Reversible Data Hiding by Image Encryptions and Spatial Correlation Comparisons*

YA-LIN LEE[1] AND WEN-HSIANG TSAI[2,3]
[1]*Institute of Computer Science and Engineering*
National Chiao Tung University
Hsinchu, 30010 Taiwan
[2]*Department of Computer Science*
National Chiao Tung University
Hsinchu, 30010 Taiwan
[3]*Department of Information Communication*
Asia University
Taichung, 41354 Taiwan

A reversible data hiding scheme based on double image encryptions and spatial correlation comparisons is proposed, which improves the performance of two previous methods when dealing with flat cover images. A grayscale cover image is first encrypted. The four LSBs of each block pixel in the resulting encrypted image are encrypted once more to embed a message bit, thereby solving the problem of being unable to handle flat images encountered by the previous methods which flip three LSBs of each involved pixel at this stage. Data extraction and image recovery are achieved by a scheme of checking gray-value differences between blocks, which utilizes spatial correlations existing in nature images. Moreover, a side-match scheme that makes use of the spatial correlations of both recovered and unrecovered blocks is proposed to decrease the bit extraction error rate, in contrast with the two previous methods that consider only recovered blocks. Experimental results showing the feasibility of the proposed method are also included.

Keywords: image encryption, image recovery, reversible data hiding, spatial correlation, side-match

1. INTRODUCTION

Reversible data hiding [1]-[5] is a type of method by which secret messages can be embedded into cover media, and the original content of the cover media can be recovered *losslessly* after the message is extracted later. Such methods may be utilized for applications like covert communication, copyright protection, document authentication, secret keeping, etc. Many reversible data hiding methods have been proposed. Celik *et al.* [1] proposed a generalized-LSB method that utilizes a lossless compression scheme to create an additional space for data embedding. Tian [2] and Hu *et al.* [3] proposed difference expansion methods that explore data redundancy in images to

Received March 11, 2013; revised September 13, 2013.
* This work was supported in part by the NSC, Taiwan under Grant No. 101-3113-P-009-006 and Grant No. 102-2218-E-009-003, and in part by the Ministry of Education, Taiwan under the 5-year Project II of "Aiming for the Top University" from 2011 through 2015.

achieve high embedding capacities. Ni *et al.* [4] and Lee and Tsai [5] proposed histogram modification methods that shift values around histogram peaks to embed secret messages.

Furthermore, in some applications, the contents of cover media, such as those used in military and medical applications, need be protected to prevent leaking important information existing in the cover media to the public. Therefore, some methods jointing data hiding and encryption techniques have been proposed [6]-[8], in which a part of the cover media is encrypted and the other part is used for data embedding. Such methods, however, reveal undesirably the content of the second part of the cover media. To provide higher security, Zhang [9] proposed a method that can prevent people, including the data hider, from realizing the cover media content before or after the data embedding process is performed. Specifically, the cover image is encrypted *entirely*, instead of partially, by the content owner using a key, and the result is then delivered to the data hider for data embedding. Also, the original cover image can be recovered after the embedded data are extracted. Hong *et al.* [10] improved Zhang's method [9] by using a side-match scheme based on uses of spatial correlations of adjacent blocks.

Either [9] and [10] embeds a message into an encrypted image by flipping the three least significant bits (LSBs) of a portion of the pixels of each image block to embed a message bit. Data extraction and image recovery are achieved by using spatial correlations. However, such a message embedding scheme suffers from a problem which occurs when the cover image is a *flat image*, i.e., when the cover image has lots of *smooth* regions with the characteristic that the most significant bits (MSBs) of the pixels in each of such regions are all the same. Recently, Hong *et al.* [11] proposed another method that can embed messages into a flat image. However, their method is based on LSB flipping as in [9]-[10], and the spatial similarity of the original LSBs of the pixels in each block will be kept undesirably by their method using the flipping function.

In this paper, the LSBs of each block pixel in an encrypted image are *encrypted further*, rather than flipped, to embed a message bit, thereby solving the aforementioned problem encountered in [9] and [10] which is caused by flat cover images. Also, the spatial similarity of the original LSBs of the pixels in each block is broken by the encryption function. Moreover, for each pixel of a block in the encrypted image, *four* LSBs, instead of three, are utilized for message embedding, and a side-match scheme that utilizes the spatial correlations of *both* recovered and unrecovered blocks are proposed to decrease the bit-extraction error rate, in contrast with [10] and [11] which utilize the spatial correlations of recovered blocks only.

In the remainder of this paper, a review of the methods by Zhang [9] and Hong *et al.* [10] is given in Sec. 2. The proposed method is described in Sec. 3. Experimental results are presented in Sec. 4 to show the feasibility of the proposed method, followed by conclusions in Sec. 5.

2. REVIEW OF METHODS BY ZHANG [9] AND HONG ET AL. [10]

The reversible data hiding scheme proposed in Zhang [9] for grayscale images includes three phases: 1) image encryption, 2) message embedding, and 3) message

extraction and image recovery. In the first phase, a cover image I is encrypted by performing the exclusive-OR (XOR) operation $\oplus$ on all bits and their corresponding *random bits* generated by the use of an *encryption key* K_e and a random number generator PR_e. Specifically, by denoting the value of each pixel $P_{i,j}$ in I by $p_{i,j}$, each bit of $P_{i,j}$ by $b_{i,j,k}$, and the generated random bit corresponding to $b_{i,j,k}$ by $r_{i,j,k}$, the encryption of I is conducted by replacing $b_{i,j,k}$ by $b_{i,j,k}' = b_{i,j,k} \oplus r_{i,j,k}$ for all i, j, and k, resulting in an *encrypted image* I' with pixels $p_{i,j}'$ and bits $b_{i,j,k}'$.

In the message embedding phase, I' is divided into blocks $B_{m,n}$ of size $s \times s$, with each block used to carry a bit. Specifically, firstly each pixel $P_{i,j}'$ in $B_{m,n}$ is assigned into two *random sets* S_0 and S_1 using a *data-hiding key* K_h and another random number generator PR_h. Then, if the bit to be embedded into $B_{m,n}$ is 0, the three LSBs $b_{i,j,k}'$ of each pixel $P_{i,j}'$ in S_0 are flipped to be their complements $\overline{b_{i,j,k}}'$, resulting in a new pixel $P_{i,j}''$ with value $p_{i,j}''$; else, the three LSBs of each pixel $P_{i,j}'$ in the other random set S_1 are flipped. The resulting image is denoted by I''.

In the last phase — message extraction and image recovery, firstly I'' is decrypted to obtain a decrypted *image* I''' by performing an XOR operation $\oplus$ on every bit $b_{i,j,k}''$ in I'' and the corresponding random bit $r_{i,j,k}$ re-generated by the encryption key K_e. By denoting the resulting *decrypted pixels* and *decrypted bits* as $P_{i,j}'''$ and $b_{i,j,k}'''$, respectively, it can be seen that the five MSBs of $P_{i,j}'''$ in I''' are *identical* to the corresponding ones of the original pixel $P_{i,j}$ in I but the *three LSBs are not*. Next, by using the data-hiding key K_h, the random sets S_0 and S_1 may be re-generated for each *decrypted block* $B_{m,n}'$ of I'''. Then, the three LSBs of the pixels in S_0 and those in S_1 are both flipped to form blocks $H_{m,n,0}$ and $H_{m,n,1}$, respectively. It can be figured out that one of $H_{m,n,0}$ and $H_{m,n,1}$ is identical to the *original* block $B_{m,n}$ in I, and the other is an *interfered* version of $B_{m,n}$ since all the three LSBs of the latter have been flipped. Because of the spatial correlations of pixels in natural images, the original block is usually smoother than the interfered version. So the embedded bit can be extracted, as done in [9], by using a block smoothness measure f as follows:

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

where $p_{u,v}$ denotes the value of a pixel $P_{u,v}$ in the block. Specifically, by denoting the smoothness values calculated of $H_{m,n,0}$ and $H_{m,n,1}$ as $f_{m,n,0}$ and $f_{m,n,1}$, respectively, bit extraction is conducted by the rule: if $f_{m,n,0} < f_{m,n,1}$, a bit 0 is extracted and $H_{m,n,0}$ is taken as the original block; else, a bit 1 is extracted and $H_{m,n,1}$ is taken as the original block.

The reversible data hiding scheme proposed in Hong *et al.* [10] is the same as described above except that a different smoothness measure as follows with a better effect is used:

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

Moreover, a side-match scheme is adopted, using additionally the spatial correlations of *adjacent recovered blocks* to reduce the error rate of bit extraction.

A problem with [9] occurs when the five MSBs of the pixels in a block are *all identical* — a case encountered when the cover image is *flat*. The problem is: the smoothness measures $f_{m,n,0}$ and $f_{m,n,1}$ computed of $H_{m,n,0}$ and $H_{m,n,1}$, respectively, become *the same* in such a case, as proved below, and the resulting bit extraction will so be *no better than random guesses*, i.e., the probability to extract a correct bit is about 1/2.

Let the five MSBs of *each* pixel $P_{i,j}$ in the *original* block $B_{m,n}$ be denoted identically as b_8~b_4 under the assumption of image flatness. Also, let the pixels in $H_{m,n,0}$ and $H_{m,n,1}$ be denoted as $P_{u,v,0}$ and $P_{u,v,1}$ with values $p_{u,v,0}$ and $p_{u,v,1}$, respectively. Without loss of the generality, let $H_{m,n,0}$ be the one identical to the original block $B_{m,n}$ and $H_{m,n,1}$ the interfered one. Thus, $p_{u,v,0}$ in $H_{m,n,0}$ are all identical to the corresponding pixel values $p_{u,v}$ of $B_{m,n}$, and the three LSBs of $p_{u,v,1}$ in $H_{m,n,1}$ are the flipped versions of $p_{u,v}$ in $B_{m,n}$. That is, the five MSBs of each pixel in $H_{m,n,0}$ and $H_{m,n,1}$ are identically b_8~b_4, respectively; and if the three LSBs of each pixel $P_{u,v,0}$ in $H_{m,n,0}$ are $b_{u,v,3}$, $b_{u,v,2}$, and $b_{u,v,1}$, then the three LSBs of the corresponding pixel $P_{u,v,1}$ in $H_{m,n,1}$ are just $\overline{b_{u,v,3}}$, $\overline{b_{u,v,2}}$, and $\overline{b_{u,v,1}}$. Note that $b_{u,v,k} + \overline{b_{u,v,k}} = 1$. Now, the block smoothness values $f_{m,n,0}$ and $f_{m,n,1}$ can be computed respectively according to (1), leading the following derivation:

$$
\begin{aligned}
f_{m,n,1} &= \sum_{u=2}^{s-1}\sum_{v=2}^{s-1}\left|p_{u,v,1} - (p_{u-1,v,1} + p_{u,v-1,1} + p_{u+1,v,1} + p_{u,v+1,1})/4\right| \\
&= \sum_{u=2}^{s-1}\sum_{v=2}^{s-1}\left|\sum_{k=1}^{3}\left[\overline{b_{u,v,k}} - \left(\overline{b_{u-1,v,k}} + \overline{b_{u,v-1,k}} + \overline{b_{u+1,v,k}} + \overline{b_{u,v+1,k}}\right)/4\right]\times 2^{(k-1)}\right| \\
&= \sum_{u=2}^{s-1}\sum_{v=2}^{s-1}\left|\sum_{k=1}^{3}[(1-b_{u,v,k}) - ((1-b_{u-1,v,k}) + (1-b_{u,v-1,k}) + (1-b_{u+1,v,k}) + (1- \right. \\
&\qquad \left. b_{u,v+1,k}))/4]\times 2^{(k-1)}\right| \\
&= \sum_{u=2}^{s-1}\sum_{v=2}^{s-1}\left|\sum_{k=1}^{3}\left[b_{u,v,k} - \left(b_{u-1,v,k} + b_{u,v-1,k} + b_{u+1,v,k} + b_{u,v+1,k}\right)/4\right]\times 2^{(k-1)}\right| \\
&= \sum_{u=2}^{s-1}\sum_{v=2}^{s-1}\left|p_{u,v,0} - (p_{u-1,v,0} + p_{u,v-1,0} + p_{u+1,v,0} + p_{u,v+1,0})/4\right| = f_{m,n,0}. \qquad (3)
\end{aligned}
$$

The above-mentioned problem is also found in [10], i.e., the block smoothness values $f_{m,n,0}'$ and $f_{m,n,1}'$ computed according to (2) are equal when the five MSBs of the pixels in the original block $B_{m,n}$ are *all identical.* The proof, also based on the equality $b_{u,v,k} + \overline{b_{u,v,k}} = 1$, is similar to (3) and so omitted.

Fig. 1(a) shows a flat image with 8×8 blocks where the pixel values of *each* block are reassigned artificially to be all the same. Fig. 1(b) shows the incorrectly-recovered blocks (marked in white) yielded by [9] with an error rate of 50.81%, and Fig. 1(c) shows those yielded by [10] with an error rate of 44.53%. In contrast, the proposed method (described in the next section) yields a result as shown in Fig. 1(d) with an error rate of 0%, showing the effectiveness of the proposed method. Instead of using

artificially-created images, Figs. 1(e) through 1(h) shows the recovery results using an original X-ray image, Fig. 1(e), as input, where Fig. 1(f) shows the result yielded by [9] with an error rate of 16.53%, Fig. 1(g) shows that yielded by [10] with an error rate of 14.99%, and Fig. 1(h) shows that yielded by the proposed method with an error rate of 0% again.

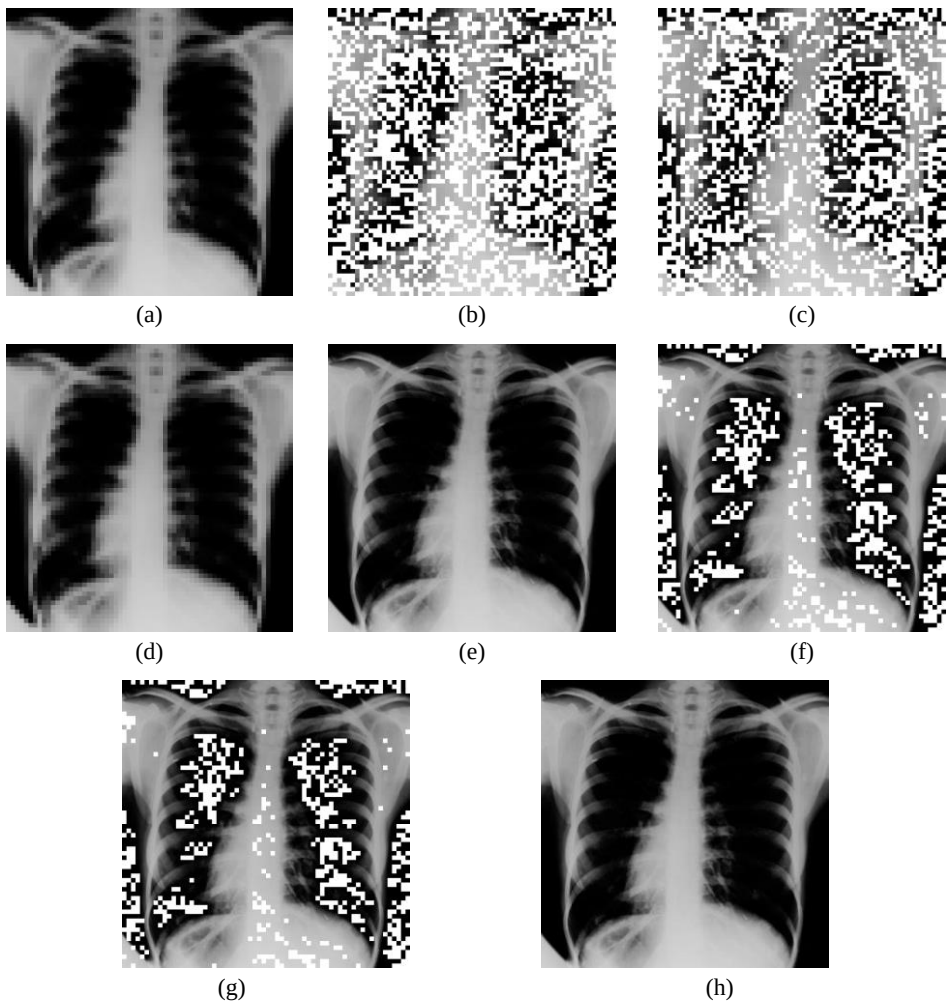


Fig. 1. Recovery results showing problems of [9] and [10] with block size 8×8, incorrectly-recovered blocks marked as white, and error rate denoted by *r*. (a) Input flat X-ray image. (b) Result with r = 50.81% yielded by [9]. (c) Result with r = 44.53% yielded by [10]. (d) Result with r = 0% yielded by proposed method. (e) Input original image of (a). (f) Result with r = 16.53% yielded by [9]. (g) Result with r = 14.99% yielded by [10]. (h) Result with r = 0% yielded by proposed method.

3. PROPOSED METHOD

3.1 Message Embedding

In the proposed method, the cover image I is encrypted using an encryption key K_e as done in [9], and each block $B_{m,n}$ of the resulting encrypted image I' is used to carry a message bit as well. However, unlike [9] and [10] which *flip* the three LSBs of each involved pixel in $B_{m,n}$, *four* LSBs are *randomized* to increase the probability for distinguishing original blocks from altered (interfered) ones using spatial correlations while the visual appearance of the decrypted image is still kept good enough. Next, four random bits $r_{i,j,k}'$ corresponding to the four LSBs $b_{i,j,k}'$ of each pixel $P_{i,j}'$ in $B_{m,n}$ are generated using the data-hiding key K_h; and if a bit to be embedded in $B_{m,n}$ is 0, then the four LSBs $b_{i,j,k}'$ of each pixel $P_{i,j}'$ in $B_{m,n}$ are replaced by $b_{i,j,k}'' = b_{i,j,k}' \oplus r_{i,j,k}'$, resulting in the new pixel value $p_{i,j}''$; else, $p_{i,j}''$ is taken to be the old value $p_{i,j}'$ of $P_{i,j}$. Let the resulting encrypted image be denoted as I''. The overall effect is: if the embedded bit in a block is 0, then the four LSBs of each block pixel are *encrypted twice* by the keys K_e and K_h; else, *once* by the key K_e only. So, the embedded bit in each block may be extracted by decrypting the block content with keys K_e and K_h and measuring the spatial correlations of the block to decide if the block has been encrypted *once* or *twice* as described next.

3.2 Message Extraction and Image Recovery

To extract the message embedded in the encrypted image I'', firstly the key K_e is used to *decrypt* image I'' to obtain another, denoted by I''', whose pixels' four MSBs are all the same as those of the pixels of the original cover image I. Next, the four corresponding random bits $r_{i,j,k}'$ are re-generated using the key K_h for the four LSBs $b_{i,j,k}'''$ of each pixel $P_{i,j}'''$ in each block $B_{m,n}'$ of size $s \times s$ in I''', and XOR operations are applied to $b_{i,j,k}'''$ and $r_{i,j,k}'$ to form another block $H_{m,n,0}$. Also, $B_{m,n}'$ itself is regarded as a *contrastive* block $H_{m,n,1}$. It can be figured out that one of $H_{m,n,0}$ and $H_{m,n,1}$ is the *original* cover block $B_{m,n}$; and the other is a *scrambled* version of $B_{m,n}$ because the four LSBs of this block's pixels have been encrypted for the second time using the key K_h. To decide which one is $B_{m,n}$, the smoothness measures of $H_{m,n,0}$ and $H_{m,n,1}$ can be utilized because: if the embedded bit is 0, then since the original *cover block* is encrypted *twice* by the keys K_e and K_h, the decrypted block $H_{m,n,0}$ using the same keys K_e and K_h will become the original block, which usually is smoother than the scrambled version $H_{m,n,1}$; and if the embedded bit is 1, then since the original cover block is encrypted *only once* by the key K_e, the decrypted block $H_{m,n,1}$ using the same key K_e will become the original block, which is usually smoother than the scramble version $H_{m,n,0}$ as well. Moreover, to compute the block smoothness, we adopt the measure used in [10] described by (2), but, unlike the side-match scheme used in [10] which utilizes only the recovered block to compute the smoothness, a new side-match scheme using both unrecovered and recovered blocks is proposed.

In more detail, for each block $B_{x,y}'$ of the four blocks adjacent to each block $B_{m,n}'$ in I''', if $B_{x,y}'$ is *unrecovered* yet, then the values of $f_{m,n,0}'$ and $f_{m,n,1}'$ computed according to (2) are augmented in the following way:

$$\text{set} \quad f_{m,n,0}' = f_{m,n,0}' + \min(\left|H_{m,n,0}' - H_{x,y,0}'\right|,\ \left|H_{m,n,0}' - H_{x,y,1}'\right|);$$

$$\text{set} \quad f_{m,n,1}' = f_{m,n,1}' + \min(\left|H_{m,n,1}' - H_{x,y,0}'\right|,\ \left|H_{m,n,1}' - H_{x,y,1}'\right|),$$

where $H_{x,y,q}'$ with $q = 0$ or 1 is generated from $B_{x,y}'$ by the same way as $H_{m,n,p}'$ with $p = 0$ or 1 is generated from $B_{m,n}'$ using the key K_h; and as illustrated in Fig. 2, $|H_{m,n,p}' - H_{x,y,q}'|$ with $p, q = 0, 1$ is defined by

$$\left|H_{m,n,p}' - H_{x,y,q}'\right| = \sum_{c=1}^{s} \left|b_{c,p} - b_{c,q}'\right|,$$

with $b_{c,p}$ denoting the value of a border pixel of block $H_{m,n,p}'$ adjacent to block $H_{x,y,q}'$; and $b_{c,q}'$ denoting the value of a border pixel of block $H_{x,y,q}'$ adjacent to block $H_{m,n,p}'$. Contrarily, if the adjacent block $B_{x,y}'$ of $B_{m,n}'$ are *recovered* as $H_{x,y,r}'$ already, then $f_{m,n,0}'$ and $f_{m,n,1}'$ are augmented in the following way:

$$\text{set} \quad f_{m,n,0}' = f_{m,n,0}' + \left|H_{m,n,0}' - H_{x,y,r}'\right|;$$

$$\text{set} \quad f_{m,n,1}' = f_{m,n,1}' + \left|H_{m,n,1}' - H_{x,y,r}'\right|.$$

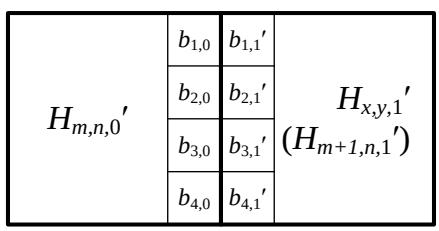


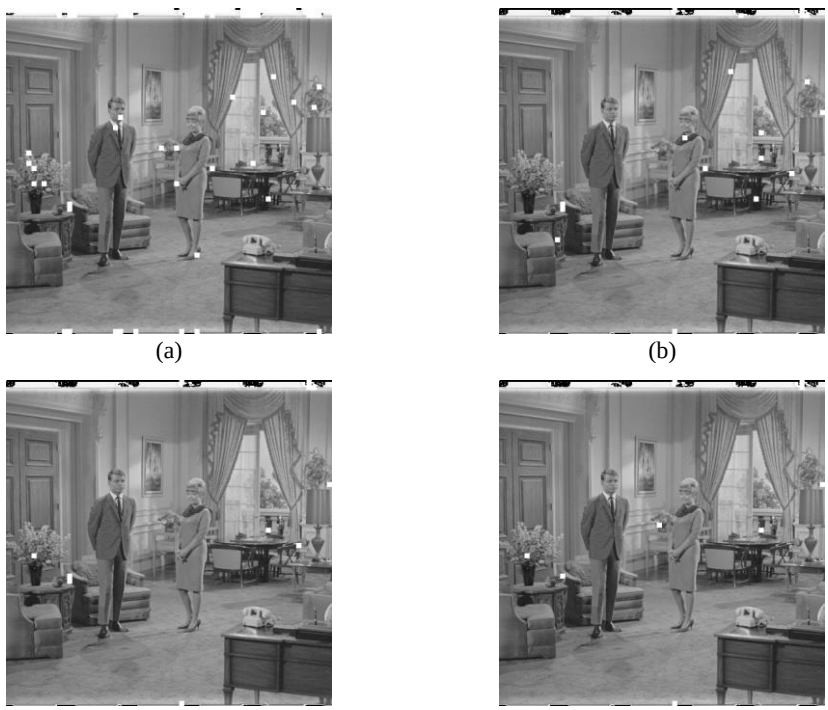
Fig. 2. Illustration of block contents for computing $|H_{m,n,0}' - H_{x,y,1}'|$ for 4×4 blocks, where currently-processed adjacent block of $B_{m,n}'$ is $B_{m+1,n}'$.

Since blocks adjacent to $B_{m,n}'$ are all used to compute the smoothness no matter whether they are recovered or not, the resulting bit-extraction error rate will be smaller than other schemes not doing so. Fig. 3 includes some results showing this effect for 8×8 blocks with a comparison with that yielded by [10].

4. EXPERIMENTAL RESULTS

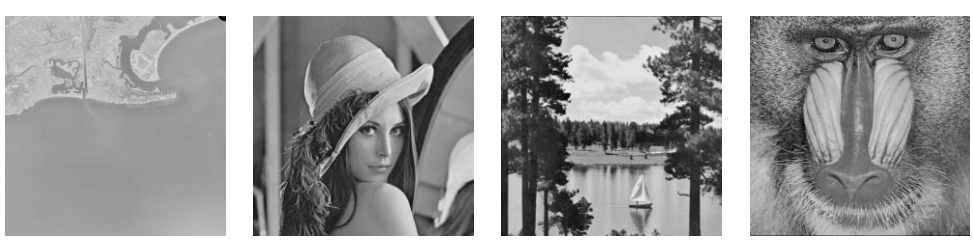
Four 512×512 test images, Figs. 4(a) through 4(d), were used in the experiments, and the results of the proposed method are compared with those yielded by [9] and [10], as illustrated in Fig. 5 which includes plots of the trends of bit-extraction error rates vesus different block sizes s×s. It is seen from Figs. 5(a) through 5(d) that the error rates yielded by the proposed method are much *smaller* than those yielded by [9] and [10]. For example, for the cover image Fig. 4(a) with block size 8×8, Fig. 5(a) shows that the bit-extraction error rates using [9] and [10] are 12.87% and 10.21%, respectively; and

that yielded by the proposed method is 0.07%. Moreover, Fig. 5(a) shows additionally that the error rate yielded by the proposed method is zero when *s* is larger than 12, but those yielded by both [9] and [10] are still larger than zero when *s* = 32.



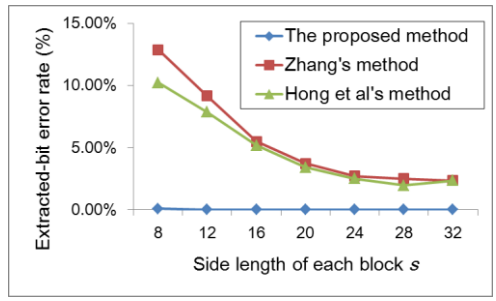
(a) (b) (c) (d)

Fig. 3. Recovery results showing effects of using both recovered and unrecovered blocks for measuring smoothness of 8×8 blocks, with incorrectly-recovered blocks marked as white, and error rate denoted by *r*. (a) Result with *r* = 2.66% yielded by [10]. (b) Result with *r* = 0.46% yielded by proposed method without using side-match. (c) Result with *r* = 0.27% yielded by proposed method using only recovered blocks in side-match scheme. (d) Result with *r* = 0.22% yielded by proposed method using both recovered and unrecovered blocks in side-match scheme.

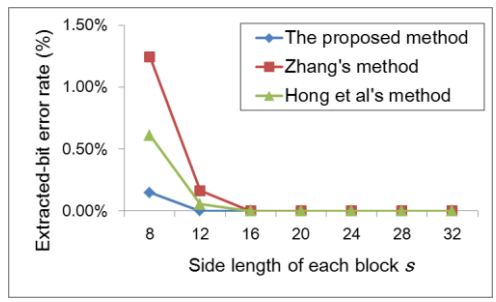


(a) (b) (c) (d)

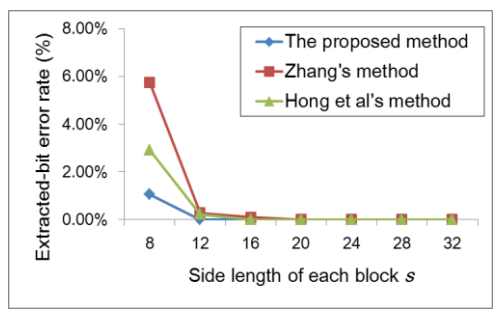
Fig. 4. Four test images of size 512×512.



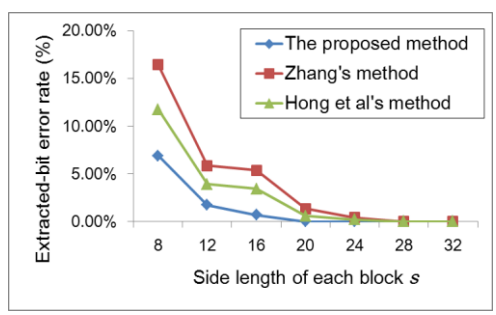
(a)



(b)



(c)



(d)

Fig. 5. Comparisons of bit-extraction error rates yielded by proposed method with those yielded by [9] and [10] versus different block sizes. (a) Error rates with cover image Fig. 4(a). (b) Error rates with cover image Fig. 4(b). (c) Error rates with cover image Fig. 4(c). (d) Error rates with cover image Fig. 4(d).

Also, experiments for comparisons of the effects of using *three or four* LSBs for data embedding have also been conducted. Results for 8×8 blocks are shown in Fig. 6, where the number of used LSBs is denoted by N_L. Specifically, the methods [9] and [10] do not perform better when $N_L = 4$ for image Fig. 4(a) as can be seen from Figs. 6(e) through 6(h). The same conclusions can be drawn for other images. Contrarily, Figs. 6(c) and 6(d) show that the proposed method performs better as N_L is enlarged from 3 to be 4.

Furthermore, the average distortion of the decrypted image with respect to the original image by using the proposed method can be computed, which is described as follows. Firstly, a decrypted pixel in the decrypted image has two possibilities: (1) correct decryption – the same as the original pixel; or (2) incorrect decryption – a scrambled version of the original pixel, where the possibility for each case is 1/2. If the decrypted pixel is of the first case, then the average squared difference between the decrypted gray value and the original one is zero; else, the average squared difference between the decrypted gray value and the original one is $\frac{1}{16}\sum_{i=0}^{15} i^2 = 77.5$, where i respresents the difference between the decrypted gray value and the original one. The value of the PSNR of the decrypted image with respect to the original image is approximately

$$\text{PSNR} = 10\log_{10}\frac{255^2}{(1/2)\times 0 + (1/2)\times 77.5} = 32.25 \text{ dB}.$$

Hence, the average PSNR value of the decrypted image with respect to the original image by using the proposed method is about 32.25 dB, which is not so good as those of [9] and [10] with the average PSNR value of 37.9 dB. However, the proposed method significantly reduces the bit-extraction error rate and solves the flat image problem without keeping the spatial similarity of the LSBs of the pixels in each block as mentioned previously.

5. CONCLUSIONS

A reversible data hiding method based on double image encryptions and refined spatial correlation comparison has been proposed, which does not have the weakness of two existing methods [9] and [10] in handling flat cover images. The weakness comes from the way of flipping the three LSBs of each pixel in part of each block in an encrypted image to embed a message bit. The proposed method improves this by encrypting the four LSBs of each pixel of every block instead of flipping three of them to embed a bit. Also, a refined side-match scheme utilizing the spatial correlations of both recovered and unrecovered blocks has been proposed to decrease the bit-extraction error rate, in contrast with Hong *et al.* [10] which utilizes only those of recovered blocks. Experimental results show the feasibility of the proposed method. Future studies may be directed to applying the proposed method for various information hiding purposes.

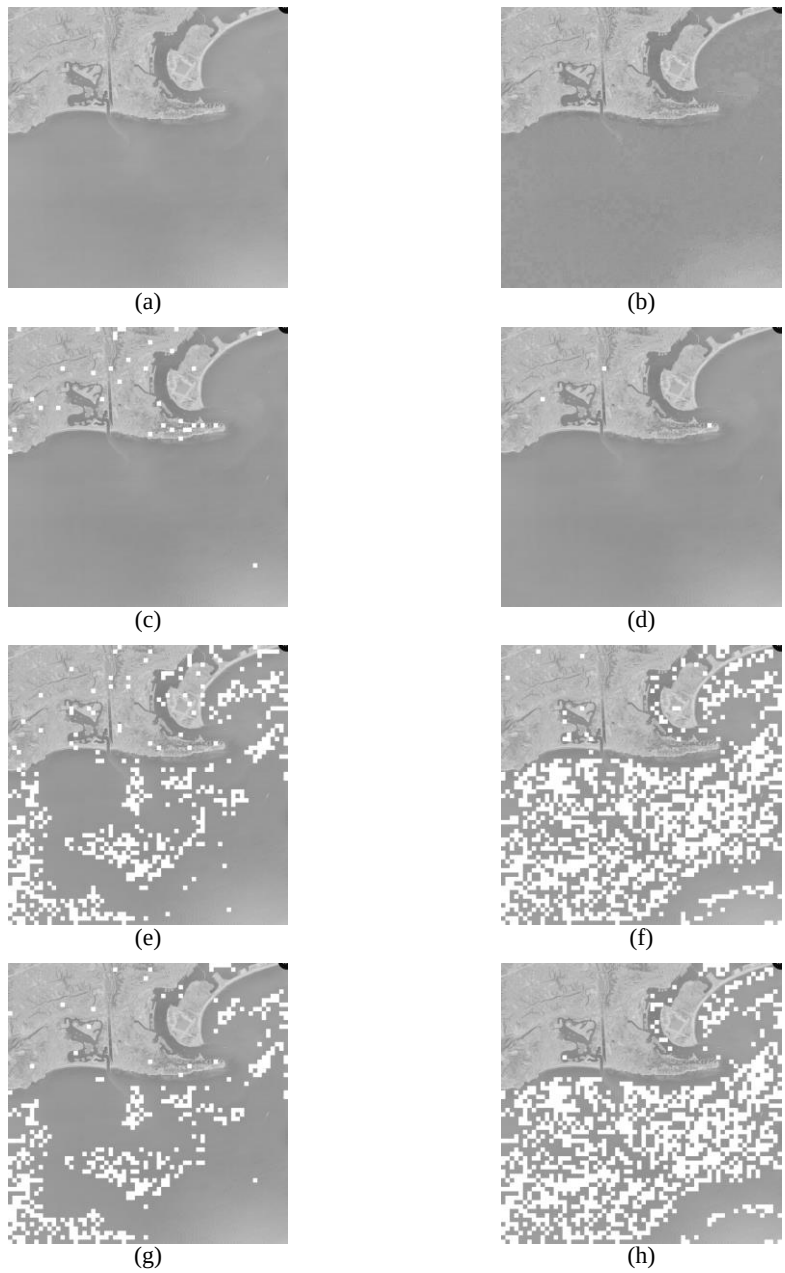
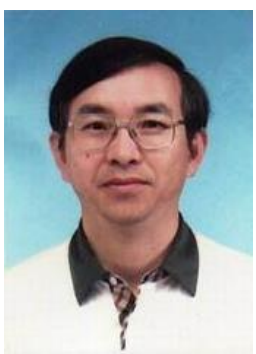


Fig. 6. Recovery results showing effects of using different numbers N_L of LSBs for 8×8 blocks with incorrectly-recovered blocks marked as white and error rate denoted by r. (a) Cover image. (b) Decrypted image with message embedded. (c) Result with r = 0.90% yielded by proposed method for N_L = 3. (d) Result with r = 0.07% yielded by proposed method for N_L = 4. (e) Result with r = 12.87% yielded by [9] for N_L = 3. (f) Result with r = 29.27% yielded by [9] for N_L = 4. (g) Result with r = 10.21% yielded by [10] for N_L = 3. (h) Result with r = 27.76% yielded by [10] for N_L = 4.

REFERENCES

1. M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, "Lossless generalized-LSB data embedding," *IEEE Trans. Image Process.*, vol. 14, no. 2, pp. 253–266, 2005.
2. J. Tian, "Reversible data embedding using a difference expansion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 8, pp. 890–896, 2003.
3. Y. Hu, H.-K. Lee, K. Chen, and J. Li, "Difference expansion based reversible data hiding using two embedding directions," *IEEE Trans. Multimedia*, vol. 10, no. 8, pp. 1500–1512, 2008.
4. Z. Ni, Y. Q. Shi, N. Ansari and W. Su, "Reversible Data Hiding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 3, pp. 354–362, 2006.
5. C. W. Lee and W. H. Tsai, "A lossless large-volume data hiding method based on histogram shifting using an optimal hierarchical block division scheme," *J. of Inform. Sci. & Eng.*, vol. 27, no. 4, pp. 1265–1282, 2011.
6. D. Kundur and K. Karthik, "Video fingerprinting and encryption principles for digital rights management," *Proc. IEEE*, vol. 92, no. 6, pp. 918–932, 2004.
7. S. Lian, Z. Liu, Z. Ren, and H. Wang, "Commutative encryption and watermarking in video compression," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 6, pp. 774–778, 2007.
8. M. Cancellaro, F. Battisti, M. Carli, G. Boato, F. G. B. Natale, and A. Neri, "A commutative digital image watermarking and encryption method in the tree structured Haar transform domain," *Signal Process.: Image Commun.*, vol. 26, no. 1, pp. 1–12, 2011.
9. X. Zhang, "Reversible data hiding in encrypted images," *IEEE Signal Process. Lett.*, vol. 18, no. 4, pp. 255–258, 2011.
10. W. Hong, T. S. Chen, and H. Y. Wu, "An improved reversible data hiding in encrypted images using side match," *IEEE Signal Process. Lett.*, vol. 19, no. 4, pp. 199–202, 2012.
11 W. Hong, T. S. Chen, J. Chen, Y. H. Kao, H. Y. Wu, and M. C. Wu, "Reversible data embedment for encrypted cartoon images using unbalanced bit flipping," *Lecture Notes in Computer Science*, vol. 7929, pp. 208–214, 2013.

Ya-Lin Lee (李雅琳) received the B. S. degree in computer science and information engineering from National Chiao Tung University, Hsinchu, Taiwan in 2009. She is currently pursuing the Ph.D. degree at the College of Computer Science, National Chiao Tung University, Hsinchu, Taiwan. Her current research interests include information hiding, image processing, pattern recognition, and artificial intelligence.



Wen-Hsiang Tsai (蔡文祥) received the B.S. degree in EE from National Taiwan University, Taiwan, in 1973, the M.S. degree in EE from Brown University, USA in 1977, and the Ph.D. degree in EE from Purdue University, USA in 1979. Since 1979, he has been with National Chiao Tung University (NCTU), Taiwan, where he is now a Chair Professor of Computer Science. At NCTU, he has served as the Head of the Dept. of Computer Science, the Dean of General Affairs, the Dean of Academic Affairs, and a Vice President. From 1999 to 2000, he was the Chair of the Chinese Image Processing and Pattern Recognition Society of Taiwan, and from 2004 to 2008, the Chair of the Computer Society of the IEEE Taipei Section in Taiwan. From 2004 to 2007, he was the President of Asia University, Taiwan.

Dr. Tsai has been an Editor or the Editor-in-Chief of several international journals, including *Pattern Recognition*, the *International Journal of Pattern Recognition and Artificial Intelligence*, and the *Journal of Information Science and Engineering*. He has published 158 journal papers and 247 conference papers and received many awards, including the Annual Paper Award from the Pattern Recognition Society of the USA; the Academic Award of the Ministry of Education, Taiwan; the Outstanding Research Award of the National Science Council, Taiwan; the ISI Citation Classic Award from Thomson Scientific, and more than 40 other academic paper awards from various academic societies. His current research interests include computer vision, information security, video surveillance, and autonomous vehicle applications. He is a Life Member of the Chinese Pattern Recognition and Image Processing Society, Taiwan and a Senior Member of the IEEE.