symmetry

MDPI

Article

Reversible Dual-Image-Based Hiding Scheme Using Block Folding Technique

Tzu-Chuen Lu [1,*] and Hui-Shih Leng [2]

[1] Department of Information Management, Chaoyang University of Technology, Taichung 41349, Taiwan
[2] Department of Mathematics, National Changhua University of Education, Changhua 50058, Taiwan; lenghs@cc.ncue.edu.tw
* Correspondence: tclu@cyut.edu.tw; Tel.: +886-4-23323000 (ext. 4558)

Received: 19 September 2017; Accepted: 9 October 2017; Published: 12 October 2017

Abstract: The concept of a dual-image based scheme in information sharing consists of concealing secret messages in two cover images; only someone who has both stego-images can extract the secret messages. In 2015, Lu et al. proposed a center-folding strategy where each secret symbol is folded into the reduced digit to reduce the distortion of the stego-image. Then, in 2016, Lu et al. used a frequency-based encoding strategy to reduce the distortion of the frequency of occurrence of the maximum absolute value. Because the folding strategy can obviously reduce the value, the proposed scheme includes the folding operation twice to further decrease the reduced digit. We use a frequency-based encoding strategy to encode a secret message and then use the block folding technique by performing the center-folding operation twice to embed secret messages. An indicator is needed to identify the sequence number of the folding operation. The proposed scheme collects several indicators to produce a combined code and hides the code in a pixel to reduce the size of the indicators. The experimental results show that the proposed method can achieve higher image quality under the same embedding rate or higher payload, which is better than other methods.

Keywords: dual stego-images; information hiding; center-folding strategy; block folding

1. Introduction

Information hiding is a technique that conceals secret messages in digital media. The sender embeds secret messages in a cover image to generate a stego-image and then sends it to the receiver. The receiver can extract the secret messages from the stego-image (Figure 1).

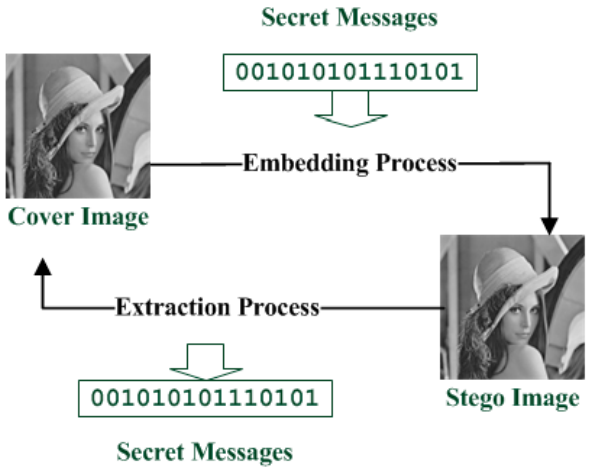


Figure 1. Diagram of the information hiding process.

In general, information hiding schemes can be classified into two categories, i.e., reversible data hiding and irreversible data hiding, as shown in Figure 2. The most commonly used irreversible data hiding methods include the least significant bit (LSB) substitution method, the pixel-value differencing (PVD) method, and the exploiting modification direction method (EMD) [1–14]. The LSB method is a well-known irreversible data hiding technique because of its high payload and low distortion. It directly replaces the bits of the cover pixel with secret bits for embedding. Mielikainen (2006) proposes a modification to the LSB method called LSB matching [1]. In his method, the secret bits are embedded by using the binary function and four embedding rules.

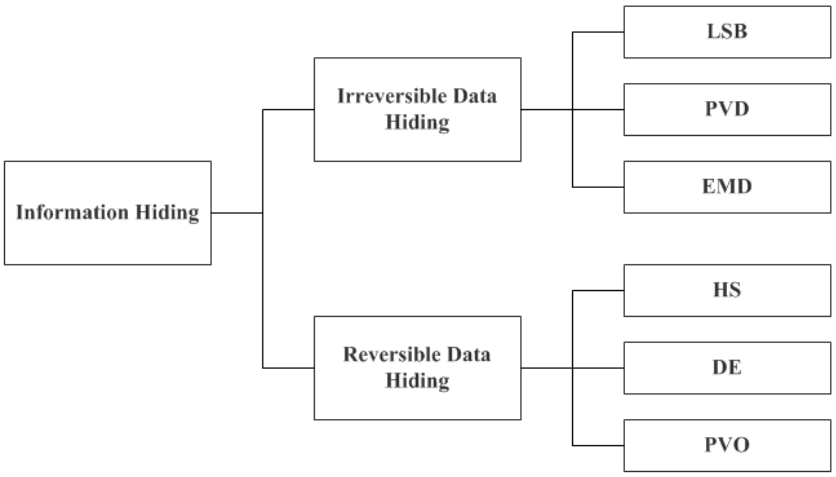


Figure 2. Information hiding categories. LSB: least significant bit substitution method, PVD: the pixel-value differencing method, EMD: the exploiting modification direction method, DE: difference expansion, HS: histogram shifting, PVO: pixel-value ordering method.

Wu and Tsai (2003) proposed the PVD method [2]. In their method, the difference of two pixels in the cover image is calculated to determine the number of bits to be embedded in these two pixels and a pre-defined range table. The technique can embed a large number of secret bits into the cover image with high imperceptibility as it makes use of the characteristic of human vision sensitivity. Chang and Tseng (2004) utilized the difference of the predicted value and the target pixels to estimate the degree of smoothness or contrast of pixels to determine the number of bits to be embedded in the target pixels [3].

Zhang and Wang (2006) proposed the EMD method [4]. The EMD method provides a good image quality for the stego-image with a peak signal-to-noise ratio (PSNR) of more than 52 dB, since, at most, only one pixel in each pixel group needs to be increased or decreased by one. Kieu and Chang (2011) improved the EMD method by exploiting eight modification directions to embed several secret bits into a cover pair at a time [5].

Compared with irreversible data hiding schemes, reversible data hiding schemes can recover the cover image without any distortion from the stego-image after the secret messages have been extracted. The most commonly used reversible data hiding methods include the difference expansion (DE) method, histogram shifting (HS) method, and pixel-value ordering (PVO) method. Tian (2003) proposed the DE method [6] that embeds a secret bit into the LSB of the expanded difference of each pixel pair of the cover image. The scheme provides a high payload; however, the distortion caused by the DE is significant. Alattar (2004) improved Tian's scheme with double the respective differences between four neighboring pixels and achieved more secret bits with the expanded difference [7].

Ni et al. (2006) proposed the HS method in 2006 [8]. Their method is to firstly generate the histogram by the pixel intensity value and shift the bins between the zero and peak point to create empty bins for data embedding. The advantage of the HS method is its low distortion; however, its drawback is a low payload, because the embedding capacity is determined by the number of points

in the peak point of the bin. Li et al. and Gui et al. proposed an adaptive embedding technique that divides pixels into different types to enhance the embedding capacity of a prediction error [9,10].

Li et al. (2013) proposed the PVO method [11]. In their method, for each block, the pixels are reordered into a pixel vector, then the smallest pixel is predicted by the second smallest pixel, and the largest pixel is predicted by the second largest pixel. It uses prediction errors 1 and -1 to embed data, whereas prediction error 0 is unchanged. Peng et al. (2014) improved the PVO method to use larger blocks for embedding and take better advantage of image redundancy to yield a higher PSNR [12]. Qu and Kim (2015) modified the PVO method so that each pixel is predicted using its sorted context pixels to achieve a better embedding capacity in smooth image regions [13].

Wang et al. (2015) used a dynamic blocking strategy to divide the cover image adaptively into various-sized blocks. Thus, the flat image areas are preferentially divided into smaller blocks to retain a high embedding capacity and the rough areas are divided into larger blocks to avoid decreasing PSNR value [14].

The dual-image-based hiding scheme is a new technology in the information hiding field. The concept of dual-image, based on information sharing, consists of concealing secret messages in two of the same cover images; only someone who has both stego-images can extract the secret messages. A diagram of the dual-image based hiding scheme is shown in Figure 3. There are many advantages to using dual-image in data hiding, such as a high payload, reversibility, and strong robustness.

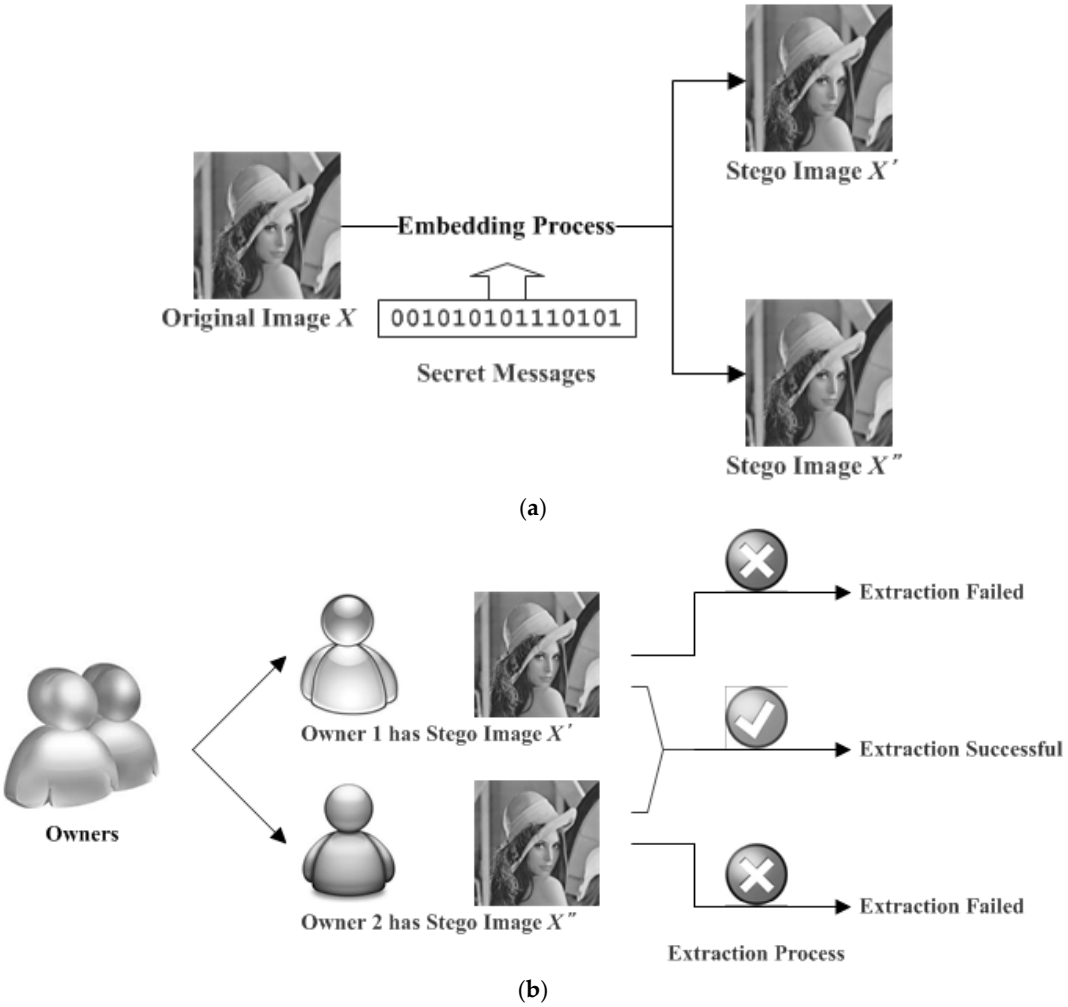


Figure 3. Dual-image based hiding scheme. (**a**) Embed secret messages into two of the same cover images; (**b**) Extract secret messages from the two stego-images.

In the dual-image-based hiding scheme, image quality and payload are affected by embedding rules [15–21]. Chang et al. (2007) combined the EMD method with the dual-image technique to achieve a high payload and reduce distortion [15]. Lee et al. (2009) used combinations of pixel orientation locations with dual-image to enhance embedding capacity and preserve good visual quality [16]. Lee and Huang (2013) converted secret messages into quinary-based secret symbols and combined every two secret symbols as a set embedded in the dual-image [17]. Chang et al. (2013) converted secret messages into decimal-based secret symbols, then embedded secret symbols in a right diagonal line [18]. Qin et al. (2014) embedded secret messages in the first image using the EMD method and in the second image using other rules that were dependent on the first image [19]. Lu et al. (2015) used the LSB matching method and modified the non-reversible stego-pixels based on a rule table to restore the cover image [20]. They proposed the center-folding strategy to reduce the value of the secret symbols. Then, they embedded secret symbols in two stego-images through an averaging method [21]. Lu et al. (2016) proposed a frequency-based encoding method to reduce the distortion derived by the maximum secret digit [22].

In [21], Lu et al. propose a center-folding strategy in which each secret symbol is folded into the reduced digit before the embedding procedure to reduce the distortion of the stego-image. The folding strategy is simple and effective, to the extent that the image quality of the stego-image is very good. Because the folding strategy can obviously reduce the value, the proposed scheme performs the folding operation twice to further decrease the reduced digit.

Furthermore, in [22], Lu et al. use a frequency-based encoding strategy to reduce the distortion of the frequency of occurrence of the maximum absolute value. The re-encoded technique also can be used to reduce the number of the secret digit and narrow down the distance between the stego-pixel and the original pixel.

Therefore, the proposed scheme first uses a frequency-based encoding strategy to encode the secret message and then uses the block folding technique by including the center-folding operation twice to embed secret messages. In addition, several steganalysis techniques are used to prove the strong robustness of the proposed scheme, including histogram steganalysis, Regular and singular groups (RS) steganalysis [23], primary sets, Chi square, sample pairs RS analysis, and fusion detection [24].

The rest of this paper is organized as follows. Section 2 describes related works. Section 3 introduces the proposed scheme. Section 4 summarizes the experiment results. The conclusions are presented in Section 5.

2. Related Works

In this section, we briefly introduce Lee et al.'s scheme, the center-folding strategy, and the frequency-based encoding strategy.

2.1. Lee et al.'s Methods

Lee et al. [16,17] proposed a direction-based dual-image method. In their scheme, two pixels P_1 and P_2 are used to conceal four secret bits and a direction map is used to represent the embedding rules. Figure 4 shows the direction map. Suppose that two pixels are $P_1 = 15$, $P_2 = 20$ and the secret bits are "00". The direction map indicates that $P'_1 = P_1 + 1 = 15 + 1 = 16$ and $P'_2 = 20$ for embedding "00" in the first stego-image. If the next two secret bits are "10", then the stego-pixels are $P'_1 = P_1 = 15$ and $P'_2 = P_2 + 1 = 20 + 1 = 21$ for the second stego-image.

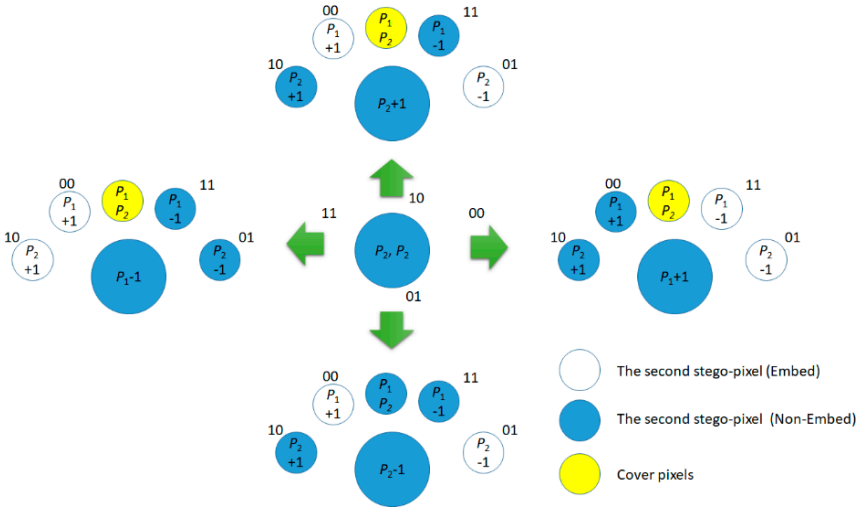


Figure 4. Direction-based embedding rules.

However, in some cases, the method cannot conceal secret data in the second stego-image. Therefore, only the first image is modified to embed the first two secret bits.

In 2013, Lee and Huang improved the above method by increasing the number of direction rules from four directions to five directions and redesigned the embedding rule to increase the embedding payload. The direction map is shown in Figure 5.

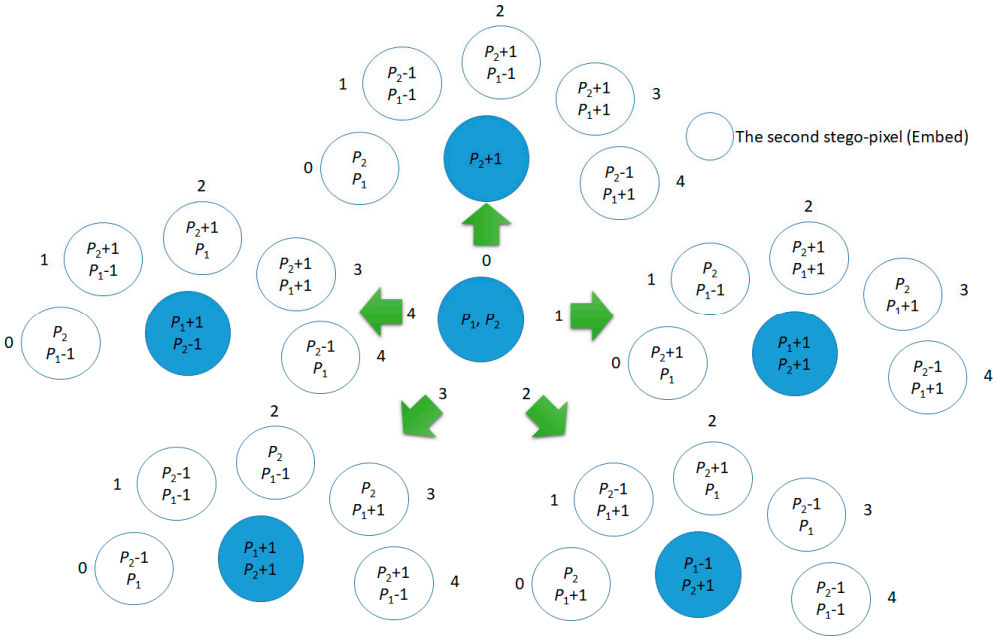


Figure 5. Data embedding method using five directions.

2.2. Center-Folding Strategy

In [21], Lu et al. proposed a center-folding strategy where each secret symbol is folded into the reduced digit before the embedding procedure to reduce the distortion of the stego-image. For example, K bits of secret data were taken as a set and converted into the secret symbol d.

The center-folding strategy changed the range of secret symbols from $R = \{0, 1, 2, \ldots, 2^{K-1}\}$ to $\overline{R} = \{-2^{K-1}, -2^{K-1}+1, \ldots, -1, 0, 1, \ldots, 2^{K-1}-2, 2^{K-1}-1\}$. The formula is as follows:

$$\overline{d} = d - 2^{K-1}, \tag{1}$$

where $\overline{d}$ is a folded secret symbol and 2^{K-1} is an intermediate value.

After the folding, the value range of the secret symbol changed to $[-2^{K-1}, 2^{K-1}-1]$. Figure 6 shows an example where K is set to be 3. The maximum value of the value range is 7. Assume that a pixel value is 138. If the secret symbol d = 7 is added directly to the pixel to get the stego-pixel 138 + 7 = 145, then the image distortion inflicted is $(145-138)^2 = 7^2 = 49$. However, in Lu et al.'s scheme, the symbol is folded as $\overline{d}$ = $7 - 2^2 = 3$. The folded symbol $\overline{d} = 3$ instead of the original symbol 7 is added with the pixel 138 to get the stego-pixel 138 + 3 = 141. The image distortion caused by Lu et al.'s method is $(141-138)^2 = 3^2 = 9$. The image distortion is reduced from 49 to 9.

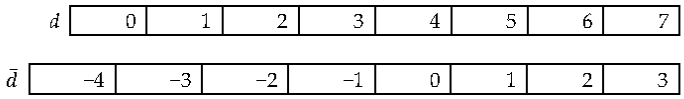


Figure 6. Example value range of the central-folding strategy with K = 3.

It is obvious that the center-folding strategy is simple and effective; it can reduce the distortion of the stego-image after the embedding procedure. Therefore, in the proposed scheme, we perform the folding operation twice to further decrease the reduced digit.

2.3. Frequency-Based Encoding Strategy

In [22], Lu et al. used a frequency-based encoding strategy to reduce the distortion of the frequency of occurrence of the maximum absolute value. For example, suppose that the secret bit stream is "110 011 111 011 101 010 110 111". First, the scheme converts each three bits (K = 3) as a group to a decimal digit stream as "7 3 7 3 5 2 6 7". Next, it uses the center-folding strategy to reduce the digit stream as "3 −1 3 −1 1 −2 2 3". Then, the frequency table can be made as Table 1, which records the rank of each reduced digit and map number of its new index in descending order by occurrence frequency.

Table 1. Example of the frequency-based encoding table (K = 3).

Decimal Digit	Reduced Digit	Occurrence Frequency	Order	Indices
0	−4	0	5	−3
1	−3	0	6	3
2	−2	1	2	1
3	−1	2	1	−1
4	0	0	7	4
5	1	1	3	−2
6	2	1	4	2
7	3	3	0	0

Compared with embedding the re-encoded secret digit and the original secret digit, the frequency-based encoding strategy can be used to reduce the number of the secret digit and narrow down the distance between the original pixel and the stego-pixel.

3. Proposed Method

In Lu's scheme, each secret symbol is reduced to the reduced digit using the center-folding strategy. A diagram of Lu's center-folding strategy is shown in Figure 7. In the figure, the center value 5 is subtracted from each decimal message d to generate the reduced digit $\widetilde{d}$. The value range of

Figure 7 can be seen as a band. The maximum value of the band is 7 in Figure 7. After the folding, the band is separated into two sub-bands. The maximum absolute value of the two sub-bands is 4.

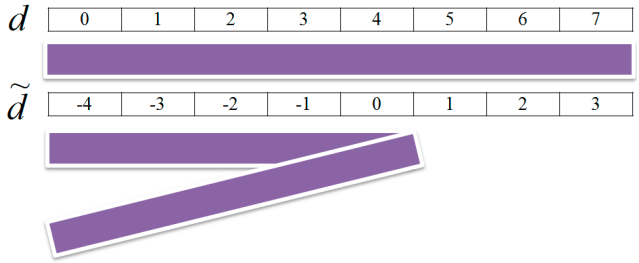


Figure 7. Diagram of Lu's center-folding strategy.

Because the folding strategy can obviously reduce the value, the proposed scheme includes the folding operation twice to further decrease the reduced digit. The concept of the proposed scheme is shown in Figure 8. In the figure, the value range of d is first divided into two sub-bands *SB1* and *SB2*. Each sub-band performs a one-time center-folding strategy. For example, the value range of sub-band *SB1* is shown in Figure 8a. The center value of *SB1* is 2. The center value 2 is subtracted from the values in *SB1* to generate the reduced digit $\hat{d}$. We can see that the maximum absolute value of the two sub-bands is 2, which is smaller than the value 4 in Figure 8.

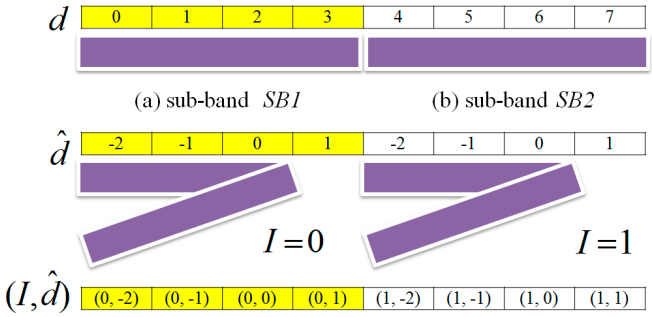


Figure 8. Diagram of the proposed block folding technique.

However, the sub-bands *SB1* and *SB2* have the same reduced values. We cannot distinguish the original value d from $\hat{d}$. Hence, an indicator is needed to identify which sub-band the reduced value $\hat{d}$ is located in. If the reduced value $\hat{d}$ is located in the sub-band *SB1*, then the indicator I is set to be 0, where $I = 0$. In contrast, if the reduced value $\hat{d}$ is located in the sub-band *SB2*, then the indicator I is set to be 1, where $I = 1$. The original value d is mapped to a code pair $(I, \hat{d})$ to reduce the image distortion. For example, the decimal value $d = 7$ is mapped to the code pair $(I, \hat{d}) = (1, 1)$.

However, the indicator is extra information that also needs to be concealed in the cover image and will decrease the hiding capacity. To solve this problem, the proposed scheme collects several indicators to produce a combined code and hides the code in a pixel. In the proposed scheme, the cover image is divided into several blocks. Each block has B pixels in it. The last pixel in the block is used to embed the combined code. The other pixels are used to conceal the reduced digit $\hat{d}$.

Furthermore, the proposed scheme considers the occurrence frequency of the decimal digit d to reassign proper code to the reduced digit $\hat{d}$ that can significantly shrink the image distortion.

3.1. Embedding Procedure

In the proposed scheme, a cover image is divided into several blocks. Each block has B pixels in it. Let $BK = \{BK_1, BK_2, \ldots, BK_B\}$ be the block. A secret image is formed as a binary string. The string is separated into several substrings the size of K. Each substring is transformed into a decimal digit d.

The scheme computes the occurrence frequency of each decimal digit d to generate a histogram and sort the histogram in descending order. The digit d with the maximum frequency is encoded with the smallest distortion code pair $(I', \hat{d}')$.

Each block BK can be used to hide $(B-1)$ code pairs. The scheme conceals the reduced digit in the pixel BK_i by using an averaging method to generate the stego-pixels BK'_i and BK''_i, where $1 \leq i \leq B-1$. The indicators are collected to form a combined code, and the code is concealed into the last pixel of the block to generate BK'_B and BK''_B.

Figure 9 shows the information hiding diagram of the proposed method. More details of the procedures are given below.

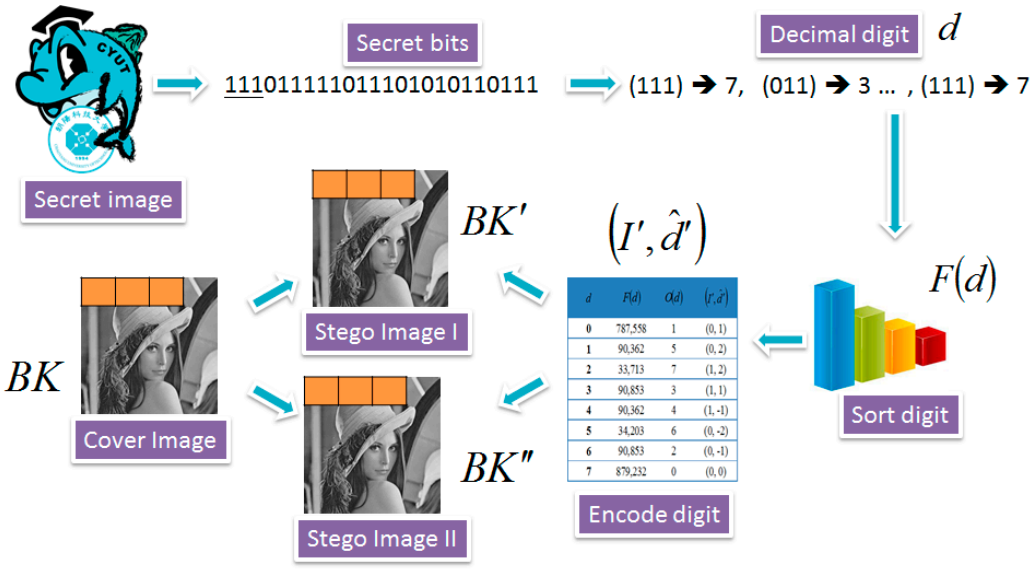


Figure 9. Information hiding diagram of the proposed scheme.

Step 1: Transform a set of K secret bits into a decimal digit. Let d be the decimal digit that is computed by

$$d = \sum_{j=1}^{K} 2^{j-1} s^j, \tag{2}$$

where $d \in [0, 2^K - 1]$ and s^j denotes the jth secret bit in the set.

Step 2: Compute the occurrence frequency of each decimal digit. Let $F(d)$ be the occurrence frequency of each decimal digit. These frequencies are sorted in descending order, and the index of sorted results is denoted as $O(d)$.

Step 3: Compute the reduced code $\hat{d}'$ for each decimal digit. In Figure 6, the decimal digit d has been folded twice to generate the reduced code $\hat{d}$ and the indicator I. Table 2 shows the reduced codes and indicators with $K = 3$. The decimal digits are reduced effectively from $[0, 2^K - 1]$ to $[-2^{K-2}, 2^{K-2} - 1]$.

Table 2. Example code pairs of Figure 6 with $K = 3$.

d	0	1	2	3	4	5	6	7
I	0	0	0	0	1	1	1	1
$\hat{d}$	−2	−1	0	1	−2	−1	0	1
$(I, \hat{d})$	(0, −2)	(0, −1)	(0, 0)	(0, 1)	(1, −2)	(1, −1)	(1, 0)	(1, 1)

However, the reduced digits in Table 2 do not show the feature of the secret image. In Table 2, the absolutely reduced code $\left|\hat{d}\right| = 2$ is the maximum reduced code, which causes maximum distortion when directly concealed into the pixel. If the maximum reduced code's occurrence frequency is high, then directly embedding it in the image will decrease the visual quality. Hence, the proposed scheme further re-encodes the reduced code and indicator code according to the order of $O(d)$. The re-encoded code $\hat{d}'$ is computed by

$$\hat{d}' = sign(O(d)) \times \left\lceil \frac{O(d)}{4} \right\rceil, \tag{3}$$

where

$$sign(v) = \begin{cases} -1, & \text{if } v \text{ is an odd number,} \\ 1, & \text{otherwise.} \end{cases} \tag{4}$$

Step 4: Compute the re-encoded indicator I' by

$$I' = \begin{cases} 0, & \text{if } O(d) = 0, \\ 1, & \text{if } \bmod((O(d)+1),4) = 0 \text{ or } \bmod(O(d),4) = 0, \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

The above procedure re-encodes the reduced digits and indicators, which occurs most frequently as the minimum absolute value. The re-encoded code pairs of Table 2 are shown in Table 3.

Table 3. The re-encoded code pairs of Table 2.

$O(d)$	0	1	2	3	4	5	6	7
I'	0	0	0	1	1	0	0	1
$\hat{d}'$	0	−1	1	−1	1	−2	2	−2
$(I', \hat{d}')$	(0, 0)	(0, −1)	(0, 1)	(1, −1)	(1, 1)	(0, −2)	(0, 2)	(1, −2)

Step 5: Generate a mapping table for further embedding and recovery processes. Integrate the re-encoded code pairs $(I', \hat{d}')$ with the indices d to form a mapping table. The mapping relationship must be recorded for use in the recovery phases. Table 4 shows an example mapping table with K = 3.

Table 4. Example mapping table with K = 3.

d	$F(d)$	$O(d)$	I'	$\hat{d}'$	$(I', \hat{d}')$
0	787,558	1	0	1	(0, 1)
1	90,362	5	0	2	(0, 2)
2	33,713	7	1	2	(1, 2)
3	90,853	3	1	1	(1, 1)
4	90,362	4	1	−1	(1, −1)
5	34,203	6	0	−2	(0, −2)
6	90,853	2	0	−1	(0, −1)
7	879,232	0	0	0	(0, 0)

Step 6: Divide a cover image into server blocks. Let $BK = \{BK_1, BK_1, \ldots, BK_B\}$ be a block.

Step 7: Pick $(B-1)$ code pairs $\{(I'_1, \hat{d}'_1), (I'_2, \hat{d}'_2), \ldots, (I'_{B-1}, \hat{d}'_{B-1})\}$ to embed into the block BK.

(1) Conceal the re-encoded reduced digit $\hat{d}'_i$ in the pixel BK_i by using

$$BK'_i = BK_i + \left\lfloor \frac{\hat{d}'}{2} \right\rfloor, \tag{6}$$

and

$$BK''_i = BK_i - \left\lceil \frac{\hat{d}'_i}{2} \right\rceil, \tag{7}$$

where $1 \leq i \leq B-1$, BK'_i denotes the ith pixel of the first stego-block and BK''_i denotes the ith pixel of the second stego-block. Figure 10 shows that the proposed method can control the distortion within $\left\lceil \frac{\hat{d}'}{2} \right\rceil$.

(2) Collect the indicators to form a combined code $(ID)_{10} = (I'_1, I'_2, \ldots, I'_{B-1})_2$. The combined code is computed by

$$ID = \sum_{i=1}^{B-1} 2^{i-1} I'_i \tag{8}$$

(3) Embed the combined code ID in the pixel BK_B

$$BK'_B = BK_B + \left\lfloor \frac{ID}{2} \right\rfloor, \text{and} \tag{9}$$

$$BK''_B = BK_B - \left\lceil \frac{ID}{2} \right\rceil \tag{10}$$

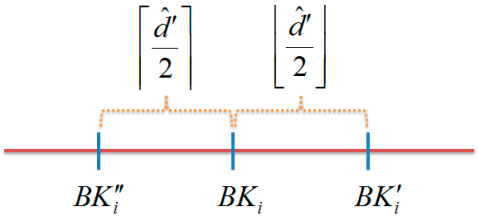


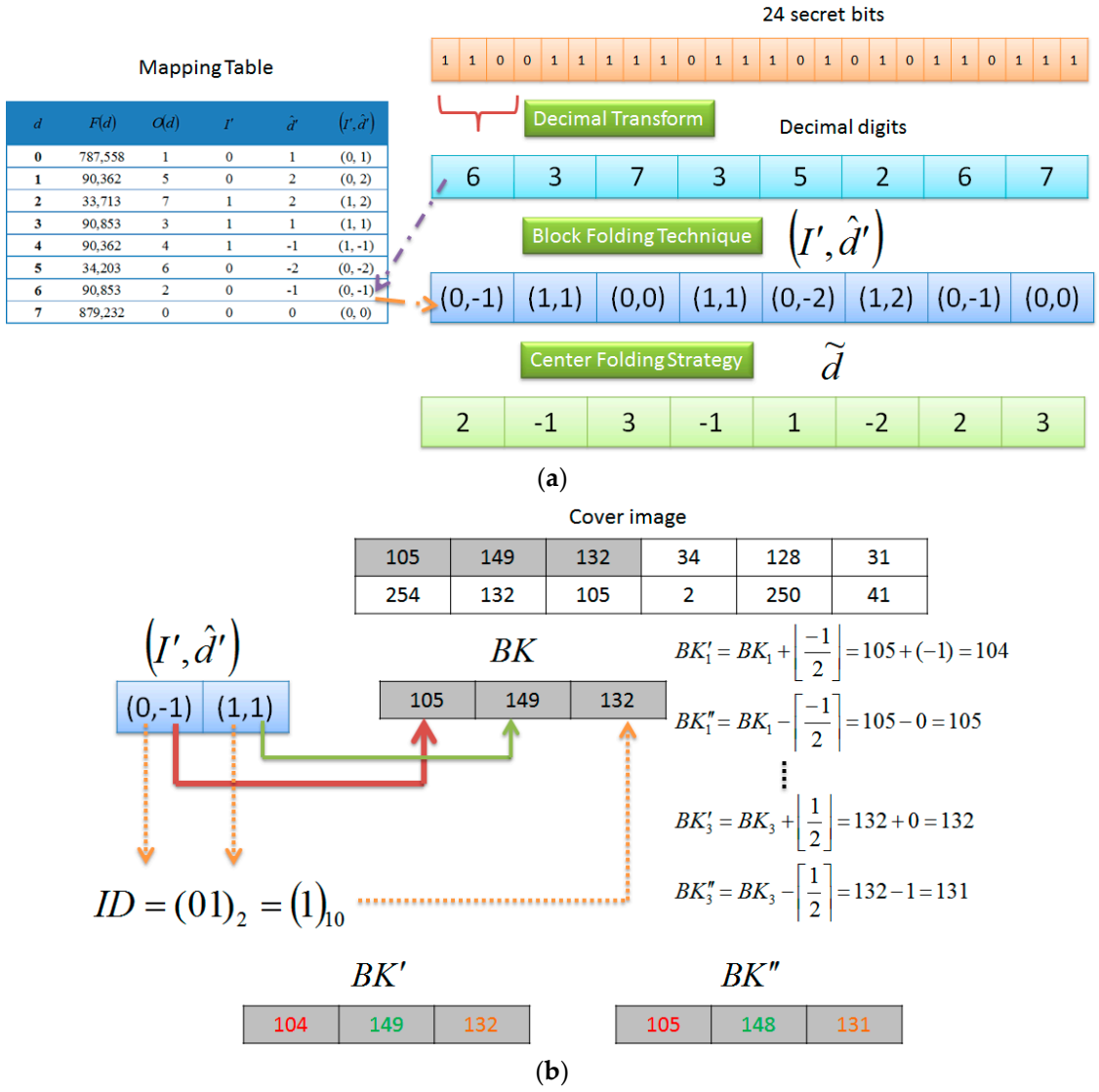
Figure 10. Diagram of the averaging embedding method.

Step 8: Repeat Step 6 until all blocks have been processed.

Figure 11 shows an example to illustrate the embedding process. Figure 11a is a secret string with 24 secret bits. Let K = 3, and the first three secret bits "110" are transformed into a decimal digit $d = \sum_{i=1}^{3} 2^{i-1} s^i = 2^{1-1} \times 1 + 2^{2-1} \times 1 + 2^{3-1} \times 0 = 6$. The scheme maps the decimal digit d to the mapping table as shown in Table 4 to obtain the re-encoded code pair $(I'_1, \hat{d}'_1) = (0, -1)$. The transformation of the other secret bits follows the same procedure described above.

Assume a cover image is divided into several blocks sized B = 3. Each block can be used to hide $(B-1) = (3-1) = 2$ code pairs. In Figure 11a, the code pairs are $\{(I'_1, \hat{d}'_1), (I'_2, \hat{d}'_2)\} = \{(0, -1), (1, 1)\}$. Figure 8b shows a cover image. The first block is $BK = \{105,\ 149, 132\ \}$. The reduced digits $\hat{d}'_1 = -1$ and $\hat{d}'_2 = 1$ are concealed into $BK_1 = 105$ and $BK_2 = 149$, respectively. The stego-pixels of the first cover pixel $BK_1 = 105$ are computed by $BK'_1 = BK_1 + \left\lfloor \frac{-1}{2} \right\rfloor = 105 + (-1) = 104$ and $BK''_1 = BK_1 - \left\lceil \frac{-1}{2} \right\rceil = 105 - 0 = 105$. The stego-pixels of the second cover pixel $BK_2 = 149$ are computed by $BK'_2 = 149 + \left\lfloor \frac{1}{2} \right\rfloor = 149$ and $BK''_2 = 149 - \left\lceil \frac{1}{2} \right\rceil = 148$. Then, the scheme collects the indicators to form a combined code $ID = (I'_1\ I'_2)_2 = (01)_2 = (1)_{10}$. The combined code is embedded in the pixel $BK_B = 132$ by $BK'_3 = BK_3 + \left\lfloor \frac{1}{2} \right\rfloor = 132 + 0 = 132$ and $BK''_3 = BK_3 - \left\lceil \frac{1}{2} \right\rceil = 132 - 1 = 131$. Two stego-blocks are shown in Figure 11b.

The embedding procedure is executed repeatedly until all code pairs are embedded. The final stego-images, along with the mapping table, are sent to the receiver for extraction and recovery.



d	$F(d)$	$O(d)$	I'	$\hat{d}'$	$(I', \hat{d}')$
0	787,558	1	0	1	(0, 1)
1	90,362	5	0	2	(0, 2)
2	33,713	7	1	2	(1, 2)
3	90,853	3	1	1	(1, 1)
4	90,362	4	1	-1	(1, -1)
5	34,203	6	0	-2	(0, -2)
6	90,853	2	0	-1	(0, -1)
7	879,232	0	0	0	(0, 0)

Figure 11. Example of the proposed method. (**a**) Secret encoding; (**b**) Embedding example.

3.2. Overflow/Underflow Problem

In the embedding process, an underflow/overflow problem might occur as a result of the reduced code. For example, if the pixel is $BK_i = 254$ and $\left\lceil \frac{\hat{d}'}{2} \right\rceil = 3$, then the two together would cause the overflow problem. In the proposed scheme, the range of $\hat{d}'$ is $[-2^{K-2}, 2^{K-2} - 1]$. If a pixel is larger than $255 - (2^{K-2} - 1) = 256 - 2^{K-2}$, then the overflow problem might occur because of the addition of $2^{K-2} - 1$. If a pixel that is less than 2^{K-2}, then it might cause the underflow problem because of the addition of -2^{K-2}. For example, assume $K = 4$, and the range of $\hat{d}'$ is $[-4, 3]$. Suppose that $BK_i = 0$, $\hat{d}' = 3$, and $\left\lceil \frac{\hat{d}'}{2} \right\rceil = \left\lceil \frac{3}{2} \right\rceil = 2$. Then, the stego-pixel is $BK_i - \left\lceil \frac{\hat{d}'}{2} \right\rceil = -2$, and there is an underflow problem.

Hence, the embeddable pixel is defined as being in the range of 2^{K-2} and $256 - 2^{K-2}$. In the embedding process, the proposed scheme determines whether the pixel BK_i is in the range of or not. If the pixel is within the range, then it could be classified into a block. However, the pixel might cause an underflow/overflow problem and the pixel is non-embeddable. For the non-embeddable pixel, the stego-pixels are set to equal the value of the original pixels. For example, assume that $K = 4$ and

a pixel in the range of $[2^{4-2}, 256 - 2^{4-2}] = [4, 252]$ is embeddable. In Figure 12, the pixel 254 is out of the range, which means it is non-embeddable. The stego-pixels are set to equal its original pixel. The next pixels 132 and 105 are embeddable. However, the next pixel 2 is non-embeddable and cannot be classified into a block. So, the next pixel 250 is gathered with pixels 132 and 105 to form block $BK = \{132, 105, 250\}$, and the embedding process is performed.

Cover image

105	149	132	34	128	31
254	132	105	2	250	41

Figure 12. Example of a block with three embeddable pixels.

3.3. Extraction and Recovery

In this section, we describe the extraction and recovery processes by which re-encoded reduced digits and secret bits are extracted from the stego-image as well as the process of cover image recovery. There are $(B-1)$ code pairs concealed in an embeddable block. Hence, the receiver divides the stego-images into several blocks the size of B. The re-encoded reduced digits $\hat{d}'$ can be extracted by computing the differences between two stego-pixels, where $\hat{d}'_i = BK'_i - BK''_i$ and $1 \leq i \leq B-1$. The combined code is extracted from the last pixels of the stego-blocks by $ID = BK'_B - BK''_B$. Then, the receiver transforms the combined code ID into $(B-1)$ binary bits. Each bit represents an indicator I_i. One indicator I_i along with one re-encoded reduced digit $\hat{d}'_i$ forms the code pair $(I_i, \hat{d}'_i)$. After the code pair is obtained, it is then mapped to the mapping table to obtain the original decimal digit d. Each decimal digit d is transformed into K secret bits.

In the recovery process, the cover pixel BK_i can be recovered by the average between two stego-pixels, i.e., $BK_i = \left\lceil \frac{BK'_i + BK''_i}{2} \right\rceil$. Figure 13 shows the data extraction and recovery procedure of the proposed method.

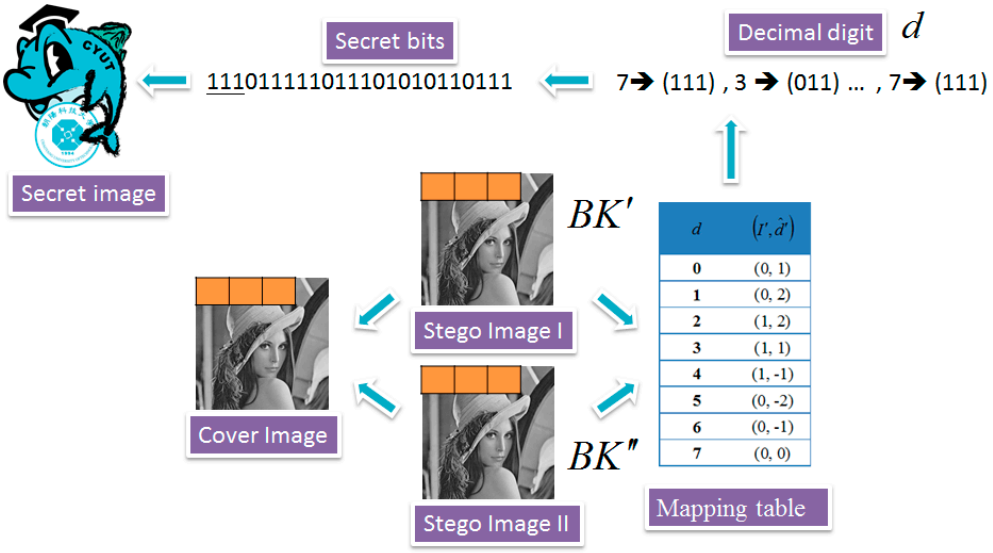


Figure 13. Diagram of the extraction and recovery processes.

Following the same example presented in Figure 11b. The stego-blocks are $BK' = \{104, 149, 132\}$ and $BK'' = \{105, 148, 131\}$. The re-encoded reduced codes are $\hat{d}'_1 = BK'_1 - BK''_1 = 104 - 105 = -1$ and $\hat{d}'_2 = BK'_2 - BK''_2 = 149 - 148 = 1$, and the combined code is $ID = BK'_3 - BK''_3 = 132 - 131 = 1$. The combined code is transformed into $(3-1) = 2$ binary bits $ID = (01)_2$, where

$I'_1 = 0$ and $I'_2 = 1$. The code pairs are $(I'_1, \hat{d}'_1) = (0, -1)$ and $(I'_2, \hat{d}'_2) = (1, 1)$. The original decimal digits $d = 6$ and $d = 3$ and can be derived by mapping them into the mapping table as shown in Figure 11a. Finally, the decimal digits $d = 6$ and $d = 3$ are transformed into three secret bits, i.e., "110" and "011". Furthermore, the original pixel can be recovered by the average between BK'_i and BK''_i, i.e., $BK_1 = \left\lceil \frac{BK'_1 + BK''_1}{2} \right\rceil = \left\lceil \frac{104+105}{2} \right\rceil = 105$, $BK_2 = \left\lceil \frac{149+148}{2} \right\rceil = 149$, and $BK_3 = \left\lceil \frac{132+131}{2} \right\rceil = 132$.

In the extraction process, the proposed scheme determines whether the two stego-pixels are both outside the range $[2^{K-2}, 256 - 2^{2-1}]$. If so, then the pixels are non-embeddable. If the two stego-pixels are not equal and more than one pixel is within the range, then the pixels are embeddable and data are concealed within. The secret information can be extracted, and the pixels can be restored following the extraction procedure.

3.4. Re-Encoding of the Combined Code

In the embedding process, the combined code is used to indicate the number of sub-bands of the re-encoded code. The value range of ID is $[0, 2^{(B-1)} - 1]$. For example, assume that the block size is set to be $B = 5$. The value range of ID is $[0, 2^{(5-1)} - 1] = [0, 15]$. The maximum value of the range is 15. The value is directly added to the last pixel of the block to generate the stego-pixel. However, if the maximum value occurrence frequency is high, then directly embedding it in the pixel will decrease the visual quality. Therefore, the proposed scheme selectively re-encodes the combined code according to its occurrence frequency and generates a mapping table to record the relationship between the combined code and the re-encoded combined code. The re-encoding procedure is the same as the re-encoding procedure of the reduced code.

4. Results

In the proposed scheme, the block size B and the hiding bit K are key factors that influence the hiding performance. To find out proper values of B and K, eight schemes with different values are implemented. They are

BlockFolding ($B = 3$, $K = 3$, $RE = 0$), BlockFolding ($B = 3$, $K = 3$, $RE = 1$),
BlockFolding ($B = 3$, $K = 4$, $RE = 0$), BlockFolding ($B = 3$, $K = 4$, $RE = 1$),
BlockFolding ($B = 5$, $K = 3$, $RE = 0$), BlockFolding ($B = 5$, $K = 3$, $RE = 1$),
BlockFolding ($B = 5$, $K = 4$, $RE = 0$), and BlockFolding ($B = 5$, $K = 4$, $RE = 1$).

The parameters B and K are the block size and the hiding bits, respectively. The parameter RE indicates whether the combined code is re-encoded or not. RE = 0 means the combined code is the original value without re-encoding. In contrast, RE = 1 means the combined code is re-encoded according to its occurrence frequency.

Five related methods are also implemented for a comparison with the proposed schemes. These are Lee's dual steganographic scheme (Lee2009), Lee's orientation combination scheme (Lee2013), Chang's magic matrix scheme (Chang2013), Lu et al.'s center-folding scheme (Lu2015), and Lu's scheme without the center-folding strategy (NonFolding).

Two measurements are used to measure the performance of the hiding methods, the embedding rate and the image quality. The embedding rate R is calculated by

$$R = \frac{C - E}{2 \times h \times w} \cdot (\text{bpp}), \tag{11}$$

where C is the total hiding capacity of the two stego-images, E is the size of the mapping table, and $h \times w$ is the size of the image. A high embedding rate means that the proposed scheme has great embedding ability.

Image quality is calculated by using the PSNR given by

$$\text{PSNR\_z} = 10 \times \log_{10} \frac{255^2}{\text{MSE\_z}} \cdot (\text{dB}), \tag{12}$$

where PSNR_z is the PSNR value of the zth stego-image, dB represents the decibels, and MSE_z is the mean squared error between the cover image and the stego-image, and is obtained by

$$\text{MSE} = \frac{1}{hw} \sum \sum (P' - P)^2, \tag{13}$$

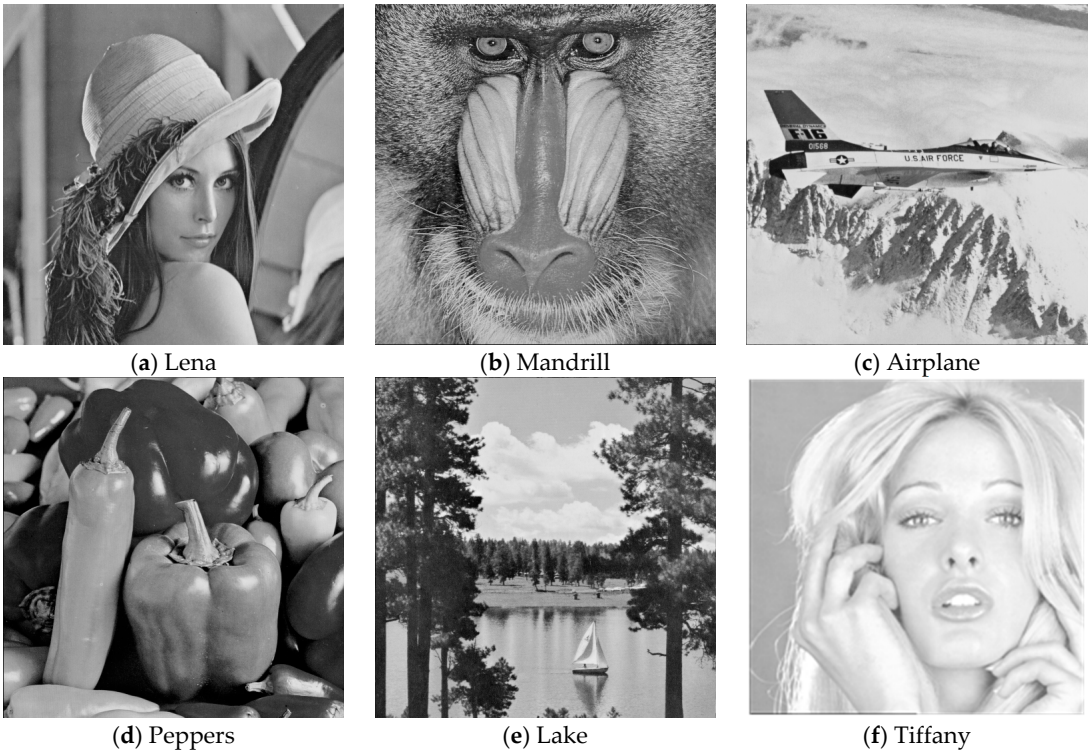
where P is the cover pixel and P' is the stego-image.

The PSNR values in the experimental results are the average values of all PSNR_z, which can be computed by

$$\text{PSNR\_avg} = \frac{1}{z} \sum \text{PSNR\_z} \cdot (\text{dB}) \tag{14}$$

In general, it is very difficult to determine the difference between the cover image and the stego-image by human eyes when the PSNR value is greater than 30 dB.

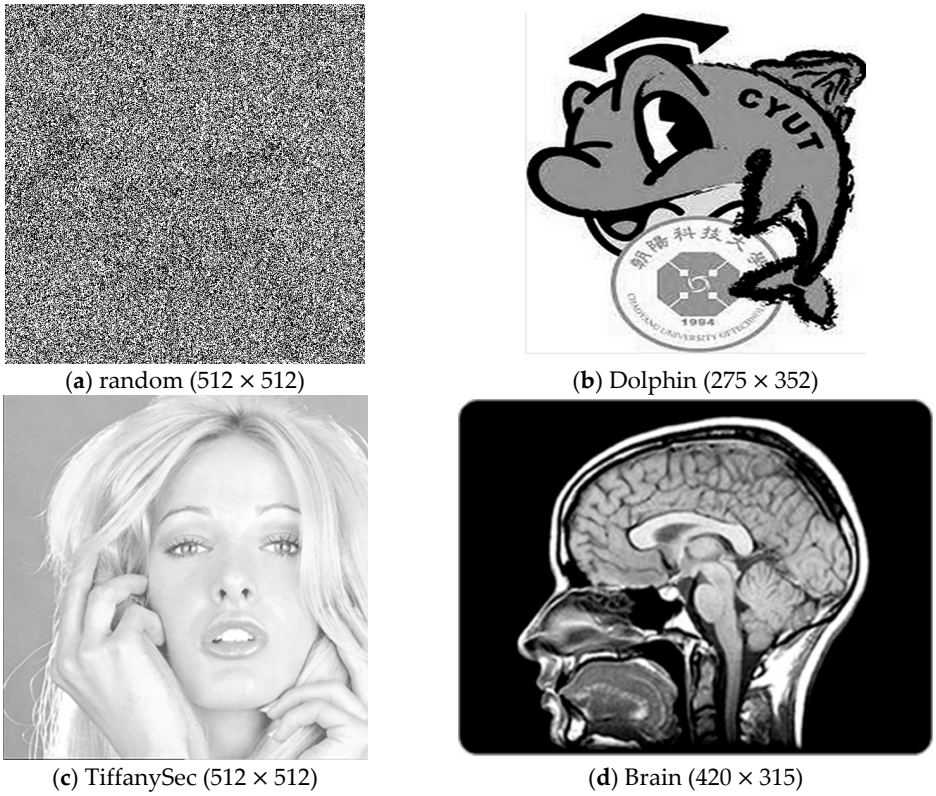
Six grayscale images were used to test the hiding performance. Figure 14 shows the images Lena, Mandrill, Airplane, Peppers, Lake, and Tiffany. The size of the image is 512 × 512. Four secret images, namely, random (512 × 512), Dolphin (375 × 352), Brain (420 × 315), and TiffanySec (512 × 512) are shown in Figure 15.



(**a**) Lena (**b**) Mandrill (**c**) Airplane

(**d**) Peppers (**e**) Lake (**f**) Tiffany

Figure 14. Six test images.



(**a**) random (512 × 512) (**b**) Dolphin (275 × 352)

(**c**) TiffanySec (512 × 512) (**d**) Brain (420 × 315)

Figure 15. Four secret images.

In the first experiment, we tested the performance of the proposed scheme with various B, K, and RE. Figure 16 shows the experimental results. "BlockFolding" is the proposed scheme. Under the same embedding rate, BlockFolding (B = 3, K = 3, RE = 0) can achieve the highest image quality. BlockFolding (B = 5, K = 4, RE = 1) can get the highest hiding payload. When B = 3 and K = 3, the image quality with RE = 0 is higher than that with RE = 1. However, when B = 5 and K = 4, the image quality with RE = 1 is higher than that with RE = 0. The reason is that if the block size is small, then the combined code is small enough that it does not need to be re-encoded. However, for a block with a large size, the combined code is usually large enough that it needs to be re-encoded. For example, if the value range of the combined code with B = 5 is $[0, 2^{(5-1)} - 1] = [0, 15]$, the maximum value 15 will cause great image distortion. The re-encoding process can effectively narrow down the distortion.

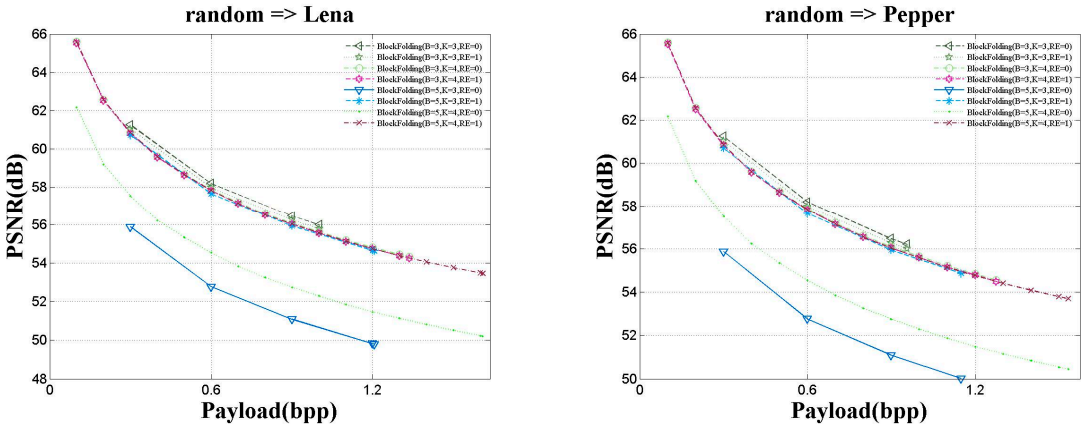


Figure 16. *Cont.*

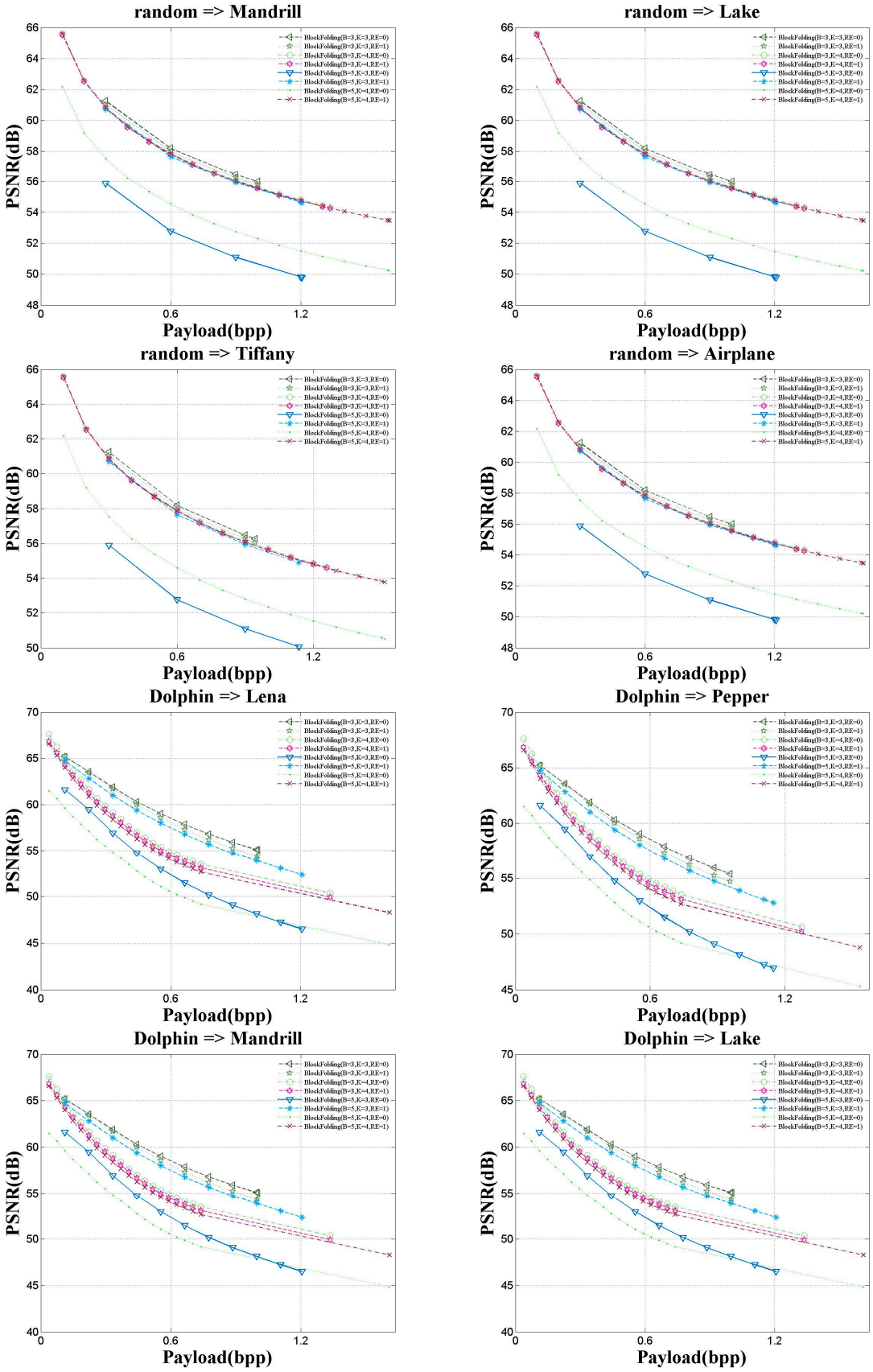


Figure 16. *Cont.*

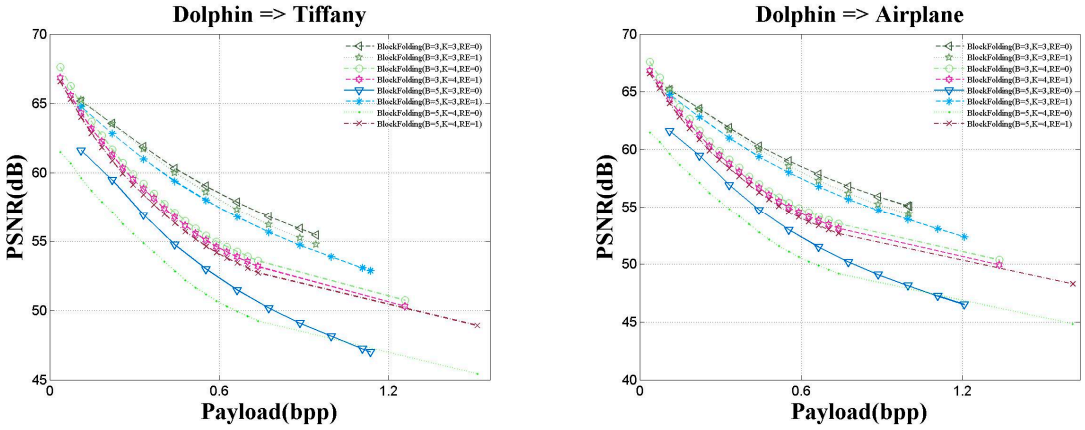


Figure 16. Results of the first experiment.

To compare for a comparison of the proposed scheme with other related methods, the value of B is set to 3 and 5, the value of K is set to 3 and 4, and RE is set to be 0 and 1. Tables 5–8 show the results of the comparison among the proposed scheme and the related works with the cover image Lena and the secret images random, TiffanySec, Dolphin, and Brain, respectively.

Table 5. Experiment results for the secret image "random".

Method	*B*	*K*	*RE*	PSNR_1	PSNR_2	PSNR_avg	Capacity	*R*	Time
BlockFolding	3	3	0	59.98	52.00	**55.99**	525,312	1.00	6.99
	3	3	1	59.04	52.52	55.78	525,312	1.00	6.99
	3	4	0	57.30	51.34	54.32	700,416	1.34	6.96
	3	4	1	56.95	51.57	54.26	700,416	1.34	6.96
	5	3	0	50.95	48.65	49.80	632,832	1.21	4.65
	5	3	1	58.13	51.15	54.64	632,832	1.21	4.65
	5	4	0	51.64	48.79	50.21	843,776	1.61	4.80
	5	4	1	56.28	50.65	53.46	843,776	1.61	4.80
Lu2015		3		45.02	42.94	43.98	786,432	1.50	2.41
		4		39.65	39.65	39.65	**1,048,576**	**2.00**	2.60
NonFolding		3		41.69	39.35	40.52	786,432	1.50	2.41
Lee2009		4		51.14	53.76	52.45	405,498	0.77	1.13
Lee2013		4		50.97	52.29	51.63	524,288	1.00	1.15
Chang2013		4		39.89	39.89	39.89	786,432	1.50	1.54

Table 6. Experiment results for the secret image Tiffany.

Method	*B*	*K*	*RE*	PSNR_1	PSNR_2	PSNR_avg	Capacity	*R*	Time
BlockFolding	3	3	0	55.37	49.79	**52.58**	525,312	1.00	7.02
	3	3	1	53.51	50.81	52.16	525,312	1.00	7.02
	3	4	0	50.62	47.79	49.21	700,416	1.34	7.47
	3	4	1	47.79	45.66	46.72	700,416	1.34	7.47
	5	3	0	46.31	44.57	45.44	632,832	1.21	4.89
	5	3	1	51.55	48.88	50.22	632,832	1.21	4.89
	5	4	0	43.84	42.69	43.26	843,776	1.61	4.99
	5	4	1	49.48	47.50	48.49	843,776	1.61	4.99
Lu2015		3		49.65	46.76	48.20	786,432	1.50	2.57
		4		42.17	42.17	42.17	**1,048,576**	**2.00**	2.74
NonFolding		3		41.93	39.80	40.87	786,432	1.50	2.57
Lee2009		4		51.14	54.65	**52.90**	378,980	0.72	1.27
Lee2013		4		50.69	49.39	50.04	524,288	1.00	1.28
Chang2013		4		39.89	39.89	39.89	786,432	1.50	1.79

Table 7. Experiment results for the secret image Dolphin.

Method	*B*	*K*	*RE*	PSNR_1	PSNR_2	PSNR_avg	Capacity	*R*	Time
BlockFolding	3	3	0	56.98	53.16	**55.07**	525,312	1.00	7.27
	3	3	1	55.53	53.16	54.34	525,312	1.00	7.27
	3	4	0	51.81	48.92	50.37	700,416	1.34	7.49
	3	4	1	51.08	48.78	49.93	700,416	1.34	7.49
	5	3	0	46.95	46.06	46.50	632,832	1.21	5.02
	5	3	1	53.56	51.17	52.36	632,832	1.21	5.02
	5	4	0	45.47	44.20	44.83	843,776	1.61	5.06
	5	4	1	49.19	47.38	48.28	843,776	1.61	5.06
Lu2015		3		47.21	43.66	45.44	786,432	1.50	2.65
		4		42.14	42.14	42.14	**1,048,576**	**2.00**	2.75
NonFolding		3		40.64	38.30	39.47	786,432	1.50	2.65
Lee2009		4		51.14	52.09	51.62	472,800	0.90	1.23
Lee2013		4		49.10	50.86	49.98	524,288	1.00	1.21
Chang2013		4		39.89	39.89	39.89	786,432	1.50	1.63

Table 8. Experiment results for the secret image Brain.

Method	*B*	*K*	*RE*	PSNR_1	PSNR_2	PSNR_avg	Capacity	*R*	Time
BlockFolding	3	3	0	61.76	**58.73**	60.24	525,312	1.00	6.85
	3	3	1	61.03	59.50	60.26	525,312	1.00	6.85
	3	4	0	50.57	48.31	49.44	700,416	1.34	7.01
	3	4	1	47.63	46.01	46.82	700,416	1.34	7.01
	5	3	0	51.58	50.64	51.11	632,832	1.21	4.64
	5	3	1	57.32	55.46	56.39	632,832	1.21	4.64
	5	4	0	43.89	42.89	43.39	843,776	1.61	4.88
	5	4	1	48.38	46.90	47.64	843,776	1.61	4.88
Lu2015		3		43.11	42.91	43.01	786,432	1.50	2.59
		4		36.95	36.95	36.95	**1,048,576**	**2.00**	2.99
NonFolding		3		48.30	46.33	47.31	786,432	1.50	2.59
Lee2009		4		51.14	**63.70**	57.42	276,696	0.53	1.26
Lee2013		4		60.84	59.35	60.09	524,288	1.00	1.17
Chang2013		4		39.87	39.87	39.87	786,432	1.50	1.55

In Table 5, with the secret image "random", the PSNR_avg of the proposed scheme with $B = 5$, $K = 4$, and $RE = 1$ for the cover image Lena is 53.46 dB, which is greater than the value of 43.98 dB for Lu2015. The image quality of the proposed scheme is higher than the value of 9.48 dB obtained by Lu's scheme. At the same time, the hiding rate of the proposed scheme is 1.61 bpp, which is higher than that of Lu2015's 0.11 bpp. Under the same hiding capacity, the image quality of the proposed scheme with $B = 3$, $K = 3$, and $RE = 0$ is 55.99 dB, which is higher than that of Lee2013's 51.63 dB. Although Lu2015 with $K = 4$ achieves the highest hiding rate, 2.0 bpp, the image quality of Lu2015 is decreased to 39.65 dB.

In Table 6, although Lee2009 has highest image quality 52.90 dB, the hiding rate of Lee2009 is only 0.72 dB. The results for different secret images have the same situation.

In Table 8, for the secret image Brain, the image quality obtained by Lee2009 is the highest. However, its hiding capacity is 0.53 bpp, which is the lowest among all results obtained by other methods. The proposed scheme with $B = 3$, $K = 3$, and $RE = 0$ has the same hiding payloads as those of Lee2013. The image quality of the proposed scheme is 58.73 dB, which is greater than that of the other methods, thereby indicating that the proposed scheme still exhibits better embedding performance compared with the others.

From the results of the experiment, we can see that the proposed scheme with a small block size $B = 3$ achieves a higher image quality. Because the combined code in the small block size is small, the code does not need to be re-encoded where $RE = 0$. In contrast, the proposed scheme with a large block

size $B = 5$ has a higher hiding capacity. The combined code in a large block size needs to be re-encoded, as the value of the code is usually very large. Hence, RE is set to be $RE = 1$ for $B = 5$.

The second experiment compared the proposed scheme with the other methods. Figure 17 shows the comparison among all five related methods and the proposed scheme in terms of embedding rate and PSNR value. Figure 17 shows that the image quality of BlockFolding ($B = 3$, $K = 3$, $RE = 0$) is higher than those of the other methods. Under the same embedding rate, the proposed method can achieve a greater PSNR value than the related methods. The hiding capacity of BlockFolding ($B = 5$, $K = 4$, $RE = 1$) is higher than those of the other methods.

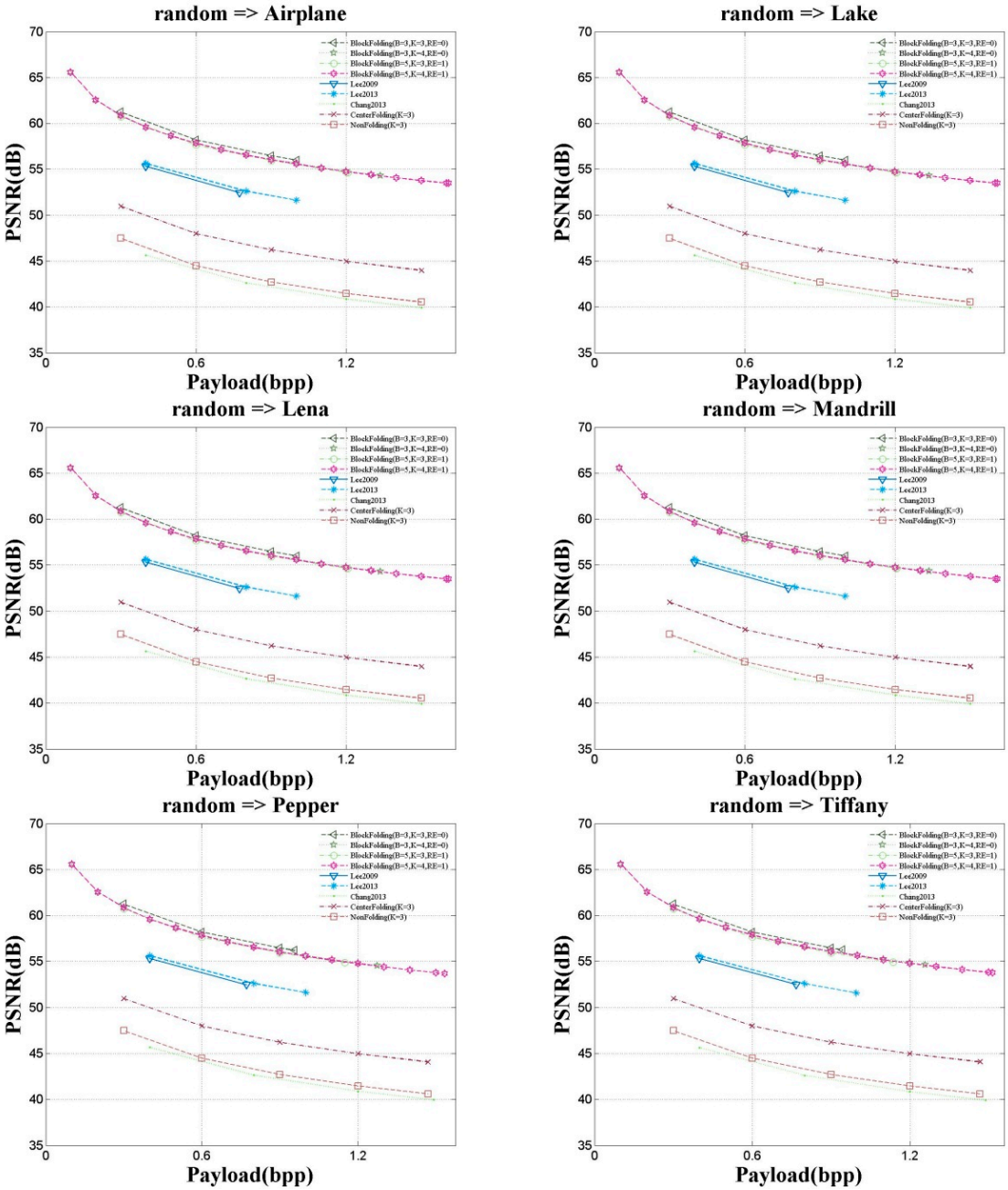


Figure 17. *Cont.*

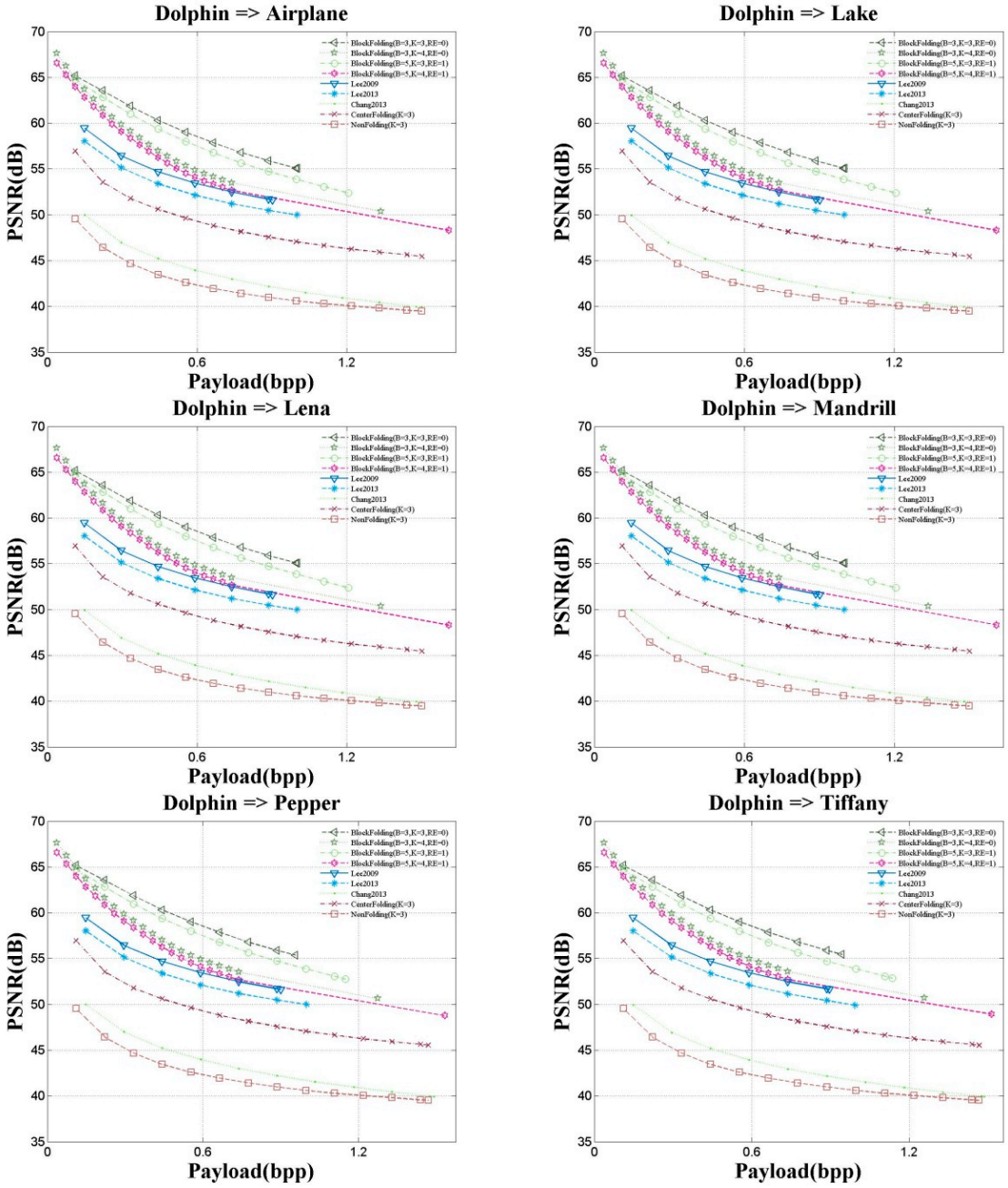


Figure 17. Comparison of the proposed and related methods.

The third experiment was aimed at proving the viability of the proposed scheme. Several steganalysis techniques such as histogram steganalysis, RS steganalysis, primary sets, Chi square, sample pairs, RS analysis, and fusion detection were used to test the performance of the proposed scheme. The histogram steganalysis method compares the shapes between the cover image and the stego-image to detect whether there is a concealed message.

Figure 18 shows histogram comparisons of Lena with different parameters. The curve starting with the symbol "*" is the histogram of the stego-image. We can see that the shape of the stego-image is almost the same as that of the cover image.

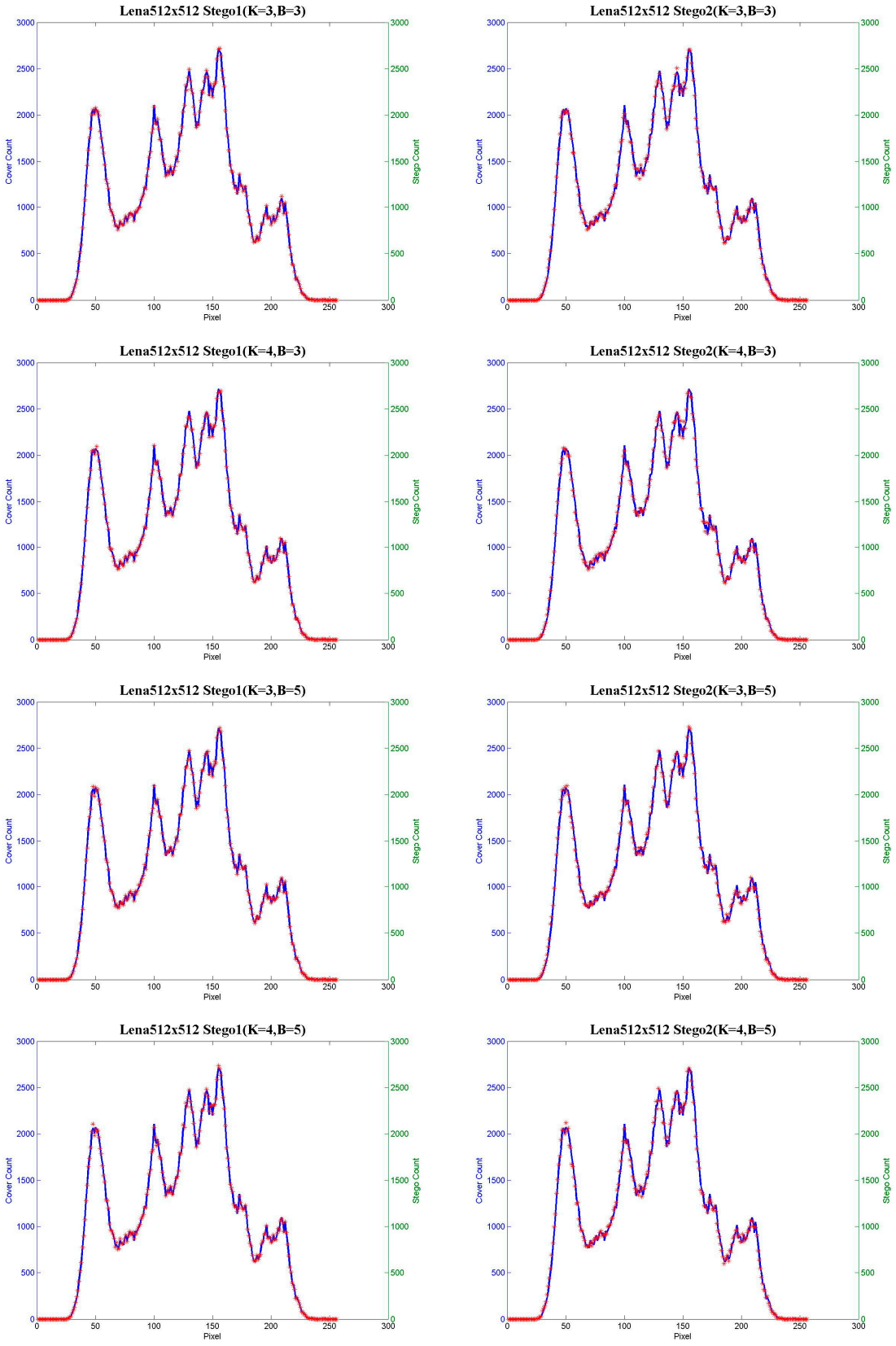


Figure 18. Histogram steganalysis of Lena with different parameters.

RS steganalysis is a kind of attack method proposed by Fridrich et al. In the method, pixels are classified into several groups. The method uses a discrimination function to quantify smoothness or

regularity and uses a flipping function to define three groups, regular (R), singular (S), and unusable (U). The percentages of each group with mask M = [1 0 0 1] and $-$M = [$-$1 0 0 $-$1] are represented as R_M_G, R_FM_G, S_M_G, S_FM_G, U_M_G, and U_FM_G, respectively. The hypotheses are R_M_G $\cong$ R_FM_G, S_M_G $\cong$ S_FM_G, and U_M_G $\cong$ U_FM_G. Figure 19 shows the RS steganalysis results for Lena with different parameters. In the figures, the curve of R_M_G is very similar to that of R_FM_G. The method judges there is no secret message hidden in the image. The curves of S_M_G, S_FM_G, U_M_G, and U_FM_G have the same situations. Hence, the proposed scheme cannot be detected by the RS steganalysis.

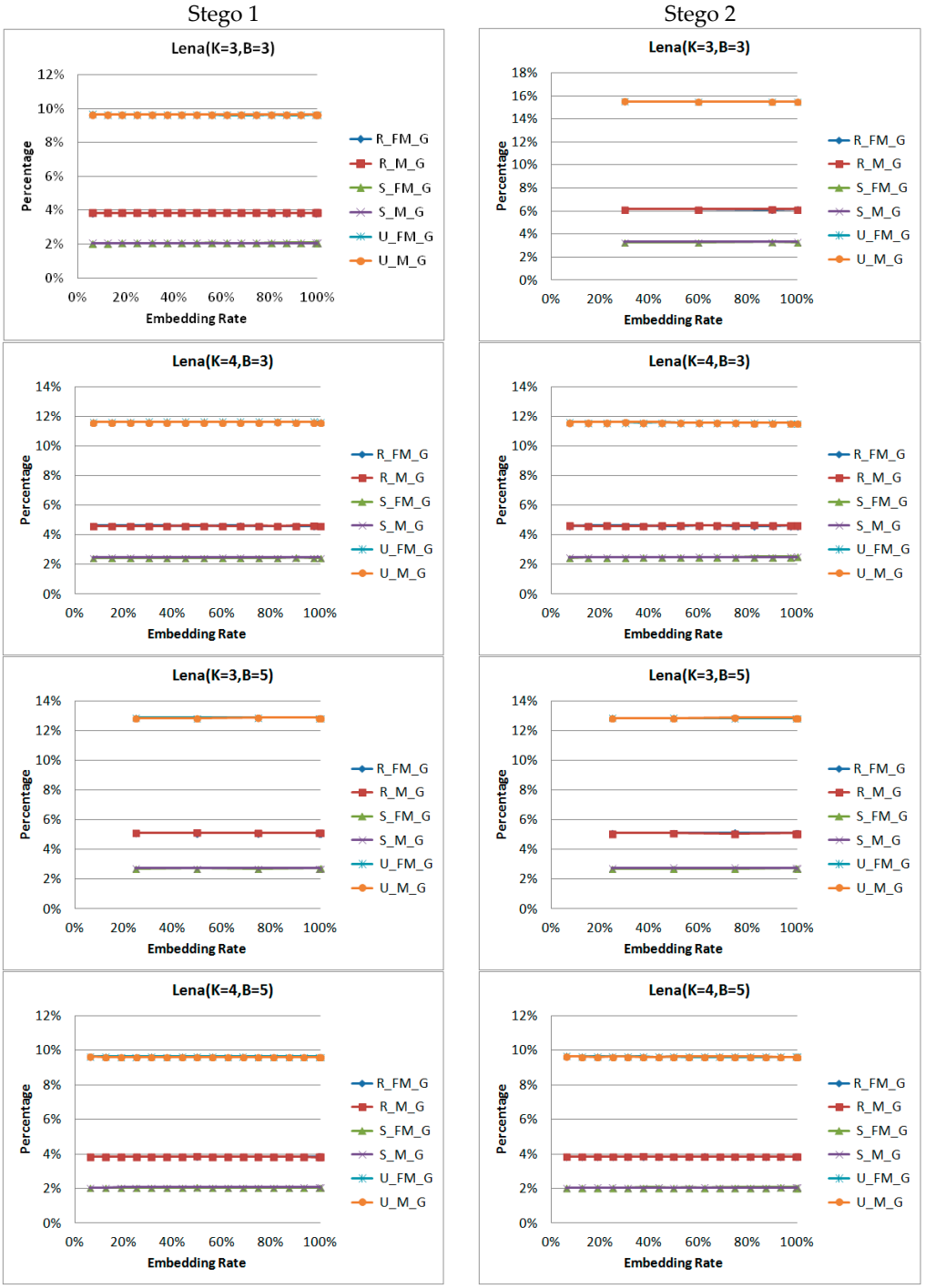


Figure 19. RS-diagram steganalysis of Lena with the secret image "random".

Other tests including primary sets, Chi square, sample pairs, RS analysis, and fusion detection were applied to examine the proposed scheme. Table 9 shows the experiment report. All numbers in the table are very small, which means the proposed scheme is robust against the steganalytic attacks.

Table 9. Steganalysis report for StegExpose.

Cover	*B*	*K*	Stego	Primary Sets	Chi Square	Sample Pairs	RS Analysis	Fusion (Mean)
Airplane	3	4	Stego 1	0.027675	0.004289	0.030703	0.024626	0.021823
			Stego 2	0.022305	0.005963	0.029186	0.017787	0.018810
	5	3	Stego 1	0.019342	0.004948	0.027352	0.020710	0.018088
			Stego 2	0.013602	0.005269	0.018627	0.017553	0.013763
	5	4	Stego 1	0.019624	0.005567	0.022817	0.016658	0.016166
			Stego 2	0.024443	0.004844	0.020266	0.015573	0.016282
Lake	3	4	Stego 1	0.050697	0.000000	0.004456	0.027477	0.020658
			Stego 2	0.031111	0.000029	0.025046	0.044235	0.025105
	5	3	Stego 1	0.041429	0.000000	0.021996	0.028785	0.023052
			Stego 2	0.042343	0.000003	0.031223	0.025238	0.024702
	5	4	Stego 1	0.043350	0.000000	0.015191	0.019025	0.019392
			Stego 2	0.031905	0.000003	0.027985	0.030218	0.022528
Lena	3	4	Stego 1	0.005693	0.001296	0.019193	0.005686	0.007967
			Stego 2	0.014598	0.001267	0.016012	0.006616	0.009623
	5	3	Stego 1	0.009356	0.001271	0.015101	0.010288	0.009004
			Stego 2	0.036424	0.001103	0.028364	0.013555	0.019861
	5	4	Stego 1	0.001998	0.001275	0.012401	0.007016	0.005673
			Stego 2	0.019515	0.001121	0.011369	0.008265	0.010068
Mandrill	3	4	Stego 1	0.054084	0.001449	0.092357	0.085073	0.058241
			Stego 2	0.046409	0.001583	0.083133	0.108457	0.059895
	5	3	Stego 1	0.071036	0.001079	0.125827	0.115084	0.078256
			Stego 2	0.077601	0.002146	0.107102	0.099520	0.071592
	5	4	Stego 1	0.075838	0.001145	0.123048	0.096351	0.074096
			Stego 2	0.053453	0.002419	0.090763	0.087010	0.058411
Pepper	3	4	Stego 1	0.080749	0.002956	0.055924	0.060841	0.050117
			Stego 2	0.058720	0.002917	0.051464	0.071926	0.046257
	5	3	Stego 1	0.104812	0.002355	0.076068	0.063953	0.061797
			Stego 2	0.024876	0.002141	0.027476	0.059740	0.028558
	5	4	Stego 1	0.093148	0.003151	0.065432	0.054871	0.054151
			Stego 2	0.083026	0.002894	0.057448	0.063565	0.051734
Tiffany	3	4	Stego 1	0.045827	0.005267	0.044256	0.046080	0.035357
			Stego 2	0.038765	0.005464	0.043139	0.040102	0.031868
	5	3	Stego 1	0.036811	0.005454	0.036731	0.040666	0.029915
			Stego 2	0.020802	0.005270	0.021787	0.033247	0.020276
	5	4	Stego 1	0.044159	0.005290	0.043003	0.042746	0.033799
			Stego 2	0.047008	0.005601	0.042099	0.033712	0.032105

5. Conclusions

The dual-image-based hiding scheme is a new technology in the data hiding field. The concept of dual-image, based on information sharing, consists of concealing secret messages in two of the same cover images; only someone who has both stego-images can extract the secret messages. There are many advantages to using dual-image in data hiding, such as its high payload, reversibility, and strong robustness.

The proposed method improves Lu's center-folding strategy by including the folding operation twice and using an indicator to identify the second folding operation as a means of distinguishing different sub-bands to further decrease the reduced digit. In addition, the proposed method effectively solves the overflow/underflow problem.

In order to evaluate the performance of the proposed scheme, eight schemes with different B and K values were implemented. Moreover, five related methods were implemented for a comparison with

the proposed scheme. The first experiment showed that a small B value achieves higher image quality, a large K value achieves a large payload, and the re-encoding process can effectively narrow down the distortion when B is larger. The second experiment compared the proposed and related methods. The results showed that the proposed scheme can achieve higher image quality ($B = 3$, $K = 3$, $RE = 0$), better image quality under the same embedding rate, and a higher payload than other schemes ($B = 5$, $K = 4$, $RE = 1$). The third experiment used several steganalysis techniques to prove the strong robustness of the proposed scheme, include histogram steganalysis, RS steganalysis, primary sets, Chi square, sample pairs RS analysis, and fusion detection.

Acknowledgments: This study was financially supported by a research grant from Taiwan's Ministry of Science and Technology (MOST 105-2221-E- 324-020).

Author Contributions: Tzu-Chuen Lu designed the algorithm, conducted all experiments, analyzed the results, wrote the manuscript, and conducted the literature review. Hui-Shih Leng conceived the algorithm and wrote the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Mielikainen, J. LSB matching revisited. *IEEE Signal Process. Lett.* **2006**, *13*, 285–287. [CrossRef]
2. Wu, D.-C.; Tsai, W.-H. A steganographic method for images by pixel-value differencing. *Pattern Recognit. Lett.* **2003**, *24*, 1613–1626. [CrossRef]
3. Chang, C.C.; Tseng, H.W. A steganographic method for digital images using side match. *Pattern Recognit. Lett.* **2004**, *25*, 1431–1437. [CrossRef]
4. Zhang, X.; Wang, S. Efficient steganographic embedding by exploiting modification direction. *IEEE Commun. Lett.* **2006**, *10*, 781–783. [CrossRef]
5. Kieu, T.D.; Chang, C.C. A steganographic scheme by fully exploiting modification directions. *Expert Syst. Appl.* **2011**, *38*, 10648–10657. [CrossRef]
6. Ni, Z.; Shi, Y.Q.; Ansari, N.; Su, W. Reversible data hiding. *IEEE Trans. Circuits Syst. Video Technol.* **2006**, *16*, 354–362.
7. Alattar, A. Reversible watermarks using a difference expansion. In *Internet and Communications Multimedia Security Handbook*; Furht, B., Kirovski, D., Eds.; CRC Press: Boca Raton, FL, USA, 2004.
8. Tian, J. Reversible data embedding using a difference expansion. *IEEE Trans. Circuits Syst. Video Technol.* **2003**, *13*, 890–896. [CrossRef]
9. Li, X.; Yang, B.; Zeng, T. Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection. *IEEE Trans. Image Process.* **2011**, *20*, 3524–3533. [PubMed]
10. Gui, X.; Li, X.; Yang, B. A high capacity reversible data hiding scheme based on generalized prediction-error expansion and adaptive embedding. *Signal Process.* **2014**, *98*, 370–380. [CrossRef]
11. Li, X.; Li, J.; Li, B.; Yang, B. High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion. *Signal Process.* **2013**, *93*, 198–205. [CrossRef]
12. Peng, F.; Li, X.; Yang, B. Improved PVO-based reversible data hiding. *Digit. Signal Process.* **2014**, *25*, 255–265. [CrossRef]
13. Qu, X.; Kim, H.J. Pixel-based pixel value ordering predictor for high-fidelity reversible data hiding. *Signal Process.* **2015**, *111*, 249–260. [CrossRef]
14. Wang, X.; Ding, J.; Pei, Q. A novel reversible image data hiding scheme based on pixel value ordering and dynamic pixel block partition. *Inf. Sci.* **2015**, *310*, 16–35. [CrossRef]
15. Chang, C.C.; Kieu, T.D.; Chou, Y.C. Reversible data hiding scheme using two steganographic images. In Proceedings of the 2007 IEEE Region 10 International Conference (TENCON), Taipei, Taiwan, 30 Ocotber–2 November 2007; pp. 1–4.
16. Lee, C.F.; Wang, K.H.; Chang, C.C.; Huang, Y.L. A reversible data hiding scheme based on dual steganographic images. In Proceedings of the 3rd International Conference on Ubiquitous Information Management and Communication (ICUIMC '09), Suwon, Korea, 15–16 January 2009; pp. 228–237.
17. Lee, C.F.; Huang, Y.L. Reversible data hiding scheme based on dual stegano-images using orientation combinations. *Telecommun. Syst.* **2013**, *52*, 2237–2247. [CrossRef]

18. Chang, C.C.; Lu, T.C.; Horng, G.; Huang, Y.H.; Hsu, Y.M. A high payload data embedding scheme using dual stego-images with reversibility. In Proceedings of the 2013 9th International Conference on Information, Communications and Signal Processing, Tainan, Taiwan, 10–13 December 2013; pp. 1–5.
19. Qin, C.; Chang, C.C.; Hsu, T.J. Reversible data hiding scheme based on exploiting modification direction with two steganographic images. *Multimed. Tools Appl.* **2014**, *74*, 5861–5872. [CrossRef]
20. Lu, T.C.; Tseng, C.Y.; Wu, J.H. Dual imaging-based reversible hiding technique using LSB matching. *Signal Process.* **2015**, *108*, 77–89. [CrossRef]
21. Lu, T.C.; Wu, J.H.; Huang, C.C. Dual-image-based reversible data hiding method using center folding strategy. *Signal Process.* **2015**, *115*, 195–213. [CrossRef]
22. Lu, T.C.; Chi, L.P.; Wu, C.H.; Chang, H.P. Reversible data hiding in dual stego-images using frequency-based encoding strategy. *Multimed. Tools Appl.* **2016**. [CrossRef]
23. Fridrich, J.; Golijan, M.; Du, R. Reliable detection of LSB steganography in grayscale and color images. In Proceedings of the 2001 Workshop on Multimedia and Security, Ottawa, ON, Canada, 5 October 2001; pp. 27–30.
24. Boehm, B. StegExpose—A Steganalysis Tool for Detecting LSB Steganography in Images. Available online: http://arxiv.org/abs/1410.6656 (accessed on 19 September 2017).