(j,m,n))) \end{cases} \tag{8}
$$

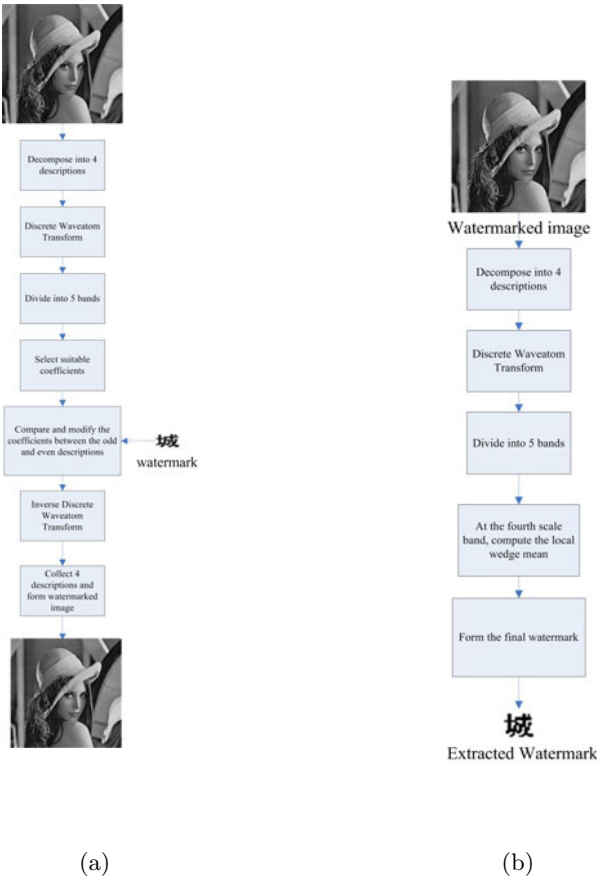


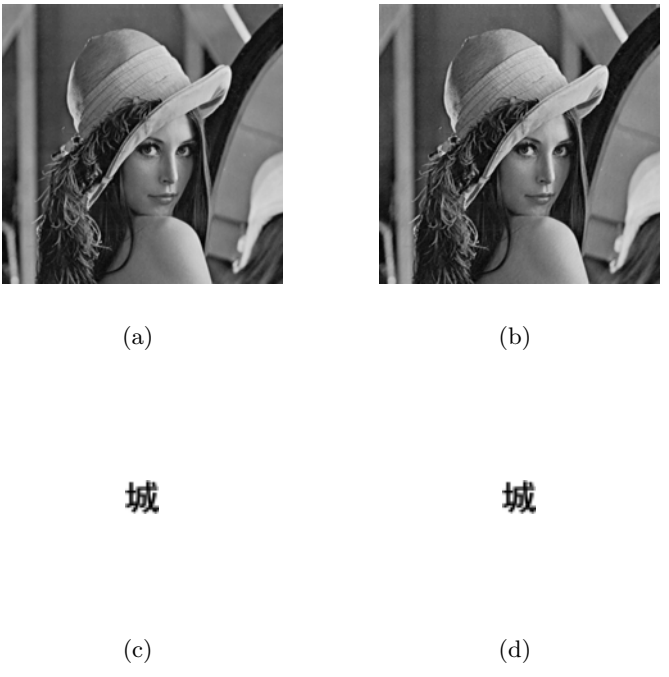
Fig. 1. (a)The Embedding Procedure, (b)The Extracting Procedure

where j is the scale, m, n represent the phase, and mean(.) represents the mean value of (.) and abs(.) is the absolute values of (.)

According to the equation 8, the first half of the watermark is extracted. Similarly, second half of the watermark is extracted by replacing $S_1^{'}$ and $S_2^{'}$ with $S_3^{'}$ and $S_4^{'}$ respectively in the equation 8. Finally, the watermark wm can be reconstructed by the merge of two half watermarks.

5 Experimental Results

In order to test the robustness of the proposed watermarking scheme, we used the 512 × 512 gray-scale image, Lena, shown in Fig. 3(a) as the test image. The watermarked image is illustrated in Fig. 3(b).The binary watermark is shown in Fig. 3(c), whose size is 16 × 16.The extracted watermark is shown in Fig. 3(d) with NC value = 1 which shows the correct watermark extraction.



(a) (b)

(c) (d)

Fig. 2. (a) Lena Image, (b) Watermarked Lena Image, (c) binary watermark, (d) Extracted watermark with NC=1

In our experiment, the threshold r is 60 and δ is 19. The embedding strength factors α_a, α_b , α_c *and* α_d are 1.6, 2.1, 0.7 and 0.4 respectively. The mean squared error (MSE) between the original and watermarked images is defined by

$$MSE = \frac{1}{M \times N} \sum_{p=1}^{M} \sum_{q=1}^{N} \left(I\left(p,q\right) - I^{'}\left(p,q\right)\right)^2 \tag{9}$$

where $I(p,q)$ and $I^{'}(p,q)$ denote the pixel value at position (p,q) of the original image I and the watermarked image $I^{'}$ with size of $M \times N$ pixels respectively.

Hence, the watermarked image quality is represented by the peak signal to noise ratio (PSNR) between I and $I^{'}$, is calculated by

$$PSNR = 10 \log_{10} \left(\frac{255^2}{MSE} \right) \left(dB\right) \tag{10}$$

To evaluate the robustness of the algorithm, the normalized cross-correlation (NC) is employed. The NC between the embedded watermark, *W(p,q)*, and the extracted watermark $W^{'}(p,q)$ is defined by

$$NC = \frac{\sum_{p=1}^{M_W} \sum_{q=1}^{N_W} [W(p,q) \bullet W'(p,q)]}{\sum_{p=1}^{M_W} \sum_{q=1}^{N_W} [W(p,q)]^2}. \quad (11)$$

where M_W and N_W denote the width and height of the watermark respectively.

Fig. 3b shows the watermarked image with the PSNR of 36.10 dB.

5.1 Robustness Tests

Several common signal processing attacks are applied to verify the robustness of the proposed scheme including Guassian low pass filtering, Gaussian additive noise, Laplacian image enhancement, jpeg compression and Salt and Pepper noises, etc. The simulation results are shown as table 1. We tested the proposed scheme on gray level images with size 512×512 pixels including "Baboon", "Lena", "Pepper" and "Boat". Table 1 shows the experimental results about PSNR of the watermarked images and the NC values between the embedded watermark and the extracted watermark with different attacks.

Table 1. Experimental results for four images ("Boat", "Lena", "Pepper" and "Baboon")

Attacks	**Boat**		**Lena**		**Pepper**		**Baboon**	
	PSNR (dB)	NC	PSNR (dB)	NC	PSNR (dB)	NC	PSNR (dB)	NC
JPEG 80	34.4	0.996	33.58	1	33.28	1	29.27	1
JPEG 50	34.12	0.920	33.08	0.933	33.23	0.932	27.00	0.996
JPEG 30	33.19	0.853	32.43	0.879	32.88	0.84	25.93	0.981
Gaussian low pass filter (Standard Variance = 0.5, Window Size =3)	34.4	0.996	33.58	1	33.28	1	29.27	1
Gaussian noise addition (Standard Variance of Gaussian Noises =20	22.04	0.940	22.11	0.959	22.06	0.982	21.71	1
Salt and Pepper noises addition (noise density =0.1)	15.40	0.774	15.09	0.771	15.28	0.781	15.50	0.989
Laplacian Sharperning	19.49	1	19.58	1	19.25	1	13.83	1
Brighter (20%)	22.82	1	21.43	1	22.47	1	23.06	1
Darker (20%)	19.60	1	21.05	1	19.69	1	19.35	1
Histogram Equalization	16.81	1	16.72	1	20.49	1	17.35	1
Median Filtering (3 × 3)	33.59	0.231	33.12	0.301	34.34	0.258	23.70	0.199

From table 1, it can be observed that the proposed scheme is quite robust to common signal processing attacks. But it fails to withstand the median filtering as shown in table1, where the NC values are only about 0.2 to 0.3.

Table 2. The comparison results of four different schemes

Attacks	Xiao et al.[17]	Leung et al.[7]	Tao et al.[18]	Proposed Method
JPEG 80	0.9553	1	0.9704	1
JPEG 60	0.9093	1	0.9245	0.933
JPEG 30	0.8074	1	0.8682	0.8792
Gaussian low pass filter (Standard Variance = 0.5, Window Size =3)	0.9889	1	0.9697	1
Gaussian noise addition (Standard Variance of Gaussian Noises =20	0.9403	0.9315	0.7405	0.959
Salt and Pepper noises addition (noise density =0.1)	0.7981	0.7648	0.9035	0.771
Laplacian Sharperning	0.9963	0.9963	0.7967	1
Cropping (20%) (half of the image)	0.8095	869	0.6758	0.8542
Histogram Equalization	0.9742	1	0.8877	1
Median Filtering (3×3)	0.9742	1	0.9232	0.301

5.2 Comparisons with Related Watermarking Methods

Besides, we compare the performance of proposed scheme with other watermarking schemes which are proposed by Xiao et al. [17], Leung et al. [7] and Tao et al. [18]. They both embed the watermark in the frequency domain which are Curvelet domain and Wavelet domain. Table 2 shows the performance of these watermarking schemes in term of the normalized cross-correlation values. From table 2, it can be seen that the proposed scheme outperforms Xiao's and Tao's scheme and is comparable to Leung's scheme. Table 3 illustrates the processing time comparison between the proposed scheme and other two schemes. It shows that the processing time of proposed scheme is the shortest one among these four schemes, which are 2.41s and 0.92s for embedding and extracting respectively.

Table 3. The processing time for watermark embedding and retrieval

Processing time	Watermark Embedding (second)	Watermark Retrieval (second)
Xiao et al.[17]	6.22	2.31
Leung et al.[7]	6.41	5.37
Tao et al.[18]	0.9	9.45
Proposed scheme	2.43	0.92

6 Conclusion

In this paper, we have proposed a robust watermarking scheme based on the Wave-atom transform is presented. The watermark is embedded in the Wave-atom domain of four descriptions which are obtained by MDC. The watermark extraction proposed is simple and does not need the original image. The main idea of our proposed method is to adjust the energy of the wave-atom wedges for different sub-images according to the watermark. By comparison of the energy difference between wave-atom wedges pairs, the embedded watermark can be extracted. The proposed method is highly robust and can withstand common signal processing. Besides, the quality of the watermarked image is good in terms of perceptibility and PSNR (over 35 dB).

References

1. Podilchuk, C.I., Delp, E.J.: Digital watermarking: algorithms and applications. IEEE Signal Process. Mag., 33–46 (2001)
2. Cox, I.J., Miller, M.L., Bloom, J.A.: Digital Watermarking. Morgan Kaufmann, San Francisco (2002)
3. Langelaar, G.C., Setyawan, I., Lagendijk, R.L.: Watermarking digital image and video data: a state-of-the-art overview. IEEE Signal Process. Mag., 20–46 (2000)
4. Cox, I.J., Kilian, J., Leighton, F.T., Shamoon, T.: Secure spread spectrum watermarking for multimedia. IEEE Trans. Image Process. 6(12), 1673–1687 (1997)
5. Candes, E.J., Donoho, D.L.: New tight frames of curvelets and optimal repesentations of objects with C2 singularities. Communications on Pure and Applied Mathematics 57(2), 219–266 (2004)
6. Candes, E.J., Donoho, D.L.: Fast Discrete Curvelet Transform. Applied and Computational Mathmatics, pp. 1–43. California Institute of Technology (2005)
7. Leung, H.Y., Cheng, L.M., Cheng, L.L.: A Robust Watermarking Scheme using Selective Curvelet Coefficients. International Journal of Wavelets, Multiresolution and Information Processing (IJWMIP) 7(2), 163–181 (2009)
8. Tao, P., Dexterb, S., Eskicioglu, A.M.: Robust Digital Image Watermarking in Curvelet Domain. In: Proceedings of the SPIE, vol. 6819, pp. 68191B–68191B-12(2008)
9. Demanet, L.: Curvelets, wave atoms, and wave equations, Ph.D. Thesis, Caltech, http://math.stanford.edu/~laurent/papers/ThesisDemanet.pdf
10. Demanet, L., Ying, L.: Wave atoms and sparsity of oscillatory patterns. Appl. Comput. Harmon. Anal. 23, 368–387 (2007)
11. Rajeesh, J., Moni, R.S., Palanikumar, S.: Noise Reduction in Magnetic Resonance Images using Wave Atom Shrinkage. The International Journal of Image Processing (IJIP) 4(2), 131–141 (2010)
12. Federico, A., Kaufmann, G.H.: Denoising in digital speckle pattern interferometry using wave atoms. Opt. Lett. 32, 1232–1234 (2007)
13. Lewis, A.S., Knowles, G.: Image compression using the 2-D wavelet transform. IEEE Transactions on Image Processing 1(2), 244–250 (1992)
14. Katzenbeisser, S., Petitcolas, F.A.P.: Information Hiding Techniques for Steganography and Digital Watermarking. Artech House, Boston (2000)

15. Chandramouli, R., Graubard, B.M., Richmond, C.R.: A multiple description framework for oblivious watermarking. In: Proc. SPIE Security and Watermarking for Multimedia Contents 2001, San Jose, USA (2001)
16. Mohan, B.C., Kumar, S.S.: Robust Digital Watermarking Scheme using Contourlet Transform. IJCSNS International Journal of Computer Science and Network Security 8, 43–51 (2008)
17. Xiao, Y., Cheng, L.M., Cheng, L.L.: A Robust Image Watermarking Scheme Based on A Novel HVS Model in Curvelet Domain. In: IIHMSP 2008, Harbin, China, August 2008, pp. 343–346 (2008)
18. Tao, P., Eskicioglu, A.M.: A Robust Multiple Watermarking Scheme in the Discrete Wavelet Transform Domain. In: Optics East 2004 Symposium, Internet Multimedia Management Systems V Conference, Philadelphia, PA (2004)

Robust Watermarking of H.264/SVC-Encoded Video: Quality and Resolution Scalability

Peter Meerwald* and Andreas Uhl

Dept. of Computer Sciences, University of Salzburg,
Jakob-Haring-Str. 2, A-5020 Salzburg, Austria
{pmeerw,uhl}@cosy.sbg.ac.at
http://www.wavelab.at

Abstract. In this paper we investigate robust watermarking integrated with H.264/SVC video coding and address coarse-grain quality and spatial resolution scalability features according to Annex G of the H.264 standard. We show that watermark embedding in the base layer of the video is insufficient to protect the decoded video content when enhancements layers are employed. The problem is mitigated by a propagation technique of the base layer watermark signal when encoding the enhancement layer. In case of spatial resolution scalability, the base layer watermark signal is upsampled to match the resolution of the enhancement layer data. We demonstrate blind watermark detection in the full- and low-resolution decoded video for the same adapted H.264/SVC bitstream and, surprisingly, can report bit rate savings when extending the base layer watermark to the enhancement layer.

Keywords: Watermarking, scalable video coding.

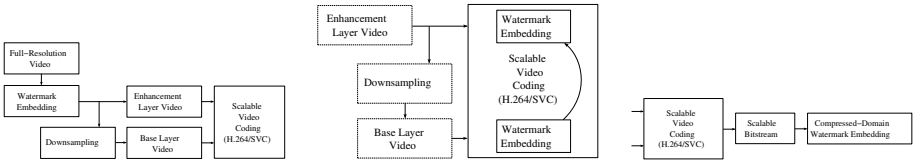
1 Introduction

Distribution of video content has become ubiquitous and targets small, low-power mobile to high fidelity digital television devices. The Scalable Video Coding (SVC) extension of the H.264/MPEG-4 Advanced Video Coding standard describes a bit stream format which can efficiently encode video in multiple spatial and temporal resolutions at different quality levels [14,15]. Scalability features have already been present in previous MPEG video coding standards. They came, however, at a significant reduction in coding efficiency and increased coding complexity compared to non-scalable coding. H.264/SVC employs inter-layer prediction and can perform within 10% bit rate overhead for a two-layer resolution scalable bitstream compared to coding a single layer with H.264.

In this work we investigate a well-known robust watermarking framework proposed by Noorkami et al. [10,11] for copyright protection and ownership verification applications of H.264-encoded video content. The aim is to provide a single scalable, watermarked bit stream which can be distributed to diverse

* Supported by Austrian Science Fund (FWF) project P19159-N13.

H.-J. Kim, Y. Shi, and M. Barni (Eds.): IWDW 2010, LNCS 6526, pp. 159–169, 2011.



(a) Watermark embedding before video encoding

(b) Integrated watermark embedding and coding

(c) Compressed-domain embedding after encoding

Fig. 1. Different embedding scenarios for watermarking resolution-scalable H.264/SVC video content

clients without the need to re-encode the video material. Scalability is provided at the bit stream level. A bit stream with reduced quality, spatial and/or temporal resolution can be efficiently obtained by discarding NAL units [14]. The watermark (i) should be detectable in the compressed domain *and* the decoded video without reference to the original content, and (ii) must be detectable in the decoded video at all scalability operation points, starting from the base layer.

In Fig. 1 we distinguish three embedding scenarios for producing a watermarked, scalable H.264/SVC bitstream: (a) embedding before encoding, (b) embedding integrated in the coding process, (c) altering the scalable bit stream (embedding in the compressed domain). The first embedding scenario offers little control over the resulting bitstream and thus makes detection in the compressed domain difficult. As watermark embedding takes place before video encoding, any robust video watermarking schemes can be applied. However, lossy compression and downsampling of the full-resolution video have an impact on the embedded watermark signal. Caenegem et al. [2] describe the design of a watermarking scheme resilient to H.264/SVC but treat the encoding only from a robustness point of view. The third scenario appears to be overly complex from an implementation point of view given the inter-layer prediction structure of H.264/SVC which necessitates drift compensation to minimize error propagation [10,4]. Zou et al. [21,20] propose a bitstream replacement watermark by altering H.264 CAVLC and CABAC symbols of HDTV video content several minutes long; scalability features are not addressed.

Integrated H.264/SVC video encoding and watermarking offers control over the bitstream; for example the watermark can be placed exclusively in non-zero quantized residual coefficients [11]. A combined encryption and watermarking-based authentication method for H.264/SVC encoding is proposed by Park and Shin [12]. Authentication information is encoded in the bits signalling the intra prediction mode, but cannot be verified on the decoded video. Many proposals for H.264 integrated watermarking have been put forward using spread-spectrum or replacement techniques for authentication and copyright protection (e.g. [13,19,16,8]), however, watermarking of a scalable bitstream and the bitrate overhead is not considered.

The present work is an extension of [9]. A robust watermark is embedded in intra-coded frames during H.264/SVC encoding and detectable in the bitstream and decoded frames. In Section 2 we briefly review the H.264 watermarking framework [10] and investigate its applicability for protecting resolution-scalable video encoded with H.264/SVC. We propose a propagation step of the base-layer watermark signal in Section 3 in order to extend the framework to H.264/SVC, including resolution and quality scalability. Experimental results are provided in Section 4 followed by discussion and concluding remarks in Section 5.

2 Watermarking of H.264-Encoded Video

Several strategies have been proposed for embedding a watermark in H.264-encoded video. Most commonly, the watermark signal is placed in the quantized AC coefficients of intra-coded macroblocks. Noorkami et al. [10] present a framework where the Watson perceptual model for 8×8 DCT coefficients blocks [18] is adapted for the 4×4 integer approximation to the DCT which is predominantly used in H.264. Other embedding approaches include the modification of motion vectors or quantization of the DC term of each DCT block [3], however, the watermark can not be detected in the decoded video sequence or the scheme has to deal with prediction error drift.

Figure 2 illustrates the structure of the watermarking framework integrated in the H.264 encoder; each macroblock of the input frame is coded using either intra- or inter-frame prediction and the difference between input pixels and

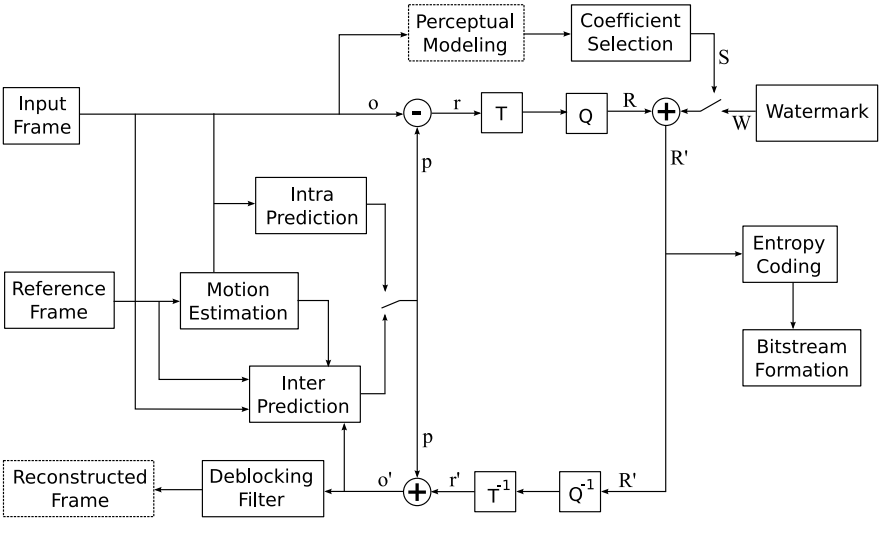


Fig. 2. Watermark embedding in quantized 4×4 DCT residual blocks

prediction signal is the residual[1]. We denote by $r_{i,j,k}$ the coefficients of 4×4 residual block k with $0 \leq i, j < 4$ and similarly by $o_{i,j,k}$ and $p_{i,j,k}$ the values of the original pixels and the prediction signal, respectively. Each block is transformed and quantized, T denotes the DCT and Q the quantization operation in the figure. Let $R_{i,j,k}$ represent the corresponding quantized DCT coefficients obtained by $R_k = \mathsf{Q}(\mathsf{T}(r_k))$. $R_{0,0,k}$ thus denotes the quantized DC coefficient of block k. After watermark embedding, described in the following paragraphs, and entropy coding, the residual information is written to the output bitstream.

For each block, a bipolar, pseudo-random watermark $W_{i,j,k} \in \{-1, 1\}$ with equiprobable symbols is generated and added to the residual block to construct the watermark block R',

$$R'_{i,j,k} = R_{i,j,k} + S_{i,j,k} \cdot W_{i,j,k}, \tag{1}$$

where $S_{i,j,k} \in \{0, 1\}$ selects the embedding locations for block k. The design of S determines the properties of the watermarking scheme and differentiates between various approaches: in [10], embedding locations are selected based on the masked error visibility thresholds derived from the Watson perceptual model. Further, the number of locations is constrained to avoid error pooling and AC coefficients of large magnitude are preferred in the selection process.

The pixels of the reconstructed, watermarked video frame are given by $o'_{i,j,k} = p_{i,j,k} + r'_{i,j,k}$ where $r'_k = \mathsf{T}^{-1}(\mathsf{Q}^{-1}(R'_k)) = \mathsf{T}^{-1}(\mathsf{Q}^{-1}(R_k) + Q_k \cdot S_k \cdot W_k)$. For simplicity, we have dropped the coefficient indices i, j.

Watermark detection is performed *blind*, i.e. without reference to the original host signal, and can be formulated as a hypothesis test to decide between

$$\begin{aligned} \mathcal{H}_0 &: Y_l = O_l \text{ (no/other watermark)} \\ \mathcal{H}_1 &: Y_l = O_l + Q_l \cdot W_l \text{ (watermarked)} \end{aligned} \tag{2}$$

where O_l denotes the selected 4×4 DCT coefficients of the received video frames, Q_l the corresponding quantization step size and W_l the elements of the watermark sequence; l indicates the l^{th} selected coefficient or watermark bit to simplify notation. We adhere to the location-aware detection (LAD) scenario [11] where the embedding positions are known to the detector. For efficient blind watermark detection, accurate modeling of the host signal is required. We assume a Cauchy distribution of the DCT coefficients [1] and chose the Rao-Cauchy (RC) detector [6] whose detection statistic for the received signal Y_l of length L and the test against a detection threshold T are given by

$$\rho(Y_l) = \frac{8\hat{\gamma}^2}{L} \left[\sum_{l=1}^{L} \frac{Y_l \cdot W_l}{\hat{\gamma}^2 + Y_l^2} \right]^2 \quad \text{and} \quad \rho(Y_l) \gtrless_{\mathcal{H}_0}^{\mathcal{H}_1} T. \tag{3}$$

$\hat{\gamma}$ is an estimate of the Cauchy PDF shape parameter which can be computed using fast, approximate methods [17]. According to [5], $\rho(Y_l)$ follows a χ_1^2

[1] Other modes are possible, e.g. *PCM* or *skip* mode, but rarely occur or are not applicable for embedding an imperceptible watermark due to lack of texture.

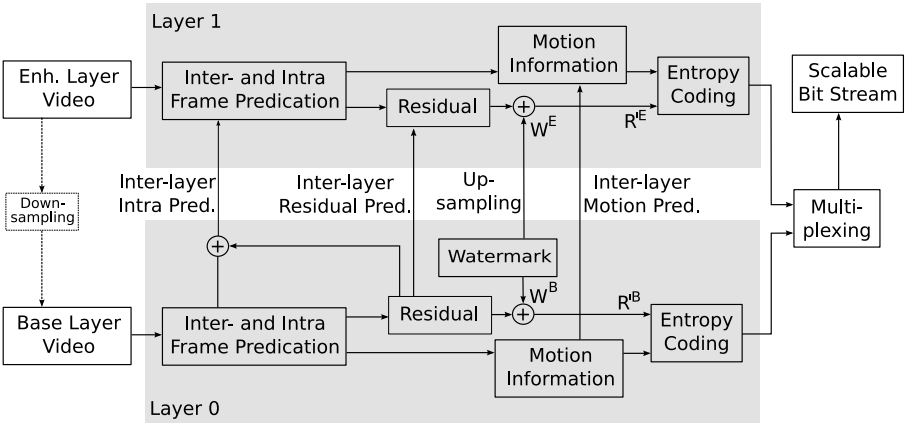


Fig. 3. Simplified H.264/SVC encoding and watermarking structure for two spatial resolution layers

distribution with one degree of freedom under $\mathcal{H}_0$ and we can write the probability of false-alarm $P_f = \mathbb{P}(\rho(Y_l) > T|\mathcal{H}_0)$ as

$$P_f = 2\,\mathrm{Q}\left(\sqrt{T}\right) \quad \text{and express} \quad T = \left[\mathrm{Q}^{-1}\left(\frac{P_f}{2}\right)\right]^2 \tag{4}$$

where $Q(\cdot)$ denotes the Q-function of the Normal distribution. Note that no parameters need to be estimated to establish the detection threshold. The Rao-Cauchy test is a constant false-alarm rate detector [5] which simplifies the experimental setup. Under $\mathcal{H}_1$, the test statistic follows a non-central Chi-Square distribution $\chi^2_{1,\lambda}$ with one degree of freedom and non-centrality parameter λ. By estimating λ from experimental detection responses, the performance of the detector can be analyzed in terms of the probability of missing the watermark,

$$P_m = 1 - \mathbb{P}(\rho > T|\mathcal{H}_1) = 1 - \mathrm{Q}(\sqrt{T} - \sqrt{\lambda}) + \mathrm{Q}(\sqrt{T} + \sqrt{\lambda}). \tag{5}$$

3 Extension to H.264/SVC

H.264/SVC resorts to several coding tools in order to predict enhancement layer data from the base layer representation [14] and exploit the statistical dependencies: (a) inter-layer intra prediction can adaptively use the (upsampled) reconstructed reference signal of intra-coded macroblocks, (b) macroblock partitioning and motion information of the base layer is carried over via inter-layer motion prediction for inter-coded macroblocks, and (c) inter-layer residual prediction allows to reduce the residual energy of inter-coded macroblocks in the enhancement layer by subtracting the (upsampled) transform domain residual coefficients of the colocated reference block. See Fig. 3 for an illustration.

In this work we focus on watermark embedding in intra-coded macroblocks of an H.264-coded base layer using the method reviewed in Section 2.

3.1 Resolution Scalability

In case a spatial enhancement layer with twice the resolution in each dimension is to be coded for SVC spatial scalability, the watermarked base-layer representation can be adaptively used for predicting the enhancement layer. In inter-layer intra prediction mode, the transform-domain enhancement layer residual of a 4×4 block k^E colocated with reference layer block k^B is given by

$$R'^E_{k^E} = \mathsf{Q}(\mathsf{T}(o^E_{k^E} - \mathsf{H}(o'^B_{k^B}))) \tag{6}$$

and the reconstructed, full-resolution video pixels are obtained by

$$o'^E_{k^E} = \mathsf{H}(o'^B_{k^B}) + \mathsf{T}^{-1}(\mathsf{Q}^{-1}(R'^E_{k^E})). \tag{7}$$

H denotes the normative H.264/SVC upsampling operation and superscripts ^{B} and ^{E} indicate base and spatial enhancement layer data, respectively. Apparently, the first right-hand term of Eq. (7) represents the upsampled, watermarked base-layer signal and the second term the quantized difference to the full-resolution, original video. Depending on the quantization parameter used to code the enhancement layer, the base-layer watermark can propagate to the decoded enhancement-layer video. Coarse quantization preserves a stronger watermark signal as illustrated in Figure 4 (a).

Watermarking only the base layer data is clearly not effective in protecting the full-resolution video. Not only does the watermark fade away, but also the bit rate for the enhancement layer increases, see Table 2, due to the added independent watermark signal which increased energy of the residual $R'^E_{k^E}$. To remedy these shortcomings, we propose to upsample the base layer watermark signal

$$W^E_{k^E} = \mathsf{Q}(\mathsf{T}(\mathsf{H}(\mathsf{T}^{-1}(Q_{k^B} \cdot S_{k^B} \cdot W^B_{k^B})))) \tag{8}$$

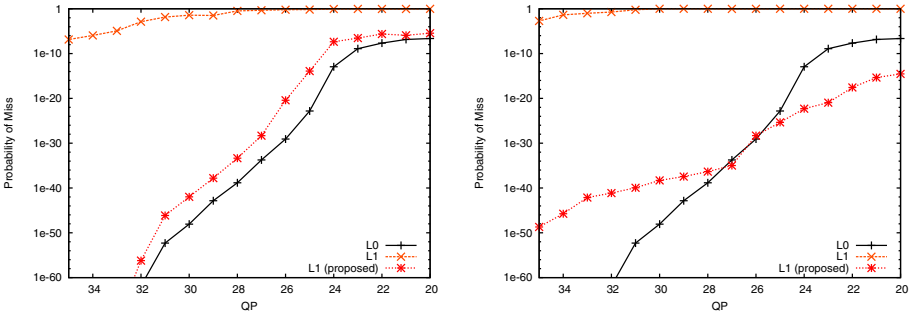
and add the resulting enhancement layer watermark $W^E_{k^E}$ to the residual blocks $R'^E_{k^E}$ to form *compensated* residual blocks

$$R''^E_{k^E} = R'^E_{k^E} + W^E_{k^E}. \tag{9}$$

Watermark detection is always performed with the base-layer watermark W, the full-resolution video is downsampled for detection.

3.2 Quality Scalability

In Fig. 4 (b) we plot the watermark transfer between two QCIF coarse-grain scalability (CGS) quality layers for a range of coding quantization parameters. The quality enhancement layer is coded using $QP - 3$ with respect to the base layer. It can be seen that the base layer watermark is effectively overshadowed by the enhancement layer video data coded with finer quantization. Simply adding the same watermark in the enhancement layer restores the watermark signal.



(a) Resolution scalability (b) Coarse grain quality scalability

Fig. 4. Transfer of the base-layer watermark to a (a) spatial resolution, and (b) coarse-gain quality enhancement layer for different quantization parameters (QP)

4 Experimental Results

Experiments have been performed using the Joint Scalable Video Model (JSVM) reference software version 9.19.9. Source code for the watermarking schemes investigated in this paper will become available at `http://www.wavelab.at/sources`. All experiments have been performed on widely-available test video sequences in CIF (352×288) and QCIF (176×144) resolution; QCIF sequences have been obtained by downsampling. The watermark is embedded in the base layer as described in Section 2 with an average target PSNR in the luminance channel of 40 dB between the original and the coded and watermarked video. We opt for always selecting the first 4×4 DCT AC coefficient in zig-zag order as the embedding location when it is non-zero; formally

$$S_{i,j,k} = \begin{cases} 1 & i = 0, j = 1 \wedge R_{0,1,k} \neq 0 \\ 0 & \text{otherwise} \end{cases} \quad \forall k.$$

The upsampled watermark signal is added to the quantized, transform-domain enhancement layer residuals as proposed in Section 3 with a target PSNR of 40 dB. The resulting watermarked, resolution-scalable bitstream can be decoded into QCIF and CIF video sequences. Watermark detection is performed on the decoded video.

Figure 4 shows the watermark detection performance for the *Foreman* sequence in terms of probability of miss (P_m) as a function of the H.264/SVC quantization parameter QP varying from 20 to 35. In the experiment, the false-alarm rate (P_f) is set to 10^{-3} and detection is performed on the first frame only; base layer and spatial resolution enhancement layer have been coded with the same QP (cf. Fig. 4a), the coarse grain quality enhancement layer (cf. Fig. 4b) with $QP-3$ relative to the base layer. The watermark can be reliably detected in the decoded base layer video (L0). Detection performance increases with coarser

Table 1. Detection results on base (L0) and resolution enhancement layer (L1)

Sequence	Probability of Miss ($P_f = 10^{-3}$)			
	L0	L1 (BL WM)	L1 (indep. WM)	L1 (proposed)
Foreman	$2.3 \cdot 10^{-25}$	0.81	0.0	$3.2 \cdot 10^{-17}$
Soccer	$2.6 \cdot 10^{-69}$	1.0	0.0	$1.1 \cdot 10^{-49}$
Bus	$1.0 \cdot 10^{-8}$	1.0	$2.5 \cdot 10^{-316}$	$6.2 \cdot 10^{-8}$
Container	$5.2 \cdot 10^{-119}$	0.44	0.0	$1.1 \cdot 10^{-91}$
Coastguard	$9.8 \cdot 10^{-133}$	0.68	0.0	$5.2 \cdot 10^{-97}$
Stefan	$8.5 \cdot 10^{-30}$	0.91	0.0	$3.2 \cdot 10^{-23}$

quantization as the watermark signal gets stronger relative to the host – remember that we added ± 1 to the quantized residual. We observe that the watermark embedded in the base layer is hardly detectable in the enhancement layer (L1). Only for coarse quantization ($QP \geq 28$) when no residual information is coded for most L1 blocks and solely the inter-layer intra prediction signal is available for reconstruction, detection becomes possible. However, using the upsampled base layer watermark, watermark detection performance in the enhancement layer is substantially improved (*L1 proposed*) and mostly restored to the level of the base layer watermark.

Table 1 provides the watermark detection results for six resolution-scalable H.264/SVC video sequences coded with $QP = 25$. The second column (*L0*) shows the probability of missing the watermark (P_m) for the decoded video in base layer QCIF resolution. When the watermark is embedded just in the base layer (column *L1 BL WM*), the watermark is not detectable using the decoded enhancement layer CIF resolution video since the base layer watermark does not propagate to the higher resolution layer. The fourth column (*L1 indep. WM*) lists the detection results for an independent watermark embedded in the enhancement layer. As the host signal is now four times larger, the probability of miss is drastically reduced. When the upsampled watermark signal is added to the enhancement layer residual (column *L1 proposed*) as presented in Section 3, the watermark can be reliably detected from the decoded CIF video sequence.

In Table 2 we examine the bit rate (in Kbit/s) of the resolution-scalable bitstream for the first 32 frames of six test sequences coded with $QP = 25$ and inter-layer prediction. Results have been averaged over 10 test runs with different watermarks. For reference, the second column (*L1 no WM*) lists the bit rates for coding the sequences without any watermark. The third column (*L1 BL WM*) contains the bit rate when watermarking the base layer only. We notice an increase of about 3% on average due to the added watermark signal. The fourth column (*L1 indep. WM*) lists the bit rate when independent watermarks are added to the base and enhancement layer; two independent watermarks in the two layer produces the highest bitrate. The rightmost column (*L1 proposed*) presents the results when adding the upsampled watermark to the enhancement

Table 2. Bit rate of the resolution enhancement layer (L1)

Sequence	Bit rate (Kbit/s)			
	L1 (no WM)	L1 (BL WM)	L1 (indep. WM)	L1 (proposed)
Foreman	883.1	939.5	1018.9	924.5
Soccer	1188.0	1239.1	1303.8	1227.0
Bus	1693.0	1732.0	1779.0	1721.0
Container	906.6	957.7	982.1	944.7
Coastguard	1506.6	1557.8	1572.6	1534.2
Stefan	1621.4	1657.0	1715.0	1651.0

Table 3. Bit rate of the coarse-grain quality layer (L1)

Sequence	Bit rate (Kbit/s)			
	L1 (no WM)	L1 (BL WM)	L1 (indep. WM)	L1 (proposed)
Foreman	287.4	330.9	342.9	320.2
Soccer	342.7	380.6	401.3	371.6
Bus	463.8	500.0	507.4	490.5
Container	258.6	307.8	315.8	298.2
Coastguard	359.5	396.0	404.8	387.0
Stefan	483.1	525.8	536.0	517.5

layer residual as proposed. Surprisingly, the bit rate can be reduced compared to the previous two columns and is lower than having no watermark in the decoded enhancement layer at all.

Table 3 lists the bit rates in Kbit/s for the coarse-grain quality (CGS) enhancement layer. The QCIF base layer is coded with $QP = 30$ and the enhancement layer of the same resolution with $QP = 24$. We can observe that watermarking the enhancement layer with the same watermark as the base layer (column *L1 proposed*) slightly reduces the bit rate over the case where the enhancement does not carry a watermark (column *L1 BL WM*) and only the base layer (BL) is watermarked, or – to a larger extent – when a different watermark (column *L1 indep. WM*) is embedded in the two quality scalability layers.

H.264/SVC also supports so-called medium-grain scalability (MGS) to enable quality adaptation without the need to code separate layers. MGS is realized by grouping the DCT coefficients in zig-zag order and allowing to discard the endmost coefficient groups. Since the watermark in this work is embedded in the first AC coefficient, MGS does not impair the watermark detection results unless all AC coefficients are discarded.

5 Discussion and Conclusion

In this work, we considered the application of a robust H.264-integrated watermarking method [10] in the context of H.264/SVC. A watermark embedded in the base layer data of a resolution-scalable bitstream is not detectable in the full-resolution decoded video sequence. We can resolve the issue by adding a compensation watermark signal to the enhancement layer residual. Note that the base layer watermark can be detected in the decoded video *and* the compressed domain, i.e. after entropy decoding. In contrast, the enhancement layer watermark can be either detected in the compressed domain residual data, *or* the decoded video due to inter-layer prediction of H.264/SVC. The aim of this work is to achieve the latter which seems more relevant for robust watermarking. Li et al. [7] discuss watermarking of a scalable audio bitstream and focus on the first case.

The 8×8 DCT which is more efficient for coding high-resolution frames can be permitted for coding the enhancement layer, only the base layer watermark is constrained to embedding in the prevalent 4×4 transform blocks since the watermark detector is blind and has no information on the H.264/SVC mode decisions. Upsampling the watermark cannot be easily extended to support several resolution enhancement layers as the watermark signal looses its high-pass characteristic; on the other hand, multi-layer H.264/SVC bitstreams have increasingly higher bit rate compared to non-scalable coding and are not likely to be adopted.

References

1. Altunbasak, Y., Kamaci, N.: An analysis of the DCT coefficient distribution with the H.264 video coder. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2004, vol. 3, pp. 177–180. IEEE, Montreal (2004)
2. van Caenegem, R., Dooms, A., Barbarien, J., Schelkens, P.: Design of an H.264/SVC resilient watermarking scheme. In: Proceedings of SPIE, Multimedia on Mobile Devices 2010, vol. 7542. SPIE, San Jose (2010)
3. Gong, X., Lu, H.M.: Towards fast and robust watermarking scheme for H.264 video. In: Proceedings of the IEEE International Symposium on Multimedia, ISM 2008, pp. 649–653. IEEE, Berkeley (2008)
4. Hartung, F., Girod, B.: Watermarking of uncompressed and compressed video. Signal Processing 66(3), 283–301 (1998)
5. Kay, S.M.: Fundamentals of Statistical Signal Processing: Detection Theory, vol, 2nd edn. Prentice-Hall, Englewood Cliffs (1998)
6. Kwitt, R., Meerwald, P., Uhl, A.: A lightweight Rao-Cauchy detector for additive watermarking in the DWT-domain. In: Proceedings of the ACM Multimedia and Security Workshop (MMSEC 2008), pp. 33–41. ACM, Oxford (2008)
7. Li, Z., Sun, Q., Lian, Y.: Design and analysis of a scalable watermarking scheme for the scalable audio coder. IEEE Transactions on Signal Processing 54(8), 3064–3077 (2006)

8. Lin, S., Chuang, C.Y., Meng, H.C.: A video watermarking in H.265/AVC encoder. In: Proceedings of the 5th International Conference on Intelligent Information Hiding and Multimedia Signal Processing, IIH-MSP 2009, Kyoto, Japan, pp. 340–343 (September 2009)
9. Meerwald, P., Uhl, A.: Robust watermarking of H.264-encoded video: Extension to SVC. In: Proceedings of the Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, IIH-MSP 2010, Darmstadt, Germany (October 2010) (accepted)
10. Noorkami, M., Mersereau, R.M.: A framework for robust watermarking of H.264 encoded video with controllable detection performance. IEEE Transactions on Information Forensics and Security 2(1), 14–23 (2007)
11. Noorkami, M., Mersereau, R.M.: Digital video watermarking in P-frames with controlled video bit-rate increase. IEEE Transactions on Information Forensics and Security 3(3), 441–455 (2008)
12. Park, S.W., Shin, S.U.: Combined Scheme of Encryption and Watermarking in H.264/Scalable Video Coding (SVC). SCI, pp. 351–361. Springer, Heidelberg (2008)
13. Qiu, G., Marziliano, P., Ho, A.T.S., He, D., Sun, Q.: A hybrid watermarking scheme for H.264/AVC video. In: Proceedings of the 17th International Conference on Pattern Recognition, ICPR 2004, pp. 865–868. IEEE, Cambridge (2004)
14. Schwarz, H., Wien, M.: The scalable video coding extension of the H.264/AVC standard. IEEE Signal Processing Magazine 25(2), 135–141 (2008)
15. Segall, C.A., Sullivan, G.J.: Spatial scalability within the H.264/AVC scalable video coding extension. IEEE Transactions on Circuits and Systems for Video Technology 17(9), 1121–1135 (2007)
16. Shahid, Z., Meuel, P., Chaumont, M., Puech, W.: Considering the reconstruction loop for watermarking of intra and inter frames of H.264/AVC. In: Proceedings of the 17th European Signal Processing Conference, EUSIPCO 2009, pp. 1794–1798. EURASIP, Glasgow (2009)
17. Tsihrintzis, G., Nikias, C.: Fast estimation of the parameters of alpha–stable impulsive interference. IEEE Transactions on Signal Processing 44(6), 1492–1503 (1996)
18. Watson, A.B.: DCT quantization matrices visually optimized for individual images. In: Proceedings of SPIE, International Conference on Human Vision, Visual Processing and Display, pp. 202–216. SPIE, San Jose (1993)
19. Zhang, J., Ho, A.T.S., Qiu, G., Marziliano, P.: Robust video watermarking of H. 264/AVC. IEEE Transactions on Circuits and Systems 54(2), 205–209 (2007)
20. Zou, D., Bloom, J.: H.264 stream replacement watermarking with CABAC encoding. In: Proceedings of the IEEE International Conference on Multimedia and Expo, ICME 2010, Singapore (July 2010)
21. Zou, D., Bloom, J.: H.264/AVC substitution watermarking: a CAVLC example. In: Proceedings of the SPIE, Media Forensics and Security, vol. 7254. SPIE, Jan Jose (January 2009)

Reversible Watermarking Using Prediction Error Histogram and Blocking

Bo Ou, Yao Zhao, and Rongrong Ni

Institute of Information Science, Beijing Jiaotong University, Beijing, P.R. China
{09112055,yzhao,rrni}@bjtu.edu.cn

Abstract. This paper presents a novel method using block-based predictive error histogram for reversible watermarking in spatial domain. This algorithm employs prediction error to embed data into the cover image. A pixel predictor based on Euclidean distance uses four neighboring pixels to predict the current pixel according to Euclidean distance and hides data into the selected pixels with a special predictive error by histogram shifting. Different from the existing histogram shifting schemes, where only one pair of peak points is used in the given image histogram, we divide the image into non-overlapping blocks and provide a pair of peak points for each block to take full advantage of embedding capacity. The set of peak point values will be compressed by arithmetic coding as a part of overhead information. Comparing with the existing spatial domain reversible watermark methods, the proposed method can achieve much better PSNR value at the high embedding rate. Experimental results prove the effectiveness of the proposed watermarking scheme.

Keywords: reversible watermarking, prediction error histogram, Euclidean distance.

1 Introduction

The increasing popularity of watermarking applications in medical and military fields has made reversible digital watermarking a research hotspot. Especially in recent years, high-capacity digital watermarking technology has drawn more and more attentions for its unique properties: the large embedding capacity and low distortion. The first well-known reversible watermarking was proposed by Tian[1] based on difference expansion(DE), where a location map is employed for restoring the original image. Tian's method gained a lower mathematical complexity and a larger embedding capacity at best 0.5b/pixel. However the uncompressed location map also consumes large embedding capacity. Soon Alattar [2] improved Tian's scheme by using the difference expansion of vectors of adjacent pixels to hide watermark, which hides two bits in every vector and decreases the location map size from one-half of the image resolution to one-third. Recently, Hu et al. [3] proposed a DE scheme with an improved overflow map based on prediction error to reduce the overflow map size. How to reduce the size of the location map become the goals in this field. In order to compress the location map, a novel reversible embedding scheme based on

H.-J. Kim, Y. Shi, and M. Barni (Eds.): IWDW 2010, LNCS 6526, pp. 170–180, 2011.

invariability and adjustment on pixel pairs (PDA) is proposed by S. Weng[4]. Kim et al.[5]presented a novel scheme to reduce the size of location map. Lin et al. [6] proposed a DE scheme and removed the location map completely.

Another classical algorithm for reversible watermarking is histogram shifting, which was first raised by Ni[7]. By shifting the histogram of an image, Ni created a gap in the given histogram for data embedding, where the embedding capacity is equivalent to the number of the peak point. Then Xuan[8]proposed a method using histogram shifting based on integer wavelets. It hides data into wavelet coefficients of high frequency sub-bands in integer wavelet transform domain. Under the same capacity, it can averagely increase PSNR 7~8dB compared with Ni's. In Hwang et al.'s paper [9], they improved Ni's scheme and applied a location map to restore original image without the knowledge of the peak point and zero point. Lin[11] utilizes the histogram of three-pixel block differences, while Tsai[10] using a residue image.

In this paper, we propose a novel data embedding method based on prediction error and blocking, which features a much better PSNR value in the high embedding rate. Different from the previous watermarking schemes, we utilize a new predictor based on Euclidean distance to get the prediction error histogram and apply a set of peak points for histogram shifting. The strategy is efficient since each block has its own peak points which results in a larger capacity for the whole image.

The rest of paper is organized as follows. In Section 2, the proposed reversible watermarking scheme is described. Experimental results are given in Section 3 followed by the conclusion drawn in Section 4.

2 Proposed Algorithm

This section presents our novel reversible data hiding scheme that uses prediction error shifting and blocking to obtain both a high payload embedding capacity and good image quality. Watermarking Framework is showed in Fig.1.A linear prediction is employed to obtain the prediction error image of the given image at first, then the image is partitioned into M ($M = (512\times512)/(N\times N)$)non-overlapping blocks sized $N\times N$ as illustrated in Fig.2. So there will be M different histograms for embedding. The pair of peak points in each block is searched for histogram shifting.

2.1 Prediction Error Shifting

Prediction error is the difference between the original pixel value and its predictive value. As showed in Fig.3, we utilize four half-enclosing casual pixels to predict the current pixel. x and $x_k, k = 1,...,4$ denote the current pixel and the neighboring pixels respectively. The predicted pixel value $\tilde{x}$ is calculated as follows:

$$\tilde{x} = round(\sum_{i=1}^{4} w_i x_i) \tag{1}$$

Here function $round(\bullet)$ rounds the element into the nearest integer. The weighted values $w_k, k = 1,...,4$ are determined by the Euclidean distance d_k between x_k and x. The computation of w_k is defined as:

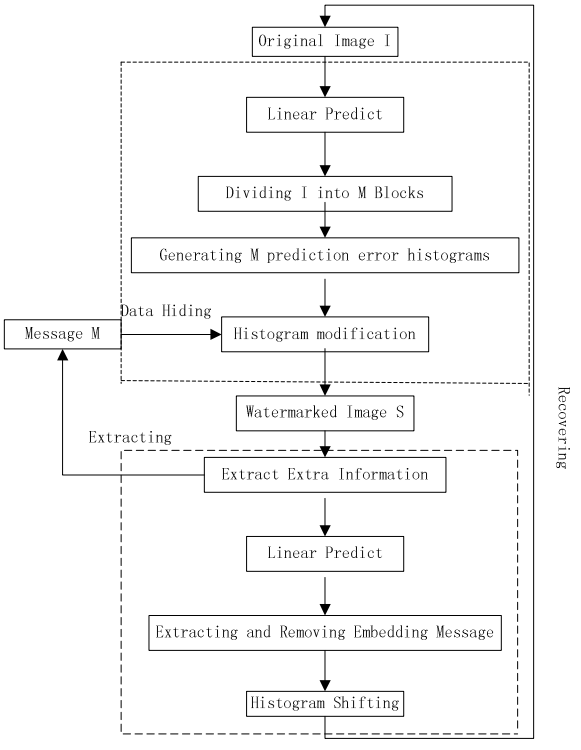


Fig. 1. Watermark framework: embedding process and extracting process

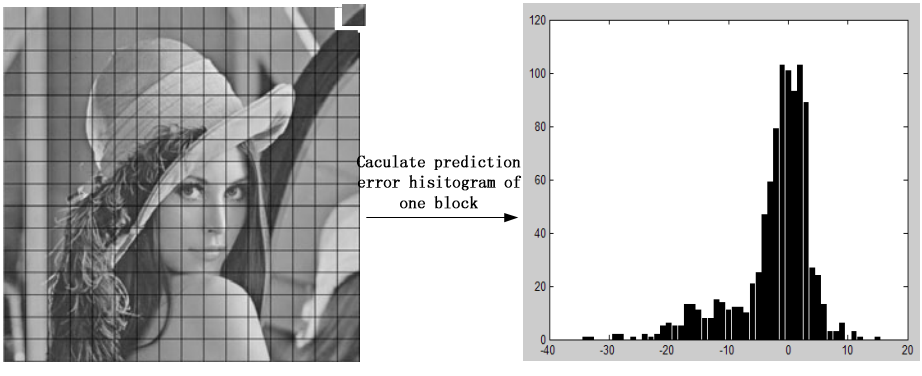


Fig. 2. The left represents the prediction error histogram is divided into $N \times N$ sized blocks; the right represents prediction error histogram of one block. Horizontal-axis denotes the value of prediction error and vertical-axis denotes the number of corresponding pixels.

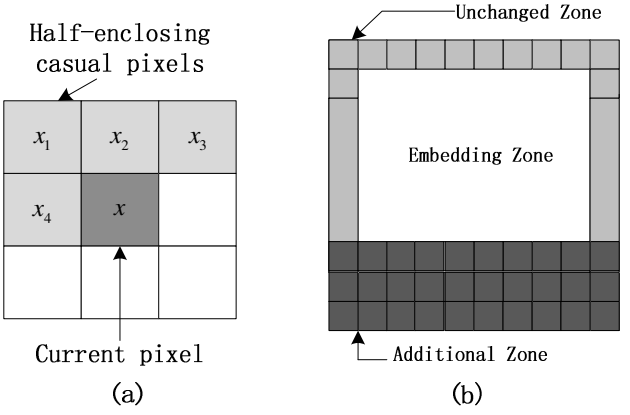


Fig. 3. (a) Prediction of half-enclosing casual pixel (b) Image structure: the white part represents the area for watermark embedding; the gray part represents the pixels in this area keep unchanged; the black part represents the area for overhead information embedding by LSB replacement

$$w_k = \frac{\frac{1}{d_k}}{\sum_{j=1}^{4}\frac{1}{d_j}} \tag{2}$$

Then predict error e is obtained via the Eq.(3)

$$e = x - \tilde{x} \tag{3}$$

We create the prediction error histogram, and find the largest two peak points LP_i and RP_i for one block. i denotes the position of the block in the whole image. ($i \in [1, M]$).Watermark bit b is embedded via Eq. (4)

$$e' = \begin{cases} e + symbol(e) \times b, & e = LP_i \text{ or } RP_i \\ e + symbol(e) \times 1, & e < LP_i \text{ or } e > RP_i \\ e, & else \end{cases} \tag{4}$$

The function $symbol(\bullet)$ defines as:

$$symbol(e) = \begin{cases} 1, & e \geq RP_i \\ -1, & e \leq LP_i \end{cases} \tag{5}$$

The watermarked pixel x_w becomes:

$$x_w = \tilde{x} + e' \tag{6}$$

We can restore original pixel x via Eq.(7)(8)

$$e = \begin{cases} e' - sign(e') \times b \ , \ e' \in [LP_i\text{-}1, LP_i\text{-}1] \ or \ e' \in [RP_i, RP_i + 1] \\ e' - sign(e') \times 1 \ , \ e' < LP_i\text{-}1 \ or \ e' > Rp_i + 1 \\ e' \qquad\qquad\qquad , \ else \end{cases} \tag{7}$$

$$x = \tilde{x} + e \tag{8}$$

When LP_i and RP_i are encountered, "0"is extracted; while LP_i-1 and $RP_i + 1$ are encountered, "1" is extracted.

$$b = \begin{cases} 0 & e' = LP_i \text{ or } e' = RP_i \\ 1 & e' = LP_i\text{-}1 \text{ or } e' = RP_i + 1 \end{cases} \tag{9}$$

2.2 Overhead Information

Assume the to be embedding bit stream is S . S is composed of three parts: payload P , overhead information O and the LSBs of the pixels to be replaced in additional zone. Referring to the Fig.3, the given image is classified into three type area: the unchanged zone; the embedding zone; the additional zone. The embedding zone is used for payload embedding, while overhead information is embedded in additional zone by LSB replacement. The LSBs is appended after the payload. After the payloads are embedded, the overhead information is compressed by arithmetic coding and embedded for restoring the original image. Overhead information contains the following:

(1) The information about the position of the pixels with the value of 0 or 255, which may occur overflow/underflow by +1/-1. These pixels are not embedded any bit and keep unchanged. The only thing we need to do is restore the positions of these pixels and tell whether the pixel is this kind when extracting the secret bits.

(2) A set of LP_i, RP_i values which are used to tell whether the pixel is embedded watermark bit.

An EOS symbol is added at both the end of payload and overhead information.

2.3 Embedding Process

The detailed description of the embedding process is given as follows:

(1) As discussed in Section 2.1, calculate predictive errors and generate a predictive error map as the same size of embedding area. Find out all the LP_i and RP_i .

(2) Then restore compressed extra information, which is embedded into the LSBs of the pixels in additional area by simple LSB replacement. The LSBs to be replaced need to be appended to the pure payload.

(3) Scan the Embedding Zone row by row from the beginning, if $x \in \{0, 255\}$ go to the step (5), else go to step 4).

(4) As the e being selected, according to Eq. (4) (6), modify x into x_w , e into e' .

(5) Let c_1 denote the condition when the embedding capacity is completed, let c_2 denote the condition when the watermark payload is finished. If c_2 is satisfied ,go to step (6);If c_1 is satisfied and c_2 is not satisfied ,start to multi-level embedding; if both of them are not satisfied ,go to step(3).
(6) Stop embedding and the watermarked image is created.

2.4 Extracting Process

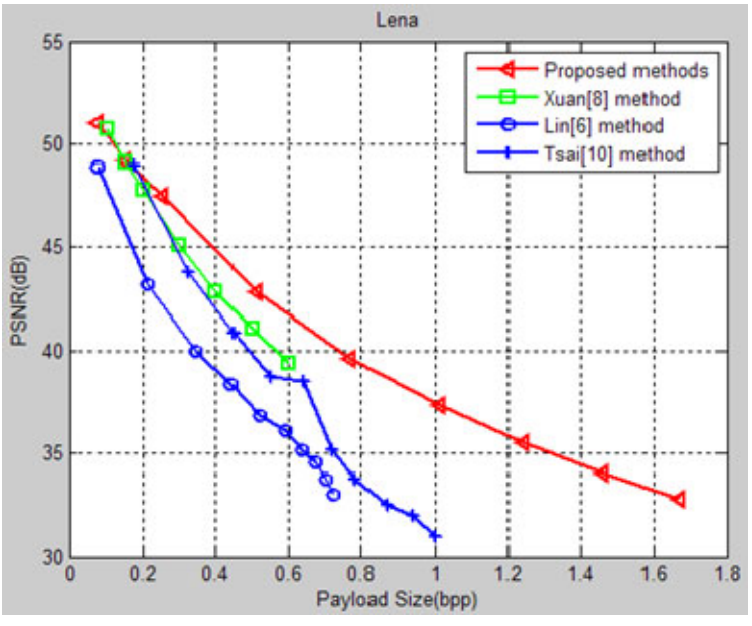
(1) Obtain overflow information from the additional area.
(2) Scan the embedding zone as the former order according to overhead information, use equation(1) to calculate $\tilde{x}$ and e';
(3) Use Eq. (7)(8)(9) to restore the current pixel and extract the watermark bit.
(4) Check the watermark bits. If EOS is encountered, stop recovery, else go to step (2).

3 Experimental Results

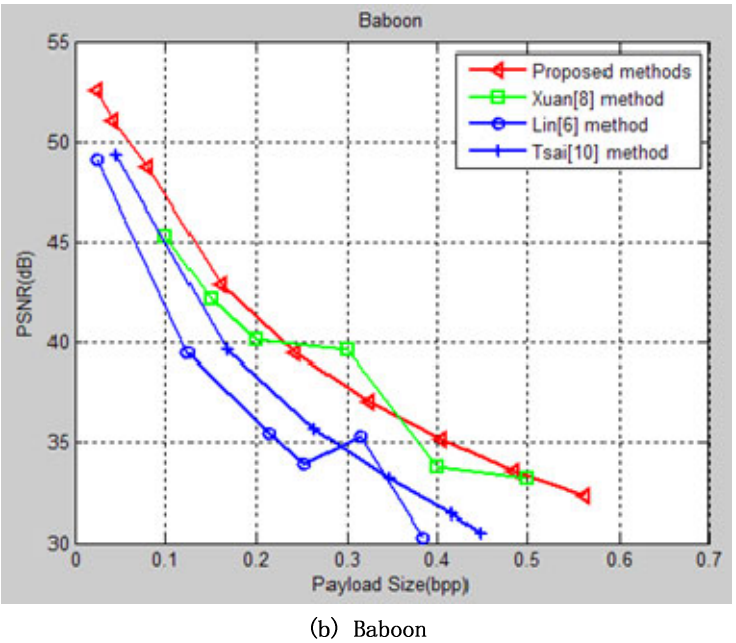
Experiments on two frequently used 512 ×512 grayscale standard images (Lena, Baboon) are reported here. In our experiments, the number of payload bits and the peak signal-to-noise ration are adopted as the measurements for embedding capacity and image distortion. The embedding capacity in the experiment only refers to the payload, not including the overhead information. For evaluation of the proposed method, we have included the performances of three reversible watermarking schemes proposed by Lin [6], Xuan[8], Tsai[10].Lin's[6] method is based on DE while Xuan's[8] and Tsai's[10] methods based on histogram shifting using integer wavelets and predictive coding respectively are similar with ours.

In our experiments, the block size is 32 ×32 for Lena image and 64 ×64 for Baboon image. As showed in Fig.4, the single-layer embedding capacity of our scheme is about 0.25bpp for Lena image and 0.08bpp for Baboon image. By multiple-layer embedding when higher embedding capacity is required, the performance of the proposed scheme can easily outperform the comparative methods, achieving 32.77dB at embedding rate of 1.67bpp for Lena image, and 32.27dB at embedding rate of 0.56 bpp for Baboon image. When embedding the same payload for Lena image, the proposed method is about 2-7dB greater than Tsai's method [10]. For example, when the embedding rate is nearly 1.0bpp, the PSNR can achieve 37dB. However, Tsai's is 31dB at the same embedding rate. Baboon is the complex texture image, so the PSNR is lower and is about 2-4dB better than Tsai's at same embedding rate.

Two reasons make our scheme outperform the other schemes. The first reason is our embedding method. Since we divide the image into blocks, the indexes are selected according to the prediction error histogram of each block, which making the entire capacity of all blocks more than the capacity of image without blocking. The Fig.5 provides the comparison with Hu's method [3] and performance of the proposed method for different block size. As the Fig.5 shows, more blocks mean a larger embedding capacity, but it also makes overhead information larger. For Baboon image,



(a) Lena



(b) Baboon

Fig. 4. The performance evaluation of multi-layer embedding over standard test images

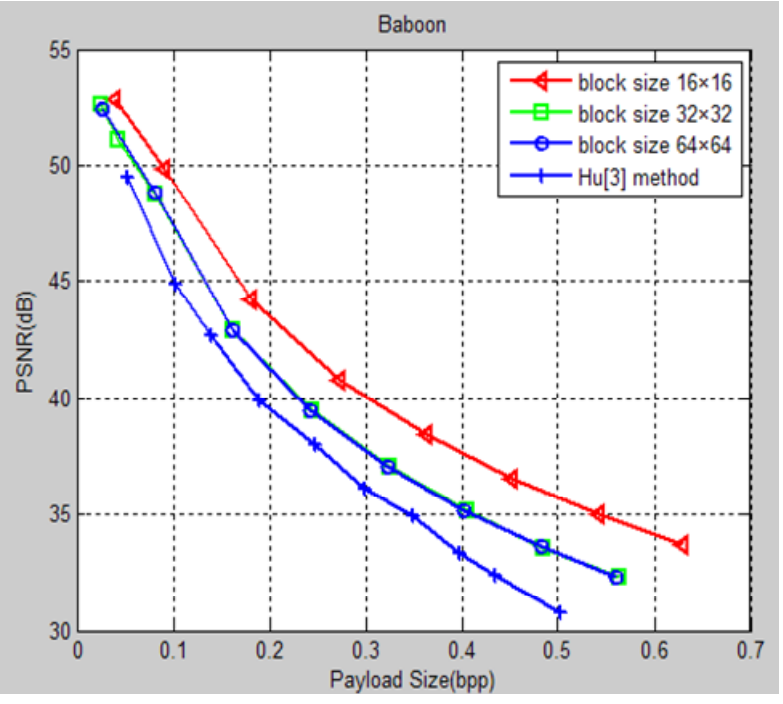


Fig. 5. The comparison with Hu's method [3] and performance of the proposed method for different block size for baboon image

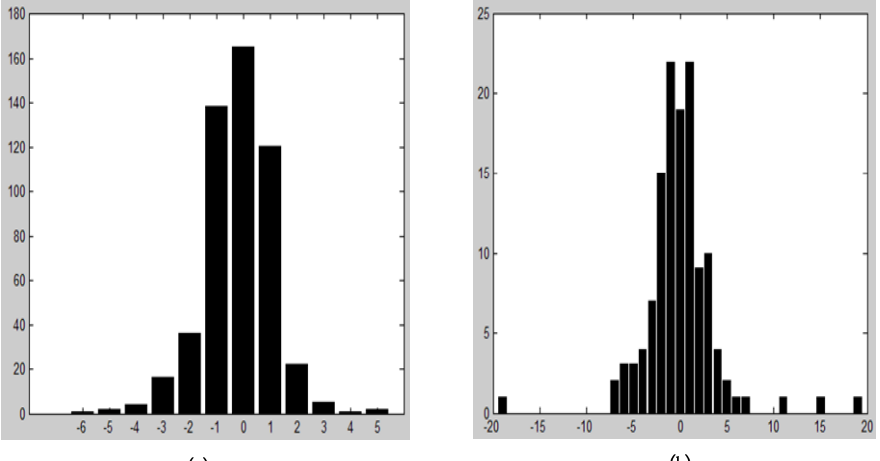


Fig. 6. The distribution of indexes for different images:(a) Lena, block size 32 ×32; (b) Baboon, block size 64 ×64

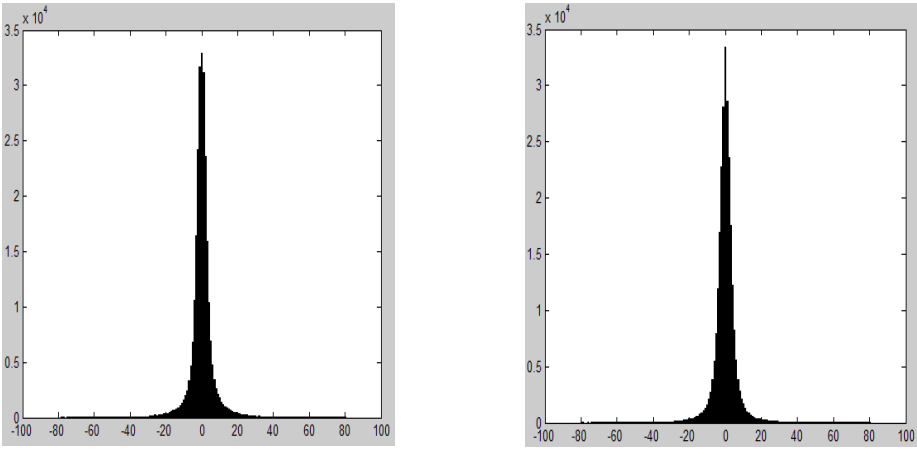


Fig. 7. The comparison of prediction error histogram for Lena image with the proposed prediction scheme and the JPEG-LS prediction scheme

Table 1. The overhead information vs payload for test image

	Lena Image			Baboon Image		
Multiple-layer Embedding	Pure Capacity (bits)/bpp	Block Size (N×N)	Overhead Information Size (bits)	Pure Capacity (bits)/bpp	Block Size (N×N)	Overhead Information Size (bits)
One-layer	67.2k/0.26	32×32	1288	21.4k/0.08	64×64	536
Two-layer	134.6k/0.51	32×32	1352	42.7k/0.16	64×64	536
Three-layer	202k/0.77	32×32	1272	64.0k/0.24	64×64	552
Four-layer	265.3k/1.01	32×32	1304	85.2k/0.33	64×64	560
Five-layer	325.6k/1.24	32×32	1400	106.3k/0.41	64×64	560
Six-layer	383.3k/1.46	32×32	1456	127.1k/0.48	64×64	512
Seven-layer	438.1k/1.67	32×32	1416	147.6k/0.56	64×64	552

Table 2. The size of overhead information before and after compression

Lena Image				Baboon Image		
Block Size	16×16	32×32	64×64	16×16	32×32	64×64
Size of overhead information before compression (bits)	8192	2048	512	12,288	3072	768
Size of overhead information after compression (bits)	5936	1280	296	10,400	2392	536

when the number of the blocks is 0-256, the embedding capacity is improved little, while the number is 256-1024, the embedding capacity obviously changes. If the given image is divided into 4096 or more blocks, the single-layer embedding capacity is over 28k bits, but the size of overhead information is too large for additional zone embedding. It is proved through abundant experiments that how many the blocks are partitioned determines the embedding capacity in our scheme. The Fig.6 gives the distribution of indexes for Lena image (block size 32 ×32) and Baboon image (block size 64 ×64), and Table.2 provide the size of the overhead information before and after compression. The Fig.6 and Table.2 show that the compression rate of overhead information depends on the distribution of the indexes of given image. For example the distribution of Lena image is more concentrated than Baboon image, so the fewer bits in binary are needed to represent the indexes. The second reason is due to our prediction method. As showed in Fig.7, compared with the JPEG-LS prediction scheme, the prediction error histogram of ours produces a better shape. Although the maximum value is almost the same, the second and third highest value are much higher, which would produce a larger capacity for single-layer embedding.

4 Conclusions

In this paper, a novel reversible watermarking scheme is presented. Different from the recent schemes, the proposed scheme uses a new predictor based on Euclidean distance to get the prediction error histogram and apply a set of peak points for histogram shifting. According to the experimental results, our scheme achieves higher capacity and better image quality for watermarked images.

Acknowledgements

This work was supported in part by National NSF of China (No. 60776794, No.60702013, No. 61073159), 973 Program (No. 2006CB303104), 863 program (No. 2007AA01Z175), PCSIRT (No. IRT0707), Fundamental Research Funds for the Central Universities (2009JBZ006).

References

1. Tian, J.: Reversible data embedding using a difference expansion. IEEE Trans. Circuits Systems and Video Technology 13(8), 890–896 (2003)
2. Alattar, A.M.: Reversible watermark using difference expansion of a generalized integer transform. IEEE Trans. Image Process. 3(8), 1147–1156 (2004)
3. Hu, Y., Lee, H.-K., Li, J.: De-based reversible data hiding with improved overflow location map. IEEE Trans. Circuits and Systems for Video Technology 19(2), 250–260 (2009)
4. Weng, S., Zhao, Y., Pan, J.S., Ni, R.: IEEE Signal Processing Letters 15, 721–724 (2008)
5. 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)

6. Lin, C.C., Yang, S.P., Hsueh, N.L.: Lossless data hiding based on difference expansion without a location map. In: 2008 Congress on Image and Signal Processing, pp. 8–12 (2008)
7. Ni, Z., Shi, Y.Q., Ansari, N., Wei, S.: Reversible data hiding. IEEE Trans. Circuits and Systems for Video Technology 16(3), 354–362 (2006)
8. 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)
9. Hwang, J., Kim, J.W., Choi, J.U.: 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)
10. Tsai, P., Hu, Y.C., Yeh, H.L.: Reversible image hiding scheme using predictive coding and histogram shifting. Signal Processing 89, 1129–1143 (2009)
11. Lin, C.C., Hsueh, N.L.: A lossless data hiding scheme based on three-pixel block differences. Pattern Recognition 41(4), 1415–1425 (2008)

An Efficient Pattern Substitution Watermarking Method for Binary Images

Keming Dong and Hyoung-Joong Kim

Graduate School of Information Management and Security,
Korea University, Seoul 136-701, South Korea
{dongkeming,khj-}@korea.ac.kr

Abstract. In this paper, a method to decrease the size of location map for non-overlapping pattern substitution method is presented. Original pattern substitution (PS) method has been proposed by Ho et al.[1] as a reversible watermarking scheme for binary images. They use a pair of two patterns to embed data. Unfortunately, their location map is huge in size. In our method, we propose an efficient mechanism which can decrease the size of location map considerably for un-overlapping version of the PS method. Experiment results show that our method works well on decreasing the size of location map. Comparison results with the original PS method demonstrate that the proposed method achieves more embedding capacity and higher PSNR value due to the reduced size of the location map.

Keywords: Binary image watermarking, reversible watermarking, pattern substitution.

1 Introduction

Digital watermarking and data hiding techniques are used to embed special/secret data into digital *cover content* to produce *stego-content* with the least amount of distortion. They have aroused great interest due to their wide application areas such as copyright protection, covert communication, annotation and authentication. There are many reversible watermarking methods on gray scale or color images. However, since only a single bit plane is allowed for binary images, as a result, a single bit modification can easily cause noticeable artifacts in the black-and-white binary images if the modified position is not on the boundary. Therefore, binary reversible data hiding techniques are relatively difficult to design. As a result, only a small number of reversible watermarking techniques are available for binary images.

The pair-wise computation (PWLC) method has been proposed by Tsai et al. [3]. This method firstly uses a mechanism to detect the patterns such as '000000' or '111111' on the boundary and flip the third place of the two patterns to embed data. The two patterns should be on the boundary in the image. In order to embed one bit, they use the 6-bit patterns. Thus, the embedding capacity of this method is not so high. Ho et al. [1] have presented another

H.-J. Kim, Y. Shi, and M. Barni (Eds.): IWDW 2010, LNCS 6526, pp. 181–188, 2011.

reversible data hiding scheme called pattern substitution (PS) method. The PS method also firstly tries to find the boundary (i.e., edge), and uses two patterns 'PF' (least probable pattern) and 'PM' (most probable pattern) to embed data which is called as a *pattern pair*. The size of PF and PM patterns is just 4 bits in length, and generally PF and PM patterns are more in numbers than '000000' or '111111' patterns in the PWLC scheme. In [1], the comparisons with the PWLC method show superiority of the PS method. Recently, another type of reversible data hiding technique has been proposed by Xuan et al. [5]. In this method, they exploit a run-length histogram modification to embed data.

In the PS method, the location map usually is huge in size, especially for binary images transformed from gray scale or color images such as 'Lena'. In this paper, we concentrate on decreasing the size of the location map for non-overlapping PS method. First, we define a pattern for replacement of PF in pattern pairs called "PFR" which means the second least probable pattern for replacement. Second, we change the PF into the PFR patterns, and we flag the patterns in the location map to distinguish two patterns. Then, we use the PM patterns only to embed secret data. Thus, the location map between the PF and PFR patterns will be smaller than the map between PF and PM patterns.

The remainder of this paper is organized as follows: Section 2 examines the PS method and its drawback. The proposed scheme is described in detail in Section 3 and analyzed in Section 4. Finally, conclusions are presented in Section 5.

2 Pattern Substitution Method

In the pattern substitution method, it firstly transforms the binary image into the difference value matrix denoted as D based on Eq.1.

$$D(i,j) = \begin{cases} H(i,j) & i = 1, j = 1 \\ H(i,j) \oplus H(i-1,j) & i \neq 1, j = 1 \\ H(i,j) \oplus H(i,j-1) & j > 1 \end{cases} \quad (1)$$

where $\oplus$ represents the exclusive-OR logic operation, i and j are the row and column positions in the cover image, respectively, and H denotes the image matrix consisting of the pixel values of the original binary image. Differencing operation produces edges or boundaries where black and white pixels meet. An illustration of the image difference value matrix is shown in Figure 1.

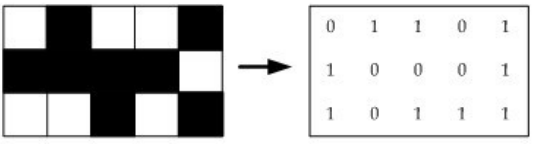


Fig. 1. An example of binary image matrix,H(left) and its difference value matrix, D(right)

Table 1. Entire substitutable difference value patterns

Difference value pattern	Substitutable difference value pattern
P_{0001}	$P_{0010}, P_{0111}, P_{1101}$
P_{0010}	$P_{0001}, P_{0100}, P_{1110}$
P_{0011}	$P_{0000}, P_{0101}, P_{1111}$
P_{0100}	$P_{0010}, P_{0111}, P_{1000}$
P_{0101}	$P_{0011}, P_{0110}, P_{1001}$
P_{0110}	$P_{0000}, P_{0101}, P_{1010}$
P_{0111}	$P_{0001}, P_{0100}, P_{1011}$
P_{1000}	$P_{0100}, P_{1011}, P_{1110}$
P_{1001}	$P_{1000}, P_{1010}, P_{1111}$
P_{1010}	$P_{0110}, P_{1001}, P_{1100}$
P_{1011}	$P_{1000}, P_{0111}, P_{1101}$
P_{1100}	$P_{0000}, P_{1010}, P_{1111}$
P_{1101}	$P_{0001}, P_{1011}, P_{1110}$
P_{1110}	$P_{0010}, P_{1000}, P_{1101}$
P_{1111}	$P_{0011}, P_{1001}, P_{1100}$

From the difference value matrix D, we can define a set of four consecutive pixels as a pattern. As we can see, there are 16 patterns from P_{0000} to P_{1111}. Among them, the pattern P_{0000} indicates that the consecutive pixels are all same. In other words, it is not on the boundary. Thus, we skip this pattern. As a result, only 15 patterns are considered as a valid pattern for hiding data.

We firstly divide the patterns into two groups depending on the parity of 1's in the pattern: either even or odd patterns. The substitution should occurs between the same group of patterns. In order to minimize the distortion, a binary bit pattern should be substituted by other binary bit patterns having the Hamming distance 1 between the original pixel value of them. For example, we consider the pattern P_{0001}, then the binary bit pattern can be B_{0001} or B_{1110} depending on whether the first neighboring pixel is 0 or 1. We can verify that if we substitute P_{0001} with P_{0010}, the distance between the original bit pattern and the resultant bit pattern is just 1. This pattern, P_{0010}, is called as the substitutable difference value pattern of P_{0001}. Table 1 enumerates all the substitutable pairs of difference value patterns.

In the difference value matrix, the most probable pattern except P_{0000} is called PM, and the least probable pattern PF is decided associated with the PM from Table 1. Embedding rule is simple. Once the pair of PM and PF is decided, the PS algorithm keeps the PM pattern as it is to hide a bit 0 or turns it into the PF pattern to hide a bit 1 or keeps the PF as it is or turns it into PM. Before data embedding process, the position of the original PF patterns is recorded into the location map. Thus, we can also hide data into the PF patterns as well.

Ho et al. [1] have introduced two types of the PS method. One is the non-overlapping pattern substation method where the difference value patterns are distinct and independent of each other. In this case, we embed data into each independent pattern and move to the next independent pattern. In order words,

the possible embedding position can only in the rows 1, 5, 9, ... The other one is called overlapping pattern substitution method. In this approach, the next patterns are decided dynamically after processing each pattern. They [1] propose two rules in order to overcome some overlapping problems. In this paper, we focus on the non-overlapping version.

3 The Proposed Scheme

Ho et al.'s [1] method marks all the positions of the PF patterns into the location map. Therefore, the size of the location map is the length of position information times the number of PF patterns. Thus, the size of the location map is proportional to the number of PF patterns. Ho et al. [1] recorded all the PF patterns' positions in x and y coordinates. The position of the i_{th} PF pattern, PF_i, is recorded by a tuple (x_i, y_i). When the size of the binary image or the number of PF patterns is large, this approach is not desirable. For example, for a 512*512 image, each tuple for one PF pattern requires 18 bits. Thus, the size of the location map can be $18 \times N_{PF}$ (number of PF patterns).

The most important concept in the proposed method is the second least probable pattern for replacement *PFR* to pattern pair PF. Before the embedding process, we firstly transform all the PF to PFR, and the location map records the original PF positions and the PFR positions. We scan the difference value matrix D from left to right and from top to bottom. If the pattern we encounter is a PF pattern, then we transform the PF into PFR and record 1 in the location map. If the pattern is PFR, we record 0 in the location map. The algorithm for generating location map is as follows:

Pseudocode: Location map generation;
1. **LOOP**
2. **IF** an encountered pattern is PF
3. Transform the PF into PFR;
4. Mark 1 to the location map;
5. **ELSE IF** an encountered pattern is PFR
6. Mark 0 to the location map;
7. **ELSE**
8. To next pattern;
9. **END LOOP**

After the first step (generation of the location map), there are no PF patterns in the original difference matrix D. Due to the location map, the whole PF patterns can be removed at the encoder and recovered perfectly at the decoder. The location map is automatically generated once the pair of PM, PF, and PFR is decided. Note that overflow/underflow problem does not happen in the PS method. According to the given secret message, embedding procedure starts. In the embedding step, we use only the PM patterns to embed data. Since there are no more PF patterns, we can keep the PM patterns as they are or change them into PF patterns. For example, if the secret data bit to embed is 1, the PM will

be left as it is, and otherwise, the PM will be turned into PF. Encoder should send the whole length of the embedded message, total length of the location map, pair information of PM, PF, and PFR, and finally the location map.

Decoder extracts the relevant information such as the length of the secret message embedded, length of the location map, and the pair information of PM, PF, and PFR, and the location map. Using these pieces of information, we can extract the secret message.

Assuming that the size of the image is $n_x \times n_y$. Then, the length of the location map by the original PS method is $(\log n_x + \log n_y) \times N_{PF}$. However the length of the location map by the proposed scheme is $N_{PF} + N_{PFR}$, where N_{PFR} is the number of PFR in the original difference value matrix D. The factor of $\log n_x + \log n_y$ is always more than 1, and because PFR is the least probable pattern except PF, generally, the proposed method can generate smaller location map, and the result is more considerable when N_{PN} is huge in numbers.

4 Experimental Results

Compared with the original non-overlapping PS method, the experimental results of the method in this paper show that our method can generate smaller location map, so it can offer not only a higher maximum data hiding capacity but also better visual quality. Fig. 2 shows three binary images we used in our

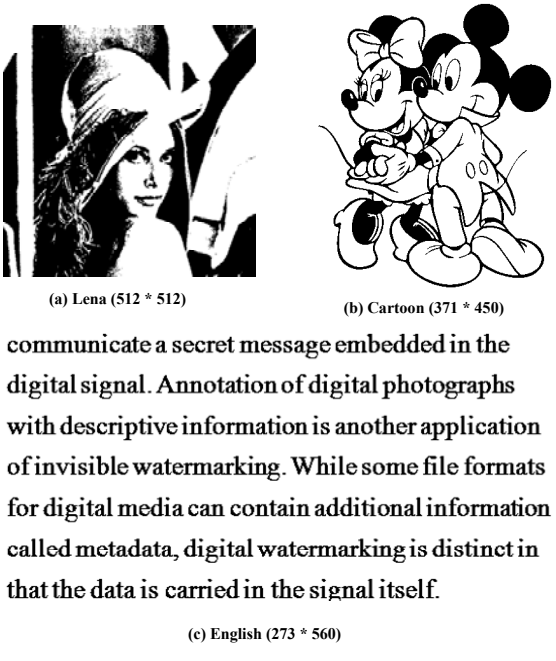


Fig. 2. Binary images: (a) Lena (b)Cartoon (c)English

Table 2. The pattern pairs used in experiments

	Original			Proposed		
	Pair tuple	NO.	LM	Pair triad	NO.	LM
Lena	$(P_{0001}, P_{1101}, P_{1011})$	(1972,102,147)	249	(P_{0001}, P_{1101})	(1972,102)	1836
Cartoon	$(P_{1000}, P_{1110}, P_{1111})$	(2032,0,0)	0	(P_{1000}, P_{1110})	(2032,0)	0
English	$(P_{0100}, P_{0111}, P_{1011})$	(2660,27,11)	38	(P_{0010}, P_{1110})	(2458,19)	361

experiments as test images. Fig. 2(a) is obtained from gray scale Lena image. Binary image in Fig. 2(b) is original. Fig. 2(c) is an English document.

The pattern pairs used in the experiments and the comparison of the length of location map between the original non-overlapping PS method and the proposed method are shown in Table 2. Fig. 3 shows the distortion in terms of PSNR where various sizes of secret data are embedded into diverse types of cover images using the proposed scheme and the original non-overlapping PS method. As the results reveal, the length of the location map by the proposed method is always not bigger than the original PS method, and the PSNR performance of the proposed method is always better/equal than/to the original one no matter which test image is used. Fig. 4 is the stego-images from the proposed method.

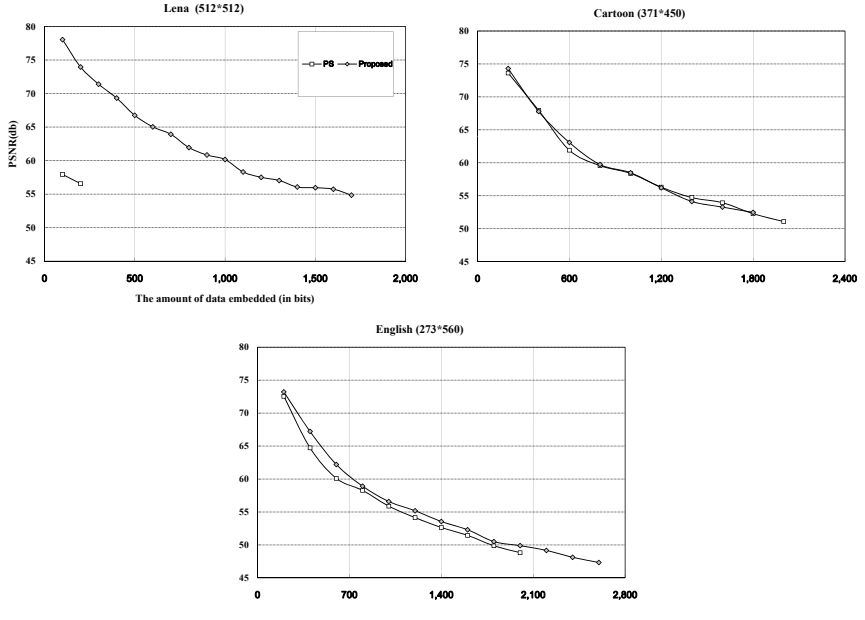


Fig. 3. Comparison of PSNR for various types of images

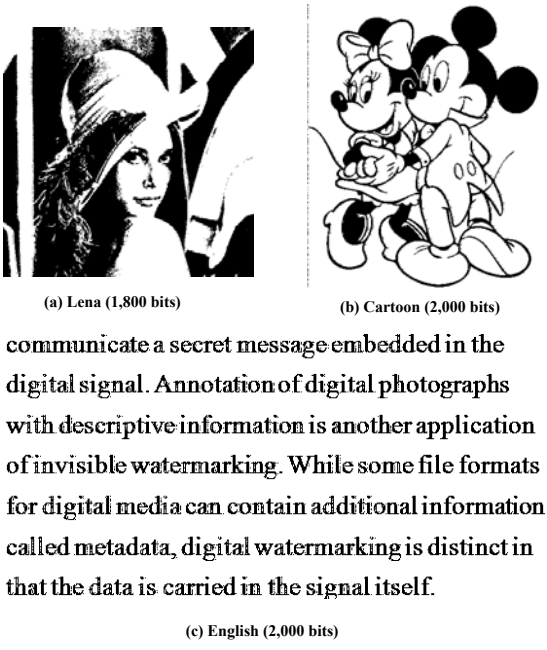


Fig. 4. Stego-images of various types of images: Lena is embed by 1,800 bits, and the rest are embedded 2,000 bits

5 Conclusions

In this paper, a scheme to decrease the length of location map for non-overlapping PS method is proposed. We show that our method can always get better results than original non-overlapping PS method, specially it can decrease the length of the location map considerably for the complex binary images, such as the binary images obtained from gray scale images like Lena. The experimental results show that the proposed method can always achieve not only higher maximum data hiding capacity but also less visual distortion on such binary images.

References

1. Ho, Y.-A., Chan, Y.-K., Wu, H.-C., Chu, Y.-P.: High-capacity reversible data hiding in binary images using pattern substitution. In: Computer Standards and Interfaces, pp. 787–794 (2009)
2. Mei, Q., Wong, E.K., Memon, N.: Data hiding in binary text documents. In: Proceedings of SPIE - The International Society for Optical Engineering, pp. 369–375 (2001)

3. Tsai, C.-L., Chiang, H.-F., Fan, K.-C., Chung, C.-D.: Reversible data hiding and lossless reconstruction of binary images using pair-wise logical computationmechanism. In: Pattern Recognition, pp. 1993–2006 (2005)
4. Wu, M., Tang, E., Liu, B.: Data hiding in digital binary image. In: IEEE International Conference on Multi-Media and Expo. (2000)
5. Xuan, G., Shi, Y.Q., Chai, P., Tong, X., Teng, J., Li, J.: Reversible binary image data hiding by run-length histogram modification. In: 19th International Conference on Pattern Recognition (2008)

New JPEG Steganographic Scheme with High Security Performance

Fangjun Huang[1], Yun Qing Shi[2], and Jiwu Huang[1]

[1] School of Information Science and Technology, Sun Yat-Sen University, Guangzhou, GD 510006, China
{huangfj,isshjw}@mail.sysu.edu.cn
[2] Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102, USA
shi@njit.edu

Abstract. In this paper, we present a new JPEG steganographic scheme. Three measures are taken in our method: 1) The secret message bits are not spread into the quantized block discrete cosine transform (BDCT) coefficients of all frequencies, and only those coefficients (including those of value 0) belonging to relatively low frequencies are selected for data embedding; 2) For any coefficients selected for embedding, the rounding error in JPEG quantization is utilized directly to guide the data embedding; 3) Matrix embedding. The experiments have demonstrated that these three measures can help to achieve small distortion in spatial domain, preserve the histogram of quantized block discrete cosine transform coefficients, and enhance the embedding efficiency of matrix embedding, etc. Consequently, the proposed steganographic scheme has achieved a high security performance. It can resist today's most powerful JPEG steganalyzers effectively.

Keywords: JPEG steganographic scheme, BDCT, Frequency.

1 Introduction

Steganography is such a secret communication approach that it can transmit information without arousing suspicion of the existence of the secret communication. The carrier of steganography can be various kinds of digital media such as text, image, audio, and video. Due to the common use of JPEG images, JPEG steganography has recently attracted more and more attention. Some JPEG steganographic methods have been proposed, e.g., J-Steg [1], JPHide [2], Outguess [3], F5 [4], MB [5], PQ [6], MME [7] and YASS [8], etc.

The initial steganographic algorithms such as J-Steg and JPHide concentrate on the imperceptibility. Both of these two LSB substitution based JPEG steganographic schemes can resist the visual attacks presented in [9] successfully. However, because the Pairs of Values (PoVs) will be generated in the quantized BDCT coefficient histogram during message embedding, these initial JPEG steganographic schemes such as J-Steg and JPHide can be detected easily by the chi-square attack [9] and the extended chi-square attack [10, 11]. For

H.-J. Kim, Y. Shi, and M. Barni (Eds.): IWDW 2010, LNCS 6526, pp. 189–201, 2011.

the sake of security, some steganographic algorithms such as Outguess and MB focus on preserving the histogram of quantized BDCT coefficient of the cover image as much as possible. However, they still leave some artifacts. Through observing the block artifacts [12] and over fitting effect of quantized BDCT coefficient histogram [13] in the stego images of Outguess and MB, these two JPEG steganographic schemes can be detected effectively.

Among all approaches that have been adopted to enhance the security performance of JPEG steganography, matrix embedding is very effective. With the utilization of matrix embedding, less alternation needs to be made to the cover image while embedding the same amount of information bits. The idea of importing matrix embedding into steganography is first proposed by Crandall [14]. Westfeld [4] implemented it into F5, which resorts to the Hamming codes to reduce the modifications to the quantized BDCT coefficients. In order to preserve the characteristics of histogram of quantized BDCT coefficients, F5 decrements the magnitude of the coefficient by 1 while the embedded message bit does not match the LSB of the corresponding quantized BDCT coefficient. If the subtraction leads to a zero coefficient, the same message bit must be embedded in the next coefficient again at the transmitting end. Thus the number of coefficients with value equal to 0 will increase in the stego image, which is referred to shrinkage effect. Resorting on the calibration technique, this shrinkage effect is analyzed and the message length is possible to be estimated [15].

Another important approach for improving the security performance of JPEG steganography is to embed the secret message bits into those coefficients that may introduce minimal distortion. One of the most important minimal distortion rules is called perturbed quantization (PQ) [6], in which the secret message bits are embedded into those changeable coefficients whose rounding errors are close to 0.5. With the help of wet paper codes, Fridrich et al. have exemplified the PQ embedding strategy based on double JPEG compression. They take a singly compressed JPEG file as the cover image, and embed the secret message bits while recompressing the JPEG image for the second time with a lower quality factor. Some improved versions such as PQt (text-adaptive PQ) and PQe (energy-adaptive PQ) can be found in the recent work [16].

In [7], Kim et al. combined matrix embedding and minimal distortion embedding strategies. Specifically, they applied the minimal distortion rule in a simple and practical way and applied it with modified matrix encoding (MME). According to the allowable changing bits in each block, the MME schemes are called MME2, MME3, etc. Since the secret message bits are embedded into those coefficients which may introduce minimal distortion, MME has a better security performance than F5 even though they both use the Hamming codes for matrix embedding. The MME3 is considered currently the most secure steganographic scheme [16].

Recently, Solanki et al. [8] presented a JPEG steganography called YASS (Yet another steganographic scheme). The secret message bits are embedded into those randomized 8×8 blocks which may not coincide with the 8×8 grid used in standard JPEG compression. This new randomized embedding approach

can effectively disable some universal JPEG steganalyzers based on calibration technique [17, 18]. However, as Huang et al. pointed in [24, 25], with YASS' complicated embedding procedure, the intra- and inter-block dependencies among the quantized BDCT coefficients was still disturbed after the secret message embedding. Thus some non-calibration based universal JPEG steganalyzers [19-21] can break it. In [26], Bin et al. pointed out that YASS' embedding procedure are not randomized enough and hence the possible locations and impossible locations of embedding blocks could be identified. Consequently, the fact that extra zero coefficients having been introduced into the selected blocks by the QIM (Quantization Index Modulation) embedding can be exploited, and a new specific steganalyzer is designed to detect it.

In this paper, we present a new JPEG steganographic scheme. In addition to the utilization of matrix embedding, three measures are taken in our method to improve the security performance: 1) The secret message bits are only embedded into those coefficients belonging to relatively low frequencies (including all the zero-valued coefficients); 2) For any coefficients selected for embedding, the rounding error in JPEG quantization is utilized to guide the data embedding; 3) Matrix embedding. Via these three measures, some good properties such as small distortion in spatial domain, preservation of the histogram of quantized BDCT coefficient, and high embedding efficiency of matrix embedding can be obtained. Consequently, our algorithm has a high security performance and it can effectively resist the most powerful JPEG steganalyzers nowadays.

The rest of this paper is organized as follows. In Section II, we introduced our new steganographic scheme. Experimental results and analysis are given in Section III, and the conclusion is drawn in Section IV.

2 The Proposed Embedding Algorithm

2.1 JPEG Compression

The standard JPEG compression procedure is illustrated in Fig. 1. The encoding process consists of several steps: splitting the input image into consecutive and non-overlapped blocks of 8×8 pixels, applying two dimensional (2D) BDCT to each block, dividing and rounding, followed by entropy encoding.

As seen, in JPEG compression procedure, two different kinds of BDCT coefficients will be obtained, namely, the un-rounded BDCT coefficients and quantized BDCT coefficients. The coefficients that have been divided by quantization steps and not yet rounded are called un-rounded BDCT coefficients, and those

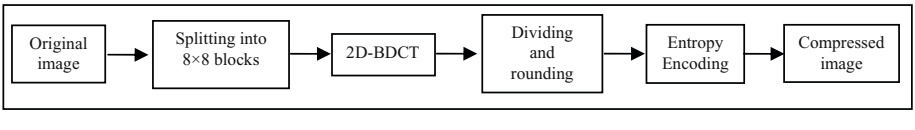


Fig. 1. The block diagram of JPEG compression process

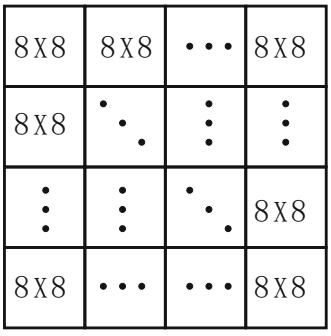


Fig. 2. Structure of un-rounded BDCT 2-D array and quantized BDCT 2-D array

having been divided by the quantization steps and rounded are called quantized BDCT coefficients. The corresponding matrixes which consist of the 8×8 un-rounded BDCT coefficients and quantized BDCT coefficients are called un-rounded BDCT 2-D array, and quantized BDCT 2-D array, respectively, which are shown in Fig. 2. These two matrixes both have the same size as that of the input uncompressed image.

2.2 Embedding Process

We suppose that the raw, uncompressed image is available to the sender, thus the knowledge of un-rounded BDCT coefficients can be utilized to guide the data hiding, and hence minimize the distortion that may be introduced in the embedding process. In addition, in order to further minimize the embedding distortion, in our algorithm all the secret message bits are embedded into those coefficients (including zero coefficients) belonging to relatively low frequencies, in which the quantization steps are smaller than that of the higher frequencies according to the zig-zag scanning order. Suppose that the input uncompressed image is of dimension $N_1 \times N_2$ in spatial domain. Without loss of generality, we assume that both N_1 and N_2 are the multiples of 8. There are in total $N_B (N_B = (N_1/8) \times (N_2/8))$ blocks in it. In our algorithm, the secret message bits are embedded through modifying the LSBs of the selected coefficients in quantized BDCT 2-D array, and the corresponding coefficients in un-rounded BDCT 2-D array are used as the guidance to determine how to modify the corresponding coefficients. In order to improve the embedding efficiency (increase the number of bits embedded per embedding change), the matrix embedding is utilized in our algorithm. It is noted that some other error correction codes such as BCH (Bose, Chaudhuri and Hocquenghem) and RS (Reed-Solomon) codes [27] all can be utilized to improve the efficiency of matrix embedding. For simplicity, we exemplify our embedding strategy with $[2^k - 1, k] (k \geq 1)$ Hamming codes. That is, k secret message bits will be embedded into $2^k - 1$ coefficients by making at most 1 embedding change. The embedding process of our proposed steganographic scheme is as follows.

1) Start JPEG compression from an uncompressed image. Produce the un-rounded BDCT 2-D array and quantized BDCT 2-D array, respectively.

2) According to the zig-zag sequencing, select the first n AC coefficients from the un-rounded BDCT 2-D array and quantized BDCT 2-D array. Randomize them with the same secret key, and we can obtain the quantized BDCT coefficient sequence $C = (c_1, c_2, ..., c_L)$ and un-rounded BDCT coefficient sequence $C' = (c'_1, c'_2, ..., c'_L)$, respectively, where $L = n \times N_B$ is the total number of elements in C and C'.

3) Determine the parameter k of $[2^k - 1, k](k \geq 1)$ Hamming codes according to the length, l, of the secret message bit sequence $M = (m_1, m_2, ..., m_l)$, and the length, L, of quantized BDCT coefficient sequence $C = (c_1, c_2, ..., c_L)$. The parameter k should satisfy $\frac{k}{2^k-1} > \frac{l}{L}$, where l and L, as defined above, represent the length of M and C, respectively. Otherwise, some of the secret message bits cannot be embedded. Whenever necessary, k can be adjusted to meet this requirement. Generally, the maximum k that satisfies $\frac{k}{2^k-1} > \frac{l}{L}$ is selected in our method to obtain the best embedding efficiency.

4) For simplicity, we assume that the length, L, of the quantized BDCT coefficient sequence length is a multiple of $2^k - 1$ and message bit length l is a multiple of k (In practice, we can append some redundant bits to M, or neglect some surplus elements in C and C' to ensure this assumption). Divide the secret message bit sequence M into segments of length k, which are represented as

$$M_i = (m_{k(i-1)+1}, m_{k(i-1)+2}, ..., m_{ki})(i = 1, 2, ..., l/k) \tag{1}$$

The coefficient sequence C and C' are divided into segments of length $2^k - 1$, which are represented as

$$C_i = (c_{(2^k-1)(i-1)+1}, c_{(2^k-1)(i-1)+2}, ..., c_{(2^k-1)i})(i = 1, 2, ..., L/(2^k - 1) \tag{2}$$

and

$$C'_i = (c'_{(2^k-1)(i-1)+1}, c'_{(2^k-1)(i-1)+2}, ..., c'_{(2^k-1)i})(i = 1, 2, ..., L/(2^k - 1) \tag{3}$$

5) Via matrix embedding, embed each k message bits (in one segment) into the corresponding quantized BDCT coefficient segment (with the length of $2^k - 1$) sequentially. Let H be the parity check matrix of $[2^k - 1, k](k \geq 1)$ Hamming codes, which is shown below.

$$H = \begin{bmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,2^k-1} \\ \vdots & \vdots & \vdots & \vdots \\ h_{k-1,1} & h_{k-1,2} & \cdots & h_{k-1,2^k-1} \\ h_{k,1} & h_{k,2} & \cdots & h_{k,2^k-1} \end{bmatrix} \tag{4}$$

All the elements in H are either binary bit "0" or "1", and each binary column vector $h_q = (h_{1,q}, h_{2,q}, ..., h_{k,q})(1 \leq q \leq 2^k - 1)$ corresponds to the decimal value

q. Before message embedding, we need to compute the LSB values for all elements in each separated quantized BDCT coefficient segment $C_i (i = 1, 2, ..., L/(2^k-1))$. The LSB segment is represented as

$$B_i = (b_{(2^k-1)(i-1)+1}, b_{(2^k-1)(i-1)+2}, ..., b_{(2^k-1)i}) (i = 1, 2, ..., L/(2^k - 1)) \quad (5)$$

For any LSB segment B_i and the parity check matrix H, we can compute the syndrome X_i according to

$$X_i = (H \cdot {B_i}^T) \mod 2 \quad (6)$$

where ${B_i}^T$ is the transpose of B_i . For the message bit segment M_i to be embedded, the location of the element that needs to be changed in segment C_i is computed as

$$P_i = binvec2dec(X_i \bigoplus {M_i}^T) \quad (7)$$

where ${M_i}^T$ is the transpose of M_i, $\bigoplus$ represents the bitwise exclusive or operation, and *binvec2dec* is the function that converts the binary vector to the equivalent decimal number. Note that if $P_i = 0$, no change is needed for any element in segment C_i, and if $P_i \neq 0$, through switching the LSB of P_i-th element in coefficient segment C_i, the k secret message bits in M_i can be embedded. The LSB of P_i-th($P_i \neq 0$) element in C_i is switched in the following way.

$$s_{(2^k-1)(i-1)+P_i} = \begin{cases} c_{(2^k-1)(i-1)+P_i} + 1 & \text{if } c'_{(2^k-1)(i-1)+P_i} \geq c_{(2^k-1)(i-1)+P_i} \\ c_{(2^k-1)(i-1)+P_i} - 1 & \text{if } c'_{(2^k-1)(i-1)+P_i} \leq c_{(2^k-1)(i-1)+P_i} \end{cases} \quad (8)$$

where the un-rounded BDCT coefficient $c'_{(2^k-1)(i-1)+P_i}$ is utilized as the guidance to determine how to modify the P_i-th element in C_i, and $s_{(2^k-1)(i-1)+P_i}$ is the P_i-th element in the modified BDCT coefficient segment S_i. As seen, when $P_i = 0$, there is no difference between C_i and S_i, and when $P_i \neq 0$, all the elements in C_i and S_i are the same expect the P_i-th element. After all the secret message bits having been embedded, we can get the modified coefficient segments $S_1, S_2, ..., S_{L/(2^k-1)}$, which are represented as

$$S_i = (s_{(2^k-1)(i-1)+1}, s_{(2^k-1)(i-1)+2}, ..., s_{(2^k-1)i}) (i = 1, 2, ..., L/(2^k - 1)) \quad (9)$$

Concatenate all these modified coefficient segments, and we can get the quantized BDCT coefficient sequence S(corresponding to the quantized BDCT coefficient sequence C).

6) Relocate the modified quantized BDCT coefficient sequence S to its original location in the quantized BDCT 2-D array, and continue the JPEG compression (entropy encode etc.) to generate the stego image.

It is noted that when implementing our embedding algorithm, the parameter k and the secret message length l can be transmitted secretly between the

transmitter and receiver, or embedded into the carrier as side information. If k and l are embedded as the side information, they should be extracted first in the receiving end, and then all the secret message bits can be extracted correctly.

2.3 Message Extraction

To extract the embedded message bits, we need to read the quantized BDCT coefficients belonging to the first n frequencies in the stego image first. Then use the same key as in the embedding procedure to obtain the stego coefficient sequence S and divide it into segments of length $2^k - 1$ to get the coefficient segments $S_1, S_2, ..., S_{L/(2^k-1)}$. Suppose the LSB segment corresponding to S_i is represented by ${B_i}^S$, the embedded information can be extracted as

$$ {M_i}^S = (H \cdot ({B_i}^S)^T) \mod 2 \tag{10} $$

where $({B_i}^S)^T$ is the transpose of ${B_i}^S$, and ${M_i}^S$ is the secret message bits that is extracted from this segment. It can be proved [27] that the extracted ${M_i}^S$ will be equal to the embedded M_i if no attack has been made to the stego image.

3 Experimental Results and Analysis

In this section, experimental results and analysis are presented to demonstrate the performance of our proposed JPEG steganographic scheme, including small distortion, histogram preservation, high embedding efficiency and high security performance. The test image set consists of 5000 uncompressed images. Among them, 2631 images were taken by members of our group in different scenario with different cameras, 1543 images were downloaded from NRCS [28], and the remaining 1096 images are from CorelDraw image data set [29]. All the 5000 images are central-cropped into the size of 512×512. According to the zig-zag order, we select the coefficients belonging to the first 18 AC frequencies for data hiding, and the generated stego images are with the quality factor 80. The secret message bits are randomly generated, and the embedding rates are represented in terms of *bpnc* (bits per non-zero quantized BDCT coefficients) values. For comparison, we also applied MME2 and MME3 with the same embedding rate to generate stego images in our experiments. Note that these three JPEG steganographic schemes all adopted Hamming codes to improve the embedding efficiency.

3.1 Small Distortion

The PSNR of stego image verses cover image is an indication that how many pixel values of the cover image have been changed and how big the changes are due to data embedding. In Table 1, the average values of peak signal to noise ratio (PSNR) between the cover image (i.e., the JPEG image without data hiding) and its corresponding stego image are presented. The data with underline represent that the highest PSNR values having been achieved. It is

Table 1. The PSNR values of different steganographic schemes with different embedding rates

bpnc	MME2	MME3	MSS(proposed)
0.05	53.62	52.55	56.85
0.10	49.79	49.14	53.16
0.15	47.86	47.34	50.89
0.20	46.21	45.91	49.25

observed from Table 1 that in each case, less distortion has been left by the proposed steganographic scheme, which is represented by MSS (Mode-Selective Steganography). The higher PSNR achieved by our proposed MSS implies the proposed method causes smaller changes on the cover image during the data embedding.

3.2 Histogram Preservation

The preservation of the quantized BDCT coefficient histogram is another important property that can be utilized to evaluate the security performance of the JPEG steganographic schemes. For example, in most of today's JPEG steganalyzers [9, 13, 15, 17, 18], some of the features for classification are extracted from the quantized BDCT coefficient histogram directly. In this section, some experimental results are given to demonstrate that our new proposed JPEG steganographic scheme can preserve the histogram of quantized BDCT coefficients better than the other JPEG steganographic schemes such as MME2 and MME3. Let ${h_c}^{i,j}(d), {h_s}^{i,j}(d)(i,j = 1,2,...,8, d \in Z)$ denote the number of quantized BDCT coefficients of value d in mode (i,j) of cover image and its corresponding stego image, respectively. The number of un-rounded BDCT coefficients in the cover image belonging to the interval $[d-0.5,d)$ and $[d,d+0.5)$ are represented by $h_{c^-}^{i,j}(d)$ and $h_{c^+}^{i,j}(d)$, respectively. Note that $h_c^{i,j}(d) = h_{c^-}^{i,j}(d) + h_{c^+}^{i,j}(d)$ and $h_{c^-}^{i,j}(d) = h_{c^+}^{i,j}(d)$ in general. Without loss of generality, we assume that the message to be embedded is encrypted first, and the bits "0" and "1" are uniformly distributed. Suppose the embedding rate (i.e., the *bpnc* value) is $p(0 \le p \le 1)$. Then without matrix embedding, the probabilities that a quantized BDCT coefficient with value d in the cover image (whose corresponding un-rounded BDCT coefficient value is in the interval $[d-0.5,d+0.5)$) will be changed to $d-1$ and $d+1$ are equal to $p/4$, and the probability that it will be preserved is $1-p/2$. Via the utilization of matrix embedding, the embedding efficiency will be improved to $k2^k/(2^k-1)$ bits per change (note that without matrix embedding the embedding efficiency is 2 bits per change) of non-zero quantized BDCT coefficients when $[2^k-1,k](k \ge 1)$ Hamming codes is adopted [16]. Thus for the quantized BDCT coefficient with value of d, the probabilities that it will be changed to $d-1$ and $d+1$ are equal to $\frac{p}{2}(\frac{2^k-1}{k2^k})$, and the probability that it will be preserved is $1-\frac{(2^k-1)p}{k2^k}$.

As seen, the effect of the proposed algorithm on the selected quantized BDCT coefficient histogram is identical to filtering it with a low-pass filter. It is noted that in our method, only the coefficients belonging to the relatively low frequencies are selected as the data hiding band. In these quantized BDCT coefficient histograms of individual modes, the peak value ${h_c}^{i,j}(0)$ is not much bigger than ${h_c}^{i,j}(1)$ and ${h_c}^{i,j}(-1)$ in general. Thus the aforementioned low-pass filtering effect will not change ${h_c}^{i,j}(0)$ much, and the quantized BDCT coefficient histogram can be well preserved in our method. Let us define the distortion that has been introduced into quantized BDCT coefficient histogram as

$$D = \sum_i \sum_j \sum_d |{h_c}^{i,j}(d) - {h_s}^{i,j}(d)| \quad i,j = 1,2,...,8, d \in Z \tag{11}$$

The average values of D for all the aforementioned 5000 images are exemplified in Table 2, where the data with underline represent the least distortion having been introduced. It is observed from Table 2 that the proposed MSS can preserve the quantized BDCT coefficient histogram better than the other steganographic schemes MME2 and MME3.

Table 2. The distortion introduced into the BDCT coefficient histogram for different steganographic schemes with different embedding rates

bpnc	MME2	MME3	MSS(proposed)
0.05	583.3	670.8	429.5
0.10	990.1	1064.6	815.3
0.15	1344.4	1407.4	1229.0
0.20	1837.5	1834.7	1675.0

3.3 Embedding Efficiency

With the usage of $[2^k - 1, k](k \geq 1)$ Hamming codes, we can embed k message bits into $2^k - 1$ quantized BDCT coefficients by changing at most one of the quantized BDCT coefficients. That is, the larger the k, the smaller distortion will be introduced, and the matrix embedding will be conducted more efficiently. The parameter k of Hamming codes are determined by the length of secret message bit sequence $M = (m_1, m_2, ..., m_l)$, and the length of quantized BDCT coefficient sequence $C = (c_1, c_2, ..., c_L)$. As seen, if the message length to be embedded is determined, the only determination factor of k is the length of the quantized BDCT coefficient sequence C that can be utilized for data hiding. Different from MME2 and MME3, in our proposed method, only the coefficients belonging to the relatively low frequencies are selected for embedding. It looks as if that the number of coefficients chosen for embedding is less than that of MME2 and MME3 (Note that in MME2 and MME3, only the non-zero coefficients can be utilized for data hiding). However, since in general the number of zero coefficients belonging to the relatively low frequencies is more than the non-zero coefficients belonging to the relatively high frequencies (e.g., when the first 18

Table 3. The number of coefficients that can be utilized for different steganographic schemes

MME2	MME3	MSS(proposed)
67,408	67,408	73,728

AC frequencies are considered as the relatively low frequencies), when all the coefficients (including the numerous zero coefficients) belonging to the relatively low frequencies are selected for embedding, more coefficients can be utilized in our method. In Table 3, the number of quantized BDCT coefficients that can be utilized for different steganographic schemes are illustrated.

3.4 Security Performance

The security performance of MME2, MME3 and our proposed MSS are tested against three state-of-the-art universal JPEG steganalyzers presented in [17, 21, 23], denoted by ClbJFMP-274 [17], MP-486 [21] and ClbMP-324 [23], respectively, where the numbers 274, 486 and 324 denote the total number of features utilized, Clb stands for calibration technique [12, 15], JF stands JPEG features, and MP for Markov process based features. To the best of our knowledge, these three steganalyzers are among the most effective universal JPEG steganalyzers in detecting today's JPEG steganographic schemes. In our experiments, the LibSVM [30] is adopted as the classifier. For every classifier, the randomly selected 3/4 cover images and the corresponding 3/4 stego images are trained with the second polynomial kernel. The remaining 1/4 cover and 1/4 stego images are used for test. The TNR, TPR represent the true negative rate and true positive rate respectively, and AR (AR = (TNR+TPR)/2) represents the accuracy rate. To eliminate the randomness effect caused by image selection, we individually conduct each random experiment 10 times. Results reported in the next are the average of these 10 experiments.

The detection results of MME2, MME3 and MSS against the aforementioned three universal JPEG steganalyzers are shown in Table 4. The data with underline represent that the least detection accuracy rates obtained by the four steganographic schemes with the same embedding rate. It is observed from Table 4 that our proposed MSS has the best security performance except in the cases while the embedding rate is 0.05 *bpnc* and MP-486, ClbJFMP-274 are selected as the detectors. Note that when the embedding rate is 0.05 *bpnc*, the detection accuracy rates of MME2, MME3 and MSS against all the aforementioned staganalyzers are near random guessing. When the embedding rate is no less than 0.10 *bpnc*, the detection accuracy rates corresponding to MSS are much smaller than that to the other three steganographic schemes. Note that MME2 and MME3 are among the most secure steganographic schemes that can resist today's universal JPEG steganalyzers [16]. The experiments results illustrated in Table 4 have demonstrated the high security performance of our proposed method.

Table 4. The detection results of MME2, MME3 and MSS against different universal JPEG steganalyzers (TNR, TPR and AR denote the true negative rate, true positive rate, and accuracy rate respectively. The data with underline represent that the least detection accuracy rates of MME2, MME3 and MSS against MP-486, ClbJFMP-274 and ClbMP-324).

		MME2			MME3			MSS(proposed)		
	bpnc	TNR	TPR	AR	TNR	TPR	AR	TNR	TPR	AR
ClbJFMP-274 [17]	0.05	52.5	52.9	52.7	51.3	51.6	51.5	54.0	53.5	53.8
	0.10	66.6	67.9	67.3	63.3	64.1	63.7	59.9	59.3	59.6
	0.15	81.2	79.1	80.2	77.4	76.7	77.1	64.9	64.6	64.8
	0.20	96.5	96.7	96.6	94.7	95.3	95.0	70.1	68.7	69.4
MP-486 [21]	0.05	52.3	52.2	52.3	52.1	51.0	51.6	52.3	53.7	53.0
	0.10	63.9	64.7	64.3	61.1	62.3	61.7	56.6	56.8	56.7
	0.15	76.7	77.2	77.0	73.8	73.6	73.7	61.9	60.7	61.3
	0.20	95.8	95.7	95.8	93.4	93.9	93.7	66.3	65.8	66.1
ClbMP-324 [23]	0.05	52.1	52.3	52.2	50.6	51.8	51.2	51.0	51.3	51.2
	0.10	64.3	65.6	65.0	62.2	62.0	62.1	53.9	52.3	53.1
	0.15	77.8	77.9	77.9	74.1	75.2	74.7	55.7	54.5	55.1
	0.20	96.4	96.8	96.6	94.7	94.7	94.7	57.2	57.7	57.5

4 Conclusion

In this paper, we have presented a new JPEG steganographic scheme. Via three measures, some excellent performance can be obtained by our method. The conclusion is as follows.

- In our algorithm, the secret message bits are only embedded into quantized BDCT coefficients belonging to the relatively low frequencies. Compared with other JPEG steganographic scheme in which the secret message bits are spread into all frequency coefficients, less distortion are introduced in the spatial domain and the histogram of quantized BDCT coefficients will be preserved better.
- More coefficients (including the zero coefficients in the selected frequencies) can be utilized for data hiding and the matrix embedding will be conducted more efficiently.
- Our new JPEG steganography has a high security performance. It can resist the most powerful universal JPEG steganalyzers effectively.

Acknowledgements

This work was supported by 973 Program of China (Grant No. 2011CB302204), the National Natural Science Foundation of China (Grant No. 60633030), and the Research Fund for the Doctoral Program of Higher Education of China (Grant No. 20070558054). The work was mostly conducted when the first author visited New Jersey Institute of Technology, Newark, New Jersey, USA.

References

1. http://zooid.org/~paul/crypto/jsteg/
2. JP Hide & Seek, http://linux01.gwdg.de/~alatham/stego.html
3. Provos, N.: Defending against statistical steganalysis. In: 10th USENIX Security Symposium, Washington DC, USA (2001)
4. Westfeld, A.: High capacity despite better steganalysis (F5-a steganographic algorithm). In: Moskowitz, I.S. (ed.) IH 2001. LNCS, vol. 2137, pp. 289–302. Springer, Heidelberg (2001)
5. Sallee, P.: Model based methods for steganography and steganalysis. International Journal of Image Graphics 5(1), 167–190 (2005)
6. Fridrich, J., Goljan, M., Soukal, D.: Perturbed quantization steganography with wet paper codes. In: Proc. the ACM Workshop on Multimedia and Security, Magdeburg, Germany, September 20-21, pp. 4–15 (2004)
7. Kim, Y., 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)
8. 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)
9. Westfeld, A., Pfitzmann, A.: Attacks on steganographic systems. In: Pfitzmann, A. (ed.) IH 1999. LNCS, vol. 1768, pp. 61–75. Springer, Heidelberg (2000)
10. Provos, N., Honeyman, P.: Detecting steganographic content on the Internet. CITI Technical Report 01-11 (August 2001)
11. Provos, N., Honeyman, P.: Hide and seek: an introduction to steganography. In: IEEE Security and Privacy, May/June, pp. 32–44 (2003)
12. Fridrich, J., Goljan, M., Du, R.: Attacking the OutGuess. In: Proc. the ACM Workshop on Multimedia and Security, Juan-les-Pins, France, December 6, pp. 3–6 (2002)
13. Böhme, R., Westfeld, A.: Breaking Cauchy model-based JPEG steganography with first order statistics. In: Samarati, P., Ryan, P.Y.A., Gollmann, D., Molva, R. (eds.) ESORICS 2004. LNCS, vol. 3193, pp. 125–140. Springer, Heidelberg (2004)
14. Crandall, R.: Some notes on steganography. Steganography Mailing List (1998), http://os.inf.tu-dresden.de/westfeld/crandall.pdf
15. 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)
16. Fridrich, J., Pevny, T., Kodovsky, J.: Statistically Undetectable JPEG Steganography: Dead Ends, Challenges, and Opportunities. In: Proc. the ACM Workshop on Multimedia and Security, Dallas, TX, September 20-21, pp. 3–14 (2007)
17. Pevny, T., Fridrich, J.: Merging Markov and DCT features for multi-class JPEG steganalysis. In: Proc. SPIE Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents, vol. 6505, pp. 650503.1–650503.13 (2007)
18. Fridrich, J.: Feature-based steganalysis for JPEG images and its implications for future design of steganographic schemes. In: Fridrich, J. (ed.) IH 2004. LNCS, vol. 3200, pp. 67–81. Springer, Heidelberg (2004)
19. Fu, D., Shi, Y.Q., Zou, D., Xuan, G.: JPEG steganalysis using empirical transition matrix in blick DCT domain. In: Proc. IEEE International Workshop on Multimedia Signal Processing, Victoria, BC, Canada (2006)

20. 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)
21. Chen, C., Shi, Y.Q.: JPEG image steganalysis utilizing both intrablock and interblock correlations. In: Proc. IEEE International Symposium on Circuits and Systems, Seattle, WA, May 18-21 (2008)
22. Huang, F., Li, B., Huang, J.: Universal JPEG steganalysis based on microscopic and macroscopic calibration. In: Proc. IEEE International Conference on Image Processing, San Diego, California, U.S.A, October 12-15 (2008)
23. Huang, F., Huang, J.: Calibration based JPEG steganalysis. Science in China Series F: Information Sciences 52(2), 260–268 (2009)
24. Huang, F., Shi, Y.Q., Huang, J.: A study on the security performance of YASS. In: Proc. IEEE International Conference on Image Processing, San Diego, California, U.S.A, October 12-15 (2008)
25. Huang, F., Huang, J., Shi, Y.Q.: An experimental study on the security performance of YASS. IEEE Trans. Informaiton Forensics and Security 5(3), 374–380 (2010)
26. Li, B., Huang, J., Shi, Y.Q.: Steganalysis of YASS. IEEE Trans. Information Forensics and Security 4(3), 369–382 (2009)
27. Moon, T.K.: Error correction coding. Mathematical methods and algorithms. John Wiley & Sons, Inc., Hoboken (2005)
28. NRCS Photo Gallery, http://photogallery.nrcs.usda.gov
29. CorelDraw Image CD, http://www.corel.com
30. Chang, C.-C., Lin, C.-J.: LIBSVM: a library for support vector machines (2001), http://www.csie.ntu.edu.tw/~cjlin/libsvm

Ternary Data Hiding Technique for JPEG Steganography

Vasily Sachnev and Hyoung-Joong Kim

Center of Information Security Technologies,
Graduate School of Information Security and Management,
Korea University, Seoul 136-701, Korea
bassvasys@hotmail.com, khj-@korea.ac.kr

Abstract. In this paper we present JPEG steganography method based on hiding data to the stream of ternary coefficients. In the proposed method each nonzero DCT coefficient is converted to the corresponding ternary coefficient. The block of $3^m - 1$ ternary coefficients is used for hiding m ternary messages by modifying one or two coefficients. Due to higher information density of the ternary coefficients, the proposed method has many solutions for hiding necessary data. Such a big choice enables to choose coefficients with lowest distortion impact. As a result, the proposed methods have better data hiding performance compared to the existing steganographic methods based on hiding data to stream of binary coefficients like matrix encoding (F5) and modified matrix encoding (MME). The proposed methods were tested with steganalysis method proposed by T. Pevny and J.Fridrich. The experimental results show that the proposed method has less detectability compared to MME (modified matrix encoding).

1 Introduction

Steganography is a group of data hiding techniques, which hides data in undetectable way. Here, the features extracted from modified images (stegos) and original images have to be statistically undistinguishable. Steganography enables secure undetectable communication by sending several modified and original images.

In general, steganography can be used in many areas for different host signals (i.e. images, audio, video, text). Among them the digital compressed images (JPEG) is the most popular. Hence, the developing of JPEG steganography methods becomes an important research direction in the steganography area.

The one of the first steganography method for JPEG images was JSteg. This method utilizes very popular LSB substitution technique for hiding data to the quantized DCT coefficients. JSteg significantly distorts the histogram of the DCT coefficients. As result, this method can be easily detected by estimating the shape of the histogram of modified DCT coefficients. The next generation of the JPEG steganography methods tried to remove drawback of the JSteg and keeps the histogram of the DCT coefficients just slightly modified [13], [18].

H.-J. Kim, Y. Shi, and M. Barni (Eds.): IWDW 2010, LNCS 6526, pp. 202–210, 2011.

Provos [13] keeps histogram of the DCT coefficients by compensating distortion caused after data hiding. He divided the set of the DCT coefficients into two disjoint subsets. The first subset is used for hiding data. The second subset is used for compensating histogram's changes after data hiding to the first subset. As result, the histogram of the DCT coefficients after data hiding has the same shape as original histogram. Methods presented in [2] and [11] use a similar approach.

Another way to decrease detectability of the steganography methods is reducing total distortion caused after data hiding. Normally one DCT coefficient is used for hiding one bit of data. Westfeld [18] increased the efficiency of embedding by using the matrix encoding. The key idea of his method is to hide n bits by changing one coefficient among m. The distortion after data hiding using the matrix encoding may significantly decreased.

Later Kim et. al [9] improved the performance of the matrix encoding by modifying coefficients with less distortion impact. In fact, the proposed modified matrix encoding method (MME) modified more coefficients compared to the matrix encoding. They show that the distortion after modifying one coefficients 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. Note that Kim et. al method requires the original bitmap image for data hiding.

Solanki et. al. [17] utilized the robust watermarking scheme for steganography purposes. They embed data to image in spatial domain by using method robust against JPEG compression. Their scheme provides less degradation of the features of DCT coefficients, and, as result, less detectability.

In this paper we improved the existing steganographic data hiding methods by replacing the binary computation to ternary. Data hiding method based on ternary computation operates with ternary coefficients. Due to higher information density of the ternary coefficients, the proposed method has larger number of possible solutions (i.e., hiding the same data by modifying different coefficients). Steganographic method with larger number of possible solutions may always choose better solution with minimum distortion impact and, finally, may better survive against powerful steganalysis.

The paper is organized as follows. The section 2 describes the ternary embedding techniques in detail. The encoder and decoder are presented in the section 3. The section 4 provides the experimental results. Section 5 concludes the paper.

2 Ternary Embedding Techniques for JPEG Steganography

In general, JPEG steganographic methods use a stream of DCT non zero coefficients from JPEG image (see Equation 1) for data hiding purposes.

$$c = \frac{DCT(I)}{Q_t}, \qquad C = \lfloor c + 0.5 \rfloor \tag{1}$$

where $DCT(X)$ is the Digital Cosine Transform for $8\times$ 8 block of image I, c and C are the non rounded and rounded DCT coefficients; Q_t is the quantization table.

Matrix encoding, modified matrix encoding and BCH based methods convert DCT coefficients to binary numbers. Those methods hide data by modifying several DCT coefficients (i.e., $C \pm 1$), such that the corresponding binary coefficient change the value.

Note that the choice between $C+1$ and $C-1$ is not utilized well for F5(matrix encoding). Modified matrix encoding (MME) always has several solutions and may choose the best one which causes the lowest distortion. MME also chooses the best modification between $C+1$ and $C-1$. The number of possible solutions for MME N_{MME} is computed as follows:

$$N = 2 + 2 \cdot \left\lfloor \frac{n}{2} \right\rfloor, \tag{2}$$

where $n = 2^m - 1$ is the block's size. Coefficient 2 implies the choice between $C + 1$ and $C - 1$.

The proposed data hiding strategy based on ternary computation increases the number of possible solutions compared to MME. In the proposed ternary embedding JPEG coefficients are converted to the ternary coefficients (i.e., 0,1,2) as follows:

$$v_i = |C_i| \bmod 3 \tag{3}$$

Data hiding method uses vector $v = \{v_1, v_2, ..., v_n\}$ (where $n = 3^m - 1$) of ternary coefficients for hiding m ternary messages as follows:

$$\mathbf{M} = H \cdot V^T, \tag{4}$$

where $\mathbf{M} = \{M_1, M_2, ..., M_m\}$ ($M_m \in [0, 1, 2]$) is the hidden message, V is the vector of modified coefficients v, H is the parity check matrix:

for $m = 2$,

$$H = \begin{pmatrix} 0\,0\,1\,1\,1\,2\,2\,2 \\ 1\,2\,0\,1\,2\,0\,1\,2 \end{pmatrix} \tag{5}$$

Proposed method requires to find the position(s) of the coefficient(s) to be modified by plus or minus 1. We developed two algorithms for hiding data by modifying one or two coefficients.

There are two solutions for hiding data by modifying one coefficient:

First solution is $C'(j^+) = C(j^+) + 1$, where index j^+ is computed as follows:

$$j^+ = (M + 2 \cdot M_{org})_{10} \tag{6}$$

Second solution is $C'(j^-) = C(j^-) - 1$, where index j^- is computed as follows:

$$j^- = (2 \cdot M + M_{org})_{10} \tag{7}$$

Table 1. Ternary arithmetic

Multiplication: $a \cdot b$				
		b		
		0	1	2
a	0	0	0	0
	1	0	1	2
	2	0	2	1

Summation: $a + b$				
		b		
		0	1	2
a	0	0	1	2
	1	1	2	0
	2	2	0	1

where $C'(j)$ and $C(j)$ are modified and original DCT coefficients, M_{org} (see Equation 8) is the original message computed from vector v, operator $(X)_{10}$ converts ternary vector X to decimal number. Note that, the Equations 6 and 7 uses ternary arithmetic (see Table 2).

$$M_{org} = H \cdot v^T \tag{8}$$

Example:
Assume $m = 2$, $C = \{1, -1, 3, 5, -3, 1, 2, 1\}$, $M = \{0\ 2\}^T$. The stream of ternary coefficients computed from C is $v = \{1\ 1\ 0\ 2\ 0\ 1\ 2\ 1\}$. Find vector C' such that $H \cdot V^T = M$, where V is the vector of corresponding ternary coefficients from C'.

The original message is:

$$M_{org} = H \cdot v^T = \begin{pmatrix} 0\ 0\ 1\ 1\ 1\ 2\ 2\ 2 \\ 1\ 2\ 0\ 1\ 2\ 0\ 1\ 2 \end{pmatrix} \cdot \{1\ 1\ 0\ 2\ 0\ 1\ 2\ 1\}^T = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{9}$$

The index j^+ for first solution is:

$$j^+ = (M + 2 \cdot M_{org})_{10} = \left(\begin{pmatrix} 0 \\ 2 \end{pmatrix} + 2 \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right)_{10} = \begin{pmatrix} 2 \\ 2 \end{pmatrix}_{10} = 8 \tag{10}$$

The modified vectors C' and V for first solution are $C' = \{1, -1, 3, 5, -3, 1, 2, \mathbf{2}\}$ and $V = \{1\ 1\ 0\ 2\ 0\ 1\ 2\ \mathbf{2}\}$, correspondingly, and $H \cdot V^T = M$.

The index j^- for second solution is:

$$j^- = (2 \cdot M + M_{org})_{10} = \left(2 \cdot \begin{pmatrix} 0 \\ 2 \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right)_{10} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}_{10} = 4 \tag{11}$$

The modified vectors C' and V for second solution are $C' = \{1, -1, 3, \mathbf{4}, -3, 1, 2, 1\}$ and $V = \{1\ 1\ 0\ \mathbf{1}\ 0\ 1\ 2\ 1\}$, correspondingly, and $H \cdot V^T = M$.

The modified coefficients for the first and second solutions in vectors C' and V are marked by bold font.

Data hiding by modifying two coefficients requires to compute M_{org}, j^+ and j^- using Equations (8), (6), and (7). All two flip solutions can be divided into 4 groups. Each solution contains two indexes p_1 and p_2. The coefficients with

Table 2. Two flip solutions

	Flip pattern	Solutions (p_1,p_2)	Modification
Group 1	$(+1,-1)$	$(a,a+j^+)$	$C'(p_1) = sign(C(p_1))(\lvert C(p_1)\rvert + 1)$ $C'(p_2) = sign(C(p_2))(\lvert C(p_2)\rvert - 1)$
Group 2	$(-1,+1)$	$(a,a+j^-)$	$C'(p_1) = sign(C(p_1))(\lvert C(p_1)\rvert - 1)$ $C'(p_2) = sign(C(p_2))(\lvert C(p_2)\rvert + 1)$
Group 3	$(+1,+1)$	$(a,2\cdot(a+j^+))$	$C'(p_1) = sign(C(p_1))(\lvert C(p_1)\rvert + 1)$ $C'(p_2) = sign(C(p_2))(\lvert C(p_2)\rvert + 1)$
Group 4	$(-1,-1)$	$(a,2\cdot(a+j^-))$	$C'(p_1) = sign(C(p_1))(\lvert C(p_1)\rvert - 1)$ $C'(p_2) = sign(C(p_2))(\lvert C(p_2)\rvert - 1)$
$a \in [1,2,...,3^m-1]$, skip solutions $(a,0)$, and (a,a)			

indexes p_1 and p_2 have to be modified according to the group's flip pattern, i.e., (+1,-1), (-1,+1), (+1,+1), (-1,-1) (See Table 2).

Example:
Find all two flip solutions for the previous example. Knowing data: $C = \{1,-1,3,5,-3,1,2,1\}$, $M = \{0\ 2\}^T$, and $v = \{1\ 1\ 0\ 2\ 0\ 1\ 2\ 1\}$. Computed data: $M_{org} = \{0\ 1\}^T$, $j^+ = 8$, and $j^- = 4$.

Group 1:

According to equation $(a,a+j^+)$, two flip solutions are: (1,5), (2,3), (3,7), (4,8), (5,6), (6,1), (7,2). Flip pattern is $(+1,-1)$, then for solution (1,5) we have $C' = \{\mathbf{2},-1,3,5,\mathbf{-2},1,2,1\}$, $V = \{\mathbf{2}\ 1\ 0\ 2\ \mathbf{2}\ 1\ 2\ 1\}$. Verification: $H \cdot V = M$. Note that all calculations has to be done by using ternary arithmetic (see Table 2).

Group 2:

According to equation $(a,a+j^-)$, two flip solutions are: (1,6), (2,7), (3,2), (5,1), (6,5), (7,3), (8,4).

Group 3:

According to equation $(a,2\cdot(a+j^+))$, two flip solutions are: (1,7), (2,3), (3,7).

Group 4:

According to equation $(a,2\cdot(a+j^+))$, two flip solutions are: (1,3), (2,5), (6,7).

Note that, there are 22 possible one and two flip solutions for this example.

According to the equations for computing indexes p_1 and p_2 (see Table2), the number of all possible solutions for the proposed method N_{TE} can be computed as follows:

$$N_{TE} = 2 + 2\cdot(n-1) + 2\cdot(\frac{n}{2}-1) \tag{12}$$

where $(n-1)$ is the number of solutions for Group 1 or 2, $\frac{n}{2}-1$ is the number of solutions for Group 3 or 4.

Note that the number of possible solutions is larger than that for MME (i.e., $N_{MME} < N_{TE}$). Thus, the proposed method has higher flexibility to choose better solution with lowest distortion.

Distortion can be computed as follows:

$$D_{plus} = e_{plus} \cdot Q_t, \tag{13}$$

$$D_{minus} = e_{minus} \cdot Q_t, \tag{14}$$

$$e_{plus} = \begin{cases} |C| + 1 - |c|, & \text{if } C \neq -1, \\ NaN, & \text{if } C = -1, \end{cases}$$

$$e_{minus} = \begin{cases} |c| - (|C| - 1), & \text{if } C \neq 1, \\ NaN, & \text{if } C = 1, \end{cases}$$

The total distortion for each solution has to be defined as a sum of the distortions according to the corresponding flip pattern.

If $e = NaN$, such solutions have to be skipped from data hiding process.

3 Encoder and Decoder

Encoder:
For bitmap image I and binary message M process following:

1) Divide image into 8 by 8 blocks. Compute non rounded c and rounded C DCT coefficients using Equation 1.
2) Define stream of ternary coefficients using the Equation 3. Compute distortions D_{plus} and D_{minus}.
3) Convert binary message M into ternary M_t. Find maximum possible parameter m such that:

$$\frac{m}{3^m - 1} > \frac{|M_t|}{N}, \tag{15}$$

where N is the number on non zero rounded DCT coefficients C.

4) Divide stream of computed ternary coefficients into blocks of $n = 3^m - 1$ coefficients.
5) Hide data to each block
   5.1) Compute all one and two flip solutions using guidelines from the section 2.
   5.2 Using the corresponding flip pattern and distortion D_{plus} and D_{minus}, choose solution which causes the lowest distortion.
   5.3 According to the flip pattern of the chosen solution, modify the corresponding DCT coefficients.
6) Using the stream of the modified DCT coefficients build a stego image I_{stego}.

Decoder:
For stego image I_{stego} and parameter m process following:

1) Get the stream of modified DCT coefficients C' from the stego image I_{stego}.
2) Define stream of ternary coefficients from the C'. Divide stream of ternary coefficients to blocks of $3^m - 1$ coefficients.
3) Recover the hidden message from each block using the Equation 4.
4) Convert ternary hidden message to binary.

4 Experimental Results

Proposed method and examined modified matrix encoding (MME) were tested by powerful steganalysis algorithm proposed by T. Pevny and J. Fridrich [12]. It uses 274 different features of the DCT coefficients and allows deeply investigate artificial changes in the tested images. The union of the 274 features from the original and modified images uses for making a model in support vector machine (SVM).

The set of the 1174 test images distributed by CorelDraw was used in our experiments. The proposed and examined method were tested for 4 different payloads (5, 10, 15, 20 bit per coefficient, or bpc) and quality factor 75. We used model adapted to quality factor 75 and tested the 4 sets of test images for each examined payload. Each set of test images had 587 cover and 587 stego images. The result shows the accuracy of the steganalysis (Error probability) for each set of the stego images (see Figures 1).

The error probability e is computed as follows:

$$e = \frac{1}{2}(P_a + P_b), \tag{16}$$

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).

The results show relatively high detectability for MME for all examined payloads. For small payloads (0.05 - 0.1 bpc) MME has 0.46 and 0.32 in terms of error probability points. Such results may keep necessary protection against steganalysis. For bigger payloads the performance of the MME is dramatically decreased. For payloads 0.15 - 0.2 bpc MME shows 0.19 and 0.045 error probability points, respectively. High detectability rate (error probability is closed to 0) means a successful stego image detection.

The proposed method shows improvement over the MME in terms of error probability points for all tested payloads. The proposed ternary embedding method has at 0.02, 0.16, 0.13, and 0.1 higher achievement in terms of error probability points for payloads 0.05, 0.1, 0.15, and 0.2 bpc, respectively. Steganographic methods with higher error probability may successfully survive against steganalysis for larger payloads. The proposed method has acceptable detectability rate even for payloads 0.1 - 0.15 bpc, whereas MME can not successfully survive for such payloads. The performance of the proposed method is decreased for large payloads. The error probability for payload 0.2 bpc is 0.15, which results unacceptable high detectability rate.

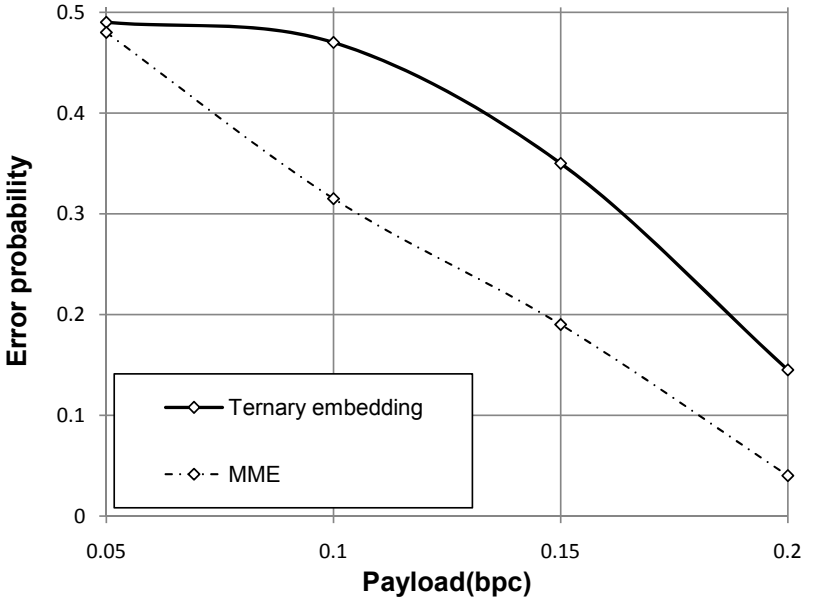


Fig. 1. Steganalysis accuracy for sets of images with quality factor 75

5 Conclusion

In this paper we present an improved data hiding technique for steganography. Compared to the existing methods, the presented data hiding method uses the ternary coefficients instead of binary. The proposed data hiding method can easily get all one and two flip solutions. The proposed ternary embedding provides larger number of possible solutions, and, as a results, shows much lower detectability for powerful steganalysis. The experiments shows that the proposed method is always better than the examined MME. Thus, shift from binary computation to ternary may be a new promising direction for improving existing data hiding methods for steganography.

Acknowledgment

This work was in part supported by Information Technology Research Center (ITRC), Korea University.

References

1. Chandramouli, R., Kharrazi, M., Memon, N.D.: Image steganography and steganalysis: Concepts and practice. In: Kalker, T., Cox, I., Ro, Y.M. (eds.) IWDW 2003. LNCS, vol. 2939, pp. 35–49. Springer, Heidelberg (2004)

2. Eggers, J., Bauml, R., Girod, B.: A communications approach to steganography. In: SPIE, Electronic Imaging, Security, and Watermarking of Multimedia Contents, San Jose, CA (2002)
3. Farid, H., Lyu, S.: Steganalysis using higher-order image statistics. IEEE Transactions on Information Forensics and Security, 111–119 (2006)
4. Farid, H., Siwei, L.: Detecting hidden messages using higher-order statistics and support vector machines. In: Petitcolas, F.A.P. (ed.) IH 2002. LNCS, vol. 2578, Springer, Heidelberg (2003)
5. Fridrich, J.: Minimizing the embedding impact in steganography. In: Proceedings ACM Multimedia and Security Workshop, Geneva, Switzerland, pp. 2–10 (2006)
6. 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)
7. Fridrich, J., Filler, T.: Practical methods for minimizing embedding impact in steganography. In: Proceedings SPIE Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX, San Jose, CA, vol. 6505, pp. 2–3 (2007)
8. Hetzl, S., Mutzel, P.: A graph–theoretic approach to steganography. In: Dittmann, J., Katzenbeisser, S., Uhl, A. (eds.) CMS 2005. LNCS, vol. 3677, pp. 119–128. Springer, Heidelberg (2005)
9. 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)
10. Lee, K., Westfeld, A.: Generalised category attack—improving histogram-based attack on JPEG LSB embedding. In: Furon, T., Cayre, F., Doërr, G., Bas, P. (eds.) IH 2007. LNCS, vol. 4567, pp. 378–391. Springer, Heidelberg (2008)
11. Noda, H., Niimi, M., Kawaguchi, E.: Application of QIM with dead zone for histogram preserving JPEG steeganography. In: Processing ICIP, Genova, Italy (2005)
12. Pevny, T., Fridrich, J.: Merging Markov and DCT features for multi-class JPEG steganalysis. SPIE, San Jose (2007)
13. Provos, N.: Defending against statistical steganalysis. In: 10th USENIX Security Symposium, Washington, DC (2001)
14. Sachnev, V., Kim, H.J., Zhang, R., Choi, Y.S.: A Novel Approach for JPEG steganography. In: Kim, H.-J., Katzenbeisser, S., Ho, A.T.S. (eds.) IWDW 2008. LNCS, vol. 5450, pp. 209–217. Springer, Heidelberg (2009)
15. 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)
16. Sallee, P.: Model-based methods for steganography and steganalysis. International Journal of Image Graphics, 167–190 (2005)
17. 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)
18. Westfeld, A.: High capacity despite better steganalysis (F5—a steganographic algorithm). In: Moskowitz, I.S. (ed.) IH 2001. LNCS, vol. 2137, pp. 289–302. Springer, Heidelberg (2001)
19. 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)

Interleaving Embedding Scheme for ECC-Based Multimedia Fingerprinting

Xuping Zheng[1], Aixin Zhang[2], Shenghong Li[1], Bo Jin[3], and Junhua Tang[2]

[1] Department of Electronic Engineering,
Shanghai Jiao Tong University, Shanghai, P.R. China
xupingzheng@gmail.com, shli@sjtu.edu.cn
[2] School of Information Security Engineering,
Shanghai Jiao Tong University, Shanghai, P.R. China
axzhang@sjtu.edu.cn, junhuatang@sjtu.edu.cn
[3] The Third Research Institute of Ministry of Public Security,
Shanghai, P.R. China
jinbo@mail.trimps.org.cn

Abstract. In this paper, we focus on improving collusion resistance of error correcting code (ECC) based multimedia fingerprinting. Although permuted subsegment embedding (PSE) scheme has provided better interleaving collusion resistance than conventional scheme, our study shows that the resistance is still weaker than that under averaging collusion at moderate-to-high watermark-to-noise ratio (WNR). We then propose interleaving embedding (ILE) scheme to enforce interleaving collusion resistance, where user's original fingerprint is applied block interleaving before embedded to the host signal. Simulation results show that ILE scheme can resist more colluders' interleaving collusion than PSE scheme at moderate-to-high WNR. Theoretical analysis and experimental results demonstrate that the performance of ECC-based fingerprinting with ILE scheme under interleaving collusion is comparable to that under averaging collusion.

Keywords: collusion resistance, error correcting code(ECC), interleaving embedding, multimedia fingerprinting.

1 Introduction

Digital fingerprinting is an effective technique for copyright protection. In multimedia fingerprinting system, a unique signal known as fingerprint represents a subscriber. Before distributed to the subscriber, the copy will be embedded with his/her corresponding fingerprint. Thus, the user who illegally distributes the authorized copy can be traced with the help of the fingerprint. In this way, digital representations of multimedia content can be protected from piracy. However, illegal users may carry out attacks against the fingerprints to minimize the chance for them to be detected. Collusion is such a kind of attack, where a group of dishonest users (colluders) use their copies to forge an illegal copy. Colluders will try their best to make their fingerprints in that copy removed as much as

H.-J. Kim, Y. Shi, and M. Barni (Eds.): IWDW 2010, LNCS 6526, pp. 211–223, 2011.

possible [1]. Therefore, it is essential to design fingerprinting system with strong collusion resistance. Typical fingerprinting schemes developed to combat collusion attacks include [2,3,4,5,6,7]. Among them, many schemes utilize Gaussian spread spectrum sequences to construct collusion resistant fingerprints [2,3,4,5]. Fingerprinting schemes based on Gaussian signal can be further grouped into two classes: orthogonal fingerprinting and coded fingerprinting. In the orthogonal approach, the fingerprint for each user is a Gaussian spread spectrum sequence, and one user's fingerprint is orthogonal to any other's [2]. The coded fingerprinting schemes employ a two-layer structure with a code layer and an embedding layer [3,4,5]. On the code layer, each user is assigned a codeword. The code is constructed over a finite alphabet and has a strong ability to trace colluders. On the embedding layer, all the symbols of user's codeword are mapped to Gaussian sequences in a predefined manner to derive the fingerprint, and then the fingerprint will be embedded into the original copy.

Error correcting code (ECC) based fingerprinting is a typical coded fingerprinting. Compared with orthogonal fingerprinting, ECC-based fingerprinting has much higher colluder detection efficiency, but its collusion resistance under interleaving collusion is much weaker than that under averaging collusion [4]. By jointly considering the code layer and embedding layer, permuted subsegment embedding (PSE) technique was proposed in [4] to improve the interleaving collusion resistance of ECC-based fingerprinting. Combining PSE technique and trimming detection algorithm, ECC-based fingerprinting was used to design a video fingerprinting system accommodating 10 million users [5].

Although the PSE technique has improved interleaving collusion resistance of ECC-based fingerprinting, the performance is still not comparable to averaging collusion resistance, especially at moderate-to-high watermark-to-noise ratio (WNR). In this paper, we explore a new approach called interleaving embedding (ILE) scheme to enforce interleaving collusion resistance of ECC-based fingerprinting, where original fingerprint is interleaved before embedding. Simulation results show that ECC-based fingerprinting with ILE scheme can resist much more colluders' interleaving collusion than that with PSE scheme at moderate-to-high WNR. Both theoretical analysis and simulation results show that collusion resistance of ECC-based fingerprinting with ILE under interleaving attack is comparable to that under averaging attack in a wide range of WNR and number of colluders.

2 ECC-Based Multimedia Fingerprinting

2.1 Fingerprint Construction and Embedding

The basic design of ECC-based fingerprinting scheme [4,5] can be summarized as follows. First, the ECC should be determined. In [4] and [5], the ECC considered is Reed-Solomon code $\mathrm{RS}(L, k)$ over Galois Field $\mathrm{GF}(q)$, where L is the code length and k is the dimension. $\mathrm{RS}(L, k)$ can accommodate $N_u = q^k$ users at the most. Next, spread spectrum sequences $\mathbf{p}_i$ $(i = 1, 2, \ldots, q)$ representing different symbols in $\mathrm{GF}(q)$ should be generated. For a host signal $\mathbf{x}$ containing

N_s embeddable components, the length of each spread spectrum sequence is $N_p = N_s/L$. All the sequences follow independent identically distributed (i.i.d.) Gaussian distribution $N(0, \sigma_s^2)$, and are mutually orthogonal with equal energy $\|\mathbf{p}\|^2$. User j's fingerprint $\mathbf{s}_j$ can be derived by mapping all the symbols of his assigned RS codeword to the corresponding spread spectrum sequences and then concatenating them. Different users' fingerprint sequences share the same energy $\|\mathbf{s}\|^2$. To achieve good robustness and fidelity, the embedding is performed additively in transform domain using watermarking techniques [2,8]. User j's fingerprinted copy $\mathbf{y}_j$ can be modeled as

$$\mathbf{y}_j = \mathbf{x} + \mathbf{s}_j \tag{1}$$

2.2 Collusion Attacks

Colluders may carry out various collusion attacks to produce the pirated copy. Averaging is an effective collusion attack studied in a lot of existing literature [2,3,4,5,9]. The pirated copy is made by averaging all the different copies of the same content. Colluders will introduce extra noise to the averaged copy to further reduce the chance for them to be detected. The model for averaging collusion can be shown as [4,5]

$$\mathbf{z} = \frac{1}{c} \sum_{j \in C} \mathbf{s}_j + \mathbf{x} + \mathbf{n} \tag{2}$$

where C is the colluder set with c colluders, and $\mathbf{z}$ and $\mathbf{n}$ denote the pirated signal and the additional noise, respectively. The noise term $\mathbf{n}$ is assumed to follow i.i.d. Gaussian distribution $N(0, \sigma_n^2)$, Watermark-to-noise ratio (WNR) is defined as $\mathrm{WNR} = 10 \log_{10}(\|\mathbf{s}\|^2/\|\mathbf{n}\|^2)$ [9].

Interleaving collusion is another typical collusion attack. Colluders divide their own copies of the same content into segments which are called interleaving units, and each colluder contributes several segments. The pirated copy is made by concatenating all the segments contributed by different colluders. Noise may be also introduced to the pirated copy for the same purpose as in averaging collusion. Naturally, no colluder wants to take more risk of being identified, so every colluder's contribution to the pirated copy is almost the same [4,5].

2.3 Colluder Detection

In order to minimize the opportunity of accusing innocent users, non-blind detection scheme is employed to detect only on colluder when a pirated copy is found [4,5]. First the fingerprint sequence is extracted from the pirated copy, and then the correlation of the extracted fingerprint with each user's fingerprint is calculated. For user j, the detection statistic is [4,5]

$$T(j) = \frac{(\mathbf{z} - \mathbf{x})^{\mathrm{T}} \mathbf{s}_j}{\|\mathbf{s}\|}, \quad j = 1, 2, \ldots, N_u \tag{3}$$

2.4 Analysis of Collusion Resistance

From (3) we know that colluder is identified by comparing N_u detection statistics. According to central limit theorem, under averaging collusion, these N_u detection statistics follow N_u-dimensional Gaussian distribution. The mean, variance and covariance are:

$$E_{\text{ave}}[T(j)] = \begin{cases} \left[\frac{1}{c} + \left(1 - \frac{1}{c}\right)\rho\right] \|\mathbf{s}\|, & j \in C \\ \rho\|\mathbf{s}\|, & j \notin C \end{cases} \tag{4}$$

$$\text{var}_{\text{ave}}[T(j)] = \begin{cases} [c + 1 + \rho(c-1)]\frac{\sigma_s^2}{c^2} + \sigma_n^2, & j \in C \\ (1+\rho)\frac{\sigma_s^2}{c} + \sigma_n^2, & j \notin C \end{cases} \tag{5}$$

$$\text{cov}_{\text{ave}}[T(j), T(k)] = \rho\sigma_n^2, \quad j \neq k \tag{6}$$

where ρ is the average correlation between two different fingerprints. The fingerprint correlation between users i and j is defined as [4]

$$\rho_{ij} = \frac{\mathbf{s}_i^{\mathrm{T}}\mathbf{s}_j}{\|\mathbf{s}\|^2} \leq \frac{L - D}{L} \tag{7}$$

where D is the minimum distance of the ECC.

As for interleaving collusion, in conventional ECC-based fingerprinting, colluders can discover which part of the copy contains fingerprint information representing a codeword symbol through carefully examination and comparison. Colluders' interleaving unit is a segment containing information of a codeword symbol [4]. Therefore, interleaving collusion resistance is determined by combinatorial properties of ECC [4,6]. In [10], the authors prove that ECC over an alphabet with size q can resist c users' interleaving collusion if

$$D > \left(1 - \frac{1}{c^2}\right) L \tag{8}$$

According to (8), ECC can provide rather limited interleaving collusion resistance. A feasible thinking is to make it impossible for colluders to adopt interleaving collusion on the code layer [4]. An effective improvement is permuted subsegment embedding (PSE) technique [4], where each fingerprint segment representing a symbol is first divided into β subsegments and then the final signal is derived by applying random permutation to all the βL subsegments with a secrete key. In the detection procedure, inverse operation is performed to the fingerprint extracted from the suspicious copy. [4] shows that PSE scheme outperforms the conventional ECC-based fingerprint under interleaving collusion attack.

As a matter of fact, collusion resistance of ECC-based fingerprinting with PSE scheme under interleaving collusion is still weaker than that under averaging collusion. The following simulation demonstrates the limitation of the PSE

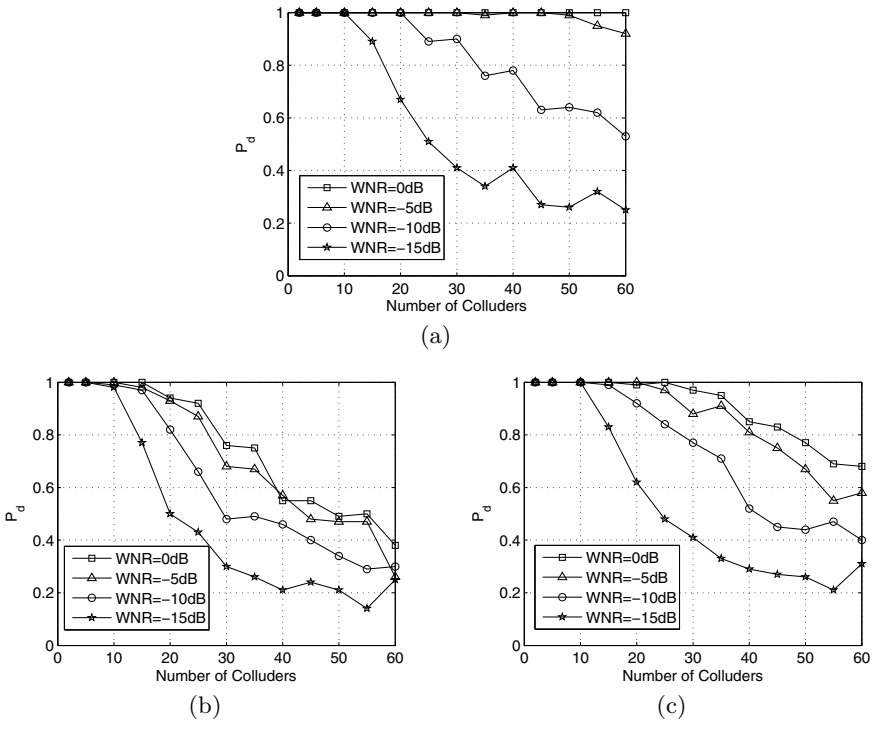


Fig. 1. (a) Colluder detection probability P_d of ECC-based fingerprinting under averaging collusion. Colluder detection probability of ECC-based fingerprinting with PSE scheme of (b) $\beta = 5$ and (c) $\beta = 10$ under subsegment wise interleaving collusion.

scheme. We choose RS code with parameters $q = 32$, $k = 2$, and $L = 30$, the same as those in [4]. We set the total embeddable components $N_s = 36,000$, thus $N_p = 1200$. Number of colluders is set from 2 to 60, and WNR ranges from -15dB to 0dB. We estimate the probability of catching one colluder P_d through 100 iterations. The results of averaging collusion and PSE scheme with $\beta = 5$ and 10 followed by interleaving collusion are shown in Fig. 1.

From Fig. 1 we observe that the performance of fingerprinting system based on ECC is similar when WNR is low (i.e., WNR ≤ -10dB). It is because noise is the dominating factor affecting detection accuracy. Since severe distortion is introduced to the host media in low WNR situation, making the host media useless, we should focus on simulation results at higher WNR, where the fingerprint information in the pirated copy dominates the detection accuracy. Unfortunately, we observe a significant performance gap under the two collusion attacks at moderate-to-high WNR (i.e., WNR> -10dB). The fingerprinting system can resist much less colluder's interleaving collusion than averaging collusion. It can be explained as follows. Under averaging collusion, as is pointed out in [4], the overall fingerprint of each colluder is well preserved in the extracted fingerprint,

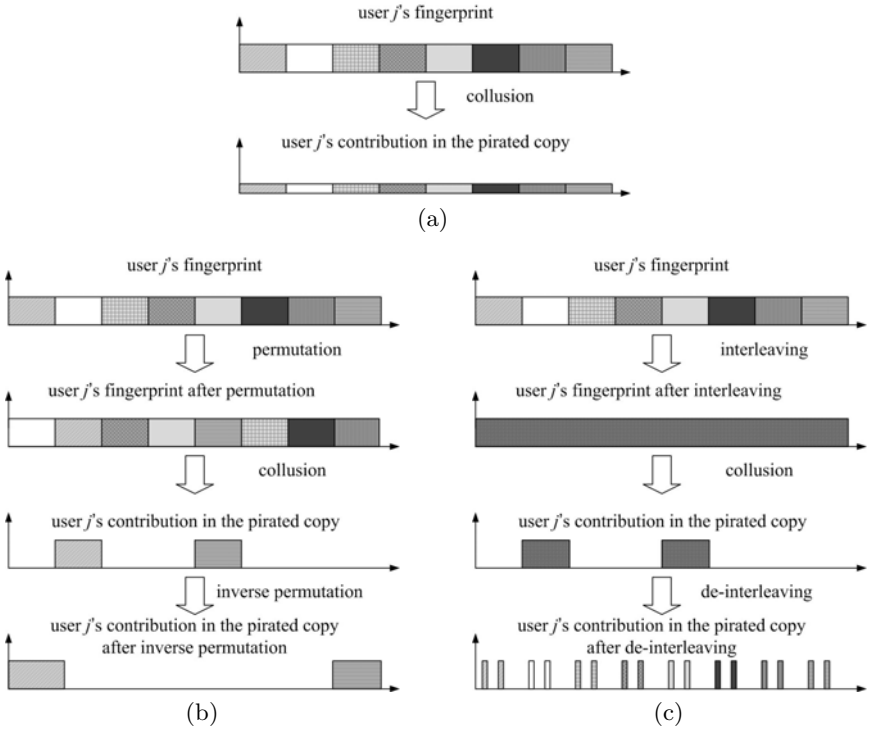


Fig. 2. Illustration of what user j's fingerprint undergoes in different circumstances: (a) ECC-based fingerprinting under averaging collusion; (b) ECC-based fingerprinting with PSE scheme under interleaving collusion; (c) ECC-based fingerprinting with ILE scheme under interleaving collusion. The horizontal axis indicates the position of the fingerprint segments, or subsegments, or elements, and the vertical axis indicates the amplitude of the fingerprint elements.

although the energy of each colluder's fingerprint is attenuated. However, it is not the case for interleaving collusion. When the number of colluders is relatively small, subsegments of each colluder' may uniformly distribute over the entire fingerprint. But as the number of colluder increases, less fingerprint subsegments of each colluder remain in the pirated copy. Thus, fingerprint elements from one colluder only concentrate in limited part of the whole fingerprint. Besides, it should be noted that the averaging collusion resistance of ECC-based fingerprinting is the same no matter what kind of fingerprint embedding scheme is adopted. Fig. 2(a) and (b) illustrate what user j's fingerprint undergoes under averaging collusion and interleaving collusion following PSE scheme. (The illustration in Fig. 2(c) will be discussed in the next section and is attached here for a clear comparison.)

The analysis above shows that the distribution of each colluder's fingerprint in the pirated copy influences the colluder identification accuracy at

moderate-to-high WNR. The more uniformly each colluder's fingerprint scatters in the colluded copy, the higher the probability of identifying the true colluder is.

3 Interleaving Embedding Scheme

3.1 Proposed Embedding Scheme

From the previous section we can see that collusion resistance of ECC-based fingerprinting with PSE scheme under interleaving collusion is still weaker than that under averaging collusion at moderate-to-high WNR. The limitation of PSE scheme motivates us to explore new approach to improve the collusion resistance of ECC-based fingerprinting. Our objective is to make each colluder's fingerprinting information spreads through the entire pirated signal under interleaving collusion.

Interleaving technique has been widely employed in communication systems where the data are transmitted in burst noise channel [11]. At the transmitter side, according to the application environment, the interleaver changes the order of the input symbols to ensure that successive symbols are separated to some extent. At the receiver side, the de-interleaver makes the interleaved symbols back to their correct order. Consequently, serial error symbols introduced during transmission will spread through the de-interleaved symbol sequence at the receiver side, resulting in uniformly distributed error symbols in the sequence.

Considering the advantage of interleaving technique, we apply block interleaving to the original fingerprint sequence before embedding. Suppose we have a matrix $M = [m_{ij}]_{L \times N_p}$, the elements of the original fingerprint sequence are laid down in M row by row and then read out column by column. That is to say, the interleaving depth is N_p, and m_{ij} is the j'th element of the spread spectrum sequence corresponding to the i'th symbol of user's codeword. In the output sequence, m_{ij} is the $[(j-1)L+i]$'th element. As a result, any L consecutive elements contain information from L different codeword symbols. In detection, all the elements of the extracted fingerprint are first input in matrix M column by column and then output row by row back to the original sequence. We call this processing to fingerprint sequence InterLeaving Embedding (ILE) scheme.

As for the collusion resistance aspect of ILE scheme, we hope that each colluder contributes segments carrying consecutive interleaved fingerprint elements, so that after de-interleaved, those elements may well spread over the entire fingerprint sequence, as shown in Fig. 2(c). In the next subsection we will show that this is just the case, which guarantees stronger interleaving collusion resistance at moderate-to-high WNR.

3.2 Security Issue of ILE Scheme

At first sight, it seems that interleaving embedding scheme is vulnerable to interleaving collusion attack: fingerprint rearrangement is conducted regularly instead of randomly, thus colluders can de-interleave the sequence and discover

the sequences corresponding to symbols and then mount symbol wise interleaving collusion. To address this problem, we should first reexamine the fingerprinting procedure in detail. Without loss of generality, we consider one dimensional host signal, for two and higher dimensional signals can be expressed in only one dimension after proper elements rearrangement. Let host signal be denoted as $\mathbf{f}$ in the signal domain and the transform domain coefficients be denoted as $\mathbf{g}$. The first step in fingerprinting is to derive $\mathbf{g}$ from $\mathbf{f}$ as

$$\mathbf{g} = A\mathbf{f} \tag{9}$$

where A is the transform matrix. Then the coefficients capable of embedding, denoted as $\mathbf{x}$ previously, are embedded with interleaved fingerprint $\mathbf{s}_j$, which can be rewritten as

$$\mathbf{g}_j = \mathbf{g} + \mathbf{w}_j \tag{10}$$

where $\mathbf{w}_j$ is the extended fingerprint sequence and all the elements in $\mathbf{w}_j$ added to coefficients without embedding capability are set to zeros. In practice, a secret key is used to decide which coefficient capable for embedding to carry which element of interleaved fingerprint sequence. And then $\mathbf{g}_j$ is inversely transform into signal domain and all the elements are rounded to the nearest positive integers to derive the fingerprinted copy $\mathbf{f}_j$,

$$\mathbf{f}_j = A^{\mathrm{H}}\mathbf{g}_j + \mathbf{e}_j \tag{11}$$

where A^{H} denotes the hermitian transpose of A, and $\mathbf{e}_j$ denotes the round off errors. Since the fingerprint elements from different symbols spread through the whole media, signal domain comparison and examination of different fingerprinted copies does not reveal the fingerprint structure. If colluders want to mount symbol wise interleaving collusion, they must transform the copies they have to the correct transform domain, determine which coefficients contain fingerprint information, and then determine the interleaving depth used in ILE. From (11) we know that user j's copy has already contained round off errors. If it is transformed into the correct transform domain, we have

$$\mathbf{g}'_j = A\mathbf{f}_j = A(A^{\mathrm{H}}\mathbf{g}_j + \mathbf{e}_j) = \mathbf{g}_j + A\mathbf{e}_j \neq \mathbf{g}_j \tag{12}$$

Obviously, the coefficients, embedded with fingerprint or not, are different to the ones just after embedding. The error term $\mathbf{e}_j$ has rather trivial impact on the fingerprint detection; however, it prevents colluders from accurately determining all the coefficients used for embedding by comparing different copies. Even if colluders do find out all the coefficients used for embedding, their ignorance of the secret key prevents them from distinguishing which coefficients contain fingerprint information from which symbol of user's codeword, let alone de-interleaving the fingerprint sequence to mount symbol wise interleaving collusion.

The above analysis reveals that the fingerprint structure has been disguised although the final fingerprint sequence for embedding is not derived by random permutation. As a result, it is extremely tough for colluders to manipulate the

fingerprints on the code layer. If colluders really want to carry out interleaving collusion, what they can do is blindly partition their copies into segments, and then each colluder contributes roughly equal amount of segments to generate the pirated one. After the fingerprint extracted from the pirated copy is de-interleaved, the information from each colluder spreads over all L symbols, which is what we expect. Whereas in PSE scheme, colluders can distinguish subsegments after carefully examining and comparing the fingerprinted copies since each subsegment of fingerprint sequence is restricted in a specific segment of host signal. That is to say, the structure of the embedding sequence is exposed to colluders. Consequently colluders can mount subsegment wise interleaving collusion. Thus, ILE scheme is securer than PSE scheme in terms of fingerprint structure.

3.3 Analysis of Collusion Resistance

When ILE is applied to ECC-based fingerprinting, detection statistics follow N_u-dimensional Gaussian distribution. The mean, variance and covariance are:

$$E_{\text{intl}}[T(j)] = \begin{cases} \left[\frac{1}{c} + \left(1 - \frac{1}{c}\right)\rho\right] \|\mathbf{s}\|, & j \in C \\ \rho\|\mathbf{s}\|, & j \notin C \end{cases} \tag{13}$$

$$\text{var}_{\text{intl}}[T(j)] = \begin{cases} [c + 1 + \rho(c-1)]\frac{\sigma_s^2}{c} + \sigma_n^2, & j \in C \\ (1+\rho)\sigma_s^2 + \sigma_n^2, & j \notin C \end{cases} \tag{14}$$

$$\text{cov}_{\text{intl}}[T(j), T(k)] = \rho\sigma_n^2, \quad j \neq k \tag{15}$$

Comparing (13)-(15) with (4)-(6), it is obvious that ILE scheme makes the distribution of detection statistics under interleaving collusion similar to that under averaging collusion, which results in similar collusion resistance.

We use simulation to verify the interleaving collusion resistance improvement of ILE scheme. Simulation settings are similar to those in the previous section. We assume that colluders divide their copies into 150 and 300 interleaving units containing equal amount of fingerprint elements in interleaving collusion, which can be compared with interleaving collusion when PSE scheme is applied to fingerprinting with $\beta = 5$ and $\beta = 10$, respectively. Simulation results in Fig. 3(a) and Fig. 3(b) show that the amount of interleaving units does not have any significant impact on collusion resistance, which is consistent with the theoretical results of (13), (14) and (15). By comparing Fig. 1(b) and Fig. 3(a), we can see that ILE scheme significantly improves interleaving collusion resistance of ECC-based fingerprinting at moderate-to-high WNR with large number of colluders. Similar conclusion can be drawn by comparing Fig. 1(c) and Fig. 3(b). From Fig. 1(a) and Fig. 3 we can see that the overall performance of ILE scheme under interleaving collusion is comparable to that under averaging collusion, which is also consistent with the theoretical analysis.

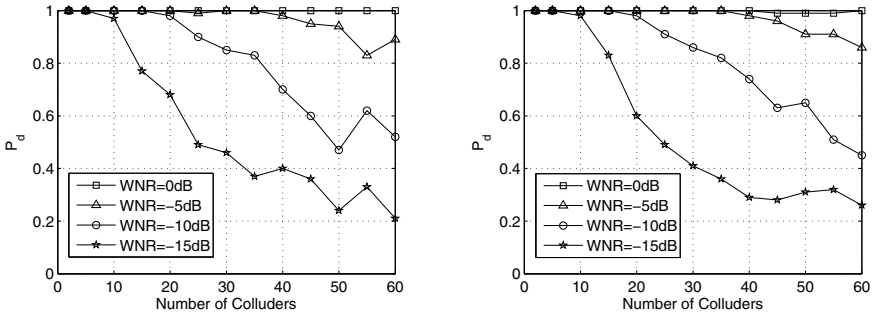


Fig. 3. Colluder detection probability P_d of ECC-based fingerprinting with ILE under interleaving collusion with (a) 150 interleaving units and (b) 300 interleaving units

3.4 Experiments on Real Images

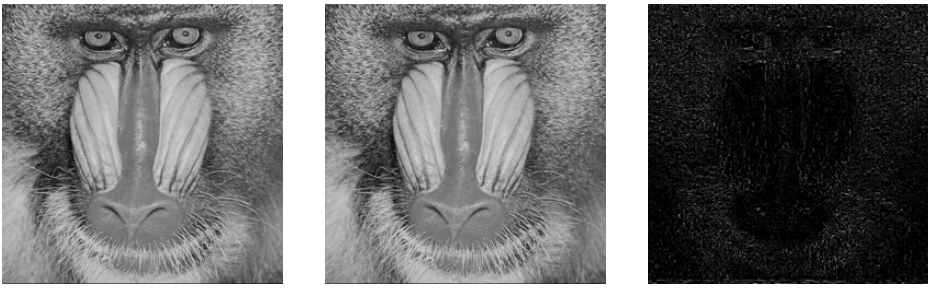
In this subsection, we further demonstrate the performance improvement of ECC-based fingerprinting with ILE scheme on real images. In order to embed fingerprint in image, we apply image adaptive watermarking technique in discrete cosine transform domain (IA-DCT) [8], where the original host image is applied 8×8 block DCT, and then just noticeable difference (JND) of every coefficient is calculated according to Watson's perceptual model, finally the fingerprint elements are embedded into the coefficients whose amplitudes are larger than their corresponding JNDs with perceptual scaling. Watermark detection is performed with the help of original host signal. Experimental results in [8] show that IA-DCT is very robust to JPEG compression, cropping, and scaling. Since the original host signal is available at the detector side, geometric distortion introduced to the watermarked copy such as rotation can be inverted [9]. Hence, we mainly focus on collusion attacks in our experiments on real images.

In our experiments, the host images are 512×512 Lena image and Baboon image. The ECC is RS(30, 2) over GF(32). For image Lena, $N_s = 37410$, $N_p = 1247$, and for image Baboon, $N_s = 87570$, $N_p = 2919$. Spread spectrum sequences representing different symbols in GF(32) follow i.i.d. Gaussian distribution $N(0, 1/9)$, such that more than 99% of the fingerprint elements fall in the range of $[-1, 1]$, guaranteeing most modifications to the DCT coefficients are not beyond the JNDs. The same secret key is used to decide which coefficient is embedded with which fingerprint element for all the copies to be generated. The PSNR of the fingerprinted images is around 41.05dB for Lena and 34.95dB for Baboon. Fig. 4 shows the original host image, fingerprinted image and the pixel wise difference image for Lena and Fig. 5 for Baboon.

We assume that colluders blindly partition the fingerprinted images they have into blocks containing 32×32 pixels, and each colluder contributes approximately equal amount of blocks of his copy to form the colluded copy in interleaving collusion. Before colluder detection, the colluded copy undergoes JPEG compression with different quality factors. For Lena image, quality factors $Q = 80$, 60, and

(a) (b) (c)

Fig. 4. (a) Original Lena image. (b) Fingerprinted Lena image. (c) Corresponding difference image (amplified by a factor of 10).



(a) (b) (c)

Fig. 5. (a) Original Baboon image. (b) Fingerprinted Baboon image. (c) Corresponding difference image (amplified by a factor of 10).

40 are considered. For Baboon image, $Q = 40$, 30, and 20. We estimate the probability of catching one colluder P_d through 100 iterations. The results are presented in Fig. 6(a) for Lena image and Fig. 7(a) for Baboon image. The results of averaging collusion are presented in Fig. 6(b) and Fig. 7(b) for comparison. From Fig. 6 and Fig. 7 we observe that the fingerprint in Baboon image is capable of surviving lower quality JPEG compressions. Actually, for Lena image, $Q = 80$, 60, and 40 correspond to WNR = 1.01dB, −5.03dB, and −8.20dB, respectively, and for Baboon image, $Q = 40$, 30, and 20 correspond to WNR = −7.20dB, −8.74dB, and −10.53dB, respectively. Since Lena image and Baboon image have quite different spatial characteristics, JPEG compression with the same quality factor introduces different strength of noise to these two images (Notice that noise introduced by JPEG compression does not follow i.i.d. Gaussian distribution). Most importantly, we can observe from Fig. 6 and Fig. 7 that collusion resistance of ILE scheme under interleaving collusion is comparable to that under averaging collusion, which again verifies the effectiveness of the proposed ILE scheme.

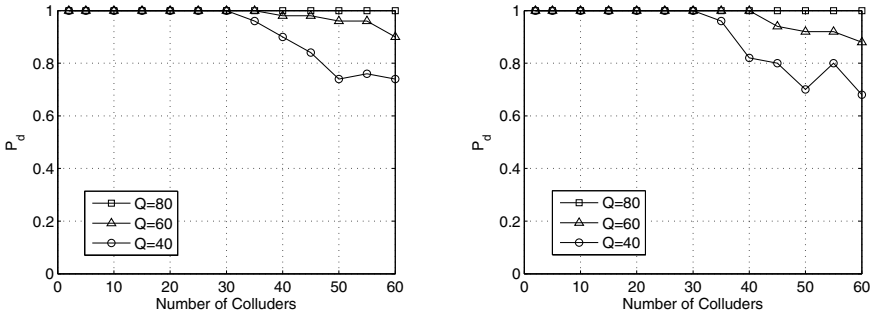


Fig. 6. Experimental results of colluder detection probability of Lena image under (a) interleaving collusion and (b) averaging collusion followed by JPEG compression with different quality factors

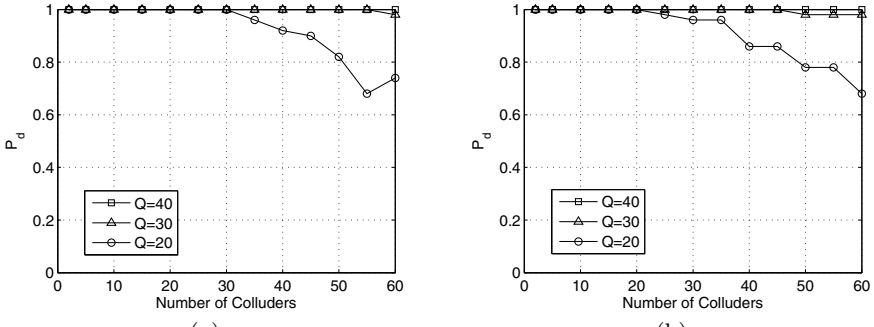


Fig. 7. Experimental results of colluder detection probability of Baboon image under (a) interleaving collusion and (b) averaging collusion followed by JPEG compression with different quality factors

4 Conclusions

Collusion resistance of ECC-based multimedia fingerprinting is studied in this paper. By evaluating the performance of existing ECC-based multimedia fingerprinting schemes under different collusion attacks, we observe that the distribution of each colluder's fingerprint in the pirated copy influences the colluder identification accuracy at moderate-to-high WNR. The more uniformly each colluder's fingerprint distributes in the colluded copy, the higher the probability of identifying the true colluder is. Based on this fact, we propose ILE scheme. The key idea is to apply block interleaving to the original fingerprint sequence, and then the interleaved fingerprint is embedded to the host signal in transform domain, and a secret key is used to control where to place each interleaved fingerprint element, whereby the fingerprint structure is well hidden and the security

issue is addressed. Simulation results show that ECC-based fingerprinting with ILE scheme has much better interleaving collusion resistance than that with PSE scheme over moderate-to-high WNR with large number of colluders. Simulation results, theoretical analysis and experiments on real images show that ECC-based fingerprinting with ILE scheme has good collusion resistance under interleaving collusion comparable to that under averaging collusion.

Acknowledgments

This work is supported by the National Science Foundation of China (No.60702047, 60772098, 60802057 and 61071152), the National Basic Research Program of China (No.2010CB731403), and the Opening Project of Key Lab of Information Network Security of Ministry of Public Security (c09607).

References

1. Wu, M., Trappe, W., Wang, Z.J., Liu, K.J.R.: Collusion-Resistant Fingerprinting for Multimedia. IEEE Signal Processing Magazine 21(2), 15–27 (2004)
2. 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)
3. Trappe, W., Wu, M., Wang, Z.J., Liu, K.J.R.: Anti-collusion Fingerprinting for Multimedia. IEEE Transactions on Signal Processing 51(4), 1069–1087 (2003)
4. He, S., Wu, M.: Joint Coding and Embedding Techniques for Multimedia Fingerprinting. IEEE Transactions on Information Forensics and Security 1(2), 231–247 (2006)
5. He, S., Wu, M.: Collusion-Resistant Video Fingerprinting for Large User Group. IEEE Transactions on Information Forensics and Security 2(4), 697–709 (2007)
6. Safavi-Naini, R., Wang, Y.: Collusion Secure q-ary Fingerprinting for Perceptual Content. In: Sander, T. (ed.) DRM 2001. LNCS, vol. 2320, pp. 57–75. Springer, Heidelberg (2002)
7. Boneh, D., Shaw, J.: Collusion-Secure Fingerprinting for Digital Data. IEEE Transactions on Information Theory 44(5), 1897–1905 (1998)
8. Podilchuck, C.I., Zeng, W.: Image-Adaptive Watermarking Using Visual Models. IEEE Journal on Selected Areas in Communications 16(4), 525–539 (1998)
9. Wang, Z.J., Wu, M., Zhao, H.V., Trappe, W., Liu, K.J.R.: Anti-Collusion Forensics of Multimedia Fingerprinting Using Orthogonal Modulation. IEEE Transactions on Image Processing 14(6), 804–821 (2005)
10. Staddon, J.N., Stinson, D.R., Wei, R.: Combinatorial Properties of Frameproof and Traceability Codes. IEEE Transactions on Information Theory 47(3), 1042–1049 (2001)
11. Rappaport, T.S.: Wireless Communications: Principles and Practice, 2nd edn. Prentice Hall, New Jersey (2002)

A Novel Collusion Attack Strategy for Digital Fingerprinting

Hefei Ling[1], Hui Feng[1], Fuhao Zou[1], Weiqi Yan[2], and Zhengding Lu[1]

[1] College of Computer Science, Huazhong University of Science Technology, Wuhan, Hubei, China
[2] Institute of ECIT, Queen's University Belfast, Belfast, BT7 1NN, United Kingdom
lhefei@hotmail.com, huifeng.email@gmail.com,
fuhao_zou@hust.edu.cn, w.yan@qub.ac.uk, zdlu@hust.edu.cn

Abstract. Digital fingerprinting is a technology which aims to embed unique marks with traceability in order to identify users who use their multimedia content for unintended purpose. A cost-efficient attack against digital fingerprinting, known as collusion attack, involves a group of users who combine their fingerprinted content for the purpose of attenuating or removing the fingerprints. In this paper, we analyze and simulate the effect of Gaussian noise with different energies added in the noise-free forgery on both the detection performance of correlation-based detector and the perceptual quality of the attacked content. Based upon the analysis and the principal of informed watermark embedding, we propose a novel collusion attack strategy, *self-adaptive noise optimization (SANO)* collusion attack. The experimental results, under the assumption that orthogonal fingerprints are used, show that the proposed collusion attack performs more effectively than the most of existed collusion attacks. Less than three pieces of fingerprinted content can sufficiently interrupt orthogonal fingerprints which accommodate many thousands of users. Meanwhile, high fidelity of the attacked content is retained after the proposed collusion attack.

Keywords: Digital fingerprinting, collusion attack, optimization, Gaussian noise.

1 Introduction

Digital fingerprinting technology has proven to be very successful in protecting multimedia content from unauthorized redistribution. Unique marks, known as fingerprints, are embedded in the content before distribution which is used to identify adversaries who leak copies of the content if a suspicious copy is found. Collusion attack is known to be a cost-effective attack, whereby a group of users combines their copies of the same multimedia content to generate a new version. With enough collected copies, the adversaries are able to attenuate or remove the fingerprints, which results in the detector being unable to trace any of the real colluders involved.

H.-J. Kim, Y. Shi, and M. Barni (Eds.): IWDW 2010, LNCS 6526, pp. 224–238, 2011.

Many kinds of fingerprinting schemes were explored for the purpose of resisting collusion attacks, as well as identifying the colluders involved. The earlier work [1] was conducted by Boneh and Shaw, two-level c-secure fingerprints were proposed under the assumption that users cannot change the state of an undetected mark without rendering the object useless. The other popular fingerprinting strategies are orthogonal spread spectrum sequence [2] which has powerful collusion-resistant properties. The codevectors which are generated for fingerprints assigned to all of the users are mutually orthogonal. Another class of codes, called anti-collusion codes (ACCs) [3], were developed based on a combinatorial design. For a given number of colluders, this scheme needs only $O(\sqrt{n})$ basis vectors to accommodate n users. Dittmann et al. [4] proposed the fingerprints based upon the theory of finite-geometries. Shan He et al. [5] proposed a group-based joint coding and embedding technique and concluded that it offers an improved tradeoff between the collusion resistance and detection efficiency when compared to the conventional ECC-based fingerprinting. In recent years, Cha et al. [6],[7] proposed MC-CDMA-based fingerprinting system and demonstrated its robust collusion-resistant performance in the presence of time-varying collusion attacks.

As for collusion attacks, most of the existing works [8], [3], [9] on multimedia fingerprinting were involved in a linear collusion attack model. A set of typical non-linear collusion attacks were studied in [10]. Zhao et al. [11],[12] analyzed the effectiveness of the linear and non-linear collusion attacks and their impact on the perceptual quality based on independent Gaussian fingerprints. Moulin et al. [13] analyzed the performance of arbitrary nonlinear collusion attacks on random fingerprinting codes. Kiyavash et al. [14] developed a mathematical analysis of the performance of order statistic collusion attacks on Gaussian fingerprinting systems. They proved that all the nonlinear attacks considered result in the same detection performance, and the linear averaging attack outperforms the other ones in the sense of minimizing mean-squared distortion. In [15], the authors studied the effect of the noise distribution on the error probability of the detection test when a class of randomly rotated spherical fingerprints is used. They concluded that the worst noise is impulsive, and the performance of the detector is dramatically worse than that obtained under i.i.d. Gaussian noise. Darko Kirovski et al. [16],[17] analyzed the collusion-resistance of a large class of spread spectrum fingerprinting systems using a new gradient attack. He et al. [18] applied the gradient attack on an existing well-engineered video fingerprinting scheme and demonstrated that the gradient attack is also effective on Laplace fingerprints.

A problem of how to design a collusion attack strategy which can efficiently defeat most of existing fingerprinting schemes, meanwhile, maintaining high fidelity of the colluded forgery, has great theoretical and practical interest. For fingerprints and signals defined over Euclidean spaces, the worst case was the linear average attack followed by addition of additive Gaussian noise under mean-squared distortion constraints [15]. However, the energies of Gaussian noise added how to affect the performance of the detector, and how to add Gaussian noise to

the noise-free forgery via an optimal approach to solve the above problem is unknown, and this is the subject of this paper.

The contributions of this paper include the following: 1) We analyze and simulate the effectiveness of Gaussian noise with different energies on both the detection performance of the detector and the perceptual quality of the attacked content. 2) We propose a *self-adaptive noise optimization (SANO)* strategy which performs much better than existing collusion attacks, meanwhile, maintaining high fidelity of the colluded content.

This paper is organized as follows. We begin, in Section 2, with the analysis and simulations of the effectiveness of Gaussian noise with different energies on the detection performance of correlation-based detector and also the impact on the visible distortion of the attacked content. Then, in Section 3, we present a detailed description regarding our proposed strategy. In Section 4, we highlight our simulation results. We draw the conclusion in Section 5.

2 The Effectiveness Analysis of Gaussian Noise

In this section, we start with the system model of digital fingerprinting and collusion attacks. Then we analyze the influence of Gaussian noise with different energies on both the detection probability of correlation detector and the perceptual quality of the attacked content. In the end, we give some simulation results based on the theoretical analysis.

2.1 System Model

Digital fingerprinting system usually consists of three parts: fingerprint embedding, collusion attacks, and fingerprint detection. In fingerprint embedding, we assume that the host media signal to be fingerprinted, $\mathbf{S}$, is a sequence of length N, and a total of M users in the system. The m-th user is specifically assigned with a unique marked copy

$$\mathbf{X}_{\mathrm{m}}=\mathbf{S}+\mathbf{W}_{\mathrm{m}}, m \in \{1, 2, ..., M\} \tag{1}$$

where $\mathbf{W}_{\mathrm{m}} = (W_m(1), W_m(2), \ldots, W_m(N)) \in R^N$ is the fingerprint of user m. The embedded fingerprint must be imperceptive and robust with respect to typical image processing attacks.

The collusion attack can be modeled as the uniform average of the colluders' marked signals, followed by addition of a noise sequence $\mathbf{e}$

$$\mathbf{Y} = \frac{1}{K}\sum_{k\in S_c}\mathbf{X}_k + \mathbf{e} = \mathbf{S} + \frac{1}{K}\sum_{k\in S_c}\mathbf{W}_k + \mathbf{e} = \mathbf{S} + \mathbf{C} \tag{2}$$

where $\mathbf{Y}$ is the attacked forgery, K out of M users participant in collusion, and S_c is the collusion set which contains the indices of colluders, $S_c \subseteq \{1, 2, ..., M\}$.

In fingerprinting applications, non-blind detection is conducted to extract the fingerprint, i.e., the host signal $\mathbf{S}$ is available and always subtracted from the

forgery $\mathbf{Y}$ to form the extracted signal $\mathbf{C}$. Many correlation statistics [11] are available to identify the presence of the original fingerprint $\mathbf{W}_m$ in the extracted signal $\mathbf{C}$. In this paper, we consider T statistic of

$$T_m = \frac{\langle \mathbf{C}, \mathbf{W}_m \rangle}{\sqrt{\|\mathbf{W}_m\|^2}} = \frac{\sum_{i=1}^{N} C(i)W_m(i)}{\sqrt{\|\mathbf{W}_m\|^2}} \tag{3}$$

For simplicity, we assume that the fingerprints assigned to the users follow i.i.d. Gaussian distribution with zero mean and variance of σ_w^2, i.e., $\mathbf{W}_m \sim N(0, \sigma_w^2)$, and the noise is also i.i.d. distributed, $\mathbf{e} \sim N(0, \sigma_e^2)$. It is easy to observe that all $\{C(i)W_m(i)\}_{i=1}^{N}$ where $m \in S_c$ have the same mean and variance, and similarly, where $m \notin S_c$ also have the same mean and variance. In the following definitions, we drop the subscript i for convenient reading. For $m \in S_c$, we define

$$\begin{aligned} \mu'_{T1} &= E[CW_m] = E[(\frac{1}{K}\sum_{k\in S_c} W_k + e)W_m] = \frac{1}{K}\sigma_w^2 \\ \sigma'^2_{T1} &= var[CW_m] = var[(\frac{1}{K}\sum_{k\in S_c} W_k + e)W_m] \\ &= E[((\frac{1}{K}\sum_{k\in S_c} W_k + e)W_m)^2] - \mu'^2_{T1} \end{aligned} \tag{4}$$

For $m \notin S_c$, we define

$$\begin{aligned} \mu'_{T2} &= E[CW_m] = E[(\frac{1}{K}\sum_{k\in S_c} W_k + e)W_m] = 0 \\ \sigma'^2_{T2} &= var[CW_m] = var[(\frac{1}{K}\sum_{k\in S_c} W_k + e)W_m] \\ &= E[((\frac{1}{K}\sum_{k\in S_c} W_k + e)W_m)^2] \end{aligned} \tag{5}$$

From the above analysis, we can find that the correlation statistic T_m can be approximated by the following Gaussian distribution

$$T_m = \frac{T'_m}{\sqrt{\|W_m\|^2}} \sim \begin{cases} N(\frac{\sqrt{N}\mu'_{T1}}{\sigma_w}, \frac{\sigma'^2_{T1}}{\sigma_w^2}), \text{if} m \in S_c \\ N(0, \frac{\sigma'^2_{T2}}{\sigma_w^2}), \text{if} m \notin S_c \end{cases} \tag{6}$$

We denote $\mu_1 = \sqrt{N}\mu'_{T1}/\sigma_w$, $\sigma_1^2 = \sigma'^2_{T1}/\sigma_w^2$ and $\sigma_2^2 = \sigma'^2_{T2}/\sigma_w^2$. There are two main performance criteria to evaluate the efficacy of the collusion attack: the probability of colluders that are successfully captured (P_d), and the probability of innocent users that are falsely accused (P_{fp}). From [11], we can see that if

$\{T_m\}_{m=1}^{M}$ are independent with each other or the correlation between them is very small, then for a given threshold h, we can approximate P_d and P_{fp} by

$$P_d = P\{\sum_{m \in S_c} T_m > h\} \approx Q(\frac{h - \mu_1}{\sigma_1}) \tag{7}$$

$$P_{fp} = P\{\sum_{m \notin S_c} T_m > h\} \approx Q(\frac{h}{\sigma_2}) \tag{8}$$

where $Q(x)$ is Gaussian tail function. The distortion introduced by collusion attack in the host signal based on *just-noticeable difference* (JND) [19] can be defined as

$$F_{\text{JND}} = \sum_{i=1}^{N} I_{|n_i| > JND_i}/N \tag{9}$$

which reflects the percentage of the noise components that exceed JND [11]. A large F_{JND} indicates large perceptual distortion introduced. Where $n_i = JND_i \cdot C(i)$, and i =1, 2,...,N. If $\{C(i)\}_{i=1}^{N}$ has the pdf of $f_{\sigma_d,K}(w)$, the perceptual quality can be approximated by

$$E[F_{JND}] = P\{|C| > 1\} = \int_{-\infty}^{-1} f_{\sigma_d,K}(w)dw + \\ + \int_{1}^{\infty} f_{\sigma_d,K}(w)dw \tag{10}$$

as in [11]. Since the fingerprints assigned to the users and the noise added to the forgery are assumed to follow i.i.d. Gaussian distribution, $\{C(i)\}_{i=1}^{N}$ are also i.i.d distributed with zero mean and variance of $\sigma_c^2 = \sigma_w^2/K + \sigma_e^2$.

2.2 Simulation Results

In this section, based on the theoretical analysis in Section 2.1, we perform two groups of simulations. In these simulations, the users of the system accommodated are 10^3. We consider the length of the fingerprints is 1024.

In the first simulation, we study the effect of Gaussian noise with different energies added to the noise-free forgery on the detection probability of correlation detector at first. We calculate the threshold h to yield the desired P_{fp} via (8) under different energies of noise and different number of colluders, and then calculate the corresponding P_d via (7). For a fingerprinting system, a reasonably low P_{fp} is required to maintain its resistance to colluders. In this paper, we consider the case when $P_{fp} = 10^{-2}$, maybe this value should be much lower in some real applications, for example, in the large scale fingerprinting schemes. Fig.1(a) illustrates the simulation result of the probability that colluders are successfully captured P_d versus the colluder number under different energies of noise. The standard deviation of the Gaussian noise added from 0.1 to 1. From Fig.1(a), we can find that the Gaussian noise has the ability of impeding the fingerprint detection effectively, and for a given number of colluders, the larger energy the noise added, the lower probability the detector captured.

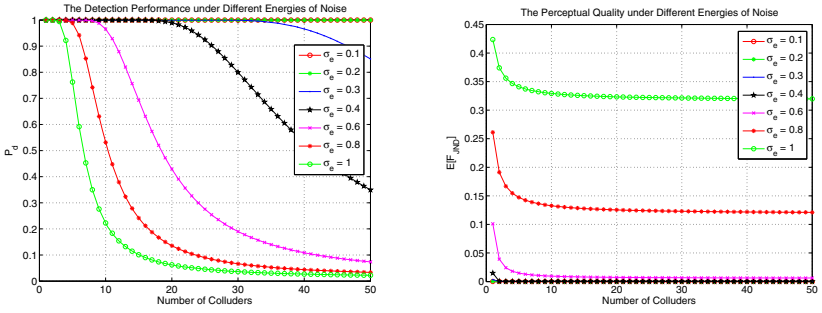


Fig. 1. The performance comparison of different energies of noise added in terms of (a) the detection probability vs. the colluder number, (b) the perceptual quality vs. the number of colluders, where the fingerprint length is 1024, $P_{fp} = 10^{-2}$

At second, in order to study the perceptual quality introduced by the noise, we calculate the corresponding distortion introduced by different energies of noise under different number of colluders according to (10). The result is shown in Fig. 1(b). From Fig.1(b), we can observe that more visible distortion in the host signal is introduced when larger energy of noise is added.

The situation in the first simulation is that the fingerprints follow Gaussian distribution. To extend the analysis to a more real application, we conduct the second group of simulation under the assumption that orthogonal fingerprints are employed. We know that orthogonal fingerprinting is one of the most efficient schemes on resisting usually used linear and nonlinear collusion attacks [8], [20]. We apply the system to a host signal that is modeled as an i.i.d. Gaussian sequence of length 256×256. The host signal is independent of the users' fingerprints and the noise added. The spread spectrum embedding scheme [21] is employed to yield the fingerprinted contents. We create a noise-free forgery via uniform averaging of different number of collected fingerprinted copies, and then add the noise with different energies to form the actual forgery. The peak signal-to-noise ratio (PSNR) is used to evaluate the perceptual quality between the forgeries and the host signal. Though PSNR can not represent the whole idea of multimedia quality, it is popularly used in the literature resorting to its intuitive property. The results of 1000 iterations are illustrated in Fig.2(a) and (b). In Fig.2(a), we plot the detection probability P_d versus the colluder number under different energies of noise. We can find that the rank of the detection probabilities under different energies of noise is similar to that of in Fig.1(a). In Fig.2(b), we display the PSNR with respect to the number of colluders. We can observe that, the distortion introduced by the noise would come into inacceptable once the energy of the noise exceeding an available value bounded by the perceptual fidelity.

As can be seen from the above analysis and simulations, the colluders add Gaussian noise to a noise-free forgery which they create by averaging their

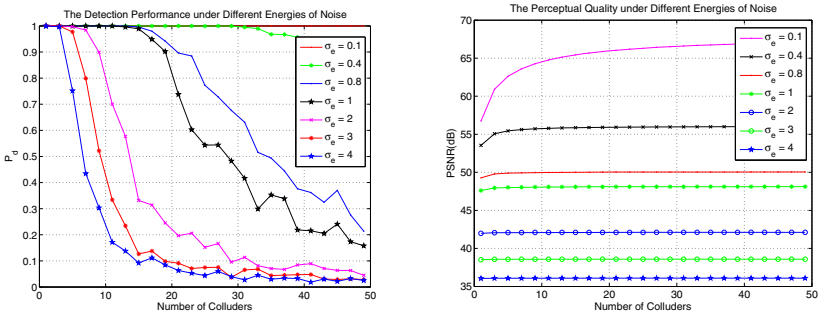


Fig. 2. The performance comparison of different energies of noise added, under the assumption that orthogonal fingerprints are used. (a) the detection probability vs. the colluder number, (b) the perceptual quality vs. the colluder number, where the fingerprint length is 1024, $P_{fp} = 10^{-2}$.

signals, this poses a great threat to the collusion resistance of the fingerprinting systems. However, it also introduces much more visible distortion in the host signal if the noise signal is not appropriately inserted. Therefore, new issue arise, i.e., how to find a strategy which is able to add the Gaussian noise into a noise-free forgery, on one hand, while maximizing the error probability of detector, on the other hand and also maintaining higher fidelity of the attacked content. With this issue in mind, we introduce the details of the proposed collusion attack strategy in the next section.

3 Proposed Collusion Attack Strategy

In this section, we propose a novel collusion attack strategy, *self-adaptive noise optimization (SANO)* collusion attack. At first, we revisit the principal of informed watermark embedding technology which inspires us to propose such sophisticated collusion attack. Then, we introduce the details of the proposed collusion attack strategy.

3.1 Inspiration

In [22], Miller et al. proposed a novel robust watermark embedding scheme, in which the watermark was embedded by an iterative method that intended to ensure the embedded message will not be confused with other messages. Geometric representation of a Voronoi diagram interprets the optimized iterative method for the informed watermark embedding very well. Fig.3 illustrates the iteration procedure. In each iteration, the cover image is modified to prevent it from being decoded as one "bad" vector. Where C_0 is the cover image, B_i is a set of "bad" vectors that should not be embedded, G is good vector that is expected to be

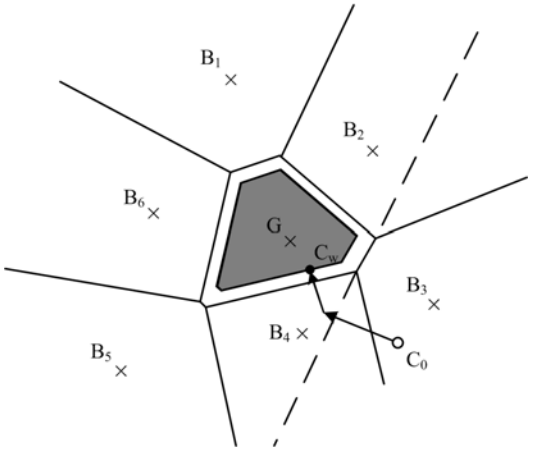


Fig. 3. Geometric representation of the Voronoi diagram for informed watermark embedding

embedded. We can see that in the first iteration, they modify C_0 so that it will not be detected as containing message B_3. In the second iteration they modify it so that it will not be detected as containing B_4. This results in a watermarked image C_w within the embedding region around G.

We found that Miller's informed embedding scheme is an optimized iterative procedure which is about to obtain a watermarked sample from the original content under the limitation of perceptual fidelity. The objective of a collusion attack is to achieve a clear copy, which does not contain the fingerprint and also makes it difficult to detect any of the colluders in the collusion set, from different marked copies of the same content. Therefore, we can observe that the goal of each party is opposite with each other, one attempts to embed the watermark, the other intends to remove it. However, the principles behind them are similar.

3.2 Self-Adaptive Noise Optimization Collusion Attack

For the purpose of implementing a more efficient collusion attack, we keep the objective of removing the fingerprints in mind, and take two steps into consideration. At the first step, in order to avoid perceptual quality degradation of the attacked content introduced by large energy of Gaussian noise, we apply a simple, iterative approach, i.e., adding a small amount of noise at each iteration. These noises are then spread over the attacked signal. However, a critical problem is how to manipulate the strength of the noises added, which makes the fingerprint signal attenuate rather than strengthen. Inspired by the informed watermark embedding, we investigate a self-adaptive optimization strategy at the second step. In this section, we first present a suitable measure function to test whether or not the fingerprint is presented in the attacked signal. Then,

combining with the two steps outlined above, we propose the *self-adaptive noise optimization (SANO)* collusion attack strategy.

Measure Function. In fingerprinting systems, the colluder and the detector are always considered as two players in a game [23]. In this paper, we assume that the fingerprint detection algorithm is undisguised for the detector and the colluders. But the host signal and the users' fingerprints are unknown to the colluders. Therefore, for the adversaries, the traditional correlation detector is not appropriate to measure the existence of the fingerprints of the participated colluders.

To develop a new measure function, we consider a simple system in which there are only two possible fingerprints extracted from the forgery $\mathbf{Y}$. We denote them as $\mathbf{W}_c$ and $\mathbf{W}_n$, which represent the signals that can be or cannot be detected as containing the colluders, respectively. The task of collusion attack is to remove the fingerprints of the colluders from the forgery, so make the extracted fingerprint yield lower correlation value on the detector side. Therefore, we can obtain

$$\mathbf{W}_c \cdot \mathbf{Y} > \mathbf{W}_n \cdot \mathbf{Y} \tag{11}$$

where $\mathbf{W}_c \cdot \mathbf{Y}$ is the correlation between $\mathbf{W}_c$ and $\mathbf{Y}$. Additional distortion may be added to the signal $\mathbf{Y}$ during the collusion, we assume that the distortion can be modeled as additive Gaussian noise. So, the detector will receive $\mathbf{Y}_e = \mathbf{Y} + \mathbf{e}$, where $\mathbf{e}$ is the noise vector whose elements follow Gaussian distribution with variance of σ_e^2. The probability of containing the fingerprints of the colluders in $\mathbf{W}_n$ is smaller than in $\mathbf{W}_c$ can be expressed as

$$\begin{aligned} &\mathrm{P}\{\mathbf{W}_c \cdot \mathbf{Y}_e > \mathbf{W}_n \cdot \mathbf{Y}_e\} \\ &= P\{\mathbf{W}_c \cdot (\mathbf{Y} + \mathbf{e}) > \mathbf{W}_n \cdot (\mathbf{Y} + \mathbf{e})\} \\ &= P\{\frac{(\mathbf{W}_c - \mathbf{W}_n) \cdot \mathbf{Y}}{|\mathbf{W}_c - \mathbf{W}_n|} > \sigma_e \mathbf{I}\} \end{aligned} \tag{12}$$

where $\mathbf{I}$ is a unit-variance random variable which follows Gaussian distribution. We define the normalized correlation value $R_0(\mathbf{W}_c, \mathbf{W}_n, \mathbf{Y})$ as

$$R_0(\mathbf{W}_c, \mathbf{W}_n, \mathbf{Y}) = \frac{(\mathbf{W}_c - \mathbf{W}_n) \cdot \mathbf{Y}}{|\mathbf{W}_c - \mathbf{W}_n|} \tag{13}$$

We can see that the larger the value of $R_0(\mathbf{W}_c, \mathbf{W}_n, \mathbf{Y})$, the smaller risk of being detected as containing the colluders. We denote $R_0(\mathbf{W}_c, \mathbf{W}_n, \mathbf{Y})$ as R_0 for simplicity. Therefore, R_0 can be served as the measure function of a simple, two fingerprints system. However, different energies of noise added to the forgery would yield different values of R_0. We take into consideration of the minimum of the R_0's to guarantee that all possible fingerprints of the colluders can be removed from the forgery. The target measure value R_t is determined by the boundary of whether the fingerprints of the colluders are presented and the limitation of the perceptual quality of the attacked content. In this paper, the value of R_t is empirical.

Self-adaptive Optimization Strategy. The best strategy for effectively attenuating or removing the fingerprint, while maintaining a higher perceptual quality of the attacked content, depends on the best amount of noise added to the noise-free forgery. Under the constraint that the host signal is unavailable to the adversaries, it is not very easy. Therefore, we propose a self-adaptive optimization strategy.

We assume we have a fingerprint detector, $Ex(\mathbf{S})$, which is used to extract the fingerprint from the signal $\mathbf{S}$. Given the noise-free forgery $\mathbf{Y}^c$, the target measure value R_t, and let $\mathbf{W}_c = Ex(\mathbf{Y}^c)$. If a small amount of noise $\mathbf{e}$ is added to $\mathbf{Y}^c$ and yield a signal $\mathbf{Y}^w$, the detector returns a fingerprint $\mathbf{W}_n$ (i.e., $\mathbf{W}_n = Ex(\mathbf{Y}^w)$) other than $\mathbf{W}_c$, then is likely to yield a low value of R_0. Our objective is to add the appropriate amount of noise to yield a fingerprint make the value R_0 close to R_t. At that point, the detector is likely to hardly detect any of the colluders.

The amount of noise to add changes in each iteration of the optimization process. In the first iteration, we do not add any noise at all, in this case, $\mathbf{W}_n$ is exactly equals to $\mathbf{W}_c$, because $\mathbf{Y}^w = \mathbf{Y}^c$. Then we need add only a small amount of noise to $\mathbf{Y}^w$, the extracted fingerprint $\mathbf{W}_n$ is likely to yield very low value of R_0. As $\mathbf{Y}^w$ is modified to be detected having less probability of containing the colluders, the extracted fingerprint yield higher values of R_0, and thus require the addition of more noise. In general, if too much noise is added, $\mathbf{Y}^w$ has a high probability of producing a fingerprint for which R_0 is much larger than the maximum available value bounded by the perceptual fidelity.

We therefore use a self-adaptive approach to adjust the amount of noise added in each iteration. At the beginning, the standard deviation of the noise, σ_e, is 0, so no noise is added to $\mathbf{Y}^w$. We increase σ_e by a small, fixed amount, δ. When $\mathbf{Y}^w$ yields a fingerprint $\mathbf{W}_n$, but R_0 is greater than or equal to R_t, we decrease σ_e by δ. If yields a fingerprint $\mathbf{W}_n$, and R_0 less than R_t, we modify $\mathbf{Y}^w$ and leave δ unchanged. In our experiments, $\delta = 0.01$. However, this approach still does not guarantee that we find the optimal fingerprint $\mathbf{W}_n$ that minimizes R_0 in each iteration, we cannot terminate the algorithm the first time R_0 that is greater than or equal to the target value R_t, there might still be some other $\mathbf{W}_n$ for which $R_0 < R_t$. We therefore create a counter to record the number of $\mathbf{W}_n$'s found for which R_0 is greater than or equal to R_t. The algorithm terminates when this counter reaches a specified number. In our experiments, the number is set to 100. Therefore, the proposed algorithm is expressed as follow.

Self-adaptive noise optimization (SANO) collusion attack algorithm:

1) Let $\mathbf{Y}^w = \mathbf{Y}^c$, $\mathbf{W}_c = Ex(\mathbf{Y}^c)$, $\sigma_e = 0$ and j=0;

2) Let $\mathbf{W}_n = Ex(\mathbf{Y}^w + \mathbf{e})$, where $\mathbf{e}$ is the noise added to the signal $\mathbf{Y}^w$ which follows Gaussian distribution with variance of σ_e^2;

3) If $\mathbf{W}_n = \mathbf{W}_c$, let $\sigma_d \leftarrow \sigma_d + \delta$ and go back to step 2);

4) Calculate R_0 according to (13). If $R_0 < R_t$, then modify $\mathbf{Y}^w$ as

$$\begin{aligned} \boldsymbol{\beta} &= (\mathbf{W}_c - \mathbf{W}_n) / |\mathbf{W}_c - \mathbf{W}_n|, \\ d &= R_t - R_0, \\ \mathbf{Y}^w &\leftarrow \mathbf{Y}^w + d\boldsymbol{\beta}. \end{aligned} \tag{14}$$

reset j to 0, and go back to step 2);

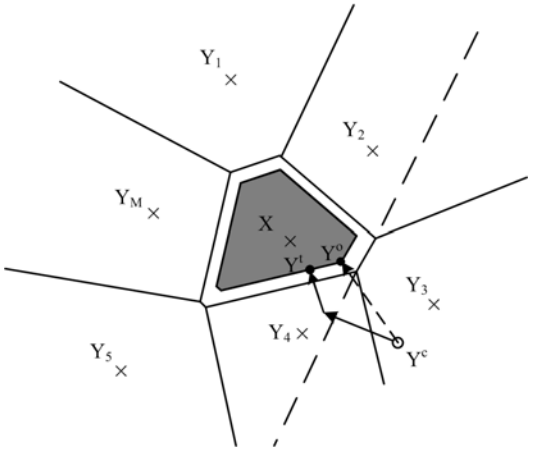


Fig. 4. Geometric representation of the proposed self-adaptive noise optimization (SANO) collusion attack

5) If $R_0 \geq R_t$, then increment j. If $j < j_{max}$, then let $\sigma_d \leftarrow \sigma_d - \delta$ and go back to step 2). Otherwise, terminate.

The rationale behind this strategy can be interpreted by geometric representation of a Voronoi diagram. The reason why we can use this theory to explain the proposed strategy is that, at first, the theory of Voronio diagram is widely applied in many kinds of areas, especially in image processing and pattern recognition. At second, the fingerprints assigned to the users can be regarded as the vectors in N-dimensional space, and they are mutually orthogonal or the correlation between them is very small. Therefore, these vectors can be considered as the Voronio sites in the N-dimensional space. The fingerprint detection can be also looked upon as finding the nearest site from the extracted fingerprint in the Voronio diagram. In Fig.4, each point corresponds to a possible marked vector. The Voronoi diagram indicates the detection regions for these vectors. $\mathbf{Y}_1$ to $\mathbf{Y}_M$ are the vectors which can be detected as containing the colluders, we use $'\times'$ to indicate them. The gray region indicates the set of vectors which make it difficult for the detector to detect any of the colluders in the collusion group. The task of the proposed optimization mechanism is to remove the fingerprint from a noise-free signal $\mathbf{Y}^c$ can be detected as containing the colluders in the collusion set, after processing of the optimization, to a signal $\mathbf{Y}^t$ which can be detected as not. The open circle indicates the initial collected vectors $\mathbf{Y}^c$, the solid dots are the closest points in the gray region. The dotted line and the solid dot illustrate the behavior of an ideal optimization, the vector $\mathbf{Y}^c$ would be directly moved to the closest point in the gray region $\mathbf{Y}^o$. In practice, it is difficult to implement a strategy to find the optimal attacked content. Instead, we use the suboptimal, iterative strategy. In the first iteration, we modify $\mathbf{Y}^c$ so that it will not be detected as containing the colluder involved signal $\mathbf{Y}_3$. In the

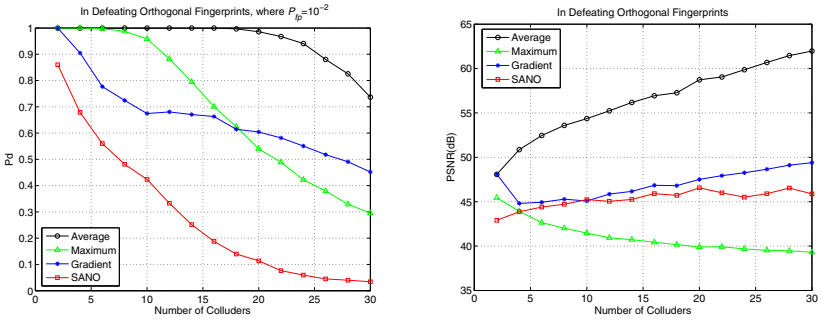


Fig. 5. Performance comparison of the average attack, the maximum attack, the gradient attack and the proposed SANO attack in terms of (a) the detection probability vs. the colluder number, (b) the perceptual quality vs. the colluder number, under the assumption that orthogonal fingerprints are used, where the fingerprint length is 1024, P_{fp} of 10^{-2}

second iteration we modify it so it will not be detected as containing $\mathbf{Y}_4$. This results in a clear content $\mathbf{Y}^t$ within the gray region around $\mathbf{X}$.

4 Experimental Results

In order to demonstrate the performance of the proposed attack in defeating orthogonal fingerprints, we conduct a group of experiment. In the experiment, we consider the fingerprinting system which accommodates as many as 10^3 users and the fingerprints length of 1024. 100 standard images with size of 512×512 from USC-SIPI Image Database [24] and Ground Truth Database [25] are selected as the test samples, including lena, baboon, airplane, couple, watch, fishingboat, etc. The spread-spectrum additive embedding technique [21] is employed to yield the fingerprinted copies. We randomly pick out specified number of colluders to participant in collusion attacks, 40 tests on each image and totally 4000 runs are performed in each iteration. The thresholding correlation detector is used to detect the colluders from the suspicious content after the proposed attack as in [8]. We rewrite the detection probability P_d at here

$$P_d = P\{ \underset{m \in S_c}{T_m} > h\} \tag{15}$$

the threshold h_0 is chosen to yield the desired probability of false positive P_{fp}. We set P_{fp} to 10^{-2} as in Section 2. We choose the target measure value $R_t = 80$.

In [11], the authors provided theoretical analysis on the effectiveness of frequently used collusion attacks and concluded that the average attack is the weakest attack, but the maximum attack as well as the randomized negative attack is the most effective attack. Recently introduced collusion attack, the gradient attack [16], has refined the definition of an optimal attack and demonstrated

strong effect on direct-sequence, uniformly distributed, and Gaussian spread spectrum fingerprints. In our experiment, we expect to make a fair comparison on the attack performance and the perceptual quality introduced between the proposed attack and that of above three collusion attacks: the average attack, the maximum attack, and the gradient attack. In the gradient attack, we select the average attack as $\mathbf{z}'$ and the max-min attack as $\mathbf{z}''$, which is the same as in [16]. Then we apply the selected attacks on the target fingerprinted images. The experimental results are illustrated in Fig.5(a) and (b). From Fig.5(a), we can observe that the SANO attack performs much better than other three attacks, less than three independently marked pieces of content can sufficiently remove the fingerprints of all the participants in the collusion group. From Fig.5(b), we can see that, though the SANO attack is more effective in defeating the fingerprinting system, it also maintains much better perceptual quality of the attacked content. After the SANO attack, the PSNR of the attacked content is above 40dB.

5 Conclusion

In this paper, we have provided theoretical analysis and simulations of the effect of different energies of Gaussian noise added in the noise-free forgery on the detection performance of correlation-based detector and the perceptual quality of attacked content. Based on the analysis and the principal of informed watermark embedding technology, we have proposed an effective collusion attack strategy, *self-adaptive noise optimization (SANO)* collusion attack. Experimental results have shown that the proposed collusion attack strategy is far more effective in defeating orthogonal fingerprints than usually used linear and nonlinear collusion attacks and recently introduced gradient attack. The adversaries can efficiently interrupt orthogonal fingerprints accommodating thousands of users with less than three independent copies of the same content, while maintaining a high perceptual quality of the attacked content after the proposed attack.

We address several remaining questions as open problems: 1) whether there exists some fingerprinting schemes that achieve better collusion-resistance to the SANO collusion attack, 2) is there a more efficient, optimal iterative mechanism that has the ability of effectively removing all fingerprints in the collusion group under the limitation of the perceptual quality, 3) can we investigate the counter measure for the proposed optimal attack strategy to improve the collusion resistance of existing fingerprinting schemes. It is worthwhile to investigate the aforementioned problems in our future work.

Acknowledgement

This work is supported by NSF of China Grants 60873226, 60803112, National 863 Hi-Tech Grant 2009AA01Z411.

References

1. Boneh, D., Shaw, J.: Collusion-secure fingerprinting for digital data. IEEE Transactions on Information Theory 44(5), 1897–1905 (1998)
2. Wang, Z.J., Wu, M., Zhao, H., Liu, K.J.R., Trappe, W.: Resistance of orthogonal Gaussian fingerprints to collusion attacks. In: International Conference on Multimedia and Expo., pp. 724–727 (2003)
3. Trappe, W., Wu, M., Wang, Z.J., Liu, K.J.R.: Anti-collusion fingerprinting for multimedia. IEEE Transactions on Signal Processing 51(4), 1069–1087 (2003)
4. Dittmann, J., Schmitt, P., Saar, E., Schwenk, J., Ueberberg, J.: Combining digital watermarks and collusion secure fingerprints for digital images. Journal of Electronic Imaging 9(4), 456–467 (2000)
5. He, S., Wu, M.: Joint coding and embedding techniques for Multimedia Fingerprinting. IEEE Transactions on Information Forensics and Security 1(2), 231–247 (2006)
6. Cha, B.H., Kuo, C.C.J.: Robust MC-CDMA-Based Fingerprinting Against Time-Varying Collusion Attacks. IEEE Transactions on Information Forensics and Security 4(3), 302–317 (2009)
7. Byung-Ho, C., Kuo, C.C.J.: Analysis of time-varying collusion attacks in fingerprinting systems: Capacity and throughput. In: IEEE International Symposium on Circuits and Systems, ISCAS 2009, pp. 493–496 (2009)
8. Wang, Z.J., Wu, M., Zhao, H.V., Trappe, W., Liu, K.J.R.: Anti-collusion forensics of multimedia fingerprinting using orthogonal modulation. IEEE Transactions on Image Processing 14(6), 804–821 (2005)
9. Wang, Z.J., Wu, M., Trappe, W., Liu, K.J.R.: Group-oriented fingerprinting for multimedia forensics. EURASIP Journal on Applied Signal Processing 14(11), 2 (2004)
10. Stone, H.: Analysis of Attacks on image watermarks with randomized coefficients. NEC Res. Inst., Tech. Rep., vol. 96-045 (1996)
11. Zhao, H.V., Min, W., Wang, Z.J., Liu, K.J.R.: Forensic analysis of nonlinear collusion attacks for multimedia fingerprinting. IEEE Transactions on Image Processing 14(5), 646–661 (2005)
12. Hong, Z., Min, W., Wang, Z.J., Liu, K.J.R.: Performance of detection statistics under collusion attacks on independent multimedia fingerprints. In: Proceedings of 2003 International Conference on Multimedia and Expo., ICME 2003, vol. 201, pp. I-205–I-208 (2003)
13. Moulin, P., Kiyevash, N.: Performance of Random Fingerprinting Codes Under Arbitrary Nonlinear Attacks. In: IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2007, pp. II-157–II-160 (2007)
14. Kiyavash, N., Moulin, P.: A Framework for Optimizing Nonlinear Collusion Attacks on Fingerprinting Systems. In: 2006 40th Annual Conference on Information Sciences and Systems, pp. 1170–1175 (2006)
15. Kiyavash, N., Moulin, P.: Performance of Orthogonal Fingerprinting Codes Under Worst-Case Noise. IEEE Transactions on Information Forensics and Security 4(3), 293–301 (2009)
16. Kirovski, D., Mihcak, M.K.: Bounded Gaussian fingerprints and the gradient collusion attack. In: Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 2, pp. 1037–1040 (2005)
17. Kirovski, D.: Collusion of fingerprints via the gradient attack. In: IEEE International Symposium on Information Theory, Citeseer, p. 2280 (2005)

18. He, S., Kirovski, D., Wu, M.: Colluding Fingerprinted Video Using the Gradient Attack. In: IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2007, vol. 2 (2007)
19. Podilchuk, C.I., Zeng, W.: Image-adaptive watermarking using visual models. IEEE Journal on Selected Areas in Communications 16(4), 525–539 (1998)
20. He, S., Wu, M.: Collusion-Resistant Video Fingerprinting for Large User Group. IEEE Transactions on Information Forensics and Security 2(4), 697–709 (2007)
21. 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)
22. Miller, M.L., Doerr, G.J., Cox, I.J.: Applying informed coding and embedding to design a robust high-capacity watermark. IEEE Transactions on Image Processing 13(6), 792–807 (2004)
23. Lin, W.S., Zhao, H.V., Liu, K.J.R.: A Game Theoretic Framework for Colluder-Detector Behavior Forensics. In: Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, vol. 2, pp. II-721–II-724 (2007)
24. The USC-SIPI Image Database. Electronic Engineering Department, University of Southern California, `http://sipi.usc.edu/database/index.html`, (accessed September 9, 2009)
25. Ground Truth Database. Department of Computer Science and Engineering, University of Washington, `http://www.cs.washington.edu/research/imagedatabase/`, (accessed September 9, 2009)

Privacy Preserving Facial and Fingerprint Multi-biometric Authentication

Esla Timothy Anzaku, Hosik Sohn, and Yong Man Ro

Korea Advanced Institute of Science and Technology, Yuseong-gu, Daejeon 305-701, South Korea

Abstract. The cases of identity theft can be mitigated by the adoption of secure authentication methods. Biohashing and its variants, which utilizes secret keys and biometrics, are promising methods for secure authentication; however, their shortcoming is the degraded performance under the assumption that secret keys are compromised. In this paper, we extend the concept of Biohashing to multi-biometrics – facial and fingerprint traits. We chose these traits because they are widely used, howbeit, little research attention has been given to designing privacy preserving multi-biometric systems using them. Instead of just using a single modality (facial or fingerprint), we presented a framework for using both modalities. The improved performance of the proposed method, using face and fingerprint, as against either facial or fingerprint trait used in isolation is evaluated using two chimerical bimodal databases formed from publicly available facial and fingerprint databases.

Keywords: Privacy preservation, Multi-Biometrics, Multi-Factor Authentication.

1 Introduction

The increase in cases of identity theft calls for the employment of more secure user authentication. User authentication is the process of verifying the identity of a user to determine if the user is the true owner of a claimed identity. Several factors can be used for user authentication. These can be categorized into three groups: knowledge-based factors (based on the knowledge of certain secret information), possession-based factors (based on the possession of certain tokens), and biometrics-based factors (based on certain human psysiological or behavioral characteristics). Two or more of these factors can be combined to form what is known as multi-factor authentication which can yield superior security.

Although multi-factor authentication yields superior security, if they are based on knowledge- and possession-based factors, they still suffer from the disadvantage of not being able to confirm if the authenticating user is the real user or just someone in possession of the valid authentication factors. This is so because these authentication factors can be stolen or even shared. This problem can be mitigated by deploying multi-factor authentication systems based on biometrics; however, the use of biometrics introduces certain challenges that should be solved to fully take advantage of the benefits that biometrics offers.

H.-J. Kim, Y. Shi, and M. Barni (Eds.): IWDW 2010, LNCS 6526, pp. 239–250, 2011.

Researcher being aware of the limitation of biometrics – the difficulty in achieving high authentication accuracy and the lack of privacy preservation – have been researching for ways to improve the accuracy of biometric authentication systems and also to preserve the privacy of users. We will briefly discuss three related works in literature that combine biometrics and secret keys for user authentication. They are the non-invertible transform method, the fuzzy vault method, and the Biohashing method.

The non-invertible transform method, also refered to as cancelable biometrics [1], [2], is based on the transformation of a biometric using a one-way transform function. The parameters of the transform function can be change leading to the generation of revocable templates. Also, template matching is carried out in the transformed domain. The fuzzy vault method [3] is similar to the fuzzy commitment method proposed in [4], except that it works with unorder data sets unlike the fuzzy commitment method that works only with ordered data sets. In the fuzzy vault method, the biometric is used to bind a user's secret key, and only a biometric that is similar to the one used for binding can be used to retrieve the secret key. Implementations of the fuzzy vault schemes were presented in [5], [6], [7], [8], and [9].

Among the above mentioned methods, the Biohashing method [10] and its variants based on user-specific random projection are promising methods. Unlike the non-invertible transform method and the fuzzy vault method that degrade the performance of the base biometric method, an improved performance over the base biometric system has been reported for Biohashing in [11]. In biohashing, user-specific secret keys are used to generate random matrices which are then used to project the extracted biometric feature unto another space to generate revocable templates. The Biohashing method has received a lot of attention, and several of its variation have since been proposed: [12] [13], and [14]. Nevertheless, the major weakness of the Biohashing method was pointed out in [15], where the authors pointed out the unrealistic assumption that no secret key is compromised. They demonstrated that under a more realistic assumption that secret keys are compromised, the performance of the Biohashing method degrades significantly.

To capitalize on the advantages of the Biohashing method, this paper exploits a way to improve the authentication accuracy under a more realistic assumption that secret keys can be compromised. The paper extends the concept of Biohashing to multi-biometrics (facial and fingerprint characteristics). It is logical that such an approach would improve the performance of the Biohashing method, but the challenge remains in finding the appropriate way to combine the facial and fingerprint characteristics in an effective way, considering the different extraction methods. Our choice of the facial and fingerprint biometric characteristics is base on the fact that they are widely used, nevertheless, to the best of our knowledge, very little research attention has been given to designing multi-biometric systems, which preserve the privacy of individuals using these characteristics. The major contributions of this paper are

1. The proposal of a framework for designing a multi-factor authentication system using facial and fingerprint multi-biometric traits,
2. The proposal of a fusion scheme that combines the descrimating abilities of the single biometric traits to form a single feature vector that yields an improved performance over the performance of either the facial or fingerprint trait used in isolation.

The outline of this paper is as follows: In section 2, we presented the framework for the proposed multi-biometric system. We also presented the fusion scheme that is paramount for the success of this framework. Experiments carried out to evaluate the performance of the proposed framework, and the results obtained were presented in section 3. Finally, conclusions were drawn in section 4.

2 Proposed Multi-biometric Authentication

The system diagram of the proposed multi-factor authentication system that uses multi-biometrics (face and fingerprint) is presented in Fig. 1. Like most biometric systems, the enrollment and verification stages are involved. In the enrollment stage, the facial and fingerprint images of an authorized user are captured. Also, a pseudo-randomly generated number, which serves as a secret key is generated for each user. In a practical implementation of this proposed method, the secret key could be stored in a physical device such as a smartcard or Universal Serial Bus (USB) device.

The feature extraction modules extract the fingerprint feature vector $\boldsymbol{x}_E$ and the face feature vector $\boldsymbol{y}_E$. These extracted feature vectors, $\boldsymbol{x}_E$ and $\boldsymbol{y}_E$, are fused

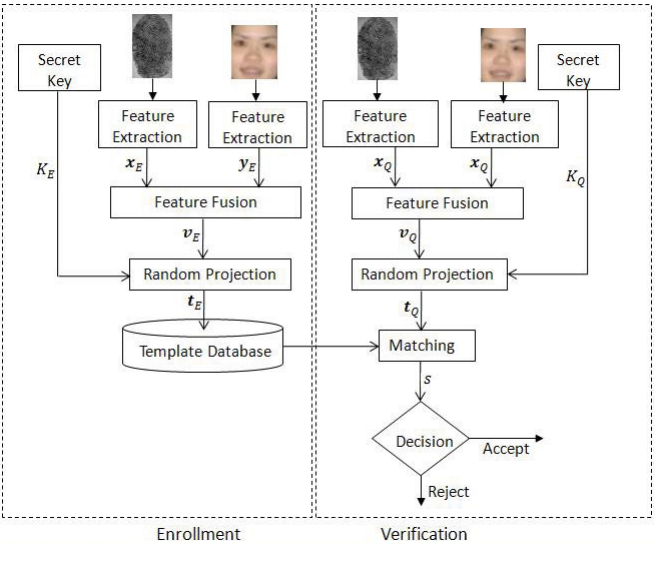


Fig. 1. Block diagram of the proposed multi-factor authentication system based on multi-biometrics

at the feature level by the feature fusion module to form a single vector $\boldsymbol{v}_E$. The proposed fusion scheme requires that $\boldsymbol{x}_E$ and $\boldsymbol{y}_E$ have the same dimension; that is $\boldsymbol{x}_E, \boldsymbol{y}_E \in \mathbb{R}^n$, where n is the dimension of the face and fingerprint feature vectors. Remember that each user is assigned a secret key (in our case, this is a pseudo-randomly generated number). This secret key is used to generate a random matrix $\boldsymbol{R} \in \mathbb{R}^{m \times n}$ which is used to project the fusion vector $\boldsymbol{v}_E$ unto a different space to form the template $\boldsymbol{t}_E$. The generated template of each authorized user is then stored in the database.

In the verification stage, the same processes – feature extraction, feature fusion and random projection – are performed on the query fingerprint and face biometrics to generate a query template $\boldsymbol{t}_Q$. Based on the identity that the user claims, the appropriate enrolled template is retrieved from the database and is compared with the generated query template. This is accomplised by the matching module of figure 1. This module outputs a matching score which is then used for decision making. By comparing the outputted matching score against a predefined system threshold, a decision to classify the user as a genuine user or an impostor is made. That is, a decision to accept or reject the user's claim is made.

In the subsequent subsections, we will briefly present the functions and algorithms used in the various modules of figure 1 .

2.1 Face Feature Extraction

For face feature vector extraction, the Principal Component Analysis (PCA) was used. The PCA is a powerful dimensionality reduction tool which is widely used in face recognition. Its goal is to compute the most meaningful basis that re-epresses a redundant or noisy data set. Given a training set of images (which are represented as vectors by concatenating the row pixel values of the image) $\boldsymbol{Z} = \{Z_i\}_{i=1}^{C}$, where $Z_i = \{\boldsymbol{z}_{ij}\}_{j=1}^{C_i}$, C is the number of classes subjects), and $S = \sum_{i=1}^{C} C_i$ denotes the total number of images in the training set.

The covariance of the dataset can be computed as

$$\boldsymbol{S}_{cov} = \frac{1}{S} \sum_{j=1}^{C} \sum_{i=1}^{C_i} (\boldsymbol{z}_{ij} - \overline{\boldsymbol{z}}) - (\boldsymbol{z}_{ij} - \overline{\boldsymbol{z}})^T, \tag{1}$$

where $\overline{\boldsymbol{z}} = \frac{1}{S} \sum_{i=1}^{C} \sum_{j=1}^{C_i} \boldsymbol{z}_{ij}$ is the mean of the set $\boldsymbol{Z}$. The first $M \leq S$ eigenvectors of $\boldsymbol{S}_{cov}$ corresponding to the largest eigenvalues are selected as the new basis that re-expresses the dataset. If we represent the selected eigenvectors using the matrix Φ, then any image vector $\boldsymbol{x}_{ij}$ from the test database can be transformed by the linear mapping: $\boldsymbol{x}_{ij} = \Phi^T(\boldsymbol{z}_{ij} - \overline{\boldsymbol{z}})$. Refer to [16] for more information on PCA.

2.2 Fingerprint Feature Extraction

The fingerprint feature extraction method is based on the texture-based fingerprint feature extraction method proposed in [17]. The extracted fingerprint

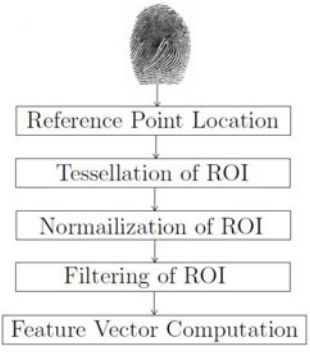


Fig. 2. Block diagram for fingerprint feature extraction

feature vectors are know as *fingercodes*. This method extracts a fixed length feature vector from the fingerprint unlike the traditional minutiae-based method, where the extracted minutiae information is variable; this is the primary reason for the adoption of this method for our research work. Our proposed method relies on the use of a fixed length fingerprint feature vector for fusion with the face feature vector.

The main processes involved in the generation of the fingercodes as proposed in [17] are reference point location, tesselation of the region of interest (ROI), normalization of the ROI, filtering the ROI in eight different directions using a bank of Gabor filters, and 5) feature vector computation from the filtered images. Figure 2 is the block diagram of the processes involved in the generation of fingercodes. The dimension of the fingercode used in this work is 96.

2.3 Fusion of Features

The main goal of the feature level fusion of the features extracted in sections 2.1 and 2.2 is to form a single feature vector $\boldsymbol{v}$ that can yield better authentication accuracy than that of either the face or fingerprint feature vectors used in isolation. The extracted fingerprint feature vector $\boldsymbol{x} \in \mathbb{R}^n$ and face feature vector $\boldsymbol{y} \in \mathbb{R}^n$, due to the different extraction processes, exhibit significant variations. Figure 3 illustrates these variations. It can be observed that the elements of the face and fingerprint feature vectors differ significantly in magnitude. Also, while the elements of the face feature vectors can assume both negative and positive values, those of the fingerprint features assume only positive values.

Our proposed fusion algorithm accounts for these variations in order to form a fusion vector with better performance than the performance obtained when only face or fingerprint features are used. Each component of the fused feature vector is computed as follows:

$$v_k = \alpha \cdot v_{\max}|x_k| + |y_k| \tag{2}$$

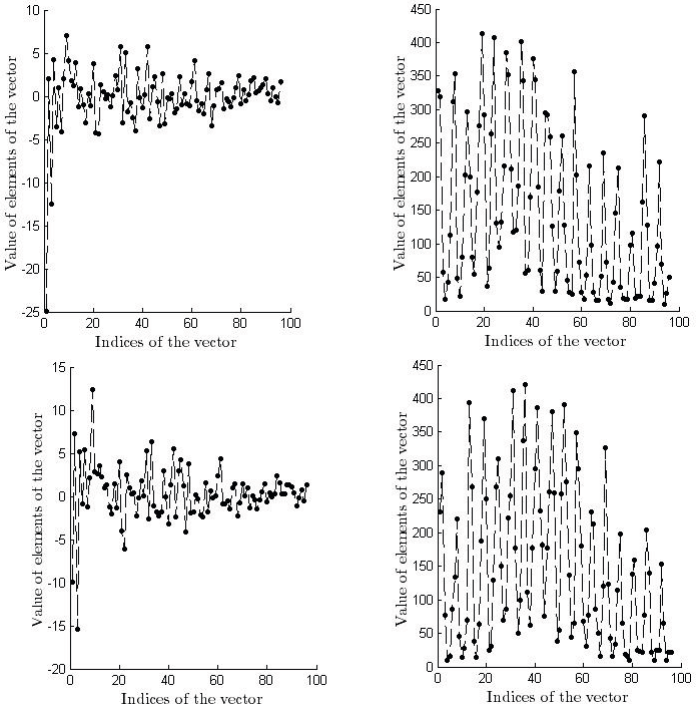


Fig. 3. Plots of the values of the elements of the extracted feature vectors belonging to two users: user A (top row) and user B (bottom row). The plots on the left are for the face feature vectors, while those on the right are for the fingerprint feature vectors.

where v_k, x_k, y_k for $k = 1, \cdots, n$, are the components of the fused feature vector, fingerprint feature vector and face feature vector, respectively. The fingerprint feature vector is normalized to the range [0,1] prior to its use in the (2), α is scalar weighing constant, n is the dimension of the feature vectors, and $v_{\max}$ is computed by averaging the maximum value of the elements of each of the face feature vectors belonging to the training set. Let $\{\boldsymbol{a}_i^m\}, \quad i = 1, 2, \cdots, S \quad and \quad m = 1, 2, \cdots, m_i$, represent the set of face feature vectors of the training set, where S denotes the total number of users, and m_i denotes the number of face images for the user i. Then, $v_{\max}$ can be computed as

$$v_{\max} = \frac{1}{S \cdot m_i} \sum_{i=1}^{S} \sum_{m=1}^{m_i} max\{\mathbf{a}_i^m\}. \tag{3}$$

2.4 Random Projection

Random projection is essentially a dimensionality reduction tool. It is the mapping of a set of points in $\mathbb{R}^n$ into $\mathbb{R}^m$, where the dimension m is usually less than n, such that the distances between all pairs of points are approximately

preserved. The origin of random projection is the Johnson-Lindenstrauss Lemma which is stated below.

Lemma 1. *For any* $0 < \epsilon < 1$ *and any interger* s*, let* m *be a positive interger such that* $m \geq \frac{4\ln(s)}{\frac{\epsilon^2}{2}+\frac{\epsilon^3}{3}}$*. Then for any set* S *of* $s = |S|$ *data points in* $\mathbb{R}^n$*, there exists a linear mapping* $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ *such that for all* $\mathbf{u}, \mathbf{v} \in S$*,*

$$(1+\epsilon)||\mathbf{u}-\mathbf{v}|| \leq ||f((u))-f(\mathbf{v})|| \leq (1+\epsilon)||\mathbf{u}-\mathbf{v}||, \tag{4}$$

where $||\cdot||$ denotes the vector 2-norm. Refer to [18] for the proof of this lemma.

According to this lemma, any set of points in $\mathbb{R}^n$ with a defined metric, can be embedded into $\mathbb{R}^m$ with distortion not greater than ϵ using a randomly generated linear mapping. In our case, the set of points are the fused feature vectors, and the linear mapping is done using user-specific randomly generated matrices, $\boldsymbol{R} \in \mathbb{R}^{m\times n}$. The metric used is the Euclidean's distance. The elements of $\boldsymbol{R}$ are drawn from a Gaussian distribution of zero mean and unit variance. Also, $m = n$; this implies that random projection was not used here as a dimensionality reduction tool, but as a means for binding secret keys to the biometric data, which leads to higher authentication accuracy and revocability. In summary, given a vector $\boldsymbol{v} \in \mathbb{R}^n$, generated by fusing face and fingerprint feature vectors and a randomly generated matrix $\boldsymbol{R}$, the template $\boldsymbol{t}$ is generated by $\boldsymbol{t} = \boldsymbol{R}\boldsymbol{v}$. Note that 5 templates for each user are generated to account for the rotation of fingerprint images.

2.5 Template Matching

Matching two templates is based on the Euclidean distance between them. The Eclidean distance between the query and enrolled templates are computed. These distances are compared against a pre-determined threshold to decide whether to accept or reject a claim.

2.6 Template Revocability and Privacy

User-specific random projection provides for the revocability of templates; if the template of a user is compromised, a new secret key can be used to generate a new template for the user. For privacy protection, it is desirable for the original biometric data to be hard to compute from the template when the template and the secret key are compromised.

Consider the random projection equation: $\boldsymbol{t} = \boldsymbol{R}\boldsymbol{v}$, $\boldsymbol{t}, \boldsymbol{v} \in \mathbb{R}^n$. If the secret key and the template are known (compromised), $\boldsymbol{v}$ can be computed by $\boldsymbol{v} = \boldsymbol{R}^{-1}\boldsymbol{t}$. But since each component of $\boldsymbol{v}$ is computed by $v_k = \alpha \cdot v_{\max}|x_k| + |y_k|$, it will be difficult to compute the constituent feature vectors $\boldsymbol{x}$, $\boldsymbol{y}$. This difficulty in computing the constituent biometric data enhances the privacy protection of the proposed method.

3 Experiments and Result

To evaluate the performance of the proposed multi-factor authentication system using multi-biometrics, we carried out experiments under two scenarios: scenario 1 and scenario 2.

- **Scenario 1:** Scenario 1 is based on the assumption that the secret keys are not compromised. This scenario measures the performance of the proposed system under an ideal condition.
- **Scenario 2:** Under this scenario, a more realistic assumption is made. Here each imposter, an unauthorized user trying to claim the identity of an authorized user, is assumed to have the correct secret key of the claimed identity.

Databases: Evaluating the performance of our proposed method requires the use of bimodal biometric databases. Since we could not lay our hands on bimodal biometric databases containing face and fingerprint images belonging to the same user, we simulated two such databases using publicly available face and fingerprint databases, namely, Bimodal DB1 and Bimodal DB2. The details of these two databases are represented in table 1.

Bimodal DB1 was created by coupling fingerprint images from the DB1 of the Fingerprint Verification Competition (FVC2000-DB1) [20] with the face images from the Carniege Mellon University's pose, illumination and expression (CMU PIE) database [19]. We randomly selected 10 frontal face images per subject from the first partition making a total of 680 images. The FVC2000-DB1 comprises of fingerprints of 110 subjects with 8 fingerprints per subject. Fingerprints from 68 subjects were randomly selected to form Bimodal DB1. Bimodal DB2 was created by coupling fingerprint images of FVC2000-DB2 with face recognition data of the University of Essex [21]. The face database consist of frontal images from 156 subjects, mostly within the ages of 18 and 20, of various races. There are 20 images per subject. For our experiments, 110 images and 10 images per subject were randomly selected. The FVC2000-DB2 comprises of fingerprints of 110 subjects with 8 fingerprints per subject. Before applying PCA, based on the coordinates of the eyes, the face images were aligned and cropped to

Table 1. Details of Bimodal DB1 and Bimodal DB2 showing the partitioning for training and testing samples

Methods	**Bimodal DB1**	**Bimodal DB2**
No. of users	68	110
No. of training samples Per user (face)	5	5
No. of training samples per user (fingerprint)	3	3
No. of test samples per user (face and fingerprint)	5	5

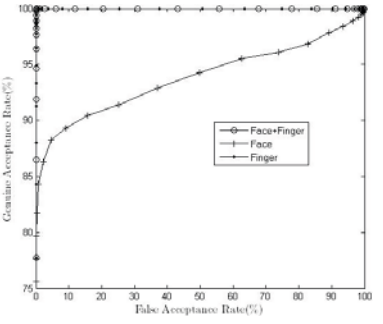


Fig. 4. ROC curve of Bimodal DB1 for scenario 1. It plots GAR against FAR for for three cases: only face, only fingerprint, and our proposed method using face and fingerprint. Here, $\alpha = 0.6$.

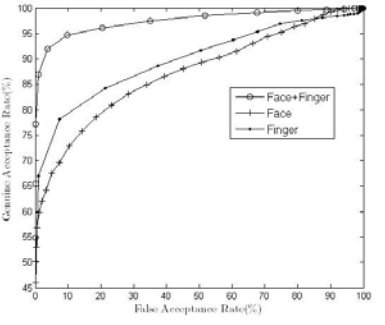


Fig. 5. ROC curve of Bimodal DB1 for scenario 2. It plots GAR against FAR for for three cases: only face, only fingerprint, and our proposed method using face and fingerprint. Here, $\alpha = 0.6$.

Table 2. EER for Bimodal DB 1 computed for only face, only fingerprint, and our proposed face and fingerprint

Methods	Scenerio 1 EER (%)	Scenerio 2 EER (%)
Face only	7.92	18.70
Fingerprint only	0	14.58
Face + Fingerprint (proposed)	0	5.92

44×44 pixels, converted to gray images, and the rows of the cropped images are concatenated to form a vector of dimension, 1936. The vectors are further normalized to zero mean and unit variance.

Evaluation Metrics: Two evaluation metrics were used to quantify the results of the experiments: the Equal Error Rate (EER) and Receiver Operation Curve (ROC), plotting the Genuine Acceptance Rates (GAR) against the False Acceptance Rates (FAR). We chose to plot the ROC curve of GAR against FAR for the evaluation of the results of our experiments because it shows vividly the tradeoff between GAR and FAR for various operating points (thresholds). On the other hand, the EER characterizes the performance of a whole biometric system using a single value; this makes for easier comparisons between various biometric systems.

Results: In computing the ROC curves and the EER of Bimodal DB1 and Bimodal DB2, each test template is used as an enrolled template once, while the others are used as query templates to compute the genuine and impostor scores (in our case the Euclidean distances). Figures 4, and 5 show the ROC curves of Bimodal DB1 for scenario 1 and scenario 2, respectively. The high GAR at

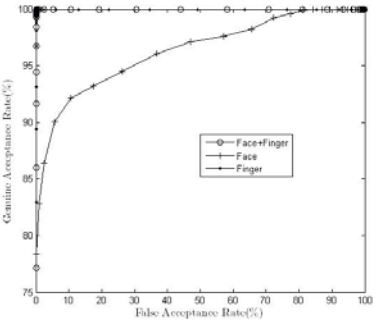


Fig. 6. ROC curve of Bimodal DB2 for scenario 1. It plots GAR against FAR for for three cases: only face, only fingerprint, and our proposed method. Here, $\alpha = 0.9$.

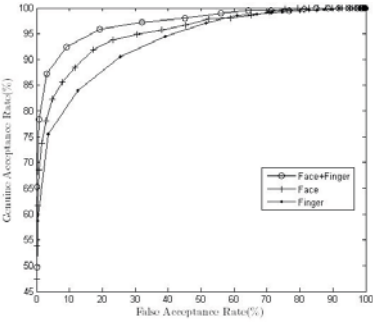


Fig. 7. ROC curve of Bimodal DB2 for scenario 2. It plots GAR against FAR for for three cases: only face, only fingerprint, and our proposed method. Here, $\alpha = 0.9$.

Table 3. EER for Bimodal DB 2 computed for only face, only fingerprint, and our proposed face and fingerprint

Methods	**Scenerio 1 EER (%)**	**Scenerio 2 EER (%)**
Face only	7.80	11.13
Fingerprint only	0	14.03
Face + Fingerprint (proposed)	0	7.89

almost zero FAR for scenario 1, especially for fingerprint and the multi-modal biometrics can be noted in figure 4. For scenario 2, we notice the degradation in performance for the face, fingerprint, and even the fused vectors (the proposed method), however, it can be seen that the performance of the fused vectors is better than those of the faces and fingerprints (see figure 5).

Likewise, the improvement in performance of the proposed method over the performances when using just the face or fingerprint for Bimodal DB2 is evident, especially under scenerio 2; see figures 6, and 7 that show the performances of Biomodal DB2 under scenario 1 and scenario 2, respectively. The EER for Bimodal DB1 and Bimodal DB2 under scenarios 1 and 2 are presentated in 3 and 3, respectively. Again, the improvement in performance when using multi-biometrics is evident.

It should, however, be noted that the parameter, α from (2, is crucial to the success of the fusion scheme; therefore, it has to be chosen carefully. The parameter, α, is highly depended on the individual classification performances of the face and fingerprint biometric subsystem. α is used to give more advantage to the feature vectors of the biometric modality that has better inter-class discrimatory power; thereby, making it contribute more in the fusion scheme. For instance, if the classification performance of the face feature vectors is

significantly higher than that of the fingerprint feature vectors, we can choose α so as to weigh down the values of the elements of the fingerprint feature vectors, or vice versa. To determine the most suitable value of α, we computed the EER values for different α values using the training data (3 face and 3 fingerprint images per user). Figures 8 and 9 show the plot of EER against α for Bimodal DB1 and Bimodal DB2 respectively. It can be observed that the EER is best when α ranges from 0.5 to 0.8 for Bimodal DB1, and from 0.8 to 1.0 for Bimodal DB2. For our experiment whose results have been shown in figures: 4, 5, 7 6, and tables: 3 and 3, α was set to 0.6 for Bimodal DB1 and 0.9 for Bimodal DB2.

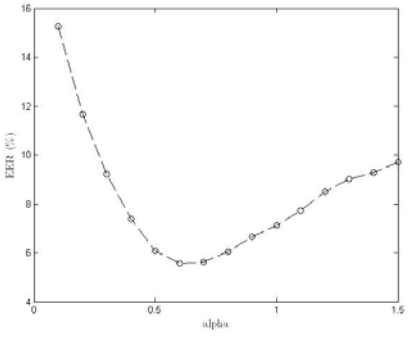


Fig. 8. A plot of EER againt alpha (α) for Bimodal DB1 showing the variation how EER varies for different values of alpha

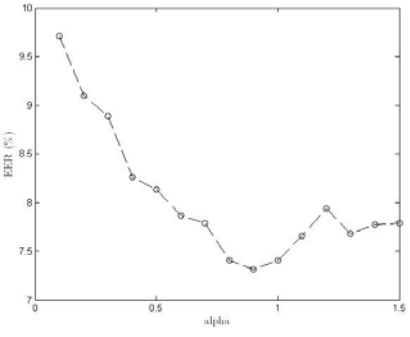


Fig. 9. A plot of EER againt alpha (α) for Bimodal DB2 showing the variation how EER varies for different values of alpha

4 Conclusion

In this paper, we presented a framework that extends the concept of Biohashing to multi-biometrics in order to design a privacy preserving multi-factor authentication system. Our major goal is to improve the performance of the Biohashing method under the assumption that secret keys are stolen by using multi-biometrics. The motivation for the choice of face and fingerprint biometric traits is founded by the wide use of these traits, howbeit, little research attention has been paid to designing multi-biometric systems, which protect the privacy of individuals and provide for revocability of templates, based on them.

The main processes involved in the presented framework include feature extraction, feature level fusion of face and fingerprint features, random projection, and template matching. The success of the presented framework is largely depended on the fusion scheme. To this end, we proposed a feature level fusion scheme that effectively combines the features to yield a fused feature vector with enhanced privacy protection and improved authentication accuracy, and the performance of the presented framework was evaluated experimentally using two chimerical databases.

References

1. Bolle, R.M., Connel, J.H., Ratha, N.K.: Biometric perils and patches. Pattern Recognition 35(12), 2727–2738 (2002)
2. Ratha, N.K., Connell, J., Bolle, R.M., Chikkerur, S.: Cancelable biometrics: a case Study in fingerprints. In: Proceedings of the 18th International Conference on Pattern Recognition (2006)
3. Juels, A., Sudan, M.: A Fuzzy vault scheme. In: Proc. International Symposium on Information Theory (2002)
4. Juels, A., Wattenbeg, M.: A Fuzzy commitment scheme. In: Sixth ACM Conference on Computer and Communications Security, pp. 28–36 (1999)
5. Clancy, T.C., Kiyavash, N., Lin, D.J.: Secure smartcard fingerprint authentication. In: ACM SIGMM, Multimedia, Biometrics Methods and Applications Workshop (2003)
6. Uludag, U., Pankanti, S., Jain, A.K.: Fuzzy vault for fingerprints. In: Kanade, T., Jain, A., Ratha, N.K. (eds.) AVBPA 2005. LNCS, vol. 3546, pp. 310–319. Springer, Heidelberg (2005)
7. Chung, Y., Moon, D., Lee, S., Jung, S., Kim, T., Ahn, D.: Automatic alignment of fingerprint features for fuzzy fingerprint vault. In: Feng, D., Lin, D., Yung, M. (eds.) CISC 2005. LNCS, vol. 3822, pp. 358–369. Springer, Heidelberg (2005)
8. Yang, S., Verbauwhede, I.: Automatic secure fingerprint verification system based on fuzzy vault scheme. In: Proc. ICASSP, vol. 5, pp. 609–612 (2005)
9. Nagar, A., Nandakumar, K., Jain, A.K.: Securing fingerprint template: fuzzy vault with minutiae descriptors. In: Proc. ICPR (2008)
10. Goh, A., Ngo, D.C.L.: Computation of cryptographic keys from face biometrics. In: Lioy, A., Mazzocchi, D. (eds.) CMS 2003. LNCS, vol. 2828, pp. 1–13. Springer, Heidelberg (2003)
11. Teoh, A.B.J., Ngo, D.C.L., Goh, A.: Biohashing: a novel approach for dual-factor authentication. Pattern Analysis and Applications 7(3), 255–268 (2004)
12. Teoh, A.B.J., Goh, A., Ngo, D.C.L.: Random multispace quantization as an analytic mechanism for bioHashing of biometric and random identity inputs. IEEE Transaction. on PAMI 28(12), 1892–1901 (2006)
13. Wang, Y., Plataniotis, Y.: Face based biometric authentication with changeable and privacy preserving templates. In: Biometric Symposium, BSYM (2007)
14. Sohn, H., Ro, Y.M.: Biometric authentication using augmented face and random projection. In: Proceedings of the 3rd IEEE International Conference on Biometrics: Theory, applications and systems, pp. 74–79 (2009)
15. Kong, A., Cheung, K.H., Zhang, D., Kamel, M., You, J. : An analysis of Biohashing and its variants. Pattern Recognition, 1359–1368 (2006)
16. Turk, M., Pentland, A.: Eigenfaces for recognition. Journal of Cognitive Neuroscience 13(1), 71–86 (1991)
17. Jain, A.K., Prabhakar, S.: Filterbank-based Fingerprint matching. In: Proc. IEEE WCMC, pp. 1435–1440 (2000)
18. Arriaga, R.I., Vempala, S.: An algorithmic theory of learning robust concepts and random project. In: Proc. 40th Annual Symposium on Foundations of Computer Science, pp. 616–623 (1999)
19. Sim, T., Baker, S., Bsat, M.: The CMU pose, illumination, and expression (PIE) Database of human faces CMU-RI-TR-01-02, pp. 1–17 (2002)
20. Maltoni, D., Maio, D., Jain, A.K., Prabhakar, S.: Handbook of fingerprint Recognition. Springer, New York (2003)
21. http://cswww.essex.ac.uk/mv/allfaces/face96.zip

Blind Linguistic Steganalysis against Translation Based Steganography

Zhili Chen[1,2,⋆], Liusheng Huang[1,2], Peng Meng[1], Wei Yang[1,2], and Haibo Miao[1]

[1] NHPCC, School of CS. & Tech., USTC, Hefei 230027, China
[2] Suzhou Institute for Advanced Study, USTC, Suzhou, 215123, China
zlchen3@ustc.edu.cn

Abstract. Translation based steganography (TBS) is a kind of relatively new and secure linguistic steganography. It takes advantage of the "noise" created by automatic translation of natural language text to encode the secret information. Up to date, there is little research on the steganalysis against this kind of linguistic steganography. In this paper, a blind steganalytic method, which is named natural frequency zoned word distribution analysis (NFZ-WDA), is presented. This method has improved on a previously proposed linguistic steganalysis method based on word distribution which is targeted for the detection of linguistic steganography like nicetext and texto. The new method aims to detect the application of TBS and uses none of the related information about TBS, its only used resource is a word frequency dictionary obtained from a large corpus, or a so called natural frequency dictionary, so it is totally blind. To verify the effectiveness of NFZ-WDA, two experiments with two-class and multi-class SVM classifiers respectively are carried out. The experimental results show that the steganalytic method is pretty promising.

1 Introduction

Enlightened by the word distribution analysis (WDA) linguistic steganalysis method proposed by Chen *et al.* [1], this paper presents an improved method which can blindly distinguish natural texts, machine translated texts and stego texts that generated by translation based steganography (TBS) [2] [3] [4]. The key idea is to examine the distribution characteristics of words that are in the same natural frequency zone (NFZ). When using a machine translator to translate texts from one language to anther, as the translator uses words somehow in a mechanical way, the translated texts have an inherent structural style determined by the machine translator. Therefore, the stego texts generated by TBS have a mixed structure style that determined by all the translators used. Similarly, the people's writing texts, which we call natural texts, also have an inherent structural style.

The paper evaluates the potential of distinguishing the structural styles of different classes of texts. Our basic observation is that the structural style can

⋆ Corresponding author.

H.-J. Kim, Y. Shi, and M. Barni (Eds.): IWDW 2010, LNCS 6526, pp. 251–265, 2011.

be well represented by the distributions of words in the same NFZs. In order to characterize the inherent structural style differences of different classes of texts, we mainly focus on the investigation of distribution characteristics of different NFZs of words. In our work, we first attribute the words in the text being analyzed into different NFZs according to their natural frequencies, which are the word frequencies obtained from a large corpus. Next, we find the positions of the words in the same NFZ and calculate the average and the variance of the distances between neighboring words. Finally, we use the distance averages and variances of all the NFZs to form the classification feature vector representing the structural style, based on which we use a SVM classifier to distinguish between different classes of texts.

The NFZ-WDA method has improved on the previous WDA method by introducing the notion of NFZs and refining the word distribution characteristics. The WDA method analyzes the testing texts entirely based on the texts themselves, while the NFZ-WDA method applies in the analysis a frequency criterion, the notion of NFZs, which provides a more correct direction. Additionally, the refinement of the word distribution characteristics preserves more structural information. As a result, the improved method makes it more possible to effectively analyze the stego texts generated by TBS.

The organization of the paper is as follows. Section 2 briefly covers the basic operations of the TBS algorithm and the previous steganalytic methods against TBS. Section 3 focuses on the description of the blind linguistic steganalysis method, NFZ-WDA. In Section 4, we present the results of our steganalytic experiments and give some related analysis. In Section 5, there are some discussions about NFZ-WDA. Finally, conclusions are presented in Section 6.

2 Related Work

2.1 Translation Based Steganography

Compared to traditional linguistic steganography methods such as nicetext and texto, translation based steganography (TBS), which was introduced by Grothoff *et al.*, is a novel and relatively secure method. TBS hides information in the "noise" created by automatic translation of natural language text. The key idea of TBS is "When translating a non-trivial text between two natural languages, there are typically many possible translations. Selecting one of these translations can be used to encode information." [2] Because there are frequent errors in legitimate automatic translated texts, it is difficult for a computer to distinguish the additional errors inserted by an information hiding algorithm and the normal noise associated with translation. Therefore, steganalysis against TBS is a challenging work.

There are two versions of TBS, lost in translation (Lit)[2] and lost in just the translation (LiJtT)[3]. Lit works as follows. At first, the sender picks up a cover text in the source language, which does not have to be kept secret and can be obtained from public sources. Then, the sender translates the source text to target language sentence by sentence using several translators and encodes

the hidden messages in this process by selecting one proper translator for each source sentence.

The work flow of LiJtT is illustrated in Fig. 1. LiJtT has improved on Lit to one that allows the receiver recovering the hidden message using only stego texts and a secret key. LiJtT works as follows. First, the sender generates multiple translations for a given cover text and uses a secret key which is shared between the sender and the receiver to hash each translated sentence into a bit string. Second, the lowest h bits of the hash string, referred to as header bits, are interpreted as an integer $b \geq 0$ and then the sentence whose lowest $[h+1, h+b]$ bits match with the bit-sequence to be encoded is selected. Finally, when the receiver receives a stego text, he breaks the received text into sentences, applies a keyed hash to each sentence and interprets the lowest $[h+1, h+b]$ bits of each hash string as the next b bits of the hidden message.

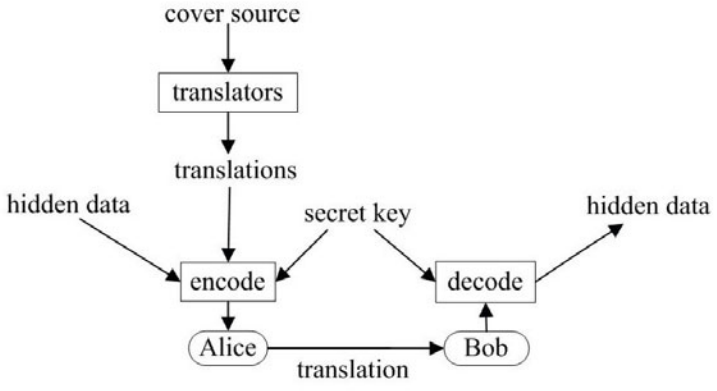


Fig. 1. Work Flow of LiJtT [3]

2.2 Previous Linguistic Steganalysis Methods of TBS

The steganalytic method on TBS presented by Meng *et al.* [5] needs to know the machine translator (MT) set and the cover text language. As the translator set is the private key of TBS, the method has to seek the possible candidate translator set before the steganalysis. This weakens the commonality of the method. Furthermore, the steganalytic process of the method has to translate the cover text two times by every translator, which may be too expensive for large-scale deployment.

The steganalytic method proposed by Meng *et al.* [6] no longer needs to know the MT set used by the TBS encoder in the steganalysis. The method is based on the fact that there are fewer high-frequency words in stego texts than in normal texts. By defining the feature vector related to the frequencies of the high-frequency words, the authors use a two-class SVM classifier to detect the stego texts and normal texts.

The latter steganalytic method described above has achieved a promising steganalysis, but it still suffers some drawbacks. Firstly, it is not a completely blind steganalysis, that is, it still needs some prepared resources, namely the sets of high-frequency words and n-grams, which seem to be more or less related to the TBS. Next, the countermeasure of this steganalytic method still seems plausible, especially by using only one-to-one words and 2-grams in the TBS. Finally, the detection accuracy needs to be improved, particularly when the text size is less than 20KB. The steganalytic method proposed in this paper can properly overcome these drawbacks.

3 Blind Linguistic Steganalysis Method

3.1 Previous WDA Method

In the WDA method [1], the spread degree (SD) of a word is defined as the variance of its positions in the testing text. Then the average and variance of the SDs of words in the testing text are used to form the classification feature vector. Finally, a two-class SVM classifier is applied to classifying the testing text to normal texts and stego texts.

Though the WDA method summarizes the structural information of the testing text so much that a lot of detailed information are lost, it still works quite effectively for detection of linguistic steganography methods such as nicetext [7], texto [8] and Markov chain based [9]. Preserving the correct syntax and coherent semantics well, the TBS generates more natural-like texts. As a result, the stego texts of TBS are more difficult to analyze. The WDA method has no effect on the steganalysis of TBS as we will see in Section 4.

3.2 The Improvement of WDA Method

The steganalytic method proposed in this paper, NFZ-WDA, has improved on the WDA method in three aspects as follows.

First, the natural frequency dictionary is used in NFZ-WDA while no language resources are used in WDA. The natural frequency dictionary is used as a guide for the NFZ partition and provides a frequency criterion by which the distribution features of words with a certain range of natural frequencies can be calculated.

Second, positions of the words in the same NFZ are used as a whole to calculate the word distribution features in NFZ-WDA while only the positions of the same word are used in WDA. By doing this, NFZ-WDA can abstract more invariant features about the word distribution.

Third, the distance average and variance of each NFZ are used to form the classification feature vector in NFZ-WDA while in WDA, the average and variance of the spread degrees of words in the testing text are used. NFZ partition makes the description of word distribution is more accurate and detailed.

3.3 NFZ-WDA Method

Definitions Before introducing the NFZ-WDA method, some definitions have to be clarified as follows.

Natural Frequency is a word's general frequency in the natural texts. The natural frequency of a word represents the occurring probability of the word in the natural texts.

Natural Frequency Dictionary is the set of natural frequencies of a certain word dictionary. It can be evaluated by processing a large corpus and calculating the word frequencies.

Natural Frequency Zone (NFZ) is the set of words of a certain natural frequency range. That is, the words in a NFZ have approximative natural frequencies.

In our research, we attribute the words to different NFZs according to their natural frequencies and investigate the distribution characteristics of words in each NFZ.

Text Formulation. As done in the paper [1], we formalize a text as follows.

$$T = \{w_0, w_1, w_2, ..., w_{n-1}\} \tag{1}$$

Here, w_i, $0 \leq i \leq n-1$ is the $(i+1)$ th word of the text.Then, the word position of word w_i is defined as.

$$l_i = \frac{i}{n} \tag{2}$$

Obviously, $0 \leq l_i \leq 1$.

In the NFZ-WDA method, we make use of the natural frequency dictionary and assign words to different NFZs according to their natural frequencies. The natural frequency dictionary in fact is a function mapping words to their natural frequencies. Given this function as $y = f(x)$, where x is any word and y is its corresponding natural frequency, the natural frequency set of the text T can be obtained as follows.

$$F = \{f(w_0), f(w_1), f(w_2), ..., f(w_{n-1})\} \tag{3}$$

Then, denoting the maximal natural frequency in the natural frequency dictionary by f_{max}, the NFZs with equal size for text T are formulated as

$$Z_k = \{w_i | kL \leq f(w_i) < (k+1)L, w_i \in T\}, k = 0, 1, ..., K-1 \tag{4}$$

Here L is the NFZ size and $K = \lceil \frac{f_{max}}{L} \rceil$ is the count of NFZs.

After having formulated the NFZs, the text T can be regarded as the constitution of words from all the NFZs. Suppose that the text T contains the words from NFZ Z_k n_k times, we have

$$\sum_{k=0}^{K-1} n_k = n \tag{5}$$

The word position set of NFZ Z_k can be denoted by

$$L(Z_k) = \{l_0^{(k)}, l_1^{(k)}, l_2^{(k)}, ..., l_{n_k-1}^{(k)}\} \tag{6}$$

subject to $l_0^{(k)} < l_1^{(k)} < l_2^{(k)} < ... < l_{n_k-1}^{(k)}$. Particularly, let $l_{-1}^{(k)} = 0$ and $l_{n_k}^{(k)} = 1$.

Let Z denote the set of NFZs, L denote the set of $L(Z_k)$. That is to say

$$Z = \{Z_k | k = 0, 1, ..., K-1\} \tag{7}$$

$$L = \{L(Z_k) | k = 0, 1, ..., K-1\} \tag{8}$$

We can represent the text T in another form, where

$$T = \langle Z, L \rangle \tag{9}$$

Classification Feature Vector. Each class of texts has a unique structural style. The more accurately we can describe the structural style, the more effectively we can distinguish from different classes of texts. Our key observation is that the structural style of a certain class of texts can be well represented by its word distribution characteristics. So, if we can accurately describe the word distribution characteristics of a text, we can identify its class with a high efficiency.

In order to measure the distribution of words, we first define the distance of words w_i and w_j in the text T as

$$d_{ij} = d_{ji} = |l_i - l_j| \tag{10}$$

Then, we define in the NFZ Z_k the average and variance of the distances of neighboring words, which are denoted by α_k and γ_k, as

$$\alpha_k = \frac{1}{n_k + 1} \sum_{i=0}^{n_k} d_{i,i-1}^{(k)} = \frac{1}{n_k + 1}(1 - 0) = \frac{1}{n_k + 1} \tag{11}$$

$$\gamma_k = \frac{1}{n_k + 1} \sum_{i=0}^{n_k} (d_{i,i-1}^{(k)} - \alpha_k)^2 \tag{12}$$

Here, $d_{i,i-1}^{(k)} = |l_i^{(k)} - l_{i-1}^{(k)}|$ denotes the distance of $w_i^{(k)}$ and $w_{i-1}^{(k)}$, which are the $(i+1)$ th and i th words in the NFZ Z_k. The borders are defined as $d_{0,-1}^{(k)} = |l_0^{(k)} - l_{-1}^{(k)}| = |l_0^{(k)} - 0| = l_0^{(k)}$ and $d_{n_k,n_k-1}^{(k)} = |l_{n_k}^{(k)} - l_{n_k-1}^{(k)}| = |1 - l_{n_k-1}^{(k)}| = 1 - l_{n_k-1}^{(k)}$.

Finally, we define the classification feature vector Γ as

$$\Gamma = \{(\alpha_k, \gamma_k) | k = 0, 1, ..., K-1\} \tag{13}$$

Therefore, vector Γ contains the word distribution information of words in each NFZ. As the size of NFZ decreases, it can describe the text structural style to an inch.

Method Description. In the steganalysis, there are normally three main processes employed: training, testing and classifying. We apply the following procedure to both training and testing processes to abstract the classification feature vector from the processed text.

Step 1: Word Position Computation. The analyzer reads the given text T, parsing it, splitting it into words and obtains the word set T in the form of Eq. (11). Then the analyzer computes the word position of each word in the given text using Eq. (12). See Alg. 1.

Step 2: NFZ Partition. The analyzer loads the natural frequency dictionary, retrieves the natural frequencies of the words in the text T and attributes them to different NFZs according to their natural frequencies. In this step we can get Z_k and then the corresponding word position set $L(Z_k)$, among which $k = 0, 1, ..., K$. See Alg. 2.

Step 3: Distance Average and Variance Computation. Using each word position set $L(Z_k)$, the analyzer computes the distances between any two neighboring words in each NFZ Z_k, namely gets the $d^{(k)}_{i,i-1}$, where $i = 0, 1, ..., n_k$. Then, the analyzer computes the distance average and variance of each NFZ Z_k according to Eq. (11) and (12). See Alg. 3.

Algorithm 1. Word Position Computation

Splits the text T into words and gets the total word count n
for all $w_i \in T$ **do**
  $l_i \leftarrow \frac{i}{n}$
end for

Algorithm 2. NFZ Partition

Loads natural frequency dictionary and retrieves the natural frequencies of the words in the text T to get natural frequency set F.
$Z_k \leftarrow \emptyset$
$L(Z_k) \leftarrow \emptyset$
for all $w_i \in T$ **do**
  $k \leftarrow \lceil \frac{f(w_i)}{L} \rceil$
  $Z_k \leftarrow Z_k \cup \{w_i\}$
  $L(Z_k) \leftarrow L(Z_k) \cup \{l_i\}$
end for

Going through all these three steps described above, the analyzer converts the given text T to a classification feature vector Γ as formulated in Eq. (13). The vector Γ is then used as the exclusive basis for the text classification.

Having introduced the classification feature extraction algorithms, we move to the description of the whole steganalytic system. Fig. 2 shows the NFZ-WDA system framework. As described previously, the framework mainly includes training, testing and classifying three parts. The former two parts are constituted of the

Algorithm 3. Distance Average and Variance Computation

for all $Z_k \in Z$ **do**
  $\alpha_k \leftarrow \frac{1}{n_k+1}$
  $\gamma_k \leftarrow 0$
  for $i = 0$ to n_k **do**
    $d_{i,i-1}^{(k)} \leftarrow |l_i - l_{i-1}|$
    $\gamma_k \leftarrow \gamma_k + (d_{i,i-1}^{(k)} - \alpha_k)^2$
  end for
  $\gamma_k \leftarrow \sqrt{\frac{\gamma_k}{n_k+1}}$
end for

same three-step classification feature extraction, while the latter part is an existent SVM classifier [10]. The arrowhead represents the flow of data, among which the dashed line arrowhead indicates that the training process can be omitted if the classification model has already been prepared. The thick dashed rectangle indicates the whole steganalytic system.

While analyzing, both the training and testing texts go through the three-step classification feature extraction, resulting in training and testing feature set. Then the training feature set is used for the training of the SVM classifier

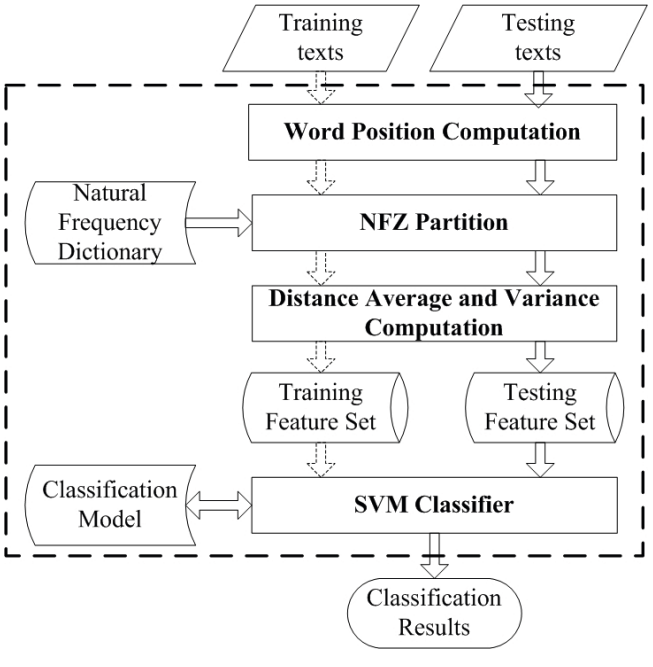


Fig. 2. NFZ-WDA System Framework

and generating of the classification model. The testing feature set is used for the classification. The classification results indicate the steganalytic conclusions.

In the system framework, apart from the training and testing texts, there is only a natural frequency dictionary used, which is applied to NFZ partition in the process of classification feature extraction. Furthermore, the natural frequency dictionary is obtained from a large corpus and it has nothing to do with TBS. As a result, the NFZ-WDA System is a totally blind steganalytic system.

4 Experiments and Analysis

In our experiments, texts were translated from German to English using the LiT prototype, with no semantic substitution, no article and preposition replacement enabled and no "badness threshhold" [2]. The translator set of LiT includes Systran, Google, Promt translators.

We have built the training and testing text sets from natural language texts, machine translated texts and stego-texts. The natural language texts which are in English were extracted from a corpus of 1000 classical English novels, the machine translated texts were generated using Systran, Google and Promt translators and the German language texts which are used as cover texts and the source language texts of translation came from the Europarl corpus [11]. All the experimental texts are in the form of text segment with a size of about 20KB. Our detector utilizes only the first thousands of bytes indicated by a text size parameter, e.g., if the text size is 5KB, it means that the detector only uses the first 5KB text of each experimental text of size 20KB.

In order to measure the detection, we use some rates that are defined as follows.

$$\text{False Rate} = \frac{\text{number of non-stego texts identified as stego}}{\text{total number of non-stego texts}}$$

$$\text{Missing Rate} = \frac{\text{number of stego texts identified as non-stego}}{\text{total number of stego texts}}$$

$$\text{Accuracy Rate} = \frac{\text{number of testing texts identified as their true type}}{\text{total number of testing texts}}$$

First of all, we use WDA method [1] to distinguish between stego-texts and natural language texts or one kind of the machine translated texts. Texts of about 20KB size are used. Tab. 1 shows the experimental results. Obviously, The WDA method has little effect on the steganalysis of TBS.

Then, in order to verify the feasibility and effectiveness of the improved steganalytic method, we have designed yet two experiments. The first one is the experiment distinguishing between stego-texts and natural language texts or one kind of the machine translated texts using a two-class SVM classifier. The second is the experiment distinguishing among the stego-texts, natural language texts and all kinds of machine translated texts using a multi-class SVM classifier.

Table 1. Experimental Results of WDA. The "Train" and "Test" columns show the numbers of texts used in the training and testing for both classes. "FR", "MR" and "AR" are short for "False Rate", "Missing Rate" and "Accuracy Rate" respectively.

Text Size	Class-1	Class-2	Train	Test	FR(%)	MR(%)	AR(%)
	Natural	Stego	60/60	172/202	72.67	31.19	49.73
20KB	Promt	Stego	60/60	211/202	58.77	39.11	50.85
(abt 3300	Google	Stego	60/60	185/202	75.14	13.37	57.11
words)	Systran	Stego	60/60	210/202	84.29	12.38	50.97

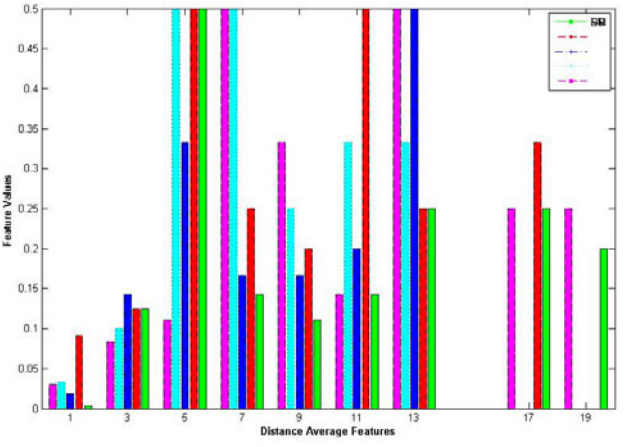


Fig. 3. The distributions of the first 10 distance average features for each text class. The X axis value $2k-1$ represents the distance average feature of kth NFZ which is evaluated by α_k, where $k = 1, 2, ..., 10$.

For both experiments, we use a natural frequency dictionary which is generated using the written English texts from the British National Corpus (BNC) [12]. In the dictionary, the maximal natural frequency f_{max} is found to be 6187927. We then let the NFZ size $L = 10$ and get the count of the NFZs is $K = \lceil \frac{f_{max}}{L} \rceil = \lceil \frac{6187927}{10} \rceil = 618793$. We use the texts with sizes varying from 5KB to 20KB, or with word counts varying from about 800 to about 3300. As an NFZ implies two classification features, namely distance average and variance of words in the NFZ, the dimensionality of the theoretical classification feature vector is very large. But as the word count of each testing text is limited, the dimensionality of the actual classification feature vector is not more than twice of the word count, that is hundreds or thousands of classification features is actually used to describe the structural style of each class of texts. Fig. 3 and Fig. 4 show the distributions of the first 10 NFZs' distance average features and distance variance features for each text class. From these pictures, we can image that each text class has a inherent, unique structural style represented by the distribution of words via which we can detect different classes of texts.

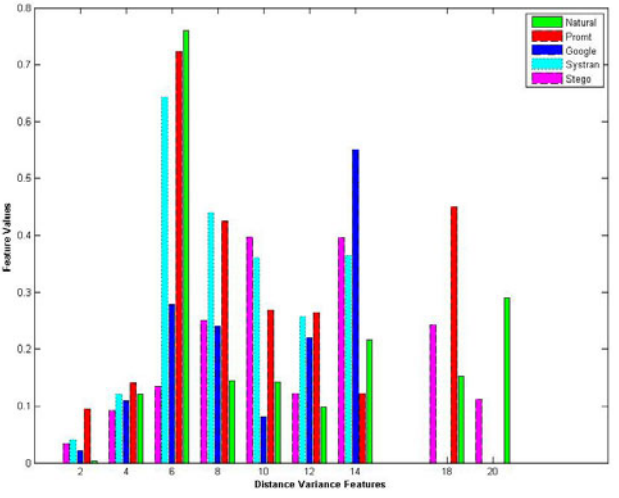


Fig. 4. The distributions of the first 10 distance variance features for each text class. The X axis value $2k$ represents the distance variance feature of kth NFZ which is evaluated by γ_k, where $k = 1, 2, ..., 10$.

Table 2. Experimental Results of NFZ-WDA using Two-class SVM classifier. The "Train" and "Test" columns show the numbers of texts used in the training and testing for both classes. "FR", "MR" and "AR" are short for "False Rate", "Missing Rate" and "Accuracy Rate" respectively.

Text Size	Class-1	Class-2	Train	Test	FR(%)	MR(%)	AR(%)
5K (abt 800 words)	Natural	Stego	60/60	172/202	0.00	0.00	100.00
	Promt	Stego	60/60	211/202	9.48	3.96	93.22
	Google	Stego	60/60	185/202	0.00	10.89	94.32
	Systran	Stego	60/60	210/202	2.38	10.40	93.69
10K (abt 1600 words)	Natural	Stego	60/60	172/202	0.58	0.00	99.73
	Promt	Stego	60/60	211/202	1.90	3.47	97.34
	Google	Stego	60/60	185/202	0.00	0.50	99.74
	Systran	Stego	60/60	210/202	0.95	0.00	99.51
15K (abt 2500 words)	Natural	Stego	60/60	172/202	0.00	0.00	100.00
	Promt	Stego	60/60	211/202	3.32	0.50	98.06
	Google	Stego	60/60	185/202	0.00	0.00	100.00
	Systran	Stego	60/60	210/202	0.00	0.00	100.00
20K (abt 3300 words)	Natural	Stego	60/60	172/202	0.00	0.00	100.00
	Promt	Stego	60/60	211/202	1.90	0.00	99.03
	Google	Stego	60/60	185/202	0.00	0.00	100.00
	Systran	Stego	60/60	210/202	0.00	0.00	100.00

Tab. 2 and Tab. 3 show the experimental results of distinguishing both between two classes of texts and among five classes of texts. On the whole, both tables show that the proposed analytic method is highly promising.

Table 3. Experimental Results of NFZ-WDA using Multi-class SVM classifier. The "Train" and "Test" columns show the numbers of texts used in the training and testing for each class.

Text Size	Class	Train	Test	Non-stego(%)	Stego(%)	Accuracy(%)
5K (abt 800 words)	Natural	60	172	100.00	0.00	91.22
	Promt	60	211	91.00	9.00	
	Google	60	185	96.76	3.24	
	Systran	60	210	94.29	5.71	
	Stego	60	202	24.26	75.74	
10K (abt 1600 words)	Natural	60	172	100.00	0.00	97.65
	Promt	60	211	98.10	1.90	
	Google	60	185	97.30	2.70	
	Systran	60	210	97.62	2.38	
	Stego	60	202	4.46	95.54	
15K (abt 2500 words)	Natural	60	172	100.00	0.00	98.88
	Promt	60	211	98.58	1.42	
	Google	60	185	99.46	0.54	
	Systran	60	210	98.57	1.43	
	Stego	60	202	1.98	98.02	
20K (abt 3300 words)	Natural	60	172	100.00	0.00	99.69
	Promt	60	211	99.53	0.47	
	Google	60	185	100.00	0.00	
	Systran	60	210	100.00	0.00	
	Stego	60	202	0.99	99.01	

In Tab. 2, the distinguishing between the stego texts and the natural texts is of very high accuracy, almost 100%, no matter how the text size is. This means that we can easily differentiate stego texts from natural language using our method. The accuracy of distinguishing between stego texts and one kind of the machine translated texts is around 93% when the text size is 5KB and above 97% when the text size is not less than 10KB, which is pretty ideal.

In Tab. 3, when the text size is 5kB, the total detection accuracy is 91.22% and the detections of natural texts and the machine translated texts as non-stego texts have accuracy rates of above 90%, but the detection of stego texts has a poor accuracy of 75.74%. This may be caused by the improper generation of the classification model in the training process of the SVM classifier. The detection accuracies of both non-stego texts and stego texts increase as the text size increases when the text size is 10kB or above. The total detection accuracies are 97.65%, 98.88% and 99.69% respectively when the text size is 10KB, 15kB and 20kB, which is also ideal.

5 Discussions

Before we complete the presentation of NFZ-WDA steganalytic method against TBS, there are some discussions about this method as follows.

First, as we have pointed out, the method uses none of the information related to TBS. Apart from the training and testing texts, the only required resource for this method is the natural frequency dictionary, which can be obtained from any one of the large enough corpuses. So it is a totally blind steganalysis against TBS.

Second, the main underlying basis of this method is that each kind of texts has a unique structural style and we use the distributions of words in all the NFZs to describe this style. The countermeasure of this method will be a goal hard to reach, for the modification of the word distributions in all NFZs is extremely difficult.

Third, the experimental results show that the steganalytic method has achieved high detection accuracies, which is superior to the previous steganalytic methods against TBS.

Finally, NFZ-WDA is very simple to achieve and can be easily applied in other natural language, e.g., the application of NFZ-WDA to the authorship attribution of Chinese novels has proved to be successful in our initial experiments.

In fact, as the texts generated by certain linguistic steganography method usually have a unique structural style, the proposed method can also be used to analyze texts generated by other linguistic steganography methods, such as nicetext, texto, spammimic, mimicry and so on. Besides, as the texts written by different men also have a unique structural style, the NFZ-WDA method also can be used to distinguish texts written by different people. This means that the proposed method can also be applied in the steganalysis of TBS method that uses manual translations and other research areas like authorship analysis and text forensics. The verification of these applications is our future work.

6 Conclusions

Stego texts generated by TBS are basically preserved syntactically correct and semantically coherent. The difficulty of detecting TBS depends on many factors such as how many translators TBS used, which translators, the source language and the target language. The previous analytic methods need to use some information more or less related to the steganographic method itself and the detection accuracies still need to be improved.

In this paper, an improved method for the steganalytic method proposed by Chen *et al.* [1] is presented. The new method called natural frequency zoned word distribution analysis (NFZ-WDA) is used as a blind steganalytic method against translation based steganography (TBS).The experimental results show that the proposed method is highly promising. Our contributions can be summarized as follows.

1) We have found a weakness in TBS: the texts generated by different translators have their inherent, unique structural style that is determined by the translator itself.

2) We have found a highly effective way to describe the unique structural style of certain class of texts, which not only can be used in the steganalysis against TBS, but also can be used in the steganalysis of other linguistic steganography methods and other research areas such as authorship analysis and text forensics.

3) We have proposed to use a two-class SVM classifier to distinguish between stego-texts and natural language texts or one kind of the machine translated texts, and to use a multi-class SVM classifier to distinguish among the stego-texts, natural language texts and all kinds of machine translated texts. Both detection accuracies are pretty high and increase as the text size increases.

Our future work includes further development of NFZ-WDA, widely verification of its applications in the steganalysis of other linguistic steganography methods and in other research areas like authorship analysis and text forensics.

Acknowledgement

This work was supported by the Major Research Plan of the National Natural Science Foundation of China (No. 90818005), the National Natural Science Foundation of China (Nos. 60903217 and 60773032), the China Postdoctoral Science Foundation funded project (No. 20090450701), the Natural Science Foundation of Jiangsu Province of China (No. BK2010255), and the Scientific and Technical Plan of Suzhou (No. SYG201010).

The authors would like to thank the anonymous reviewers for their helpful comments and suggestions.

References

1. Chen, Z., Huang, L., Yu, Z., Li, L., Yang, W.: A Statistical Algorithm for Linguistic Steganography Detection Based on Distribution of Words. In: Proc. of ARES 2008, pp. 558–563 (2008)
2. Grothoff, C., Grothoff, K., Alkhutova, L., Stutsman, R., Atallah, M.J.: Translation-based steganography. In: Barni, M., Herrera-Joancomartí, J., Katzenbeisser, S., Pérez-González, F. (eds.) IH 2005. LNCS, vol. 3727, pp. 219–233. Springer, Heidelberg (2005)
3. Ryan, S., Christian, G., Mikhail, A., Krista, G.: Lost in Just the translation. In: The Proc. of ACM Symposium on Applied Computing 2005, pp. 338–345 (2005)
4. Christian, G., Krista, G., Ryan, S., Ludmila, A., Mikhail, A.: Translation-based steganography. Journal of Computer Security 17(3), 269–303 (2009)
5. Meng, P., Huang, L., Yang, W., Chen, Z.: Attacks on Translation based teganography. In: Proc. of IEEE Youth Conference on Information, Computing and Telecommunication 2009, pp. 227–230 (2009)
6. Meng, P., Hang, L., Chen, Z., Hu, Y., Yang, W.: STBS: A statistical algorithm for steganalysis of translation-based steganography. In: Böhme, R., Fong, P.W.L., Safavi-Naini, R. (eds.) IH 2010. LNCS, vol. 6387, pp. 208–220. Springer, Heidelberg (2010)
7. Mark, C.: Hiding the Hidden: A Software System for Concealing Ciphertext as Innocuous Text (1997), http://www.NICETEXT.com/NICETEXT/doc/thesis.pdf
8. Kevin, M.: TEXTO, ftp://ftp.funet.fi/pub/crypt/steganography/texto.tar.gz

9. Wu, S.: Research on Information Hiding. Degree of master. University of Science and Technology of China (2003)
10. 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
11. Philipp, K.: Europarl: A parallel corpus for statistical machine translation. In: MT summit, vol. 5 (2005)
12. BNC database and word frequency lists, http://www.kilgarriff.co.uk/bnc-readme.html

Blind Quantitative Steganalysis Based on Feature Fusion and Gradient Boosting

Qingxiao Guan[1,2], Jing Dong[1], and Tieniu Tan[1]

[1] National Laborary of Pattern Recognition, CAS Institute of Automation
[2] Department of Automation, University of Science and Technology of China
qingxiao@mail.ustc.edu.cn, {Jdong,tnt}@nlpr.ia.ac.cn

Abstract. Blind quantitative steganalysis is about revealing more details about hidden information without any prior knowledge of steganograghy. Machine learning can be used to estimate some properties of hidden message for blind quantitative steganalysis. We propose a quantitative steganalysis method based on fusion of different steganalysis features and the estimator relies on gradient boosting. Experimental result shows that our proposed method has good performance for quantitative steganalysis.

Keywords: Steganalysis, gradient boosting, feature fusion, subspace.

1 Introduction

Steganography is an art of hiding secret message in natural images. In contrast, steganalysis is developed to break steganography by detecting weather the image has been modified to embed secret message or not. Steganalysis generally falls into two categories: specific steganalysis and blind steganalysis. Specific steganalysis tracks the trace left by a particular algorithm [1] [2], and it is able to detect the relative length of the hidden message that is embedded by certain steganography algorithm. The latter is based on pattern recognition and supervised learning, and it needs no pre-knowledge of the embedding method. Since the detection of hidden message can be regarded as a binary classification problem, it provides us with a detection result about whether the candidate image is an original image, or a stego image (embedded with secret message by certain embedding method). Besides, details of the embedding are as valuable as detection result. When we need more details about stego image after steganalysis detection, rather than using specific method, it is also suggested to cover the task of estimating detailed information in a framework of blind quantitative steganalysis. Blind quantitative steganalysis can serve as the extension of detection and require higher accuracy. This is the motivation for us to develop the blind quantitative steganalysis method.

Traditionally, quantitative steganalsis is a sequential result of specific steganalysis. As proposed in [3], embedding model that characterize F5 and Outguess are respectively built to achieve quantitative steganalysis and detecting. This situation changed when modern steganalysis method takes advantage and shown its superiority due to powerfull feature sets and machine learning. In [4], Pevný et al proposed a method

H.-J. Kim, Y. Shi, and M. Barni (Eds.): IWDW 2010, LNCS 6526, pp. 266–279, 2011.

that uses SVR (support vector regression) as estimator and input PEV-274D features [5] to estimate change rate of stego image. So far there are few works on blind quantitative steganalysis. However, this issue holds a special sense in this field. After detection, we need more quantitative details for further analysis, such as message embedding rate (bit per none-zero AC coefficient), which is related to the length of secret message in stego image. And it is known to us that higher embedding rate will bring on more artifacts in stego image, and result in more diversity as features present. The confidence of detection result is comparatively higher for stego image with higher embedding rate than that with lower one for any steganalysis method. Quantitative steganalysis in this paper can be described as solving $E(\lambda | x)$, where x is feature of image and λ is the corresponding embedding rate. Since x is a high dimension feature and we have no prior knowledge about distribution $p(x|\lambda)$ or $p(\lambda)$, we are unable to solve $p(\lambda | x)$ by Bayesian method. Instead, in this paper we focus on blind quantitative steganalysis method to estimate the embedding rate of stego-images.

The rest of this paper is organized as follows: In Section 2, an overview of our method is presented. In Section 3: three kinds of features in our experiment are introduced. In Section 4, we propose our feature fusion method based on subspace and meanwhile give a brief introduction to gradient boosting which was used in our method as an estimator. Section 5 includes a description of our experiment and an analysis for experiment result. Finally a conclusion was drawn in Section 6.

2 Overview of Proposed Method

Our proposed blind quantitative steganalysis consists of 3 significant aspects: extracting features of image, fusing features and estimating by gradient boosting. The procedure of our Blind quantitative steganalysis is illustrated in Fig. 1: Firstly we extract different features of image and concatenate them to $x \in R^n$. Secondly subspace method is used to fuse features: $x \to x' \in R^k$, $k < n$. Finally a trained Gradient boosting is used as an estimator to map the fused feature to result: $x' \to \lambda \in R$.

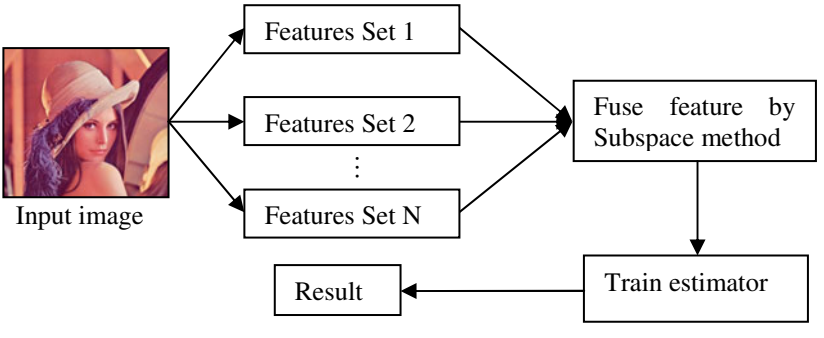


Fig. 1. Flow chart of our method

Several works on extraction of image features for JPEG image steganalysis have been proposed, such as Shi's Markov feature [6], PEV-247D feature [5]. Inspired by these ideas, in this paper we propose another set of Markov features: differential calibration Markov features. Generally we hope the feature set more representative. Different feature sets may offer different information to train a classifier. Thus a fusion of these feature sets is a straightforward method to enhance the accuracy of quantitative steganalysis. Although combining different feature sets can preserve more statistical information about candidate image than any single feature set, optimization about the feature combination is also considered here to overcome the over learning problem caused by high dimension. To incorporate advantages of different features, we adopt fusion approach to reduce the final feature dimensionality and meanwhile preserve more useful information. In particular, we focus on finding a lower subspace to keep the useful information and exclude useless component. Therefore in this paper, we proposed a subspace method to fuse different features together for quantitative steganalysis. And we will show that how our method improved the accuracy of estimation.

After fusion, fused feature is input to the estimator trained by regression to calculate final result. Gradient boosting [7] is a regression algorithm that is based on additive model [8]. We apply gradient boosting as a key part for our proposed method. Gradient boosting combines weak learners in an additive manner to construct a complex estimator. A flexible learner can fit well on training data, but sometimes may fail because of over-learning. For this problem, we adjust the parameters of gradient boosting to balance both sides and at the mean time reduce the feature dimensionality by subspace feature fusion. Next section describes the features used in the first step of our method.

3 Feature Set for Quantitative Steganography

Features which are extracted from candidate images are basic information in analysis. In steganalysis, features are usually higher order statistics of image properties. We combined two sets of proposed steganalysis features and our modified features for quantitative steganalysis. The first two proposed feature sets were named as markov feature [6] and PEV-247D [5], and our modified features named as differential calibrated Markov feature. They are introduced in this section.

3.1 Markov and PEV-247D Features

Markov feature was first proposed by *Shi* in [6]. Markov feature is extracted from JPEG coefficient array of image. It first takes absolute values of coefficient array and forms differential arrays in 4 directions: horizontal, vertical, diagonal, mirror diagonal. Elements of these four coefficient arrays, larger (or smaller) than predefined threshold T (or $-T$), will be set to T (or $-T$). Therefore the elements of these arrays range from $-T$ to T. In 4 directions, variation between adjacent coeficients can be viewed as a Markov process. For $T = 4$, the 9×9 Markov transition matrices of coefficient arrays in 4 directions are computed as Markov feature. The dimensionality of this feature set is 324.

The PEV-274D feature set we applied is a 274D set proposed by *Pevný et* al in [5] for JPEG image steganalysis. The calibration used in this method is important for

extraction of feature set. Calibration is a procedure of generating calibrated image from candidate image. It first cropping several rows and columns of candidate image and then recompresses it to a normal JPEG image using JPEG quantitize matrix of candidate image. After calibration, several features of JPEG coefficient, include histogram, co-concurrence matrix and markov feature, are respectively extracted from the candidate image and its calibrated image, and ultmately they are substracted to form PEV-274D feature. PEV-274D feature was used as quantitative steganalysis feature in [4], and achieved nice performance.

These feature sets were successfully applied to detect stego image in experiment. They are considered to be effective for blind JPEG image steganalysis, hence we include them in our feature set.

3.2 Differential Calibrated Markov Feature

Calibrated Markov feature [6] indicates the difference between origin image and stego image at the global level. Considering inter-block difference, we proposed differential calibration Markov features to capture the inter-block variance of image in four directions.

When extracting features in the horizontal direction, we only consider the change of JPEG coefficient in horizontal direction. Thus in the second step, it should crop the first α columns of pixel in spatial domain, and then compress to JPEG coefficient with the same quality factor. Differencing this jpeg array with image's JPEG array in horizontal direction, and extracting its Markov feature, we have differential calibration markov features in the horizontal direction.

For the other 3 directions, in the spatial domain, it crops the first α rows of pixels, the first α columns and rows of pixels, first α rows and last α columns of pixels, and then respectively makes difference of two JPEG arrays in 3 directions. As described above, differential calibration Markov features in these 3 directions can be extracted in similar procedure as follows:

$$M(J_1 - J_2)_i \qquad i = h, v, d, m$$

$J_1 - J_2$ denote difference of jpeg coefficient array of candidate and calibrated image in 4 directions. And we select difference step $\alpha = 1, 2, 3$, thus we have $3 \times 81 \times 4 = 972D$ differential calibrated Markov features.

These 3 type of feature sets are used in this paper. Before we conduct feature fusion, we combine them to a combined feature up to 324+274+972=1630D. They are prepared for fusion by subspace method described in the following section. We argue that this 1630D feature set is sufficient to reflect embedding rate, and is competent for our quantitative steganalysis.

4 Model for Blind Quantitative Steganalysis

4.1 Subspace Feature Fusion for Blind Quantitative Steganalysis

Naturally, combining different feature sets to a higher dimension feature set is a possible solution for preserving more information. Since regression aims for finding a

function that maps the features to its label value, it is promising to achieve higher accuracy with more features, especially when they are complemental. But there are some problems: after that simple concatenation, in the whole feature set, there may be some redundant or ineffective components that decrease the detection accuracy of steganalysis system. For a limited number of training samples, high dimension features may cause over-learning. Therefore, the feature fusion strategy is needed to overcome these defects. It is desirable if we can reduce the dimensionality of the combined feature set by keeping the useful features while skipping the redundant or ineffective components. We adopt the subspace method to feature fusion for its robustness and easy implementation. As a feature fusion method, subspace method is popular in dimensionality reduction. Many researchers have developed different subspace methods, such as PCA [10] and LDA [11]. Both methods are successful in many classification applications [12]. However, as our own concern, we want to find a subspace for quantitative steganalysis, and project high dimension feature to this subspace and fuse different feature sets. Hence we developed a specific solution for our own problem based on this method.

As proposed in previous section, original features extracted from image are concatenated to a combined feature set. The combined feature set distributes in a space with high dimensionality. This space can be decomposed into two complementary orthogonal subspaces $L1$ and $L2$. We hope the components in $L1$ are correlated to label value so as to be helpful for estimation. It is a key point how to construct $L1$ subspace. In another word, how to find basis of $L1$ subspace. This problem can be converted to supvised learning. We solve it in a numerical optimization way.

Consider a common case: to fit the feature of samples to its corresponding label, a linear multi-varieble function can be found by solving an objective function $OF(\bullet)$. Simply we know that a single linear function can hardly deal with more complex cases. Since the prototype feature data is not always linear, we will lose some information in residual subspace. It is proper to assume that non-linear properties of input samples can be retained in features of higher dimension than 1D. Considering continuously mining information in residual subspace, the second base vector can be found by optimizing objective function $OF(\bullet)$ in residual subspace. Furthermore, if n base vectors were found, it keeps on searching for the *(n+1) th* base vector in residual subspace which is orthogonal to the previous n base vectors. In this way we can further figure out remaining base vectors till all k base vectors have been found. This problem is easy to be converted to an orthogonal constrained optimization as follows:

Find k base vectors $a_1, a_2, ...a_k$ of $L1$ subspace:

$$a_1 = \arg\min_a \sum_{i=1}^{N} OF(a, x_i, y_i)$$

For j=2 to k

$$a_j = \arg\min_a \sum_{i=1}^{N} OF(a, x_i, y_i)$$

$$s.t \quad a_j^T a_1 = 0, \ a_j^T a_2 = 0 \ldots\ldots a_j^T a_{j-1} = 0.$$

End

While a_j^T denotes the transpose of j *th* vector, $OF(\bullet)$ the object function, x_i, y_i the feature vector of *i th* sample and label value, and N the number of training samples.

Constrained equations indicate that the *m th* searching step is carried out in a subspace which is complement and orthogonal to the subspace spanned by previous *m-1* base vectors. Similar approach such as CCA [11], based on linear combination, uses linear function to present the correlation of two sets of multi-variable. This kinds of subspace method could be interpreted as finding a manifold that maps the feature to a space of lower dimension. Here we actually confine the manifold to linear type.

In this paper, we use squared difference function as object function. Square distance function is a simple function in the form of $OF(a,x,y)=(a^T x-y)^2$. Since the factor $\sum_{i=1}^{N} y_i^2$ is a constant value, for *N* samples:

$$\arg\min_a \sum_{i=1}^{N} OF(a,x_i,y_i)=\arg\min_a a^T(\sum_{i=1}^{N} x_i' x_i'^T)\ a-a^T\times 2\sum_{i=1}^{N} x_i' y_i$$

Solve all the base vectors $a_1,a_2...a_k$ in k iterations, Finally, subspace $L1=span(a_1,a_2,...a_k)$. A feature vector x can be transferred to fused feature by the projection to subspace.

It is essential to select k to preserve useful information and discard useless components as much as possible.

4.2 Estimate Embedding Rate

4.2.1 The Goal of Regression

In quantitative steganalysis, it is reasonable to assume there is a relationship between features and embedding rate. Therefore we can estimate embedding rate from features by a model. Based on machine learning, regression builds this model by training data. Finally we obtain an estimating function that maps feature to an embedding rate value which we take as the estimation result.

To be general, we do not require any knowledge of the feature. But we should have training samples with their label values already known. By training on these samples, regression algorithm automatically finds a function to estimate the embedding rate value by using the features. In this way it builds the relationship between features and embedding rate in form of a multi-variable function $f(x)\to y\in R, x\in R^n$. In fact, before using regression scheme to estimator, we fused our features. It means that finally the estimation function maps $x\in R^k$, $k<n$ to a real value $\lambda\in R$.

The regeression method depends on some basic concepts. It searches that function by minimizing an objective function. Let $L(x,y)$ be a loss function with two variables [7]. This function only takes positive value and $L(x,y)=0$, if and only if $x=y$. Here $L(x,y)$ indicates the difference between two real values.

Set an objective function $E_X(E_Y(L(Y,F(X))))$, we need to find a function $\tilde{F}(X)$ that satisfies:

$$\tilde{F}(X)=\arg\min_{F(X)} E_X(E_Y(L(Y,F(X)))) \tag{1}$$

with *N* samples:

$$\sum_{i}^{N} L(Y_i, F(X_i)) \approx E_X(E_Y(L(Y, F(X)))) \tag{2}$$

Thus:

$$\tilde{F}(X) = \arg\min_{F(X)} \sum_{i}^{N} L(Y_i, F(X_i)) \tag{3}$$

4.2.2 Gradient Boosting

There are some available solutions for (3), and Gradient boosting is a feasible one. Gradient boosting was first proposed by Friedman and Hastie in [7]. It implements the Gradient descend-like process by adding a weak learner in each step. Iteration makes the estimator converge to the one with higher accuracy than any single weak learner. Ultimately we can obtain a function corresponding to the minimal value of the objective function.

It is practical to use regression tree as weak learner, and in each step we fitting regression tree to current partial deviation of object function on $F(X)$.

The algorithm is described as follows:

1. $F_0 = \arg\min_{\rho} \sum_{i=1}^{N} L(y_i, \rho)$
2. For $m = 1$ to M do
3. Fitting weak learner $h_m(x_i)$ to

   $$-\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\Big|_{F(x)=\sum_{j=0}^{m-1} F_j(x)}$$

   $i = 1, 2, 3 \ldots N$ denote the sample's index
4. $b_m = \arg\min_{b} \sum_{i=1}^{N} (F(x_i) + b h_m(x_i) - y_i)^2$
5. $F(x) = F(x) + \alpha b_m h_m(x)$
6. end

α is a shrink factor which shorten the step length to avoid stepping in local minimum point in searching steps.

Although we do not know how to design best loss function $L(x, y)$ for this problem since we do not have any details about $p(x|\lambda)$ and $p(\lambda)$, we found that squared difference function $(x-y)^2$ is proper for our task after several testing. In this paper, we use regression tree as weak learner and squared difference function as loss function in Gradient boosting. It should be noted that the depth of trees is controled to avoid overlearning in each step.

5 Experimental Results

We test our method on the UCID [13] image database. There are 1382 natural images of resolution 512×384, This database was divided into two parts: one training set having 854 images, and the rest 484 images as testing set. Respectively we embed message with 0% (cover), 4%, 5%, 6%, 7%, 8%, 9%, 10%, 11%, 12% embedding rate in these two sets by 4 different embedding methods: MB1, MB2, F5, and Jsteg. Thus

for each embedding method, we have $854 \times 10 = 8540$ samples for training, and $484 \times 10 = 4840$ for testing. Initially we extract Shi's Markov features, PEV-274D features, and differential Markov features (step =1, 2, 3). Then we combine them to one feature set of 1630D, and project it to a 300D subspace using the method described in Section 2. Actually after projection we have features of 300D for each sample in training and testing using gradient boosting. It is arguable that this method can also deal with any larger range of embedding rate as well if more training samples of higher embedding rate are collected, since density of features between different embedding rate is main factor that affect final result.

Experiment result shows that: among all single feature sets, PEV-274D feature has best performance when SVR is used as estimator. Thus, in this experiment we take the result of PEV-274D features and SVR to make a comparison with our gradient boosting and subspace feature fusion method. To make an objective comparison, we add other two schemes: PEV274 feature with Gradient boosting, and combined

Table 1. Test result of estimation accuracy on MB1 stego images

MB1 Ground truth	Mean			
	SVR+ PEV274	Our Method	Boosting+ PEV274	Boosting +1630D
0	0.0198	**0.0120**	0.0372	0.0139
0.04	0.0479	**0.0427**	0.0566	0.0459
0.05	0.0545	**0.0519**	0.0615	0.0541
0.06	0.0616	**0.0611**	0.0664	0.0620
0.07	0.0679	**0.0699**	0.0716	0.0698
0.08	0.0762	**0.0791**	0.0770	0.0778
0.09	0.0826	**0.0871**	0.0809	0.0847
0.10	0.0891	**0.0951**	0.0857	0.0914
0.11	0.0962	**0.1017**	0.0888	0.0975
0.12	0.1028	**0.1063**	0.0931	0.1017
MB1 Ground truth	Average Absolute Deviation			
	SVR+ PEV274	Our Method	Boosting+ PEV274	Boosting +1630D
0	0.0236	**0.0140**	0.0373	0.0168
0.04	0.0164	0.0180	0.0206	**0.0165**
0.05	**0.0151**	0.0170	0.0177	0.0160
0.06	**0.0149**	0.0156	0.0154	0.0157
0.07	0.0152	0.0157	**0.0141**	0.0147
0.08	0.0150	0.0152	0.0137	**0.0133**
0.09	0.0160	0.0146	0.0147	**0.0135**
0.10	0.0176	**0.0127**	0.0168	0.0131
0.11	0.0192	**0.0115**	0.0218	0.0145
0.12	0.0218	**0.0143**	0.0269	0.0185

Table 2. Test Result of estimation accuracy on MB2 stego images

MB2	Mean			
	SVR+ PEV274	Our Method	Boosting+ PEV274	Boosting +1630D
0	0.0128	**0.0110**	0.0307	0.0166
0.4	0.0457	**0.0429**	0.0541	0.0470
0.5	0.0528	**0.0516**	0.0594	0.0545
0.6	0.0615	**0.0611**	0.0658	0.0632
0.7	0.0685	**0.0698**	0.0713	0.0696
0.8	0.0755	**0.0779**	0.0762	0.0762
0.9	0.0851	**0.0873**	0.0829	0.0835
0.10	0.0917	**0.0944**	0.0875	0.0888
0.11	0.1004	**0.1011**	0.0929	0.0943
0.12	**0.1070**	0.1059	0.0962	0.0987
MB2	Average absolute deviation			
	SVR+ PEV274	Our Method	Boosting+ PEV274	Boosting +1630D
0	0.0190	**0.0129**	0.0309	0.0173
0.4	0.0154	0.0166	0.0192	**0.0151**
0.5	0.0145	0.0145	0.0164	**0.0139**
0.6	0.0146	**0.0141**	0.0153	0.0143
0.7	0.0146	0.0139	0.0145	**0.0119**
0.8	0.0151	0.0143	0.0137	**0.0118**
0.9	0.0148	0.0135	0.0135	**0.0122**
0.10	0.0163	**0.0121**	0.0151	0.0136
0.11	0.0174	**0.0113**	0.0179	0.0162
0.12	0.0186	**0.0141**	0.0238	0.0213

1630D feature (with out feature fusion) with Gradient boosting. In top columns of Table 1-4, mean value of estimation result of different embedding rate is given to show that: these methods are unbiased estimations respect to ground truth displayed in the first column of Table1-4. Average absolute deviation, which is described in (4), is also given in the last 2 columns of Table 1-4. It indicates the absolute error between estimated value and ground truth, and is used as the criterion to evaluate the accuracy of testing result in different embedding rate.

$$\frac{1}{N_k}\sum_{i=1}^{N_k}|Y_i - F(X_i)| \tag{4}$$

Where N_k denotes the number of testing samples at embedding rate k, in this experiment $N_k = 484$. Table 1-4 are our experiment results. We made a comparison between our feature fusion method and single feature set method.

Table 3. Test Result of estimation accuracy on Jsteg stego images

Jsteg	Mean			
	SVR+ PEV274	Our Method	Boosting+ PEV274	Boosting +1630D
0	0.0081	**0.0011**	0.0183	0.0094
0.4	0.0440	**0.0409**	0.0476	0.0472
0.5	0.0523	**0.0496**	0.0551	0.0548
0.6	**0.0607**	0.0582	0.0634	0.0626
0.7	**0.0698**	0.0688	0.0717	0.0711
0.8	0.0779	0.0793	0.0785	**0.0799**
0.9	0.0861	**0.0891**	0.0853	0.0883
0.10	0.0951	**0.0988**	0.0920	0.0961
0.11	0.1035	**0.1077**	0.0978	0.1021
0.12	0.1121	**0.1137**	0.1033	0.1065
Jsteg	Average absolute deviation			
	SVR+ PEV274	Our Method	Boosting+ PEV274	Boosting +1630D
0	0.0156	**0.0015**	0.0195	0.0094
0.4	0.0139	**0.0086**	0.0155	0.0095
0.5	0.0138	0.0081	0.0149	**0.0071**
0.6	0.0134	0.0088	0.0141	**0.0071**
0.7	0.0137	0.0101	0.0132	**0.0076**
0.8	0.0137	0.0100	0.0132	**0.0080**
0.9	0.0142	0.0092	0.0128	**0.0077**
0.10	0.0138	0.0088	0.0127	**0.0066**
0.11	0.0146	**0.0070**	0.0139	0.0081
0.12	0.0149	**0.0069**	0.0168	0.0134

By comparing the experimental results in Table 1-4, it is obvious that: the mean values of estimation in different embedding rate approximate to ground truth for these steganographic methods. In another word, these method are unbias estimation. As numbers in bold face in column 2 and 3, mean values of our method are closer to ground truth for most parts. While average absolute deviation is a criterion to indicate the accuracy of method, lower average absolute deviation correspond to higher accuracy. In terms of that, performance of our proposed method and Boosting+1630D are better than other method for part of MB2 and all part of F5, Jsteg. From Table 1-4, it can be seen that although the proposed method has no promotion in average absolute deviation compared to Boosting+1630D with out fusion, but it outperform other with mean value closer to ground truth for most part, especially for MB2, MB1 and F5, as well as bold numbers indicate in Table 1-4. This result proves a fact: for any stego images, combined feature with higher dimensionality and feature fusion respectively improved bias and deviation. In another word we excluded useless component by subspace feature fusion before estimating, and consequently refined data and

alleviated the over-learning problem caused by feature combination. It is understandable that feature fusion successfully removed useless components of feature sets, and distinctly improved the performance of our quantitative steganalysis system.

Fig. 2-5 are distributions of error of both methods on MB1 and F5. We observed that error of our method comforms to Laplace distribution, while that of other method conforms to Gaussian distribution. For MB2 and Jsteg, the conclusions also stand. This phenomenon is probably related to the structure of the estimator. However, how the feature or learning strategy affect error distribution or its parameter is yet unknow. As we can see in Fig. 2-5, error distribution of two methods are symmetric and centers at zero, and for our method, they are denser than that of other method. It is remarkable that error distributions of F5 have large variance. These properties are consistent with experimental result.

Table 4. Test result of estimation accuracy on F5 stego images

F5	Mean			
	SVR+ PEV274	Our Method	Boosting+ PEV274	Boosting +1630D
0	0.0301	**0.0240**	0.0530	0.0310
0.04	0.0482	**0.0472**	0.0614	0.0510
0.05	**0.0534**	0.0537	0.0637	0.0564
0.06	0.0580	**0.0593**	0.0663	0.0615
0.07	0.0659	**0.0699**	0.0699	0.0702
0.08	0.0722	**0.0767**	0.0726	0.0766
0.09	0.0773	**0.0834**	0.0750	0.0819
0.10	0.0825	**0.0888**	0.0768	0.0868
0.11	0.0887	**0.0956**	0.0793	0.0931
0.12	0.1006	**0.1047**	0.0845	0.1018
F5	Average Absolute Deviation			
	SVR+ PEV274	Our Method	Boosting+ PEV274	Boosting +1630D
0	0.0353	**0.0251**	0.0530	0.0314
0.04	0.0240	0.0236	0.0243	**0.0184**
0.05	0.0228	0.0225	0.0197	**0.0167**
0.06	0.0227	0.0212	0.0169	**0.0153**
0.07	0.0222	0.0206	0.0150	**0.0136**
0.08	0.0229	**0.0190**	0.0156	0.0198
0.09	0.0245	0.0180	0.0189	**0.0146**
0.10	0.0264	**0.0176**	0.0243	0.0178
0.11	0.0283	**0.0168**	0.0309	0.0201
0.12	0.0280	**0.0155**	0.0356	0.0221

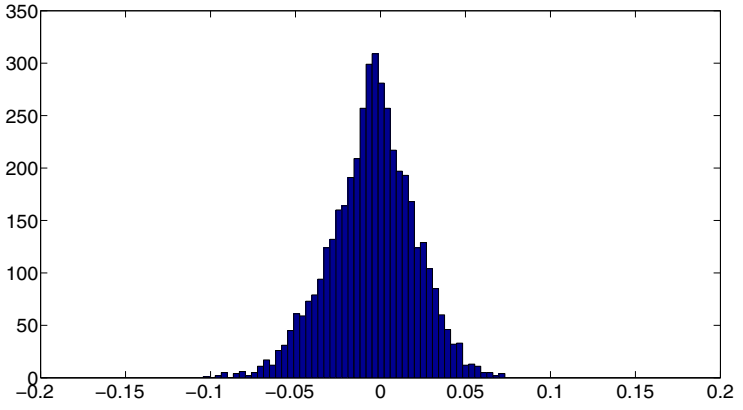


Fig. 2. Error distribution of Feature fusion+ boosting on F5

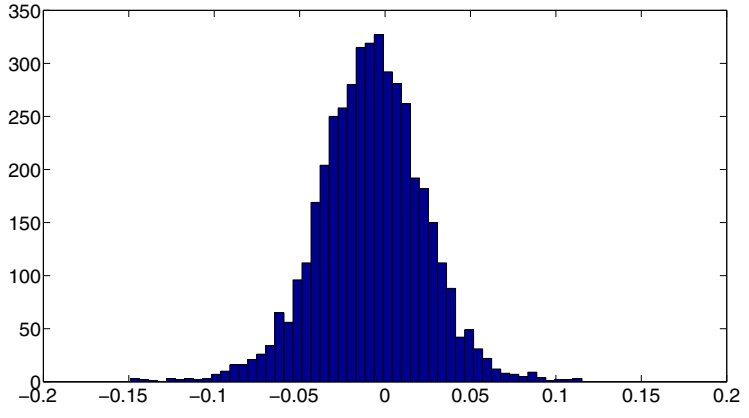


Fig. 3. Error distribution of PEV247 feature+ SVR on F5

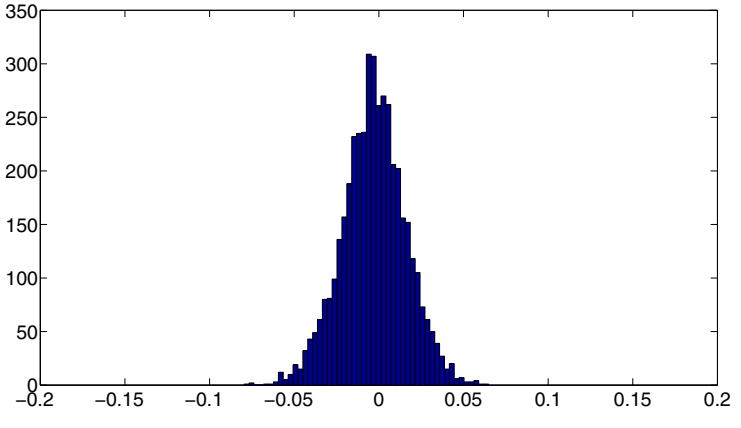


Fig. 4. Error distribution of Feature fusion+ boosting on MB1

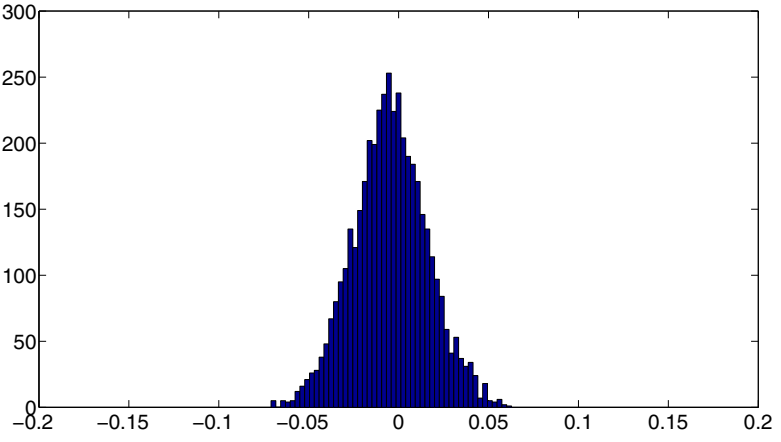


Fig. 5. Error distribution of PEV247 feature+ SVR on MB1

6 Conclusions

Quantitative steganalysis holds a special role in steganalysis. Estimation of the embedding rate of a stego image is an important task for stego-image in applications. In this paper we have proposed a novel method for blind quantitative steganalysis. We have used gradient boosting for quantitative steganalysis and used a subspace method to fuse different steganalysis features. Experiment in four kinds of stego image have demonstrated that our method has better performance. This proves that: (1) regression with boosting is effective for quantitative steganalysis. (2) different feature sets can be fused to enhance performance of quantitative steganalysis system if proper fusion scheme is applied. Besides, quantitative steganalysis has more applications in future, such as algorithm evaluation. Thus it is advisable to intensively check each kind of stego image and take further researches on quantitative steganalysis.

Acknowledgments. This work is funded by National Basic Research Program (Grant No. 2004CB318100), National Natural Science Foundation of China (Grant No. 60736018, 60702024, 60723005), and National Hi-Tech R\&D Program (Grant No. 2006AA01Z193, 2007AA01Z162).

References

1. Fridrich, J., Miroslav, G., Dorin, H.: 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)
2. Li, B., Shi, Y.Q., Huang, J.: Steganalysis of YASS. IEEE Transactions On Information Forensics And Security 4, 369–382 (2009)
3. Fridrich, J., Goljan, M., Hogea, D., Soukal, D.: Quantitative steganalysis:Estimating secret message length. ACM Multimedia Systems Journal. Special issue on multimedia Security 9(3), 288–302 (2003)

4. Pevný, T., Fridrich, J., Ker, A.D.: From blind to quantitative steganalysis. In: Media Forensics and Security, vol. 7254 (2009)
5. Pevný, T., Fridrich, J.: Multiclass Detector of Current Steganographic Methods for JPEG Format. IEEE Transactions On Information Forensics And Security 3(4), 635–650 (2008)
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. Friedman, J.H.: Greedy Function Approximation: A Gradient Boos-Ting Machine. The Annals of Statistics 29(5), 1189–1232 (2001)
8. Friedman, J., Hastie, T., Tibshirani, R.: Additive Logistic Regr-Ession: A Statistical View Of Boosting. The Annals of Statistics 28(2), 337–407 (2000)
9. 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)
10. Liu, W.-M., Chang, C.-I.: Variants of Principal Compo-nents Analysis. In: IEEE International Geoscience and Remote Sensing Symposium (July 2007)
11. Bartlett, M.: Further aspects of the theory of multiple regression. Proceedings of the Cambridge Philosophical Society (34) (1938)
12. Wang, X., Tang, X.: A unified framework for subspace face recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(9), 1222–1228 (2004)
13. Uncompressed Color Image Database, `http://www.staff.lboro.ac.uk/~cogs/datasets/UCID/u-cid.html`

IR Hiding: A Method to Prevent Video Re-shooting by Exploiting Differences between Human Perceptions and Recording Device Characteristics

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

[1] Graduate University for Advanced Studies, Japan
[2] Sharp.Ltd., Display Systems Laboratories, Japan
[3] National Institute of Informatics, Japan
{nii20081705,iechizen}@nii.ac.jp, gohshi.seiichi@sharp.co.jp

Abstract. A method is described to prevent video images and videos displayed on screens from being re-shot by digital cameras and camcorders. Conventional methods using digital watermarking for re-shooting prevention embed content IDs into images and videos, and they help to identify the place and time where the actual content was shot. However, these methods do not actually prevent digital content from being re-shot by camcorders. We developed countermeasures to stop re-shooting by exploiting the differences between the sensory characteristics of humans and devices. The countermeasures require no additional functions to use-side devices. It uses infrared light (IR) to corrupt the content recorded by CCD or CMOS devices. In this way, re-shot content will be unusable. To validate the method, we developed a prototype system and implemented it on a 100-inch cinema screen. Experimental evaluations showed that the method effectively prevents re-shooting.

Keywords: copyright protection, sensory perceptions, infrared LED, Bartley effect.

1 Introduction

High-quality digital content, such as pictures and videos shot by individuals, is now widely available thanks to the rapid growth of broadband networks and availability of high-performance consumer audio/video equipment. Anyone can easily shoot videos using camcorders and distribute the recorded content via the Internet. A serious problem, however, is caused by copyrights violation of non-personal content such as movies and photos displayed on digital signage; such content can easily be re-shot using a camcorder and distributed via the Internet or sold illegally on recording media such as DVDs. The Motion Picture Association of America (MPAA) estimates the damage caused by bootleg film recordings to be three billion dollars per year [1]. The damage is exacerbated by continuing advances in camcorder technology with better-quality recording capabilities. Preventing re-shooting of images and videos is thus essential for copyright protection. Digital watermarking technology can detect the flow of illegal distributed digital content [2-5]. However, it cannot actually prevent a person

H.-J. Kim, Y. Shi, and M. Barni (Eds.): IWDW 2010, LNCS 6526, pp. 280–292, 2011.

from re-shooting films in a movie theater with a camcorder. This paper describes a new method to stop re-shooting of pictures and videos using camcorders. No new functions need to be added to the existing user-side devices, because near-infrared signals are used to add noise to display screens. These noise signals cannot be seen by the human eye and hence theatre goers will not perceive any degradation in image quality. However, they can be picked up by image sensors such as charge coupled devices (CCD) and complementary metal-oxide semiconductors (CMOS). Hence, viewers of pirated video would see the noise. We developed a prototype system to prevent re-shooting of films in movie theaters that uses near-infrared light emitting diodes (LEDs) placed on the back side of a 100-inch cinema screen. The test results using this system proved our method effectively prevents re-shooting.

Section 2 introduces our method based on the differing sensory capabilities of humans and devices. Sections 3 and 4 describe our prototype system implemented on a 100-inch cinema screen and present the results of our experimental evaluation. Finally, Section 5 briefly summarizes the key points of this paper.

2 Proposed Method for Re-shooting Prevention

2.1 Principle

The re-shooting prevention method is based on the difference between the sensory characteristics of humans and devices. Figure 1 illustrates the perceptible areas of the sensory perceptions of humans and sensor devices (e.g. the human eye and CCDs). Sensor devices have been developed in such a way that their recording characteristics correspond to humans' visual/auditory perceptions. However, they have inherent design limitations that make it hard for their characteristics to exactly match the perception range of humans.

We propose a method to prevent re-shooting of images and videos by adding a noise signal to wavelengths of light corresponding to the shaded gray area shown in Fig. 1, that is, a range of visual characteristics that humans cannot perceive but to which sensor devices can react. The noise signal of the proposed method is implemented in the display itself, such as a movie screen or a liquid crystal display (LCDs); hence, the proposed method does not require new functions to be implemented on the recording device or on the screen showing the pirated content.

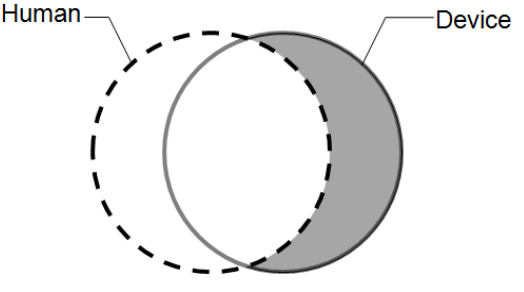


Fig. 1. Inexact overlap of sensory perceptions of humans and characteristics of sensor devices

2.2 Proposed Method

According to the International Commission on Illumination (CIE), visible wavelengths range between 380–780 nm [6]. On the other hand, the range of image sensor devices, such as CCD and CMOS, of digital cameras and camcorders, is between 200 to 1100 nm, which covers a wider range. Digital camcorders react to signals with wavelengths outside human's visible range in order to maintain luminous sensitivity in dark spaces. Figure 2 shows the different sensory ranges of the human eye and camcorders.

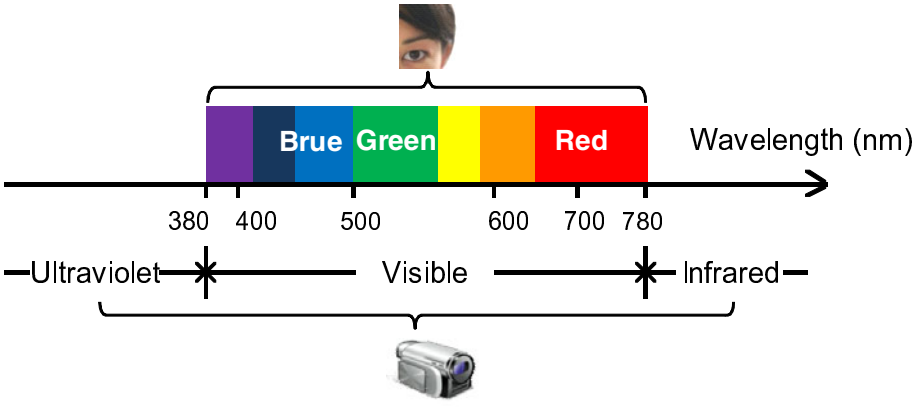


Fig. 2. Wavelength ranges 'visible' to human and digital video cameras

Three stimulus values for the human eye, B_H, G_H, and R_H, and corresponding response values of a digital camcorder, B_D, G_D, and R_D, are given by

$$\begin{aligned} B_H &= \int_{380}^{780} s(\lambda)\cdot\bar{b}(\lambda)d\lambda \\ G_H &= \int_{380}^{780} s(\lambda)\cdot\bar{g}(\lambda)d\lambda \\ R_H &= \int_{380}^{780} s(\lambda)\cdot\bar{r}(\lambda)d\lambda \end{aligned} \tag{1}$$

$$\begin{aligned} B_D &= \int_{200}^{1100} s(\lambda)\cdot b(\lambda)d\lambda \\ G_D &= \int_{200}^{1100} s(\lambda)\cdot g(\lambda)d\lambda \\ R_D &= \int_{200}^{1100} s(\lambda)\cdot r(\lambda)d\lambda \end{aligned} \tag{2}$$

where $s(\lambda)$ is the source of the signal source with a wavelength λ, $\bar{b}(\lambda)$, $\bar{g}(\lambda)$, and $\bar{r}(\lambda)$ are the color matching functions, and $b(\lambda)$, $g(\lambda)$, and $r(\lambda)$ are the spectrum

products of a digital camcorder (including the spectrum sensitivity and spectrum transmission of image sensor) [6-9]. We denote the noise signal and the light source of a video to be displayed on a screen or monitor as $n(\lambda,t)$ and $v(\lambda,t)$, respectively. The source of the signal for the proposed method, $s(\lambda,t)$, is given by

$$s(\lambda,t) = v(\lambda,t) + n(\lambda,t) \tag{3}$$

On the basis of the principal of the proposed method, the relationship between each stimulus value of the human eye and the response value of a digital camcorder is given by

$$X_H[s(\lambda,t)] = X_H[v(\lambda,t)] \tag{4}$$

$$X_D[s(\lambda,t)] \neq X_D[v(\lambda,t)] \tag{5}$$

where X represents R, G, or B, as described in formulas (4) and (5). That is, the human eye perceives $s(\lambda,t)$ and $v(\lambda,t)$ to be the same, but the digital camcorder can pick up a difference. We represent the device's difference in sensory perception between $s(\lambda,t)$ and $v(\lambda,t)$ as

$$\triangle = | X_D[s(\lambda,t)] - X_D[v(\lambda,t)] | \tag{6}$$

Then we can derive

$$\triangle = | X_D[n(\lambda,t)] | \tag{7}$$

by exploiting the linearity of formula (2). We modify the wavelength and time characteristics of a noise signal $n(\lambda,t)$ in order to increase the above difference.

2.3 Wavelength of Noise Signal

The wavelength of the noise signal should be outside the human visible range, either in the infrared (IR), which is electromagnetic radiation with a longer wavelength than visible light, or in the ultraviolet (UV), which is electromagnetic radiation with smaller wavelengths than visible light. UV can cause serious damage to people's skin, eyes, and immune system and is thus not a suitable noise signal. IR, on the other hand, is widely used in various consumer equipment such as TV remote controls and heaters, and its safety has already been established. LEDs, laser diodes, xenon lamps, and halogen lamps are considered safe IR light emitters. We used infrared LEDs because of their low cost and good reliability. Unlike light emitters with a single wavelength such as a laser, LEDs emit on multiple wavelengths distributed as a Gaussian centered on a peak wavelength. The human eye perceives as them as red if the peak wavelength is close to the human visible range. Conversely, the effect of the noise on the camcorder recorded content would be decreased if the peak wavelength is far outside the human visible range. We thus evaluated the noise effect and perceptibility of infrared LEDs using five different peak wavelengths (780, 810, 850, 870, and 940 nm) and two different digital camcorders (1/6-inch CCD-based and 1/3.2-inch CMOS-based). As a result, we found that the infrared LEDs with a peak wavelength of 870 nm had a better noise effect and was less perceivable than the others.

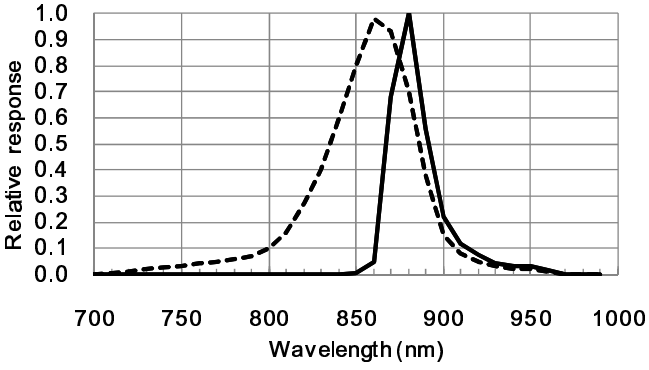


Fig. 3. Distributions of relative response (dashed line: infrared LED with peak wavelength of 870 nm, solid line: infrared LED short-wavelength cut filter with cut-on wavelength of 870 nm)

The distribution of the relative response of the infrared LEDs with a peak wavelength of 870 nm is represented by the dashed line in Fig. 3.

As shown in the distribution, the emissions are less than 780 nm, which is the upper limit of the human visible range, and only cause a slight visual degradation. To avoid this, we used a short-wavelength cut filter with a cut-on wavelength of 870 nm (cut ratio: 50%). The solid line in Fig. 3 represents the distribution after passing the IR light through the short-wavelength cut filter; the emissions causing visual degradation are eliminated and the shift of the peak wavelength at which the digital camcorder can react is minimized. On the basis of the above results, we chose to use infrared LEDs with a peak wavelength of 870 nm and a short-wavelength cut filter with a cut-on wavelength of 870 nm.

2.4 Temporal Characteristics of the Noise Signal

We had to consider the noise effect based on the noise signal's temporal characteristics. Bartley said that humans can perceive a flashing light signal most easily when it is around 10 Hz [10]. Moreover, Talbot's law says that humans can perceive a continuous light with an average intensity of a flashing light when the frequency of the flashing light is very large [10]. In light of these observations, we decided to add a flashing function with a frequency of around 10 Hz to the noise signal in the hopes of raising the level of disturbance in the recorded content.

2.5 Safety of Infrared LEDs

Originally, in 1993, LEDs were listed under the safety standard of lasers (IEC 60825-1: Safety of laser products - Part 1: Equipment classification and requirements). The standard, however, proved to be excessively severe. Many countries demonstrated the safety of the LEDs in various experimental evaluations. LEDs were subsequently proved safe and eliminated from the standard in 2006.

3 Prototype

3.1 Description

We developed a prototype system for re-shooting prevention and implemented it on a 100-inch cinema screen. Figures 4 and 5 show the overview and photographs of the prototype system. As shown in Fig. 4, the prototype system is comprised of two circuits: a flashing regulator circuit and infrared emission circuit. The details of each circuit are described as follows:

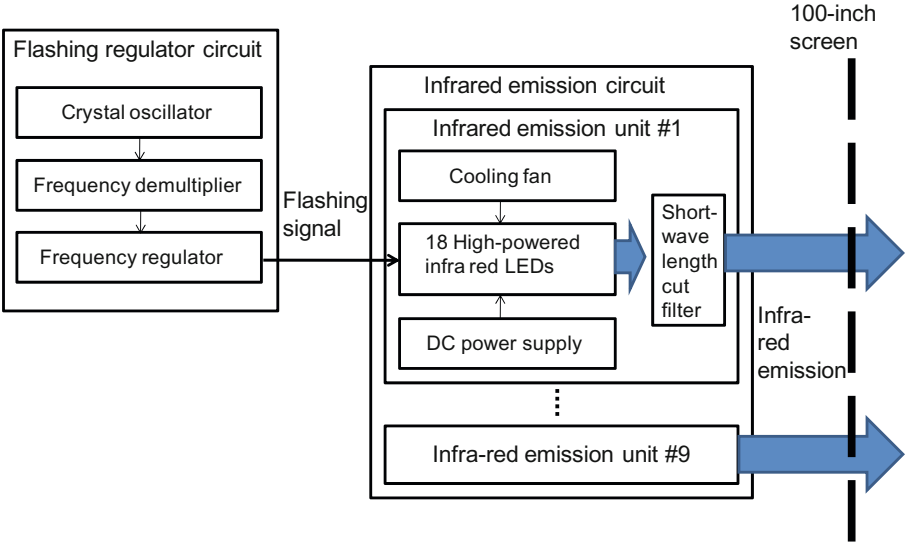


Fig. 4. System overview

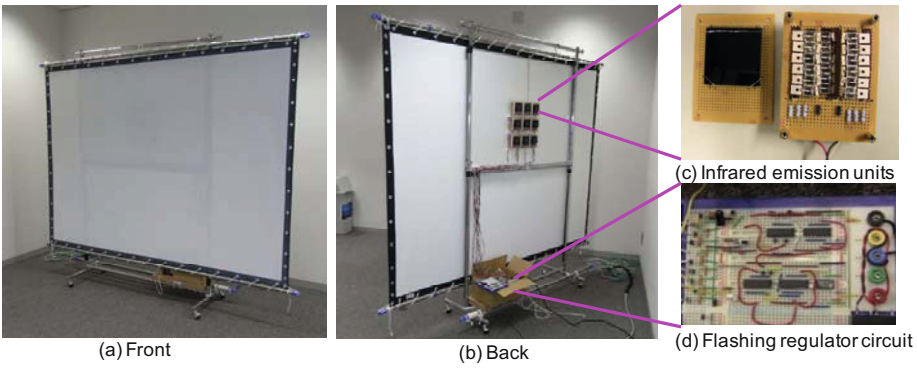


Fig. 5. Photographs of the system

Flashing regulator circuit
The flashing regulator circuit generates a flashing signal between 1 and 64 Hz by transmitting pulses from a crystal oscillator to the frequency demultiplier. These generated pulses are used for flashing the infrared LEDs on an infrared emission circuit.

Infrared emission circuit
The infrared emission circuit is comprised of 9 infrared emission units mounted behind the screen in the 3 by 3 arrangement shown in figure 5. Each unit is comprised of 18 infrared LEDs, a short-wavelength cut filter, and a cooling fan attached to the LEDs. Moreover, each unit consumes 36 W of electrical power. A cinema screen has many tiny holes 1 mm in diameter for sound from the back speakers to pass through. The infrared light from this circuit also passes through these holes.

4 Evaluation

We subjectively evaluated the picture quality of the prototype system by using the procedure described in Recommendation ITU-R BT.500 [13]. More precisely, we selected six samples from 30 standard video samples [14] and displayed them on the screen with the infrared circuit on (***proposed screen***) and with the circuit off (***normal screen***). Evaluators subjectively evaluated the level of disturbance of the proposed screen against the normal screen in two cases: (a) they directly viewed the videos on the screens, and (b) they viewed videos recorded by three different ordinary digital camcorders. For case (a), we hoped that the evaluators would not perceive any visual degradation in the proposed screen in comparison with the normal screen. For the case (b), on the other hand, we hoped the level of disturbance on the proposed screen would be as high as possible. Table 1 shows the environment of the subjective evaluation.

4.1 Standards of Subjective Evaluation

Standards of subjective evaluation can be classified into two types: evaluations defined by the International Telecommunication Union Radio Communications Sector (ITU-R) [11] and those defined by the International Telecommunication Union Telecommunication Standardization Sector (ITU-T) [12]. The evaluations of ITU-R are intended for high-quality videos for broadcast. Those of ITU-T, on the other hand, are intended for videos of multimedia communication with videophones and video conferencing systems. Since our prototype system is intended for films at theaters, we used the ITU-R evaluations.

The subjective evaluations of ITU-R are further classified into Double-Stimulus Continuous Quality-Scale (DSCQS), which evaluates the level of visual degradation in videos on communication channels, and Double-Stimulus Impairment Scale (DSIS), which evaluates the level of visual disturbance in videos.

Since the noise signal of the prototype system can be considered a disturbance in videos, we used the DSIS, so that the evaluators could subjectively evaluate the level of disturbance in the videos.

Table 1. Evaluation conditions

Screen	100-inch cinema screen
Projector	Digital projector (1000 ANSI lumen)
Video	NHK Engineering Services, Inc.
	Six selected samples from 30 standard video samples (fig.6)
	Swinging
	Flamingoes
	Buddhist images
	Driving
	Skyscrapers
	View from sky with credits
Camcorder	1/3.2-inch CMOS-based digital camcorder (207 M pixel)
	1/6-inch CCD-based digital camcorder (69 M pixel)
	Cellular phone with CMOS-based digital camcorder (8 M pixel)
Flashing frequency	Five different frequencies of flashing (continuous, 5, 10, 15, and 20H z)
Evaluators	15 non-specialists

4.2 Evaluated Video Samples

We selected six different samples from 30 standard video samples for subjective evaluation [14] in accordance with camera action (zooming and panning), object movement (slow and quick), and image processing particular to films (video with credits) (Fig. 6).

- **Swinging:** Zoomed-in scene of woman on a swing in a park.
- **Flamingoes:** Horizontally pan-scanned scene of moving flamingos.
- **Buddhist images:** Vertically pan-scanned scene of Buddha statue on the cliff.
- **Driving:** High-velocity pan-scanned scene of a car going around a curve.
- **Skyscrapers:** Horizontally pan-scanned scene of buildings.
- **View from sky with credits:** Scene of river and mountains with credits scrolling vertically.

4.3 Procedure

The evaluation procedure of the Double-Stimulus Impairment Scale (DSIS) defined by ITU-R BT.500 is as follows.

Step 1: Sample videos of the ***proposed*** and ***normal screens*** were shown to 15 evaluators.
Step 2: Each evaluator rated the picture quality of the ***proposed screen*** in accordance with the scale listed in Table 2.
Step 3: Steps 1 and 2 were done by the 15 evaluators for case (a), in which evaluators directly viewed videos on a screen, and case (b), in which evaluators viewed videos recorded by digital camcorders, and the average of the 15 scores was used as the quality level. The above steps were performed at five different flashing frequencies (continuous, 5, 10, 15, and 20 Hz).

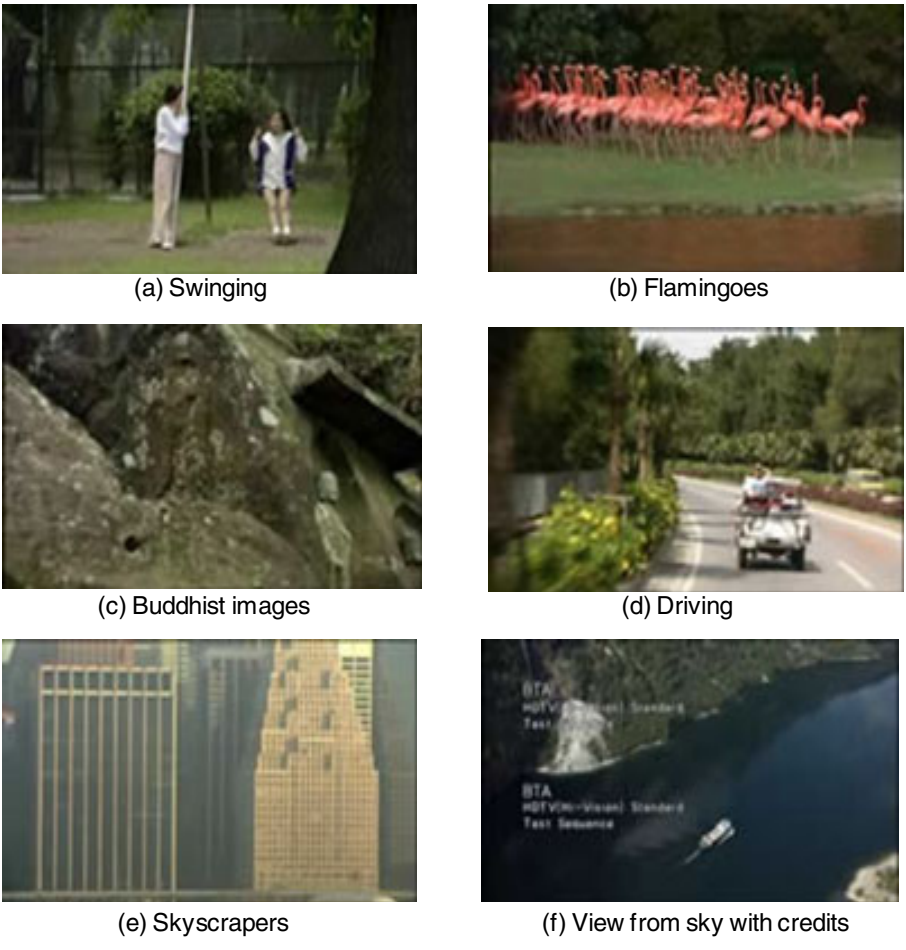


Fig. 6. Evaluated videos

Table 2. Level of disturbance and rating scale

Disturbance	Score
Imperceptible	5
Perceptible but not annoying	4
Slightly annoying	3
Annoying	2
Very annoying	1

4.4 Results

Case (a): evaluators directly viewed videos on a screen

This evaluation aimed to confirm whether the audience can see a movie displayed on the proposed screen without any visual degradation. The evaluation showed that all

evaluators rated the videos 5 (“Imperceptible”), which means the noise signal of the proposed screen was imperceptible for all the sample videos and flashing frequencies. This confirmed that the proposed method satisfied the relationship described in formula (4) and did not cause visual degradation to the human eye.

(1) CMOS-based digital camcorder

(2) CCD-based digital camcorder

(3) Cellular phone with CMOS-based camera

Fig. 7. Shot images

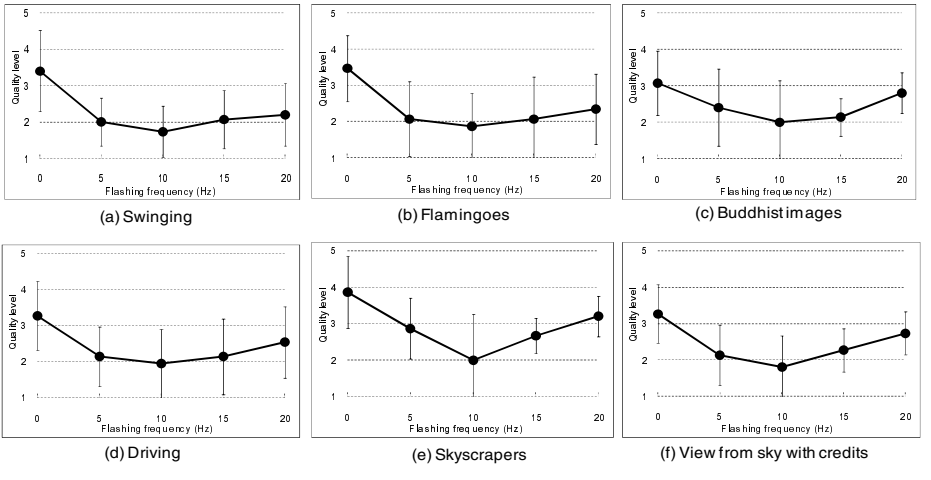


Fig. 8. Evaluation results (CMOS-based digital camcorder)

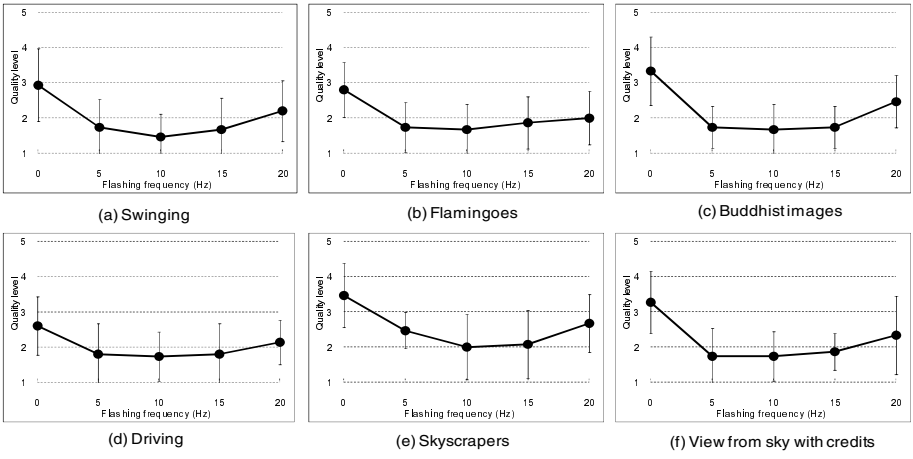


Fig. 9. Evaluation results (CCD-based digital camcorder)

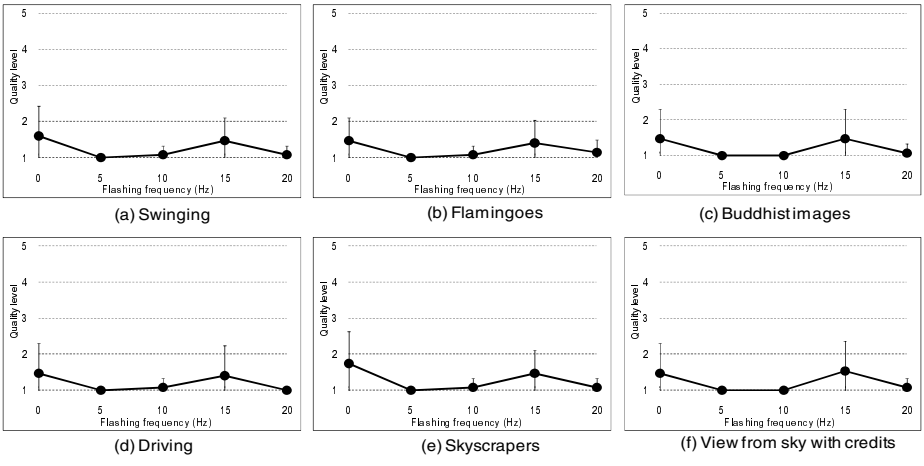


Fig. 10. Evaluation results (Cellular phone with CMOS-based camera)

Case (b): evaluators viewed videos recorded by digital camcorders

Three different ordinary digital camcorders (1/6-inch CCD based, 1/3.2-inch CMOS based, and a cell phone with CMOS-based camcorder) were used in the evaluation. Figure 7 shows the still pictures of videos shot by the three difference camcorders and projected on the screen. Figures 8 through 10 show the evaluation results for each camcorder. The horizontal axis represents the flashing frequency of the noise signal (0 Hz means continuous). The vertical axis represents quality level, μ; average of 15 scores with a range of $[\mu-\sigma, \mu+\sigma]$, where σ is the standard variation of quality level. Flashing at 10 Hz had an effect on all evaluated samples recorded by the 1/6-inch CCD and 1/3.2-inch CMOS cameras, and the evaluated quality level at 10 Hz was less than 2 (annoying). The cell phone video recorded with a CMOS-based

camcorder had the highest level of disturbance. The results for each camcorder are described in the following.

(1) CMOS-based camcorder
The proposed system had the smallest disturbances from continuous noise signals in all the evaluation images. Six evaluation images had quality levels of 3.1 to 3.9. Six evaluation images with flashing noise were rated 1.7 to 3.2. For all the evaluation images, a flashing frequency of 10 Hz was the most disturbing; the quality level was 2 ("Annoying") or less. This shows that Bartley effect is effective. Noise tended to be disturbing in the low pixel value "Swinging" images, and not so disturbing in the high pixel value "Skyscrapers" images. The evaluators' scores varied more widely than those for the cellular phone camcorder described later, and the disturbance's strength seemed to depend on the individual.

(2) CCD-based camcorder
Continuous noise signals had the lowest disturbance effect in all the evaluation images. Six evaluation images had quality levels of 2.6 to 3.5. Six evaluation images with a flashing noise signal had quality levels of 1.5 to 2.0. In all the evaluation images, the most disturbing effect was when the noise was flashed at 10 Hz. The quality level was 2 ("Annoying") or less on 6 evaluation images. Like the CMOS-based camcorder evaluation, this shows that the Bartley effect is effective. The quality level and the variation in the evaluators' scores for each image had the same trend as those of the CMOS-based camcorder.

(3) Cellular phone with CMOS-based camcorder
All 30 evaluators' scores were 2 (Annoying") or less. We can infer that camcorders implemented on cellular phones are not equipped with infrared cut filters in order to the lower their manufacturing costs and save weight. Therefore, these devices react drastically to IR noise signals. The quality level with a flash frequency of 15 Hz was high like in the case of a continuous noise signal because it appeared as continuous light; the frame rate of the cell phone camcorder has the same rate as the noise flashes (15 Hz). The variation in each evaluator's scores is smaller than in the cases of CMOS- and CCD-based camcorders. The images always appeared to be corrupted, regardless of the evaluator or type of image.

5 Conclusion

To enforce copyright protection, pirates must be prevented from re-shooting movies. Laws to prevent re-shooting of movies have been enacted, and watermarking technology has been developed. However, neither laws nor watermarking by themselves has been able to control the re-shooting problem. In this paper, we proposed a method based on the different sensory perceptions of humans and devices. The method stops images displayed on a screen or monitor from being re-shot, without adding any new functions to existing camcorders. This method uses a noise signal that can be picked up by a camcorder but not by the human eye.

The noise signal of the proposed method becomes part of the recorded image during recording. We applied it to infrared LEDs equipped with a short-wavelength cut filter and flashing the noise signal by using the Barley effect. The resulting images have no visual degradation when viewed with the naked eye whereas recording of these images contain noticeable disturbances.

We developed a functional prototype system and implemented it on a 100-inch cinema screen. We demonstrated its effectiveness in a subjective evaluation experiment. Our future work will focus on countermeasures against re-shooting with an infrared cutting filter, which can eliminate infrared noise on certain camcorders. The countermeasure will detect reflected infrared rays from the infrared cutting filter attached to the camcorder. Moreover, we will apply our method to various displays, including LCD and LED monitors.

References

1. The Motion Picture Association of America (MPAA), http://www.mpaa.org/piracy.asp
2. Haitsma, J., Kaler, T.: A Watermarking Scheme for Digital Cinema. In: Proc. International Conference on Image Processing, vol. 2, pp. 487–489 (2001)
3. 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)
4. Haruyuki, N., Seiichi, G., Ryosuke, F., Hiroshi, I., Mitsuyoshi, S., Shigenori, T., Yukari, T.: 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)
5. 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)
6. Schanda, J.: Colorimetry: understanding the CIE system, Wiley-Interscience, New York (2007)
7. BITRAN (CCD Spectral Response), https://www.bitran.co.jp/ccd/character/
8. Gonzalez, R.C., Woods, R.E.: Digital Image Processing, 3rd edn. Prentice-Hall, Englewood Cliffs (2007)
9. Holst, G.C., Lomheim, T.S.: CMOS/CCD Sensors and Camera Systems. Society of Photo Optical (2007)
10. Halstead, W.C.: A note on the Bartley effect in the estimation of equivalent brightness. Journal of Experimental Psychology 28(6), 524–528 (1941)
11. ITU Radiocommunication Sector (ITU-R), http://www.itu.int/ITU-R/index.asp?category=information&rlink=rhome&lang=en
12. ITU Telecommunication Standardization Sector (ITU-T), http://www.itu.int/ITU-T/index.html
13. Rec. ITU-R BT.500-11: Methodology for the subjective assessment of the quality of television picture (2002)
14. The Institute of Image Information and Television Engineers: Evaluation video sample (standard definition)

On Limits of Embedding in 3D Images Based on 2D Watson's Model

Zahra Kavehvash and Shahrokh Ghaemmaghami

Sharif University of technology, Tehran, Iran
kavehvash@ee.sharif.edu, ghaemmag@sharif.edu

Abstract. We extend the Watson image quality metric to 3D images through the concept of integral imaging. In the Watson's model, perceptual thresholds for changes to the DCT coefficients of a 2D image are given for information hiding. These thresholds are estimated in a way that the resulting distortion in the 2D image remains undetectable by the human eyes. In this paper, the same perceptual thresholds are estimated for a 3D scene in the integral imaging method. These thresholds are obtained based on the Watson's model using the relation between 2D elemental images and resulting 3D image. The proposed model is evaluated through subjective tests in a typical image steganography scheme.

Keywords: Human visual system; information hiding; three dimensional image, integral imaging.

1 Introduction

Objective quality models (e.g. video quality model (VQM) [1] which are closely correlated with the characteristics of the human visual system (HVS), are emerging to allow for reliable measurements of the quality of 2D video [2-8]. In this field, the most inclusive work is done by Watson [7-8]. However, development of an objective quality metric, which incorporates the perceptual aspects of 3D images, is a complex issue since the perceived attributes (e.g. depth, presence, naturalness, visual comfort) of the 3D images are multidimensional in nature.

Watson's model in 2D imaging is based on data embedding in discrete cosine transform (DCT) coefficients and their imperceptible change thresholds. In this model, the distortion resulting from embedding in the DCT coefficients is adapted to human contrast sensitivity function, luminance masking, and contrast masking within certain image blocks.

The knowledge of how the HVS operates comes from a variety of scientific fields: anatomy, physiology, and psychology. While a wide range of the HVS features, including binocular perception, are thoroughly studied, models of 3D image perception are still scarce, and, in most cases, simplistic. The typical approach to 3D image quality measurement is to estimate the quality perceived by each eye separately and to combine the two measurements into one compound quality metric (stereoscopic imaging [15, 16]). This approach may fail to predict the effects of binocular masking and facilitation on the overall perceived quality.

H.-J. Kim, Y. Shi, and M. Barni (Eds.): IWDW 2010, LNCS 6526, pp. 293–304, 2011.

In this work, based on Watson's model, a comprehensive method for study and analysis of hiding capacity in 3D images is developed. The proposed method is based on the integral imaging that yields detailed information about a given 3D image at different depths [9]. Thus, the thresholds in Watson's model could be generalized to 3D images. This metric predicts 3D image quality considering different 3D attributes such as depth information and masking. In this paper, the maximum embedding rate in 3D images, obtained through integral imaging method, is derived based on Watson's model through precise formulations. The resulting embedding rates have subjectively been evaluated on a set of test images. The result of subjective tests shows the high performance of this method in measuring the quality of 3D images.

The rest of the paper is organized as follows. Section 2, gives a review of the integral imaging method. The proposed method is introduced and explained in section 3. In section 4, the derived thresholds are applied to the test images that are evaluated through standard subjective tests. The paper is concluded in section 5.

2 3D Integral Imaging Method

Integral imaging is a 3D imaging technique in which a microlens array or a pinhole array is used for capturing the optical beams reflected from the 3D object [9]. In the following, 3D image capture and reconstruction in an integral imaging system are briefly described.

2.1 Image Recording in Integral Imaging Method

To record 3D objects using an integral imaging system, intensities and directional information of light rays that pass through each microlens, or pinhole in the lens array, are captured on a 2D image sensor. The information captured through each microlens or pinhole forms a demagnified 2D image with its own perspective of the whole 3D scene. Such a captured 2D image is referred to as elemental image (EI). Thus, to form the EIs in the pickup plane, each voxel of the 3D objects at location (x, y, z) is mapped to the imaging plane of the pickup microlens array and recorded by a CCD camera. Fig. 1, illustrates the experimental setup of the pickup process in the integral imaging system.

2.2 Image Reconstruction in Integral Imaging Method

Three-dimensional image reconstruction in integral imaging method is based on inverse optical mapping [9]. The reconstruction procedure extracts pixels from each EIs and displays the corresponding voxels at (x, y, z) coordinates. The array of 2D EIs is directly projected through a pinhole array to reconstruct the 3D scene by superposition according to geometrical optics. Fig. 3 illustrates the computational reconstruction of the 3D image on a display plane at distance z. For a fixed distance z from the display pinhole array, each EI is inversely projected through each synthesized pinhole in the array, and is magnified according to the magnification factor M. M is the ratio of the distance between the synthesized pinhole array and the reconstruction image plane, to the distance between the synthesized pinhole array and the EI plane (g), that is $M = L/g$.

The intensity at the reconstruction plane is inversely proportional to the square of the distance between the EI and the reconstruction plane. In order to form the 3D volume information, we repeat this process for different distances corresponding to all reconstruction planes of interest. Fig. 2 is the illustration of the lateral (x) axis coordinate of the reconstruction plane, according to the pth EI at (x, z). This can be extended to any voxel location at (x, y, z). Let I_{pq} be the pth row and the qth column EI, and $O_{pq}(x, y, z)$ be the inversely mapped image of the EI I_{pq} at the location (x, y, z). As seen in Fig. 2, $O_{pq}(x, y, z)$ can be represented in terms of EI I_{pq} within the boundaries of the inversely mapped EI [9]:

$$O_{pq}(x,y,z)=\frac{I_{pq}\left(s_x p-\frac{(x-s_x p)}{M},s_y q-\frac{(y-s_y q)}{M}\right)}{(z+g)^2+\left[(x-s_x p)^2+(y-s_y q)^2\right]\left(1+\frac{1}{M}\right)^2}, \quad \text{for } \begin{cases} s_x(p-M/2)\le x\le s_x(p+M/2) \\ s_y(q-M/2)\le y\le s_y(q+M/2) \end{cases} \tag{1}$$

where, s_x and s_y are the sizes of EI I_{pq} in x and y directions, respectively. The denominator of (1) is the square of the distance between the pixel of EI I_{pq} and the corresponding voxel of the inversely mapped EI at reconstruction plane z. The reconstructed 3D image at (x, y, z) is obtained from the summation of all the inversely mapped EIs:

$$O(x,y,z)=\sum_{p=0}^{m-1}\sum_{q=0}^{n-1}O_{pq}(x,y,z) \tag{2}$$

where m and n are the numbers of the EIs in x and y directions, respectively.

3 Extension of Watson's Model to 3D Images

In this section, we explain the proposed method for generalizing the Watson's model to describe the embedding distortion in 3D images.

So far, just few studies on the quality of 3D images have been reported in the literature based on the HVS [10-13]. This is while most of these studies are based on stereoscopic method, as the way of capturing and displaying the 3D images. Stereoscopy is a 3D imaging method which is based on the HVS. To capture a 3D stereoscopic image, two different views of the 3D scene are captured by a camera. The displaying stage is accomplished by sending each of the registered views of the 3D scene to one eye through spatial filters. In this way, the whole image will be processed as a 3D scene in the human brain. Hence, no details of the image at different points of the 3D space are available in this 3D capturing method. Accordingly, in this method, precise study of the effect of the data embedding distortion at different depths (the third dimension in the 3D space) is not feasible and the 3D image quality cannot be measured methodically.

Superiority of the integral imaging method over the stereoscopic imaging comes from the fact that, in the former, we have several images from different perspectives of the 3D object, which make it possible to reconstruct the 3D image at any point in the

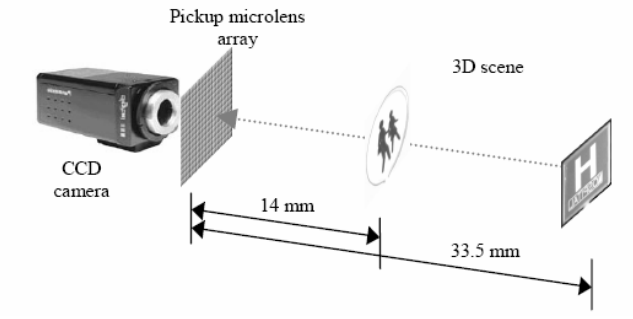


Fig. 1. Recording EIs in integral imaging method [9]

3D space, as described in section II. Therefore, in hiding information in 2D EIs, effect of the embedding distortion on the images at different depths on the z axis can be studied precisely and methodically. This has been the main concept behind the proposed method to extend the Watson's thresholds to 3D images.

The embedding thresholds in Watson's model, while the resulting distortions are not perceptually detectable, are introduced in [8]. These threshold values are based on visual sensitivity to spatial and spectral frequencies, contrast, luminance, and different masking. The threshold values derived from this model give the acceptable changes to the DCT coefficients based on the human visual system (HVS). These acceptable changes indicate the amount of message that could be embedded in the DCT coefficients of the 2D image and thus its embedding rate.

In the proposed 3D model, changes to the DCT coefficients of the EIs at different rates are made based on a hiding algorithm. Here, we just change each DCT coefficient by a certain amount computed from the integral imaging relations, as described later. Then, at each embedding rate, the DCT coefficients at different depths are computed from the embedded EIs. Next, the amount of change made to the DCT coefficients at each z is compared to Watson's thresholds. The embedding rate, at which the change to the DCT coefficients goes beyond the Watson's thresholds at least at one z value, will be considered as the embedding threshold of the DCT coefficients in the 3D image. In this way, the highest rate of imperceptible embedding in the DCT coefficients, at any z values, could be estimated which indicates the rate of embedding in the corresponding 3D image.

The crucial issue to discuss is the relation between the DCT coefficients of each EI with the DCT coefficients of the resulting 3D image at different depths. The distinct element of the integral imaging method is a lens array for capturing and displaying images. An important parameter of each lens is its frequency response characterized by its Optical Transfer Function (OTF). The OTF gives the frequency coefficients of the output image at each z distance, based on the frequency coefficients of the input image. Suppose that the input image of a single lens imaging system has the frequency spectrum of $F_i(f_x, f_y)$ (f_x and, f_y are the vertical and horizontal spatial frequencies, respectively) and the output image spectrum is $F_o(f_x, f_y)$. These two functions are related as:

$$F_O(f_x, f_y) = OTF(f_x, f_y) \times F_i(f_x, f_y) \tag{3}$$

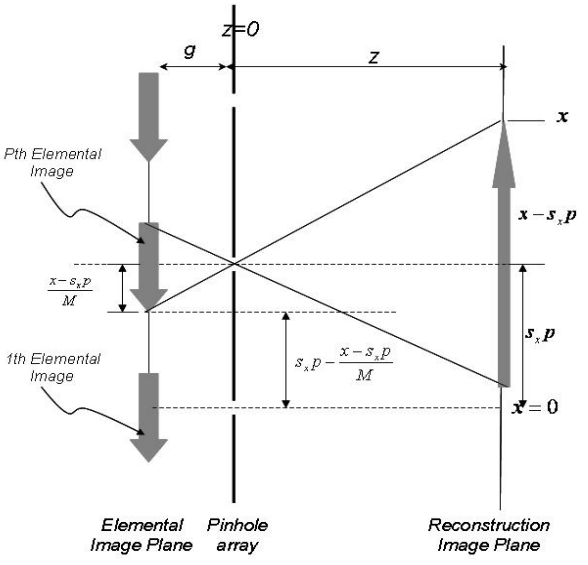


Fig. 2. Image formation in reconstruction of image plane through each virtual pinhole in the array

where $OTF(f_x, f_y)$, denotes the lens frequency response, given as [14]:

$$OTF(f_x, f_y) = \frac{\int_{-\infty}^{\infty}\int P\left(x + \frac{\lambda z_i f_x}{2}, y + \frac{\lambda z_i f_y}{2}\right) P\left(x - \frac{\lambda z_i f_x}{2}, y - \frac{\lambda z_i f_y}{2}\right)}{\int_{-\infty}^{\infty}\int P(x, y) dx dy} \tag{4}$$

Here $P(x,y)$ is the lens pupil function, which is equal to one in the lens aperture and is zero elsewhere. For instance, for a spherical lens with aperture radius of r_0, the pupil function is equal to $circ(\sqrt{x^2 + y^2} / r_0)$.

The DCT coefficients are indeed the real frequency coefficients of the image. Thus, the relation between the DCT coefficients of each EI with its related part of the image at each depth is determined by the OTF of the associated lens. It should be mentioned here that in the Watson's model, and also in other methods, the DCT transform is applied to each 8x8 block of the image, separately. In this case, each 8x8 block of an EI can be considered as an individual image. An important point to remark is that each 8x8 block, considered as an individual image, has a lateral shift with respect to the center of the image plane. Therefore, the frequency spectrum of each block should be multiplied with a phase shift that apparently does not affect their final amplitude values.

At the other side, based on (3), each point in the 3D image at depth z is affected by a number of EIs. Thus, each 8x8 block of the 3D image at depth z is the sum of a number of 8x8 blocks resulting from the related blocks in a number of EIs.

Consequently, due to the linearity of the DCT, the DCT of the output image at depth z, is the sum of the DCTs of 8x8 blocks of a number of O_{pq}:

$$DCT(BO_k) = \sum_p \sum_q DCT(BO_{i(pq)}) \tag{5}$$

where BO_k is the kth block of the output image at depth z and $BO_{i(pq)}$ is the output image related to ith block of I_{pq}, $BI_{i(pq)}$, where its DCT is determined as:

$$DCT(BO_{i(pq)}) = OTF_{pq}.DCT(BI_{i(pq)}) \tag{6}$$

This means that the relation between frequency coefficients of output image in any spatial frequency (in the given 8x8 block) and input image is determined through the frequency response of the system which is a well known fact. Now, referring back to the Watson's model based on what stated in the previous section, we can extract the maximum undetectable changes with the HVS in every 8x8 blocks, in the 3D image, at each z value. Subsequently, we are able to obtain the threshold values of the acceptable changes to the DCT coefficients of each $BO_{i(pq)}$ block. Given that, this block has overlap with some other blocks based on (5), the threshold value of each DCT coefficient in $BO_{i(pq)}$ block is considered in average as $1/N$ of the related DCT coefficients in block BO_k. The parameter N in this relation indicates the number of $BO_{i(pq)}$ blocks that overlap with the BO_k. block.

$BO_{i(pq)}$ is indeed the image of the input $BI_{i(pq)}$ block in the (p,q) EI and our final goal is to derive the threshold value of the acceptable changes in these input blocks. Hence, we derive the embedding rates of input $BI_{i(pq)}$ based on the derived threshold values for the corresponding $Bo_{i(pq)}$ output blocks. This is while each input $BI_{i(pq)}$ block is related to a number of $Bo_{i(pq)}$ in different z values. Accordingly, the derived threshold values for each $BO_{i(pq)}$ block at depth z, is called the *candidate threshold* for this block, as it only accounts for the effect of image changes at depth z. The same procedure is applied to images at different depths. For each block, $BO_{i(pq)}$, the minimum value of the candidate thresholds is taken as the threshold for imperceptible changes to the DCT coefficients. This is because any changes larger than these thresholds will disturb the visual quality of the 3D image at least at one depth, z:

$$\begin{aligned} &DCT(BO_{i(pq)}) = OTF_{pq}.DCT(BI_{i(pq)}) \;\Rightarrow \\ &\Delta\big(DCT(BO_{i(pq)})\big) = OTF_{pq}.\Delta\big(DCT(BI_{i(pq)})\big) \;\Rightarrow \\ &Th_{DCT(BO_{pq(i)})} = \min_z \left\{ Th_{DCT(BO_k)} \right\} \end{aligned} \tag{7}$$

At the final step, using (7), the threshold value for changes to the DCT coefficients of the related block in the EI_{pq}, $Th_{DCT(BI_{i(pq)})}$, is obtained, as:

$$Th_{DCT(BI_{i(pq)})} = Th_{DCT(BO_{i(pq)})} / OTF_{pq}. \tag{8}$$

This way, the acceptable threshold values for the DCT coefficients of each EI is obtained in a way that the resulting changes to the DCT coefficients of the 3D image, at any value of z, is undetectable by the HVS. The derived invisibility threshold values determine the amount of message that could be embedded in the corresponding EI arrays, while is still undetectable by the HVS. In this way, the acceptable embedding rate in an array of EIs could be computed using the presented method.

4 Simulation Results and Subjective Test

In this part we first reconstruct the 3D image using the reconstruction approach given in section 2.2 for a sample array of EIs. In Fig. 3.a, an EI from the two test sets, toy cars, and statues, are shown. Toy cars consist of 16x16 EIs and statues consist of 5x5 EIs. In Fig. 3.b, the reconstructed 3D images of the toy cars, obtained using (1) and (2), are shown at different values of z. Given that the images in Fig. 3.b, show the 3D image at different depths, we can analyze the effect of embedding in EIs on the quality of these images. We first extract the perceptual thresholds of the EIs' DCT coefficients using (3) through (8). Then, the DCT coefficients of the input EIs are changed based on the extracted thresholds, and the resulting 3D image is computed at different values of z.

As an example, the original 3D output image of the toy cars at depth z = 38 mm is shown in Fig. 4.a, while the same output image resulting from the embedded EIs with the extracted thresholds is shown in Fig. 4.b. As observed from this figure, by applying the DCT changes up to the extracted threshold values, the resulting output 3D image at the selected depth is of no perceptible difference from the original image. The amounts of changes to the DCT coefficients of the resulting 3D images of the embedded EIs for the two test sets were computed. These values were similar to the Watson's thresholds of the images to a large extent, as expected. This was evaluated through an informal subjective test detailed in the following.

For the subjective evaluation of the derived thresholds, we changed each EI's DCT coefficients with the derived thresholds. Then the corresponding 3D images were computed in 12 different depths (z values). Thus; we have 24 embedded images to be compared with the original images. We have also chosen some 2D images and embedded them with different values (less and more than Watson's threshold) for training the subjects. Nineteen subjects (12 female, 7 male) participated in the test. They were all non-expert viewers with a marginal experience of 3D image and video viewing. The age distribution ranged from 20 to 53 with an average of 32.

For subjective evaluation of 2D visual quality several recommendations have been issued by the International Telecommunication Union (ITU) including the widely used ITU-R BT.500.7. It describes methods for the subjective quality assessment of standard definition television (SDTV) pictures. The most prominent methods are the double stimulus continuous quality scale (DSCQS), the double stimulus impairment scale (DSIS) and the single stimulus continuous quality evaluation (SSCQE). With respect to the subjective quality evaluation of future multimedia data such as high definition (HDTV) and 3-dimensional TV (3DTV) the same methods are recommended in ITU-R BT.7108 and ITU-R BT.1438,9 respectively. The DSIS can be used for a direct comparison of impaired and unimpaired stimuli. Thus here, for determining

the amount of difference between embedded and original 3D images we should use the DSIS method. In DSIS method, a series of 3D images are presented in time and the assessors are asked to judge only the test image, "keeping in mind the reference". The usually applied rating scales are given in Table 1.

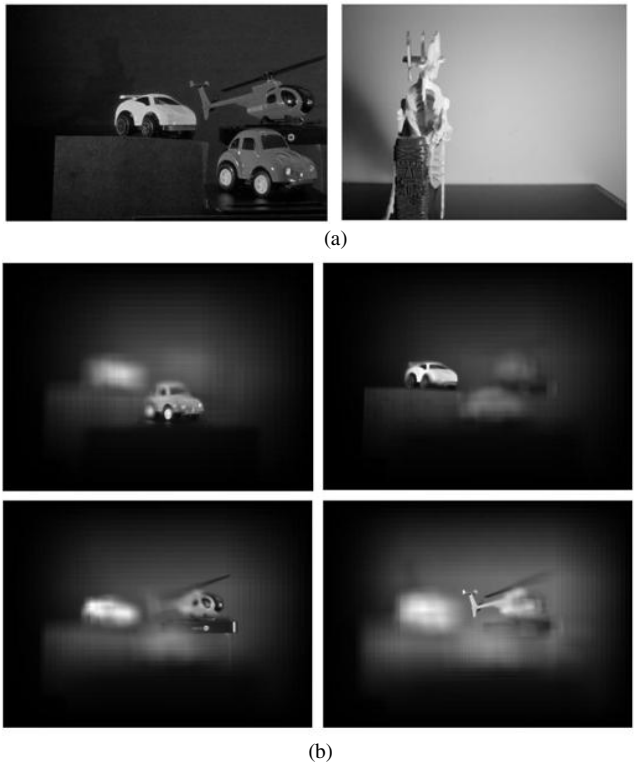


Fig. 3. (a) EIs in the 4 corner of the EI array (b) The reconstructed images in different depths

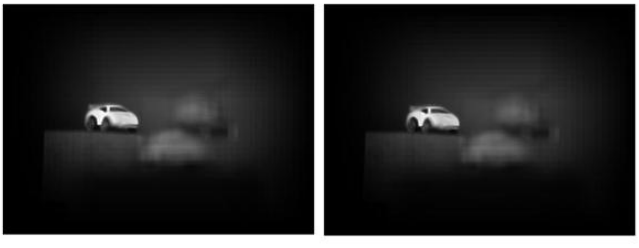


Fig. 4. (a) Original 3D output image in depth z = 38 mm and (b) The same image resulting from embedded EIs with the extracted thresholds

Table 1. ITU-R BT.500.7 recommendation rating scales

Comparison scale for DSIS
-3 much worse
-2 worse
-1 slightly worse
0 the same
1 slightly better
2 better
3 much better

In the training session, the subjective test methodology was introduced to the subjects and the range of quality levels was explained through a set of training stimuli. The training stimuli were presented in the same way as the test material to familiarize the subjects with the methodology. In the testing session, the subjects evaluated the quality of the 24 pairs (embedded and original image) of test stimuli, which are displayed in random order. Each stimulus is shown once with duration of 10 sec. and a 5 sec. break between the stimuli, during which the subjects provide their scores. Given that the images were embedded with values a bit less than threshold levels, we expected most of the subjects to give number 0 (the same). Thus, the percentage of the 0 scores was evaluated for each image in different depths. The resulting curves are shown in Fig. 5, where more than 80 percent of the viewers have voted for the same quality of the images meaning that the derived thresholds are correctly the just noticeable distances.

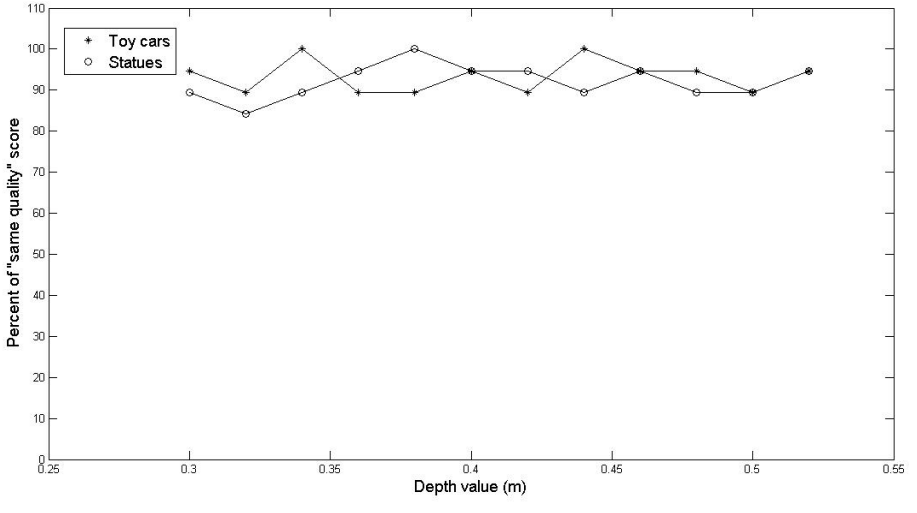


Fig. 5. Percent of "same quality" score versus different depth values (3D images in different depths) for two test images: Toy cars and Statues

We also conducted another subjective test to find out value of the JND (Just Noticeable Difference) for each DCT coefficient of the EIs. These JND values are then compared to the corresponding values of the JND derived using the proposed method.

The subjective test procedure was as follows: we first watermarked the two example sets of EIs (toy cars and statues) by changing the DCT coefficients by values quite smaller than the JND values derived using the proposed method. Then, we reconstructed the 3D image in different depth values using these watermarked EIs. Also, we built the original 3D image in different depths using the clear EIs for comparison to be made by the subjects. This subjective comparison, between the two 3D images in different z values, was made by ten subjects familiar with the test methodology. The subjects were trained by showing them both the original images and a set of images watermarked at different embedding rates, lower than or higher than the JND.

For each subject, we changed strength of the embedded message (the amount of change made to each DCT coefficient) based on the following scenario. In each depth, and for each 3D test image, the observer was presented with two stimuli; one watermarked and one original. The subject was then asked to identify the watermarked image. Upon three correct guesses, strength of the watermark was decreased, while it was increased in case of a wrong guess. The watermark strength eventually showed to be oscillating around a point that was considered as the true JND in each test. The ten individual JNDs, obtained from this subjective test for each DCT coefficient, were then averaged to compute the mean JND for that DCT coefficient.

The JNDs obtained from this subjective test were then compared to the corresponding values computed using the proposed method. To do this, we calculated the root mean square (RMS) of the difference between the two JND values, normalized by the RMS of the whole DCT coefficients of the corresponding EI array. These normalized

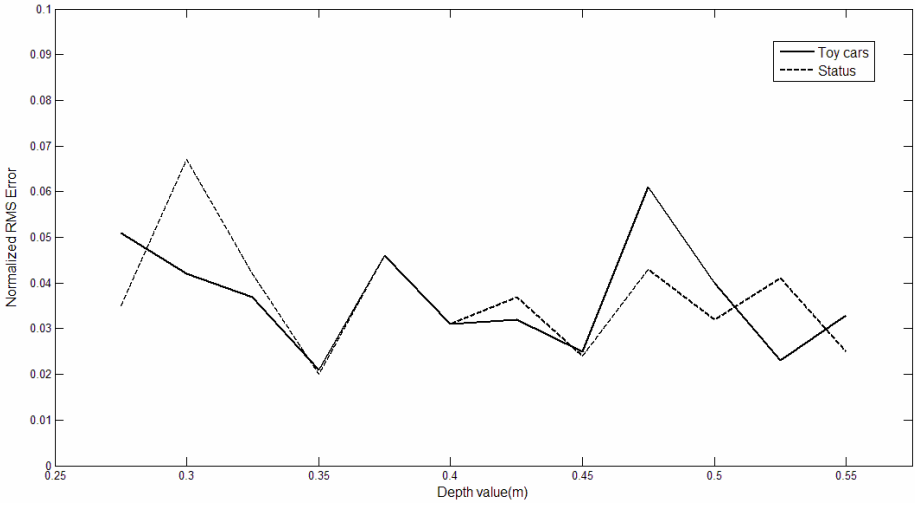


Fig. 6. Normalized RMS error between subjective and proposed JNDs in different depths for two test images, Toy cars and Statues

RMS errors were calculated in different depth values for both arrays of the test EIs. The results are shown in Fig.6, indicating that amount of the error in different depths does not exceed 7%.

5 Conclusions

A perceptual model for the embedding in 3D images has been addressed in this paper. The work done so far in this field has solely been based on the stereoscopic imaging that is inadequate to yield precise and inclusive thresholds for the embedding rate in 3D images. In this work, the perceptual thresholds for embedding in EIs of the 3D image in the integral imaging method have been derived based on the Watson's model. Using the existing precise relation between the DCT coefficients of the input EIs and the output 3D image in integral imaging method, the derived thresholds are shown to be compatible with the Watson's HVS based thresholds. This is also confirmed by our subjective evaluation of the embedded 3D images.

Acknowledgments. The authors thank Iran National Elite Foundation for its support.

References

1. Pinson, M., Wolf, S.: A new standardized method for objectively measuring video quality. IEEE Trans. Broadcast. 50, 312–322 (2004)
2. Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E.: Image quality assessment: From error visibility to structural similarity. IEEE Trans. Image Processing 13(4), 600–612 (2004)
3. Katkovnik, V., Egiazarian, K., Astola, J.: Adaptive window size image de-noising based on intersection of confidence intervals (ICI) rule. J. of Math. Imaging and Vision 16(3), 223–235 (2002)
4. Chandler, D.M., Hemami, S.S.: VSNR: A Wavelet-Based Visual Signal-to-Noise Ratio for Natural Images. IEEE Transactions on Image Processing 16(9), 2284–2298 (2007)
5. Sheikh, H.R., Bovik, A.C.: Image information and visual quality. IEEE Transactions on Image Processing 15(2), 430–444 (2006)
6. Winkler, S.: Quality metric design: A closer look. In: Proc. SPIE Human Vision and Electronic Imaging, San Jose, CA, January 22-28, vol. 3959, pp. 37–44 (2000)
7. Watson, A.B., Hu, J., McGowan, J.F.: Digital video quality metric based on human vision. Journal of Electronic Imaging 10(1), 20–29 (2001)
8. Watson, A.B.: DCT quantization matrices visually optimized for individual images. In: Rogowitz, B.E. (ed.) Human Vision, Visual Processing, and Digital Display IV. Proc. SPIE 1913-14 (1993)
9. Hong, S.H., Jang, J.S., Javidi, B.: Three-dimensional Volumetric Object Reconstruction Using Computational Integral Imaging. J. OSA, Optics Express 12(3), 483–491 (2004)
10. Benoit, A., Le Callet, P., Campisi, P., Cousseau, R.: Quality assessment of stereoscopic images. EURASIP Journal on Image and Video Processing, special issue on 3D Image and Video Processing, Article ID 659024 2008, 13 pages (2008)
11. Hewage, C.T.E.R., Worrall, S., Dogan, S., Kondoz, A.M.: Prediction of stereoscopic video quality using objective quality models of 2-D video. Electronics Letters 44(16), 963–965 (2008)

12. Gorley, P., Holliman, N.: Stereoscopic Image Quality Metrics and Compression. In: Proc. SPIE Stereoscopic Displays and Applications XIX, vol. 6803 (2008)
13. Lu, F., Wang, H., Ji, X., Er, G.: Quality Assessment of 3D Asymmetric View Coding Using Spatial Frequency Dominance Model. In: Proc. IEEE 3DTV Conference, Potsdam, Germany (Mai 2009)
14. Goodman, J.W.: Introduction to Fourier Optics, 2nd edn. MC. Graw-Hill, New York (1996)
15. Benton, S.A.: Selected Papers on Three-Dimensional Displays. SPIE Optical Engineering Press, Bellingham (2001)
16. Jang, J.S., Javidi, B.: Depth and Lateral size control of three-dimensional images in projection integral imaging. J. OSA, Optics Express 12(16), 3778–3790 (2004)

A Reversible Acoustic Steganography for Integrity Verification

Xuping Huang[1], Akira Nishimura[3], and Isao Echizen[1,2]

[1] School of Multidisciplinary Sciences
The Graduate University for Advanced Studies(SOKENDAI)
[2] National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, Japan 101-8430
{huang-xp,iechizen}@nii.ac.jp
[3] Department of Information Systems
Tokyo University of Information Sciences
4-1 Onaridai, Wakaba-ku, Chiba, Japan 265-8501
akira@rsch.tuis.ac.jp

Abstract. Advanced signal-processing technology has provided alternative countermeasures against malicious attacks and tampering with digital multimedia, which are serious issues. We propose a reversible acoustic steganography scheme to verify the integrity of acoustic data with probative importance from being illegally used. A hash function is used as a feature value to be embedded into original acoustic target data as a checksum of the data's originality. We compute the target original signal with an Integer Discrete Cosine Transform (intDCT) that has low computational complexity. Embedding space in the DCT domain is reserved for feature values and extra payload data, enabled by amplitude expansion in high-frequency spectrum of cover data. Countermeasures against overflow/underflow have been taken with adaptive gain optimization. Experimental evaluation has shown the distortion caused by embedding has been controlled under a level that is perceptible. Lossless hiding algorithm ensures this scheme is reversible.

Keywords: Acoustic steganography, Reversibility, Feature extraction, Integrity Verification.

1 Introduction

Digital multimedia content is widely available with the optimized development of broadcasting services, information storage devices, and data-transmitting technology through the Internet. Maintaining the integrity of digital content and verifying whether there has been illegal tampering are serious issues, especially for acoustic data that must be kept as material evidence for probative purposes. Such data may include police-investigation tapes, last wills and testament tapes, telephone recordings, phone banking records, emergency calls, and air-traffic communications. There are two requirements for these scenarios: (1) the scheme must be able to determine whether malicious tampering has occurred; and (2)

H.-J. Kim, Y. Shi, and M. Barni (Eds.): IWDW 2010, LNCS 6526, pp. 305–316, 2011.

the scheme must be reversible since the original is required for probative purposes. Many verifiable or reversible watermarking schemes have been proposed in the field of image processing. These methods take advantage of Human Visual System and can not be applied to acoustic data, which will cause auditory distortion.

The paper is organized as follows. The approaches taken here and those in the previous studies are introduced in Section 1. Section 2 details the proposed method, including feature value (hash function) extraction, embedding, and extraction processes, and verification of tampering. Section 3 introduces the implementation of acoustic steganography, and Section 4 summarizes the results from an objective experiment. Section 5 concludes this paper.

1.1 Conventional Works and Their Problems

Alternative reversible hiding methods have been studied [1],[2],[3],[4],[5],[6]. Technologies have also been utilized to verify the integrity of the transmitted speech signal [7], enhance the security of speaker identification system [8], and etc. However, in some application scenarios, any modification of sensitive acoustic data is not allowed: biometric system [9], legal evidence and military communication[10]. Methods have also been proposed to detect audible modifications or to identify uses[11],[12],[13].

Popular reversible hiding policies for integrity verification have been classified into two main classifications by Mehmet et al.[6] i.e., (a) During decoding a spread spectrum signal corresponding to the information payload (embed data) is superimposed on the host signal (cover data), and during decoding the payload is subtracted from the stego data, and (b) the features of the original data are embedded as the watermark payload. The first classification offers bit robustness while the second offers high capacity. In both these ways, the watermarked signal is visible (images), audible (acoustic data) and perceptible in the payload.

Algorithms for reversible watermarking can mainly be categorized into three classifications: (1) Data compression based methods[14]: the original portions of cover data will be replaced by payload using compression and data alteration, and during decoding, feature information is extracted and decompressed. (2) Methods based on modifying difference expansion[15]: these schemes represent features of original data with small values, then the value is expanded to embed the payload in LSB. (3) Histogram bin shifting methods[16]: the embedding target is replaced by the histogram of a block. These methods are used to enhance the robustness of reversible watermarks.

The approach of imperceptible payload hiding has been a difficult issue in conventional work. A distortion-free scheme for embedding has been proposed [21]; however, it cannot be applied for integrity verification.

According to our investigations, none of the past studies have met the requirements to combine reversible and verifiable approaches for integrity verification for acoustic data, and there has been a lack of applications to maintain the integrity of acoustic data or verify authenticity. Comparing it with the conventional works on algorithms, the proposed approach adapts the gain factor (-3dB)

to make it possible to control the distortion of host data. Flexible amplitude expansion and countermeasures against overflow/underflow make it possible to achieve reversibility and verifiability, as two important issues in this scheme to protect the integrity of sensitive acoustic data, when the usual criteria do not apply.

1.2 Function Properties

Probative use requires digital content to be faithful to the original and be reliable, i.e., the original data should be lossless and available when required. This means three issues should be taken into account.

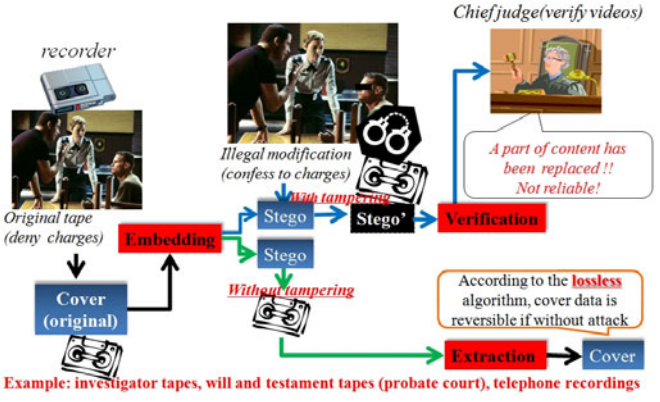


Fig. 1. Target Applications: This scheme is used to verify whether malicious modifications have occurred that may lead to unjust accusations. Integrity and reliability can be verified without original data.

1. **Verifiability.** The scheme identifies any malicious modifications to acoustic data and finds the specific domain where tampering has occurred to ensure digital acoustic data are reliable, such as those in investigator tapes, will and testament tapes, and telephone recordings.
2. **Reversibility.** The original acoustic data are required by courts and judges for particular purposes. For instance, they may be recordings that are to be used as material evidence in court trials. Therefore, the original sound (cover data) recording should be able to be recovered even after the feature data have been extracted from the stego data. Reversibility requires both the embedding and extracting algorithms to be lossless.
3. **Imperceptibility.** When the feature data are camouflaged in the cover sound, distortion needs to be kept below a perceptible level to keep the stego data audible without additional data processing. We calculated the embedding positions available in the frequency domain. Figure 1 outlines the target applications.

This paper explains the general principles behind lossless embedding and reversible and imperceptible schemes to verify whether tampering has occurred. We extract hash data as authentication information and embed the hash and payload into the high-frequency spectrum domain after computing the type-IV integer discrete cosine transform (intDCT) in the segmented host audio data. Prior to embedding, amplitude modification is applied to the DCT coefficients to achieve a totally reversible steganography scheme.

2 Proposed Method

The feature data (hash) are calculated from the original data and embedded in redundant space in the frequency domain of the cover data after intDCT. There are three phases of processing. Figures 2, and 3 are block diagrams of the proposed model. All processes are applied repeatedly to seamlessly segmented audio signals of length N.

2.1 Embedding Phase

The data to be embedded consist of hash data, positional data for embedding, stored LSB data for amplitude optimization, and an extra payload. The embedding phase is conducted in the frequency domain calculated with the intDCT of the host signal. The high-frequency DCT coefficients are doubled prior to the embedding process to make room for the embedding data, since the least bit positions of the modified high-frequency coefficient is replaced with the data to be embedded. The frequency spectrum is converted into waveform data by using an inverse intDCT to obtain the stego signal. If amplitude overflow or underflow of the stego signal occurs, the amplitude optimization process prior to the inverse intDCT is repeated until overflow and underflow no longer occur. The details are given in Subsection 3.1.

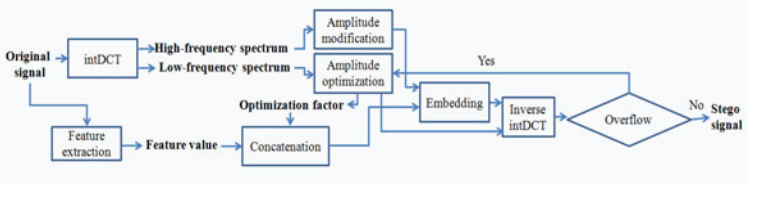


Fig. 2. Structure of embedding phase

2.2 Extraction Phase

The input stego data are converted by computing intDCT in the extraction phase. The output data are the re-extracted feature value, the re-constructed original host signal, and the extra payload data. The details are given in Subsection 3.2.

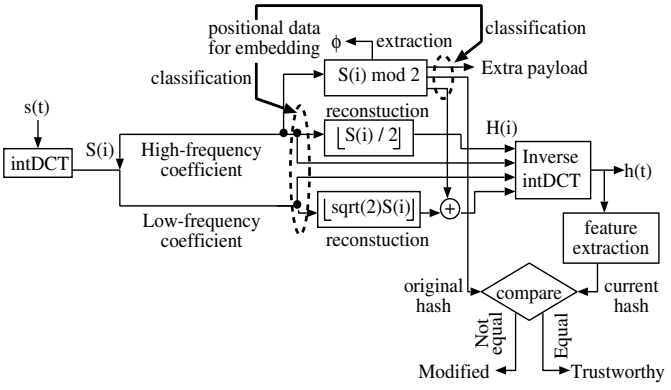


Fig. 3. Structure of extraction and verification phases

2.3 Verification Phase

The hash value is detected in the verification phase from the reconstructed original data. The current hash value is compared to the original hash value. If they differ, the stego data are modified. Figure 3 outlines the extraction and verification phases.

3 Implementation

3.1 Process Flow for Embedding Phase

intDCT. The signal representation is shown in Figure 4.

The intDCT Type IV algorithm[17] proposed by Haibin et al.[18] is used for time-frequency data conversion. There are a total of $2.5N$ rounding operations, where N is the DCT size (the length of a block).

Let $h(t)$ (t=0,1,..., $N-1$) be a real-valued input sequence (host signal waveform). We assume that $N = 2^p$, where p is a positive integer. The length N of the type-IV DCT of $h(t)$ is defined as

$$H(i) = \sum_{t=0}^{N-1} h(t) \cos \frac{\pi(2t+1)(2i+1)}{4N}, i = 0, 1, \ldots, N-1 \tag{1}$$

Let C_N^{IV} be the corresponding transform matrix, i.e.

$$C_N^{IV} = \left(\cos \frac{\pi(2t+1)(2i+1)}{4N} \right), \quad i, t = 0, 1, \ldots, N-1 \tag{2}$$

Amplitude Expansion for Embedding. First, the DCT coefficients $H(i)$ are divided into M ($M = 16$ in the current implementation) frequency regions. The

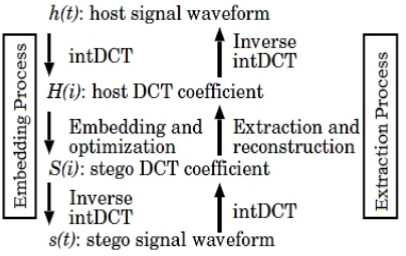


Fig. 4. Signal representations. t is discrete time; $0 < t \leq N$. i is discrete frequency; $0 < i \leq N$.

power levels are then calculated for each frequency region $p(m)(1 \leq m \leq M)$. Positional data $\boldsymbol{\varphi}$ with a length of M bits correspond to the location of the frequency region, i.e., $\boldsymbol{\varphi}(m)$ indicates the frequency region of the DCT coefficients $H(i)(i = (m-1)N/M + 1, ..., mN/M)$. Each bit value of $\boldsymbol{\varphi}$ classifies the frequency regions to be used for embedding. The first $M/2$ bits indicate low-frequency regions for amplitude optimization, and the last $M/2$ bits indicate high-frequency regions for hiding data. $\boldsymbol{\varphi}$ is initialized as:

$$\boldsymbol{\varphi}(m) = \begin{cases} 0 \text{ if } \ m \leq M/2; \text{no manipulation} \\ 1 \text{ otherwise; embedding region} \end{cases} \tag{3}$$

The high-frequency DCT coefficients, which are indicated by $\boldsymbol{\varphi}(m) = 1$, are expanded by doubling to reserve embedding space. The least bit positions of the DCT coefficients of the expanded frequency regions are replaced with the embedding data. Figure 5 has the details on embedding and amplitude expansion.

Amplitude Optimization. We have to be careful in applying inverse intDCT to the expanded DCT coefficients $S(i)$ of intense signal segments because the amplitude of the stego waveform $s(t)$ can possibly overflow or underflow, which means the amplitude of the waveform exceeds the upper bound (32767 for 16-bit signed audio and 65535 for unsigned) or the lower bound (-32768 for 16-bit signed audio and 0 for unsigned). Yan et al. [2] used a location-map technique to prevent overflow/underflow problems.

Instead of using the location-map technique, we propose the use of adaptive amplitude optimization in the DCT domain. If overflow or underflow occurs in $s(t)$, the LSB data of $H(i)$ that have maximum power level $p(m)$ in the frequency regions of $\boldsymbol{\varphi(m)}$=0 are stored. At the same time, '1' is subtracted from $\boldsymbol{\varphi}(m)$. The stored LSB data are added to the lowest expanded frequency region that is indicated by $\boldsymbol{\varphi}(m) = 1$. The frequency region indicated by $\boldsymbol{\varphi}(m) = -1$ is then divided by $\sqrt{2}$ (–3 dB) and the inverse intDCT is applied to $S(i)$.

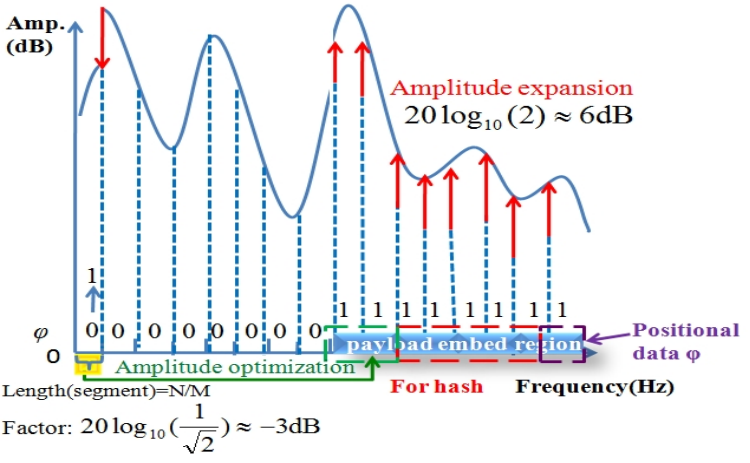


Fig. 5. Amplitude expansion and optimization: highest frequency region is reserved to embed positional data $\boldsymbol{\varphi}$, with length equals M. Embedding region is divided into three parts: extra payload (length $= N/2-128-M$ bits), hash data (length=128 bits), and positional data $\boldsymbol{\varphi}$(length $= M$).

These processes are repeated until no overflow or underflow occurs in $s(t)$. Figure 5 shows the amplitude optimization procedure.

Hash Data and Payload Data. Payload embed region is reserved due to amplitude expansion with value of $20log_{10}(2) \approx 6$dB. We used the SHA-1 standard to extract the hash code of the original data with a length of 128 bits. The hash function plays an important role in authentication schemes to verify the originality and authenticity of recordings in typical practical applications.

The highest frequency region ($m = M$) is reserved to embed the positional data ($|\boldsymbol{\varphi}|$; M bits), the hash data (128 bits), and overflow optimization data ($N/2-128-M$ bits). The location of these three data bits should be scrambled by using an embedding key to hide the information. Other frequency regions that are indicated by $\boldsymbol{\varphi}(m) = 1$ are used to store payload data. As shown in Figure 5, it is possible that there is not enough of a reserved embedding region for these three data, i.e. when N=1024 and M=16, the data length in one frequency zone is 1024/16=64. Here, it is necessary to reserve the multiple frequency domain from the highest frequency region to meet the real length of the payload in each frame.

3.2 Extraction and Reconstruction of Host Waveform

Our method focuses on the high-frequency spectrum domain. As seen in Figure 3, the input datum is the stego signal, and the output data are the re-constructed original signal and the detected feature value. After intDCT is applied to the stego signal $s(t)$, we obtain the intDCT coefficients $S(i)$.

The embedded data are extracted by applying modulo 2 to $S(i)$. The extracted and de-scrambled $|\varphi|$ data in the highest frequency region indicate which frequency region has been modified by embedding hidden data into it or by optimizing amplitude. The hash data are also extracted and de-scrambled from the same highest frequency region. The DCT coefficients of the high-frequency region that fulfills $|\varphi(m)| = 1(m > M/2)$ are divided by two to recover the original host spectrum $H(i)$. The DCT coefficients of the low-frequency region that satisfy $|\varphi(m)| = 1(m \leq M/2)$ are multiplied by $sqrt(2)$ and the extracted stored LSB data for the amplitude optimization are added to them to recover the original host spectrum $H(i)$. The reconstructed $h(t)$ is obtained by performing inverse intDCT.

Verification. We detect the feature value of the reconstructed original data, and compare it with the re-extracted feature value in the verification phase.

The extracted feature value (original hash) is compared with the feature value obtained from the reconstructed host signal. If they are not equal, the stego data are untrustworthy: i.e., they may have been modified by a third party. The main disadvantage is that signals are fragile and authentication fails when there are any attempts to modify the stego data, including regular modifications, such as sampling size conversions, compression, and re-sampling.

4 Experimental Evaluation

We evaluated our method on the RWC Music Database[19]. We did an experiment with an L-channel waveform with 44.1-kHz sampling and 16-bit quantization. The samples were cut to the initial 30 sec of playback time.

4.1 Verifiability

We marked 128 bits of the feature data extracted from the stego signal as the original *hash*, and we calculated the hash value of the reconstructed host data, *hash'*. We modified the stego data with Hex Editor, and after the extraction process, the system determined $hash \neq hash'$, which means that the modified data were verified as not being trustworthy. The precision for verification depends on the DCT size N.

4.2 Reversibility

We compared the reconstructed host data with the original data. The difference between the original data and the reconstructed host data was 0, which meant that our method was reversible. 100 tracks have been used and original cover data are reversed.

4.3 Imperceptibility

We evaluated the distortion in acoustic quality on the basis of the ITU-R BS.1387 (PEAQ) standard[20].

The impairment scale, an objective difference grade (ODG), ranged from 0 to -4 and could be interpreted as: *0* imperceptible, *-1* perceptible but not annoying, *-2* slightly annoying; *-3* annoying, and *-4* very annoying.

Acoustic data were re-sampled at 48 kHz before the evaluation. The experiments[1] were for total payloads including hash and positional data with different DCT sizes and the results are listed in Table 1. We used 100 tracks[2] for evaluation, and the ODG results are given in Figure 6; the average ODG was -0.201.

The ODG results seem to indicate poor audio-quality for some tunes. The annoying loss of quality was caused by discontinuities at the boundaries between the waveform segments in DCT operation. Figure 6 presents results obtained with the proposed steganography technique. Decreasing the number of embedding frequency regions, i.e., reducing the extra payload data, may reduce the degradation in sound quality.

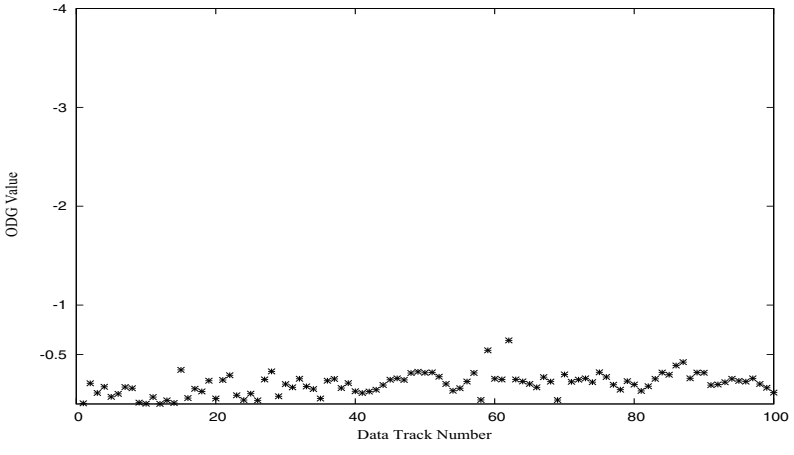


Fig. 6. ODG of 100 Tracks, N=2048

Table 1. Payload with different DCT sizes

DCT size	*Number of flames*	*Payload (Percentage)*
512	2584	620160 (23.5%)
1024	1292	640832 (24.3%)
2048	646	651168 (24.6%)

We calculated SegSNR result obtained with the proposed method using ATR 503 PB-5 from Japanese Newspaper Article Sentences (JNAS)[22] published by the Acoustical Society of Japan, with 73797 samples at 16 kHz and 16 bits, in

[1] We used RWC-MDB-G2001 No.10 with a 30 sec playback time.

[2] Samples at 44.1 kHz, and 16 bits, in mono with 30 sec of playback time (1323008 samples) and N=2048.

mono with 4.6 seconds of playback time and DCT size of $N = 2048$. We had 32.9dB as the segSNR value.

5 Conclusion

We proposed a reversible approach using steganography to verify whether acoustic data was free of tampering. A hash function was used to extract the feature value of the original cover data, and this feature value was used as the payload for verifying whether tampering had occurred. Amplitude expansions were concentrated in the high-frequency spectrum domain to make the hidden data less perceptible. A scheme to optimize amplitude prevented the amplitude of the stego signal from overflow/underflow. This scheme is a versatile way of guaranteeing the integrity and reliability of data without the need to apply additional conversion to object data.

5.1 Future Work

We intend to improve the robustness of this scheme in terms of steganalysis by increasing the amplification spectrum gradually or by using pseudo-random numbers as the key to amplitude modification.

Embedding was concentrated in the high-frequency component after it was transformed to the spectrum domain, which resulted in differences between low-frequency and high -frequency spectra. This was made more conspicuous by the boundary line caused by spectrum shifting. Two countermeasures were considered to blur boundaries and to improve the security of this scheme and we intend to implement and measure these in future work. The details on these are as follows:

I. *Gradually increase the amplification spectrum*

Instead of spectrum shifting in the high-frequency domain, the spectrum is amplified gradually. By doing this, the boundary line should theoretically be negligible.

II. *PN Key for shifting*

Use pseudo-random numbers as the shift key. This security key should be shared between the sender and trusted receiver. Using an arranged key-set of a certain length that is repeated as a loop is another option.

III. *Enhance algorithm for better audio quality*

Audio quality changes greatly with different music sources. Improvements are necessary, especially for speech contents, since most of which are most widely used as material evidence for practical purposes.

References

1. Vander, V.M., van Leest, A., Bruekers, F.: Reversible Audio Watermarking. Audio Engineering Society 5818 (2003)
2. Yan, D.Q., Wang, R.D.: Reversible Data Hiding for Audio Based on Prediction Error Expansion. In: Intelligent Information Hiding and Multimedia Signal Processing, China, pp. 249–252 (2008)

3. Gomez, E., Cano, P., Gomes, L.D., Batlle, E., Bonnet, M.: Mixed watermarking fingerprinting approach for integrity verification of audio recordings. In: International Telecommunications Symposium, ITS 2002, Natal, Brazil (2002)
4. Gulbis, M., Muller, E., Steinebach, M.: Audio integrity protection and falsification estimation by embedding multiple watermarks. In: International Conference on Intelligent Information (2006)
5. Mihçak, M.K., Venkatesan, R.: A Perceptual Audio Hashing Algorithm: A Tool for Robust Audio Identification and Information Hiding. In: Moskowitz, I.S. (ed.) IH 2001. LNCS, vol. 2137, p. 51. Springer, Heidelberg (2001)
6. Celik, M.U., Sharma, G., Tekalp, A.M., Saber, E.: Reversible Data Hiding. In: Proc. of International Conference on Image Processing, Rochester, NY, USA, vol. 2, pp. 157–160 (2002)
7. Chen, O.T.C., Liu, C.H.: Content-dependent watermarking scheme in compressed speech with identifying manner and location of attacks. IEEE Transactions on Audio, Speech, and Language Processing 15(5), 1605–1616 (2007)
8. Faundez, Z.M., Hagmuller, M., Kubin, G.: Speaker verification security improvement by means of speech watermarking. Speech Communication 48(12), 1608–1619 (2006)
9. Andreas, L., Jana, D.: Digital watermarking of biometric speech references: impact to the EER System Performance. In: Security, Steganography, and Watermarking of Multimedia Contents IX, vol. 6505, p. 650513 (2007)
10. Fridrich, J., Goljan, M., Du, R.: Invertible authentication. In: Security and Watermarking of Multimedia Contents III, USA, vol. 4314, pp. 197–208 (2001)
11. Radhakrishnan, R., Memon, N.D.: Audio content authentication based on psychoacoustic model. In: Delp, E.J., Wong, P.W. (eds.) Proc. SPIE, Security and Watermarking of Multimedia Contents IV, vol. 4675, pp. 110–117 (2002)
12. Zmudzinski, S., Steinebach, M.: Perception-Based Audio Authentication Watermarking in the Time-Frequency Domain. In: Katzenbeisser, S., Sadeghi, A.-R. (eds.) IH 2009. LNCS, vol. 5806, pp. 146–160. Springer, Heidelberg (2009)
13. Kalker, T., Haitsma, J.A., Oostveen, J.C.: Robust audio hashing for content identification. In: Content Based Multimedia Indexing (CBMI), Italy, pp. 2091–2094 (2001)
14. Celik, M.U., Sharma, G., Tekalp, A.M., Saber, E.: Localized lossless authentication watermark (LAW). In: International Society for Optical Engineering, USA, vol. 5020, pp. 689–698 (2003)
15. Tian, J.: Reversible data embedding using a difference expansion. IEEE Transactions on Circuits Systems and Video Technology 13(8), 890–896 (2003)
16. Chang, C.C., Tai, W.L., Lin, M.H.: A reversible data hiding scheme with modified side match vector quantization. In: Proceedings of the International Conference on Advanced Information Networking and Applications, Taiwan, vol. 1, pp. 947–952 (2005)
17. Bi, G.A., Zeng, Y.H.: Transforms and fast algorithms for signal analysis and representations, pp. 320–342. Birkhauser, Boston (2004)
18. Haibin, H., Susanto, R., Rongshan, Y.: A Fast Algorithm of Integer MDCT for Lossless Audio Coding. In: Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 177–180 (2004)
19. Goto, M., Hashiguchi, H., Nishimura, T., Oka, R.: RWC Music Database: Music Genre Database and Musical Instrument Sound Database. In: Proceedings of the 4th International Conference on Music Information Retrieval (ISMIR), pp. 229–230 (2003)

20. Kabal, P.: An Examination and Interpretation of ITU-R BS.1387: Perceptual Evaluation of Audio Quality. TSP Lab Technical Report, Dept. Electrical, Computer Engineering, McGill University, pp. 1–89 (2002)
21. Goljan, M., Fridrich, J., Du, R.: Distortion-free data embedding for images. In: Moskowitz, I.S. (ed.) IH 2001. LNCS, vol. 2137, pp. 27–41. Springer, Heidelberg (2001)
22. Kobayasi, T., Itahashi, S., Hayamizu, S., Takezawa, T.: ASJ continuous speech corpus for research. The Journal of the Acoustical Society of Japan 48(12), 888–893 (1992)

Author Index