J. Vis. Commun. Image R. 28 (2015) 21–27

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

Reversible data hiding in encrypted images based on absolute mean difference of multiple neighboring pixels ☆

Xin Liao [a,b,c,*], Changwen Shu [a]

[a] College of Computer Science and Electronic Engineering, Hunan University, Changsha, Hunan 410082, China
[b] Institute of Software, Chinese Academy of Sciences, Beijing 100190, China
[c] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China

ARTICLE INFO

Article history:
Received 16 August 2014
Accepted 26 December 2014
Available online 13 January 2015

Keywords:
Encrypted image
Reversible data hiding
Absolute mean difference
Multiple neighboring pixels
Data extraction
Image recovery
Extracted-bit error rate
Data embedding ratio

ABSTRACT

Recently, with the development of cloud computing, more and more secret data are stored in cloud. Reversible data hiding in encrypted images is a technique that makes contribution to cloud data management in privacy preserving and data security. In previous works, Zhang and Hong presented two reversible dada hiding methods in encrypted images, respectively. However, Zhang's work neglected the pixels in the borders of image blocks, and Hong et al.'s research only considered two adjacent pixels of each pixel. In addition, their works only considered that all image blocks are embedded into additional data. In this paper, we propose a novel method of evaluating the complexity of image blocks, which considers multiple neighboring pixels according to the locations of different pixels. Furthermore, data embedding ratio is considered. Experiments show that this novel method can reduce average extracted-bit error rate when the block size is appropriate.

1. Introduction

Reversible data hiding in images is a technique that allows the cover image to be recovered perfectly after the embedded message is extracted exactly from the marked image. In recent years, a large number of reversible data hiding methods have been proposed [1–9]. Tian embeds additional data bits by doubling the differences between two neighboring pixels [1]. Ni et al. embed data by modifying the pixel gray values using a histogram shift mechanism in [2]. Additional messages are embedded by taking advantage of the redundancy after lossless compression in Celik et al.'s work [3]. And Thodi et al. use the difference expansion in [1] and histogram shifting [2] to embed data [4]. Besides, other methods combined to the traditional reversible data hiding approaches also improve the performance [5–9].

As is known, encryption is used to protect the security and privacy of users' data. In some applications, the media used to carry additional bits is encrypted to be protected from being analyzed [10–12]. Besides, in cloud storage environment, one of the most important application scenarios, the exploit of encryption will bring a new challenge that data will lost its characters after encryption, which will make many current data processing methods no effect. As a result, signal processing of cipher text becomes one of the key issues. Nowadays, trust-management is a new security problem which cannot be solved by traditional techniques such as data backup, recovery backup, and firewall. In [13], a trust-management method based on reputation was improved by data hiding, which protect the content owner's privacy and data integrity to some extent. However, the scheme will cause data distortions when embedding messages. Therefore, we may be in favor of a reversible data hiding on encrypted media. Furthermore, we can make better use of reversible data hiding in encrypted images in other scenarios where the data hider has no right to access images from the content owner while the receiver has.

Recent years, researchers have presented some works on encrypted images after traditional reversible data hiding. In 2008, Puech et al. firstly introduced reversible data hiding in encrypted images [14]. And Zhang proposed a reversible data hiding in encrypted images which divides encrypted images into many blocks and extracts data and recover images according to the smoothness of image blocks [15]. Later Hong et al. improved Zhang's scheme by using a new method to calculate the smoothness of image blocks and exploiting the side match technique [16]. However, they both have some drawbacks. First, Zhang neglects the pixels in four borders of each image block when

☆ This paper has been recommended for acceptance by M.T. Sun.

* Corresponding author at: College of Computer Science and Electronic Engineering, Hunan University, Changsha, Hunan 410082, China.
E-mail address: xinliao@hnu.edu.cn (X. Liao).

http://dx.doi.org/10.1016/j.jvcir.2014.12.007
1047-3203/

calculating the fluctuation of each block, and Hong et al. take the pixels in the borders of each block into account, whereas only two adjacent pixels are employed in evaluating the smoothness of each block. Second, they only consider that all image blocks are embedded into additional data. Therefore, this work proposes a new more precise function to estimate the complexity of each image block which employs two, three or four adjacent pixels according to coordinate of each pixel, and it considers data embedding ratio fully.

This paper is organized as follows. Section 2 introduces some related works such as Zhang's and Hong et al.'s method briefly. Section 3 describes the detailed procedures of the proposed novel method. And we elaborate the experimental results and compare the performances among Zhang's, Hong et al.'s and the proposed one in Section 4. Conclusions are given in Section 5.

2. Related work

Reversible data hiding in encrypted images was first introduced by Puech et al., and then in 2011 Zhang proposed a novel version by dividing image into blocks and employing LSB plane. Later Hong et al. presented an improvement in data extraction and image recovery.

In Zhang's method, the content owner first encrypts the image by a bitwise exclusive-or operation. Then the data hider will divide the image into lots of blocks with size of s and embed an additional bit into each block by adopting 3 LSBs plane after segmenting each block into two parts. The receiver will first decrypt the marked encrypted image and divide the received image into blocks with the same size s, then each block will be separated into two equal-sized sets and data extraction/image recovery will be performed according to the fluctuation of each block. Zhang's fluctuation function is as follows

$$f = \sum_{u=2}^{s-1}\sum_{v=2}^{s-1}\left|p_{u,v} - \frac{p_{u,v-1}+p_{u-1,v}+p_{u,v+1}+p_{u+1,v}}{4}\right| \tag{1}$$

where $p_{u,v}$ denotes the value of a pixel that locates at (u, v).

In Hong et al.'s method, they improved data extraction/image recovery based on Zhang's method. Firstly, they proposed a new function as Eq. (2) where $p_{u,v}$ is the pixel value located at position (u, v) of image block with size $s_1 \times s_2$ to estimate the smoothness of the block. It considers more pixels so that the extracted-bit error rate is decreased.

$$f = \sum_{u=1}^{s_2}\sum_{v=1}^{s_1-1}|p_{u,v} - p_{u,v+1}| + \sum_{u=1}^{s_2-1}\sum_{v=1}^{s_1}|p_{u,v} - p_{u+1,v}| \tag{2}$$

Secondly, data extraction/image recovery is performed according to the descending order of the absolute smoothness difference between two candidate blocks. At last, side match technique is used to further decrease the extracted-bit error rate.

3. The proposed method

According to Zhang's method and Hong et al.'s method, it is not difficult to find that the evaluation of the complexity of image blocks is of big significance to decrease the extracted-bit error rate. However, their methods either ignore some pixels or do not employ all neighboring pixels when calculating the complexity of image blocks. Based on the above analysis, we propose a new more precise function to calculate the complexity of image blocks. Besides that, we consider data embedding ratio fully, that is to say, data hider can choose some blocks to embed additional data if the embedding capacity is small. Side match technique is also used to increase the correct rate and its details can be referenced in [16].

3.1. Evaluation of the complexity of image blocks

In order to further reduce the error rate, here a new calculation of block complexity is proposed. The complexity of image blocks can be estimated by calculating the absolute mean difference of pixels and their neighboring pixels. Fig. 1 shows distribution of the adjacent pixels of a given pixel. The grids circled in red in different positions stand for those pixels to be calculated, and those marked color are their neighboring pixels. Three kinds of pixels according to their coordinates are shown in Fig. 1. ① The first class is the pixels which have two neighboring pixels marked cyan. ② The second class is the pixels that have three adjacent pixels marked yellow. ③ The third class is the pixels which have four neighboring pixels marked orange. Different classes use different functions for calculating the complexity. The function f_1 is used for the pixel locating at one of four vertexes of an image block. The function f_2 is used for the pixels which locate at one of four borders except four vertexes. The function f_3 is used for the pixels in the middle of a block.

Specifically, for an image block of $s_1 \times s_2$ pixels, f_1 in Eq. (3) is used to calculate the summation of the absolute mean difference of pixels in the four vertexes and two adjacent pixels.

$$f_1 = \left|p_{1,1} - \frac{p_{1,2}+p_{2,1}}{2}\right| + \left|p_{1,s_2} - \frac{p_{1,s_2-1}+p_{2,s_2}}{2}\right| + \left|p_{s_1,1} - \frac{p_{s_1,2}+p_{s_1-1,1}}{2}\right| + \left|p_{s_1,s_2} - \frac{p_{s_1,s_2-1}+p_{s_1-1,s_2}}{2}\right| \tag{3}$$

For the pixels in four borders of each block except four vertexes, f_2 in Eq. (4) is employed to evaluate the summation of the absolute mean difference of them and their three neighboring pixels.

$$f_2 = \sum_{v=2}^{s_2-1}\left|\left(p_{1,v} - \frac{p_{1,v-1}+p_{1,v+1}+p_{2,v}}{3}\right) + \left(p_{s_1,v} - \frac{p_{s_1,v-1}+p_{s_1,v+1}+p_{s_1-1,v}}{3}\right)\right| + \sum_{u=2}^{s_1-1}\left|\left(p_{u,1} - \frac{p_{u-1,1}+p_{u+1,1}+p_{u,2}}{3}\right) + \left(p_{u,s_2} - \frac{p_{u-1,s_2}+p_{u+1,s_2}+p_{u,s_2-1}}{3}\right)\right| \tag{4}$$

For the pixels in the middle of each block, f_3 in Eq. (5) is adopted to estimate the summation of the absolute mean difference of them and their four adjacent pixels.

$$f_3 = \sum_{u=2}^{s_1-1}\sum_{v=2}^{s_2-1}\left|p_{u,v} - \frac{p_{u,v-1}+p_{u-1,v}+p_{u,v+1}+p_{u+1,v}}{4}\right| \tag{5}$$

Overall, the total function F in Eq. (6) is used to calculate the whole summation. Namely, F is employed to calculate the complexity of the image block.

$$F = f_1 + f_2 + f_3 \tag{6}$$

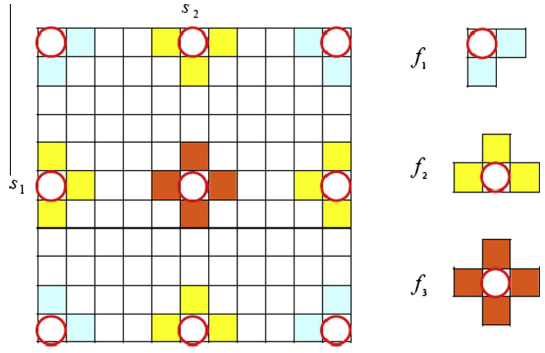

Fig. 1. Distribution of adjacent pixels of a given pixel.

3.2. Procedures of the proposed method

As shown in Fig. 2, the content owner will first encrypt the cover image according to encryption key, and then send the encrypted image to data hider. For the encrypted image, the data hider will choose an appropriate data embedding ratio to embed additional data with data hiding key, and then send the stego image (the marked encrypted image) to the receiver. With a stego image, the receiver will first decrypt it to obtain a decrypted one containing data, and then select some image blocks relying on data embedding ratio to extract data and recover image using previous data hiding key. Thus, additional data will be extracted accurately and image will be recovered successfully.

3.2.1. Image encryption

As for image encryption, lots of works have been published. Here we adopt the image encryption algorithm identical to Zhang's [15] and Hong et al.'s [16], in order to compare them conveniently and impartially. In the future, we will endeavor to do further research on the influence of image encryption on the proposed method.

The content owner encrypts the original image by calculating the exclusive-or results of the original bits of pixels and a stream cipher generated by an encryption key. Let P be the cover image of size $M \times N$, and $p_{i,j}$ be the value of a pixel locating at (i,j). Assume the pixel value $p_{i,j}$ ranges from 0 to 255 which can be represented by 8 bits $p^0_{i,j}, p^1_{i,j}, p^2_{i,j}, \ldots, p^7_{i,j}$. Thus, we have

$$p^k_{i,j} = \left\lfloor \frac{p_{i,j}}{2^k} \right\rfloor \bmod 2, \quad k = 1,2,\ldots,7 \tag{7}$$

For the encrypted image C, the encrypted bits $C^k_{i,j}$ can be calculated by the following exclusive-or operation

$$C^k_{ij} - = P^k_{ij} \oplus r^k_{ij}, \quad k = 1,2,\ldots,7 \tag{8}$$

where r^k_{ij} is generated by an encryption key using a standard stream cipher. Then the encrypted data $C_{i,j}$ can be obtained

$$C_{i,j} = \sum_{k=0}^{7} C^k_{i,j} \times 2^k \tag{9}$$

3.2.2. Data embedding

For an encrypted image, data hider cannot obtain its contents and has no right to access it. In order to manage the encrypted image well, he will embed additional secret data. The detailed embedding steps are as follows.

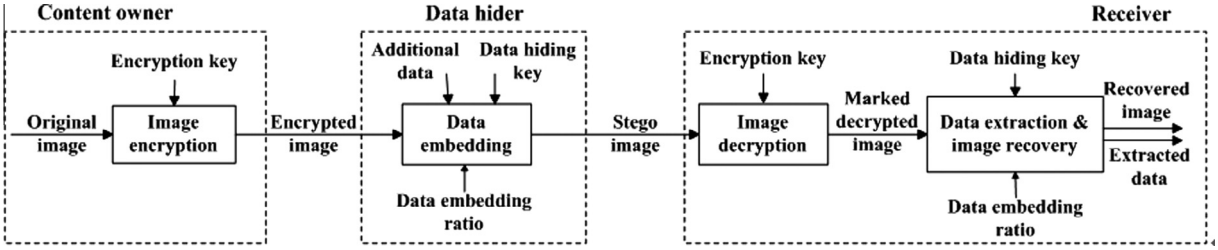

Fig. 2. Overview of the proposed method.

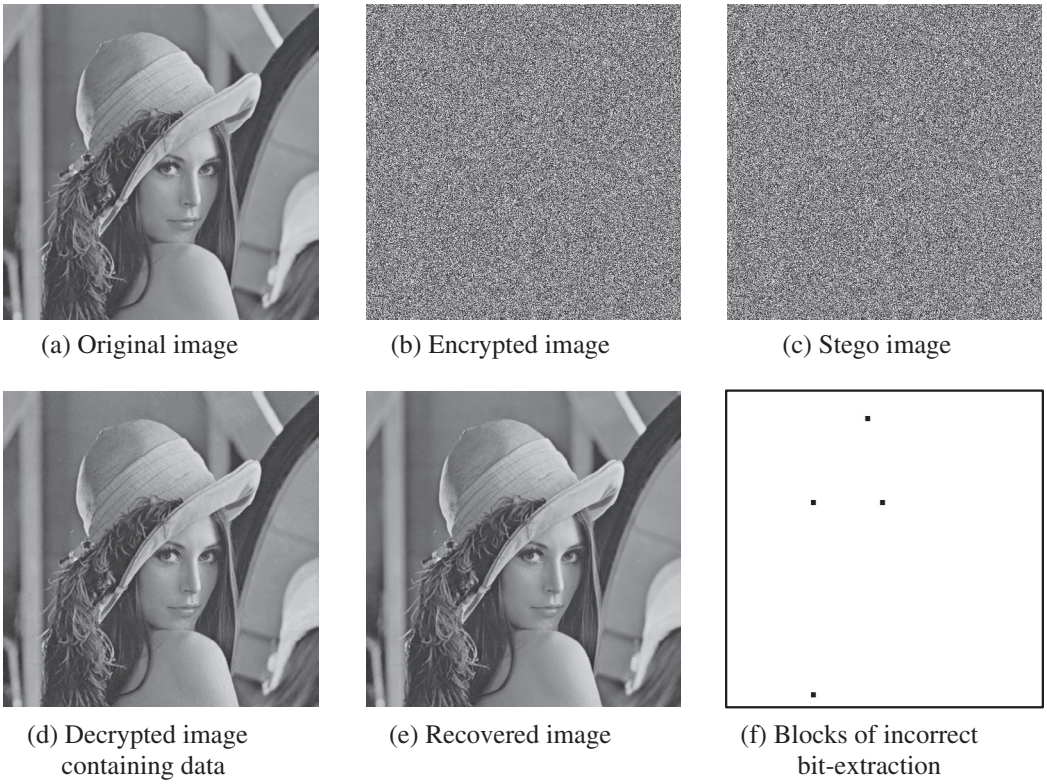

Fig. 3. The changes of the classic image "Lena" in different phases.

Table 1
PSNR values with different embedding ratios.

p	1.0	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1
PSNR (dB)	37.85	38.40	38.91	39.48	40.14	40.90	41.86	43.08	44.81	47.81

Step 1: Divide the encrypted image of $M \times N$ pixels into blocks of $s_1 \times s_2$ pixels. That is to say, $\lfloor \frac{M}{s_1} \rfloor \times \lfloor \frac{N}{s_2} \rfloor$ blocks will be used for data embedding, where $\lfloor \cdot \rfloor$ means the floor function.

Step 2: For each block, separate $s_1 \times s_2$ pixels into two equal-sized sets S_0 and S_1 according to the data hiding key. The pixels $p_{u,v}$ in S_0 satisfy $1 \leqslant u \leqslant s_1/2$ and $1 \leqslant v \leqslant s_2$, and the pixels $p_{u,v}$ satisfying $s_1/2 + 1 \leqslant u \leqslant s_1$ and $1 \leqslant v \leqslant s_2$ belong to S_1.

Step 3: Select $\lfloor \frac{M}{s_1} \rfloor \times \lfloor \frac{N}{s_2} \rfloor \times p$ blocks to embed data depending on the embedding data ratio p.

Step 4: For each block, flip the 3 LSBs in S_0 if the additional bit to be embedded is "0" and denote the marked encrypted bits (stego bits) as

$$C'^k_{ij} = \overline{C^k_{ij}}, \ (i,j) \in S_0 \quad \text{and} \quad k = 0,1,2 \tag{10}$$

Otherwise, flip the 3 LSBs in S_1 and denote the marked encrypted bits (stego bits) as

$$C'^k_{ij} = \overline{C^k_{ij}}, \ (i,j) \in S_1 \quad \text{and} \quad k = 0,1,2 \tag{11}$$

As a result, an additional bit is embedded successfully. Repeat this process until all the additional data bits are embedded.

3.2.3. Data extraction and image recovery

With a stego image, receiver can extract data and recover image without knowing the content of cover image. The following steps are used to extract data and recover image.

Step 1: Decryption of the marked encrypted image is similar to the procedures of image encryption. For those encrypted flipped bits $C'^k_{i,j}$, the marked decrypted bits can be calculated by $C'^k_{i,j} \oplus r^k_{i,j} = \overline{C^k_{i,j}} \oplus r^k_{i,j} = \overline{p^k_{i,j} \oplus r^k_{i,j}} \oplus r^k_{i,j} = \overline{p^k_{i,j}}$. In the same block, those bits that have not been flipped without embedded bits will be the same as the original bit $p^k_{i,j}$.

Step 2: Partition the secret image into blocks of $s_1 \times s_2$ pixels, which is identical to the beginning of data embedding. Select $\lfloor \frac{M}{s_1} \rfloor \times \lfloor \frac{N}{s_2} \rfloor \times p$ blocks to embed data according to the embedding data ratio p. Separate the pixels into two sets S_0 and S_1 for each selected block as did in data embedding.

Step 3: Denote the new block obtained by flipping the 3 LSBs of pixels in S_0 as H_0. Similarly, denote the new block obtained by flipping the 3 LSBs of pixels in S_1 as H_1.

Step 4: Apply Eqs. (3)–(6) to estimate the complexity of H_0 and H_1, and the results are denoted as F and F' respectively. Calculate the difference of F and F'. If the value of $|F - F'|$ is larger, H_0 is not similar to H_1, which means the image block becomes more complex than that before flipping 3 LSBs.

Step 5: The data extraction and image recovery will be more correct by using the descending order of $|F - F'|$ after calculating all image blocks. For an image block $H_{(x,y)}$ located at (x, y) to be recovered. Here we take $H_{(x,y)}$ which locates at one of four borders for an easy example. Denote its neighboring blocks as $H_{(x,y-1)}$, $H_{(x,y+1)}$ and $H_{(x+1,y)}$ respectively.

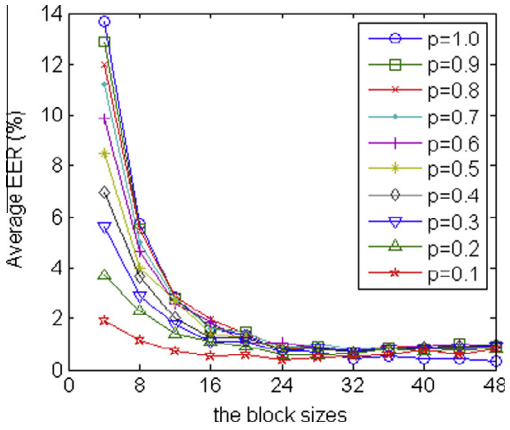

Fig. 4. Average EER with respect to block sizes.

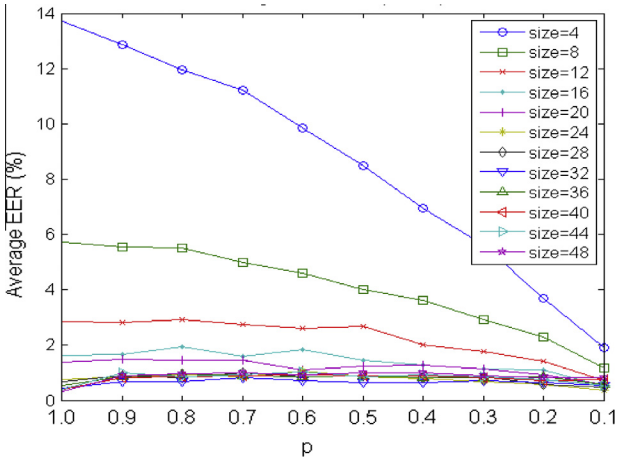

Fig. 5. Average EER with respect to embedding ratio.

Case 1: If all neighboring blocks of $H_{(x,y)}$ are not recovered, data extraction/image recovery will be executed. In other words, if $|F - F'| < 0$, the extracted bit will be "0" and H_0 is the original block. Otherwise, the extracted bit will be "1" and H_1 is the original block.

Case 2: If any of neighboring blocks of $H_{(x,y)}$ has been recovered, concatenate the border pixels of the recovered blocks to H_0 and H_1 to acquire a new block H^0_a and H^1_a respectively. Then calculate the complexity of H^0_a and H^1_a according to Eqs. (3)–(6) and the values are denoted as F_a and F'_a respectively. If $F_a < F'_a$, the extracted bit will be "0" and the original block is H^0_a. Otherwise, the extracted bit will be "1" and the original block is H^1_a.

As a result, the additional data can be obtained by combining all the extracted bits, and the original image can be recovered by concatenating all the recovered blocks according to the coordinate of image blocks.

(a) Lena (b) Peppers (c) Toy (d) Goldhill

Fig. 6. Four gray scale images.

4. Experimental results

In this section, several experimental results will be given to demonstrate the effectiveness of our proposed method.

4.1. Evaluation of the proposed method

To comprehend the proposed method fully, Fig. 3 is used to present the changes of the classic image "Lena" in different phases. Fig. 3(a) shows the original image. Fig. 3(b) and (c) presents the encrypted image without secret data and containing secret data, respectively. Fig. 3(d) shows the decrypted image containing data, and Fig. 3(e) gives the recovered image, which are visually indistinguishable from the original image. The blocks marked black in Fig. 3(f) indicate the blocks of incorrect bit-extraction, accounting for merely 0.07% of the whole image pixels.

The mean square error (MSE) between decrypted image containing data and the original image can be theoretically calculated by the following equation

$$\mathrm{MSE} = \frac{p}{2}\sum_{i=1}^{3}(2^{i-1})^2 = 10.5p \tag{12}$$

where p is data embedding ratio. Thus, the peak signal-to-noise ratio (PSNR) of the decrypted image containing data can be theoretically obtained by

$$\mathrm{PSNR} = 10 \times \log_{10}\frac{255^2}{\mathrm{MSE}} = 10 \times \log_{10}\frac{2 \times 255^2}{21p} \tag{13}$$

The smallest PSNR in theory is 37.92 dB when p is equal to 1.0, which is almost same to the following experimental results. We have used 100 images of size 384×512 or 512×384 randomly

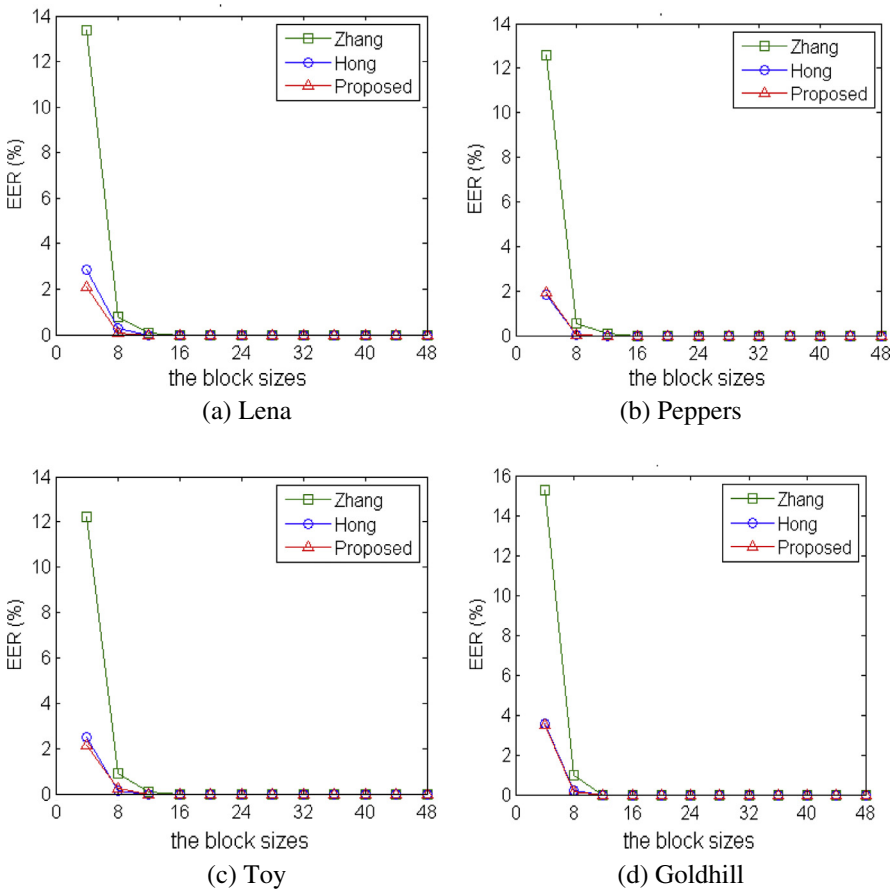

Fig. 7. Comparisons of performances based on four gray scale images.

Table 2
Comparisons of average EER (%) with respect to block sizes when $p = 1.0$.

Size	4	8	12	16	20	24	28	32	36	40	44	48
Zhang	26.33	10.11	5.17	3.17	2.17	1.58	1.41	1.16	1.06	0.81	0.74	0.68
Hong	13.01	5.59	3.13	1.96	1.87	1.25	1.29	1.14	1.04	0.94	0.93	0.75
Proposed	13.70	5.74	2.84	1.62	1.39	0.76	0.69	0.44	0.54	0.41	0.40	0.31

Table 3
Comparisons of average EER (%) with respect to block sizes when $p = 0.9$.

Size	4	8	12	16	20	24	28	32	36	40	44	48
Zhang	24.21	9.59	5.02	3.15	2.21	1.67	1.60	1.38	1.39	1.21	1.36	1.28
Hong	12.26	5.40	3.11	2.00	1.93	1.38	1.52	1.41	1.36	1.39	1.48	1.31
Proposed	12.86	5.54	2.80	1.66	1.47	0.88	0.90	0.69	0.86	0.82	1.00	0.90

Table 4
Comparisons of average EER (%) with respect to block sizes when $p = 0.5$.

Size	4	8	12	16	20	24	28	32	36	40	44	48
Zhang	14.67	6.90	4.14	2.79	2.05	1.66	1.56	1.32	1.44	1.28	1.17	1.11
Hong	8.25	4.23	2.93	2.01	1.82	1.50	1.59	1.28	1.39	1.42	1.45	1.34
Proposed	8.50	3.99	2.68	1.44	1.26	0.84	0.88	0.66	0.89	0.89	0.84	0.99

Table 5
Comparisons of average EER (%) with respect to block sizes when $p = 0.1$.

Size	4	8	12	16	20	24	28	32	36	40	44	48
Zhang	3.16	1.79	1.27	0.96	0.81	0.74	0.79	0.77	0.79	0.89	0.85	0.91
Hong	1.97	1.19	0.94	0.63	0.79	0.61	0.65	0.70	0.74	0.92	0.76	0.94
Proposed	1.91	1.17	0.71	0.50	0.57	0.38	0.46	0.53	0.56	0.77	0.56	0.81

selected from UCID database [17]. Table 1 shows the results of the proposed method in terms of PSNR values and different embedding ratios.

Extracted-bit error rate (EER) is the proportion of blocks extracted error bits in the all blocks. We analyze the relationship between average EER and block sizes in Fig. 4. It can be observed that the average EER decreases monotonically with the increase of block sizes when the data embedding ratio p is fixed. Fig. 5 indicates that the average EER decreases as data embedding ratio p decreases.

4.2. Comparisons of performances

In this subsection, we first take four gray scale images as test images, which are shown in Fig. 6. Experimental comparisons among the proposed method and these methods in [15,16] are shown in Fig. 7. It is shown that the proposed method have smaller average EER.

To further evaluate the performance, we compare these methods for test images randomly selected from UCID database [17]. The relationships of average EER with respect to block sizes or data embedding ratio are analyzed. Tables 2–5 represent comparison results in the following cases: block sizes $size \in \{4,8,12,16,20,24,28,32,36,40,44,48\}$, embedding ratio $p \in \{1.0,0.9,0.5,0.1\}$.

It is shown that the proposed method always performs better than Zhang's. Compared with Hong et al.'s, the proposed method has a better performance when the block size is bigger than 8, and has a similar performance when the block size is small. Note that in Table 2 the average EER of the proposed method increase by about 0.69% and 0.15% than Hong et al.'s when the block size is 4 and 8 respectively. When the block size is larger than 8, the average EER of the proposed one is always lower. Similar analysis results can be obtained from Tables 3 and 4. Also, we can acquire better results through adjusting the embedding ratio. For instance, in Table 5, the proposed method always performs better than Hong et al.'s method when the embedding ratio is 10%. Reversible data hiding in encrypted images is a technique that makes contribution to cloud data management in privacy preserving and data security. Usually, we just embed little data into the encrypted images for cloud managing, such as the identifications of the owner. Thus, it would be better to use the larger block size to scatter the additional data over the encrypted images, and get the lower extracted-bit error rate.

5. Conclusions

In this paper, we propose an improved method based on Zhang's and Hong et al.'s works. A new more precise function is present to estimate the complexity of each image block and increase the correctness of data extraction/image recovery, i.e., decrease the average extracted-bit error rate. The data embedding ratio is also considered when data embedding and data extraction/image recovery are performed. Our experimental results show the superiority of the proposed one, especially when the block size is large and the embedding ratio is small.

In the future, besides the achievements in this paper, we will endeavor to do further research on the influence of image encryption. Also we will try to modify and improve the proposed method, in order to further reduce the average extracted-bit error rate.

Acknowledgments

This work is supported by National Natural Science Foundation of China (Grant No. 61402162), Specialized Research Fund for the Doctoral Program of Higher Education (Grant No. 20130161120004), China Postdoctoral Science Foundation (Grant No. 2014M560123), Hunan Provincial Natural Science Foundation of China (Grant No. 14JJ7024), Young Teacher Foundation of Hunan University (Grant No. 531107040701).

References

[1] J. Tian, Reversible data embedding using a difference expansion, IEEE Trans. Circ. Syst. Video Technol. 13 (8) (2003) 890–896.
[2] Z. Ni, Y.-Q. Shi, N. Ansari, W. Su, Reversible data hiding, IEEE Trans. Circ. Syst. Video Technol. 16 (3) (2006) 354–362.
[3] M.U. Celik, G. Sharma, A.M. Tekalp, E. Saber, Lossless generalized-LSB data embedding, IEEE Trans. Image Process. 14 (2) (2005) 253–266.
[4] D.M. Thodi, J.J. Rodriguez, Expansion embedding techniques for reversible watermarking, IEEE Trans. Image Process. 16 (3) (2007) 721–730.
[5] C.-C. Chang, C.-C. Lin, Y.-H. Chen, Reversible data-embedding scheme using differences between original and predicted pixel values, Inform. Secur. 2 (2) (2008) 35–46.
[6] L. Luo, Z. Chen, M. Chen, X. Zeng, Z. Xiong, Reversible image watermarking using interpolation technique, IEEE Trans. Inform. Forensics Secur. 5 (1) (2010) 187–193.
[7] S.W. Jung, L.T. Ha, S.J. Ko, A new histogram modification based reversible data hiding algorithm considering the human visual system, IEEE Signal Process. Lett. 18 (2) (2011) 95–98.
[8] C. Qin, C.C. Chang, Y.H. Huang, L.T. Liao, An inpainting-assisted reversible steganographic scheme using a histogram shifting mechanism, IEEE Trans. Circ. Syst. Video Technol. 23 (7) (2013) 1109–1118.
[9] Y.Y. Tsai, D.S. Tsai, C.L. Liu, Reversible data hiding scheme based on neighboring pixel differences, Digital Signal Process. 23 (3) (2013) 919–927.
[10] D. Kundur, K. Karthik, Video fingerprinting and encryption principles for digital rights management, Proc. IEEE 92 (2004) 918–932.
[11] S. Lian, Z. Liu, Z. Ren, H. Wang, Commutative encryption and watermarking in video compression, IEEE Trans. Circ. Syst. Video Technol. 17 (6) (2007) 774–778.
[12] M. Cancellaro, F. Battisti, M. Carli, G. Boato, F.G.B. Natale, A. Neri, A commutative digital image watermarking and encryption method in the tree structured haar transform domain, Signal Process.: Image Commun. 26 (1) (2011) 1–12.
[13] K. Hwang, D. Li, Trusted cloud computing with secure resources and data coloring, IEEE Internet Comput. 14 (5) (2010) 14–22.
[14] W. Puech, M. Chaumont, O. Strauss, A reversible data hiding method for encrypted images, in: Proc. of Security, Forensics, Steganography, and Watermarking of Multimedia Contents X 6819, 2008.
[15] X. Zhang, Reversible data hiding in encrypted images, IEEE Signal Process. Lett. 18 (4) (2011) 255–258.
[16] W. Hong, T. Chen, H. Wu, An improved reversible data hiding in encrypted images using side match, IEEE Signal Process. Lett. 19 (4) (2012) 199–202.
[17] G. Schaefer, M. Stich, UCID – an uncompressed color image database, in: Proc. SPIE, Storage and Retrieval Methods and Applications for Multimedia, 2004, pp. 472–480.