Image and Vision Computing 26 (2008) 1670–1676

Contents lists available at ScienceDirect

Image and Vision Computing

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

An improvement of EMD embedding method for large payloads by pixel segmentation strategy

Chin-Feng Lee [a,*], Chin-Chen Chang [b,c], Kuo-Hua Wang [b]

[a] Department of Information Management, Chaoyang University of Technology, 168 Jifong E. Road, Wufong Township, Taichung County 41349, Taiwan, ROC
[b] Department of Information Engineering and Computer Science, Feng Chia University, 100 Wenhwa Rd., Seatwen, Taichung 40724, Taiwan, ROC
[c] Department of Computer Science and Information Engineering, National Chung Cheng University, 160 San-Hsing, Ming-Hsiung, Chiayi 621, Taiwan, ROC

ARTICLE INFO

Article history:
Received 22 November 2007
Received in revised form 1 April 2008
Accepted 5 May 2008

Keywords:
Covert communication
Digit steganography
Data hiding
Embedding capacity

ABSTRACT

In this paper, a novel data hiding method by using pixel segmentation strategy is proposed. The proposed paper keeps $(16 - p_m)$ MSBs of a pixel-pair unchanged and alters p_m LSBs to indicate the virtual modifications on an m-dimensional pseudo-random vectors for carrying the secret data, where $m = 2^{p_m - 1} - 1$. The embedding rate of proposed method is $R = (\log_2(2m + 1))/2$, which is greater than that of the EMD embedding method proposed by Zhang and Wang [X. Zhang, S. Wang, Efficient steganographic embedding by exploiting modification direction, IEEE Communication Letters 10 (2006) (113), pp.781–783], because the embedding rate of EMD embedding method is $R = (\log_2(2n + 1))/n$, when $m > 2$ and $n \geqq 2$. The experimental results show that the proposed method increases the number of embedded secret bits more than 1.7 times compared with the EMD embedding method. Even with such high embedding capacity, the average PSNR of 44.3 dB shows that the visual quality does not decline to an unacceptable degree.

1. Introduction

Message transmission over the Internet is quite common in almost everywhere. Some problems may arise due to the security flaws of communication channel. Digital steganography, a kind of data security technique, has been developed quickly and it receives a great deal of attentions from both the academic and industrial communities [1–11]. An original image, also called a cover image, is used to embed the secret data. The stego-image is a version of the cover image where secret messages are embedded. By means of creating stego-images that are perceptually identical to the cover images with small embedding distortion, the imperceptibility for protecting the sensitive and confidential information can be maintained without being detected or extracted. Two important issues for current data hiding techniques are to preserve the imperceptibility and the embedding capacity at the same time. However, this is an irreconcilable conflict: if we want to preserve high stego-image quality, we usually have to scarify the embedding capacity, and vice versa.

Nowadays, many steganographic methods have been proposed to hide secret messages into an image. A commonly used method is the least significant bit (LSB) replacement, which is the simplest hiding technique by using some least bits of cover pixels to embed secret data [4–6,10]. Mielikainen's LSB matching revisited method [7] can resist immune against the steganographic attacks because it has not the asymmetric property of LSB replacement methods. LSB matching revisited method also achieves the same quality of stego-images as well as LSB replacement does. However, it does not enhance the embedding capacity. For a cover image I with $M \times N$ pixels, the maximum data hiding capacity of LSB steganography is $M \times N$ bits. Define the embedding rate R as the rate of the length of embedded messages to the maximum capacity $M \times N$. If the rate R is more than one, it represents the steganographic embedding method with high embedding capacity [8,9,11].

Recently, Zhang and Wang [9] proposed a steganography to transform the binary secret data into a stream of secret digits in a $(2n + 1)$-ary notational system. Zhang and Wang's embedding method uses n cover pixels to carry one secret digit in a $(2n + 1)$-ary notational system. The embedding rate, $R = (\log_2(2n + 1))/n$, is defined for calculating the number of secret bits that can be carried by a cover pixel. Though Zhang and Wang's method could achieve high embedding efficiency and secrecy with low distortion, a larger n leads to a smaller embedding rate according to the equation $R = (\log_2(2n + 1))/n$. In other words, the EMD embedding method has its maximum embedding rate which is approximate 1.161 bits per pixel when $n = 2$.

This paper improves the EMD embedding method, especially in embedding capacity. In our proposed method, each secret digit in a $(2m + 1)$-ary notational system is carried by a pair of cover pixels. The data hider can segment the pair of cover pixels into two parts: one is used to embed the secret data and the other is used as an indicator for the decoder to extract the secret data. The embedding

* Corresponding author. Tel.: +886 4 23323000 4293; fax: +886 4 23742337.
E-mail addresses: lcf@cyut.edu.tw (C.-F. Lee), ccc@cs.ccu.edu.tw, m9503085@fcu.edu.tw (C.-C. Chang).

0262-8856/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2008.05.005

rate of our method is steadily increased as m gets larger without losing the stego-image quality and security. Experimental results show that our proposed method still keeps pretty good quality of the stego-image, while the embedding capacity is exactly enlarged.

The rest of this paper is organized as follows. In Section 2, we review previous approaches of embedding in grayscale images. Then, in Section 3, we introduce the proposed embedding and extracting procedures for grayscale images. We make comparisons of embedding rate and stego-image quality between our method and Zhang and Wang's method in Section 4. Finally, we conclude this in Section 5.

2. The EMD embedding method

An embedding method to exploit the modification directions for data hiding (also called the EMD embedding method for short) was proposed by Zhang and Wang [9]. The EMD embedding method pseudo-randomly permutes all pixels in a cover image with a secret key to partition the pixels into a series of groups. A pixel-group contains n grayscale pixels and is denoted as $(g_1, g_2, \ldots, g_n)$. The secret message to be embedded needs to be converted into a sequence of secret digits of a $(2n+1)$-ary notational system, where each secret digit falls within $[0, 2n]$. Eq. (1) depicts that a secret message of binary stream can be segmented into many pieces with L bits, and the decimal value of each secret piece is represented by K digits in the $(2n+1)$-ary notational system, where

$$L = \lfloor K \cdot \log_2(2n+1) \rfloor. \tag{1}$$

In the EMD embedding method, the data hiders can choose one of $(2n+1)$-ary notational systems and determine the length K of segmented bits to convert the secret message into a sequence of secret digits. This feature can increase the security if the attacker does not know the value of K and n. Zhang and Wang defined an embedding function f as weighted sum function modulo $(2n+1)$ for each pixel-group:

$$f(g_1, g_2, \ldots, g_n) = (g_1 \times 1 + g_2 \times 2 + \ldots + g_n \times n) \bmod (2n+1). \tag{2}$$

If a secret digit to be embedded into a given cover pixel-group $(g_1, g_2, \ldots, g_n)$ is not equal to the value calculated from the embedding function, only one of the cover pixels has to be modified by either increasing or decreasing one; otherwise, no modification needs to be done. That is why the EMD embedding method's distortion induced in the stego-image is not great.

In the extracting procedure, if the stego pixel-group is $(g'_1, g'_2, \ldots, g'_n)$, then the secret digit can be extracted by the following extraction function.

$$f(g'_1, g'_2, \ldots, g'_n) = (g'_1 \times 1 + g'_2 \times 2 + \ldots + g'_n \times n) \bmod (2n+1). \tag{3}$$

The EMD embedding method provides a pretty high stego-image quality with the PSNR value greater than 51 dB in average. This is because at most one cover pixel needs to be increased or decreased by one for a pixel-group of cover pixels. Though, the EMD embedding method gets high quality of stego-image, there is still much room to further improve the embedding capacity, because the embedding rate $R = (\log_2(2n+1))/n$ for the best case of $n = 2$ is only 1.2 secret bits per cover pixel.

3. Our proposed method

In this section, we firstly introduce the developed pixel-pair segmentation strategy. Then, the proposed data embedding and extracting procedures are presented to improve information hiding ratio than that of Zhang and Wang's method.

3.1. Pixel-pair segmentation

The message embedding is performed onto a pair of cover pixels. With two grayscale pixels, 16 bits can be segmented into four pieces (see Fig. 1(a)), which are denoted as S_1, S_2, S_3, and S_4, respectively. S_1 refers to the piece which has PV_1 most significant bits of the first cover pixels, S_2 refers to the piece which has $(8 - PV_1)$ least significant bits of the first cover pixels, S_3 refers to the piece which has PV_2 most significant bits of the second cover pixels, and S_4 refers to the piece which has $(8 - PV_2)$ least significant bits of the second cover pixels. Let p_m is the length of S_2 concatenating with S_4 i.e., $p_m = (8 - PV_1) + (8 - PV_2) = 16 - (PV_1 + PV_2)$. After the pixel segmentation is done, S_1 and S_3 are combined to be an area of a vector of coordinates (VCA) as shown in Fig. 1(b) which can be further processed to form a group of integer coordinates for logically embedding data. Moreover, S_2 and S_4 are combined to be a coordinate vector modification area (VMA) as shown in Fig. 1(c).

Example 3.1. For a given pair of cover pixels $C = (125, 101)_{10}$, we first transform C into its corresponding binary stream ($\underline{01111101}$, $\underline{01100101})_2$. Assume a key is used to obtain $PV_1 = 7$ and $PV_2 = 6$, i.e., $p_m = 3$, then the bits of C in VCA are ($\underline{0111110}$ $\underline{011001}$)$_2$ and the bits of C in VMA are $(101)_2$.

3.2. Embedding procedure

As we know, the distortion of grayscale image increases greatly if we change the most significant bits (MSBs) instead of the least significant bits (LSBs). In addition, the modification of the most significant bits is much easier to be detected. Therefore, in the proposed method, only p_m least significant bits distributed over the VMA of the given pair of cover pixels are possibly altered. In other words, we do not physically modify the $(16 - p_m)$ most significant bits to achieve the embedding of secret messages. In this way, good stego-image quality can be preserved and the embedding ratio is high.

In fact, the $(16 - p_m)$ bits of VCA are merely used as a seed of random function f_r which can be predetermined and known by both data hiders and extractors. The outcome generated by the

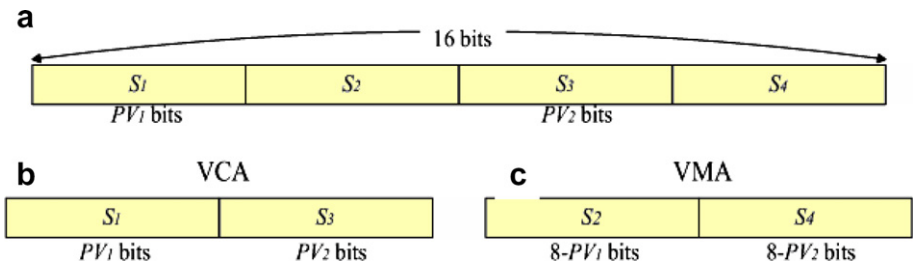

Fig. 1. Pixel-pair segmentation. (a) A grayscale pixel-pair; (b) VCA; (c) VMA.

random function is a vector of coordinates with m positive integers, which can be denoted as $(g_1,g_2,\ldots,g_m)$ and can be expressed as a unit of hyper-cube in an m-dimensional space. Here, m is determined by the value of p_m such that $m = 2^{p_m-1} - 1$. The hyper-cube related to the vector of coordinates $(g_1, g_2,\ldots,g_m)$ is inspired from the hyper-cube of a pixel-group introduced by Zhang and Wang. The difference between the vector of coordinates $(g_1, g_2,\ldots,g_m)$ proposed here and the pixel-group proposed by Zhang and Wang is that the vector of coordinates is projected from the $16 - p_m$ bits of VCA for imitating the embedding way proposed by Zhang et al. That is, m integers of vector coordinates are not cover pixels. Therefore, no "physical" modification is performed on the coordinate values. Also, no bits of VCA are modified during the embedding procedure. The embedded secret message has to be converted into a sequence of secret digits of the $(2m + 1)$-ary notational system, where each secret digit falls within $[0,2m]$. We modify the p_m bits of VMA to indicate which integer of $g_1, g_2,\ldots$, and g_m will be "logically" increased/decreased one when a secret digit is going to be embedded.

The embedding steps are presented as follows.

Step 1: Determine the p_m value as introduced in the procedure of pixel-pair segmentation.

Step 2: Pseudo-randomly permute all grayscale cover pixels with a secret key, and divide these cover pixels into a series of pixel-pairs.

Step 3: For each pair of cover pixels, say C, we further proceed to divide the pixel-pair C into two areas, VCA and VMA as shown in Fig. 1.

Step 4: Use the $(16 - p_m)$ bits in VCA as a seed of random function to generate a vector of coordinates denoted as $(g_1,g_2,\ldots,g_m)$, where g_i can be any positive integer, for $i = 1, 2,\ldots, m$, where $m = 2^{p_m-1} - 1$.

Step 5: Convert a binary secret message into a sequence of secret digits in the $(2m + 1)$-ary notational system.

Step 6: Apply the embedding function which can be calculated in Eq. (4) to embed every secret digit in the $(2m + 1)$-ary notational system.

$$F = f(g_1,g_2,\ldots,g_m) = \left[\sum_{i=1}^{m}(g_i \times i)\right] \bmod (2m+1). \tag{4}$$

Since the vector of coordinates $(g_1,g_2,\ldots,g_m)$ corresponds to a unit hyper-cube in an m-dimensional integer space, the hyper-cube can be labeled with its F value. Obviously, the F value of any hyper-cube and its $2m$ neighbors are integers within $[0,2m]$, and they are mutually different.

There are $(2m + 1)$ possible cases when embedding a secret digit s. If s is equal to F, no modification for the vector of coordinates is needed. Otherwise, one of the m coordinates, i.e., g_i needs to be "logically" modified. From Eq. (5), d can be obtained first.

$$d = s - F \bmod (2m+1). \tag{5}$$

Then, when d is greater than m, the value of $g_{(2m-1)-d}$ has to be "logically" decreased by one; otherwise, the value of g_d has to be "logically" increased by one. That is, when we embed a secret digit in the $(2m + 1)$-ary notational system, one "logical outcome" of $(g_1,g_2,\ldots,g_m)$, $(g_1 - 1,g_2,\ldots,g_m)$, $(g_1 + 1,g_2,\ldots,g_m)$, $(g_1,g_2 - 1,\ldots,g_m)$, $(g_1,g_2 + 1,\ldots,g_m),\ldots$, $(g_1,g_2,\ldots,g_m - 1)$, and $(g_1,g_2,\ldots,g_m + 1)$ is obtained after applying the embedding function in Eq. (5).

Example 3.2. According to Example 3.1, when $p_m = 3$, m is three. Assume that the bit stream $(\underline{0111110}\ \underline{011001})_2$ is an input argument of a predetermined random function f_r and the random function f_r will return a vector of coordinates, $(7,13,9)_{10}$. According to Eq. (5), the F value of hyper-cube corresponding to $(7,13,9)_{10}$ is 4. Meanwhile, the values of hyper-cubes corresponding to the six neighbors of $(7,13,9)_{10}$ are all within $[0,6]$ as shown in Fig. 2.

Example 3.3. Let's continue Example 3.2. We have gotten the hyper-cube value (say $F = 4$) by given the vector of coordinates $(7,13,9)_{10}$, where $m = 3$. If a secret digit falling in $[0,7]$ is going to be embedded, the coordinates should be "logically" modified to be one of the $(2m + 1)$ outcomes, totally depending on the value of the embedded secret digit s. The seven outcomes are (g_1,g_2,g_3), $(g_1 - 1,g_2,g_3)$, $(g_1 + 1,g_2,g_3)$, $(g_1,g_2 - 1,g_3)$, $(g_1,g_2 + 1,g_3)$, $(g_1,g_2,g_3 - 1)$, and $(g_1,g_2,g_3 + 1)$, respectively, discussed as follows.

Case 1: if we want to embed $(0)_7$, we first calculate $d = s - F \bmod (2m + 1) = 0 - 4 \bmod 7 = 3$, which is not greater than m, where $m = 3$. Therefore, we add one to g_3 such that the vector of coordinates $(7,13,9)_{10}$ can be replaced by $(7,13,10)_{10}$.

Case 2: if we want to embed $(1)_7$, we calculate $d = s - F \bmod (2m + 1) = 1 - 4 \bmod 7 = 4$, and get d is greater than m. Therefore, we decrease g_{2m+1-d} by 1 to obtain $(7,13,8)_{10}$.

Case 3: if we want to embed $(2)_7$, we do the calculation as $d = s - F \bmod (2m + 1) = 2 - 4 \bmod 7 = 5$, and see d is greater than m. Therefore, we decrease g_{2m+1-d} by 1 such that $g_2 = 12$ and the vector of coordinates is $(7,12,9)_{10}$.

Case 4: if we want to embed $(3)_7$, d is calculated as $d = s - F \bmod (2m + 1) = 3 - 4 \bmod 7 = 6$, which is greater than m. Therefore, we decrease g_{2m+1-d} by 1 such that $g_1 = 6$ and the vector of coordinates is $(6,13,9)_{10}$.

Case 5: To embed $(4)_7$ in $(7,13,9)_{10}$, according to Fig. 2, if we want to embed $(3)_7$ in $(7,13,9)_{10}$, no modification is needed.

Case 6: if we want to embed $(5)_7$, we have $d = s - F \bmod (2m + 1) = 5 - 4 \bmod 7 = 1$, which is not greater than m. Therefore, we add one to g_1 such that the vector of coordinates $(7,13,9)_{10}$ can be replaced by $(8,13,9)_{10}$.

Case 7: if we want to embed $(6)_7$, we calculate $d = s - F \bmod (2m - 1) = 0 - 4 \bmod 7 = 2$, which is not greater than m. Therefore, we add one to g_2 such that the vector of coordinates $(7,13,9)_{10}$ can be replaced by $(7,14,9)_{10}$.

As mentioned in Section 3.1, the most $(16 - p_m)$ significant bits of VCA can be used as a seed of random function and promise no bit modification will be occur during the whole process. In the last step of embedding procedure, we physically modify the p_m bits of VMA to indicate how the vector of coordinates are changed according to Eq. (4). We use the most $(p_m - 1)$ bits of VMA to which g_i of the coordinate vector $(g_1,g_2,\ldots,g_m)$ should be logically modified. The last one bit of VMA is used to be a sign indicator. If the bit

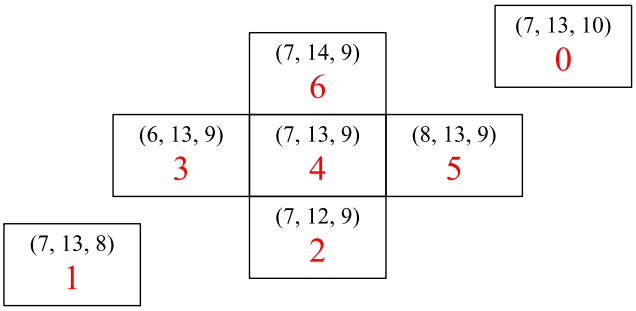

Fig. 2. the hyper-cube values of the vector of coordinates $(7,13,9)_{10}$ and its six neighbors.

of sign indicator is "0," then the ith g value should be decreased by one; otherwise, the value g_i should be increased by one.

Example 3.4. Let us continue Example 3.3 with $p_m = 3$, $m = 3$. There are (2*3 + 1) cases for physically modifying the three bits of VMA to express the logical modification of the vector of coordinates $(7,13,9)_{10}$. The seven cases are described in detail as follows.

Case 1: If the modification we have to do is $g_1 + 1$, it means that the three bits in VMA is $(01\underline{1})_2$.
Case 2: If the modification we have to do is $g_1 - 1$, it means that the three bits in VMA is $(01\underline{0})_2$.
Case 3: If the modification we have to do is $g_2 + 1$, it means that the three bits in VMA is $(10\underline{1})_2$.
Case 4: If the modification we have to do is $g_2 - 1$, it means that the three bits in VMA is $(10\underline{0})_2$.
Case 5: If the modification we have to do is $g_3 + 1$, it means that the three bits in VMA is $(11\underline{1})_2$.
Case 6: If the modification we have to do is $g_3 - 1$, it means that the three bits in VMA is $(11\underline{0})_2$.
Case 7: If we do no modification on the vector of coordinates $(7,13,9)_{10}$, it means that the three bits in VMA can be either $(000)_2$ or $(001)_2$.

3.3. Data extraction

The same as the embedding procedure, the extracting procedure has to segment a given pair of stego-pixels into two parts: the vector coordinate area (VCA) composed of $(16 - p_m)$ bits, and the vector modification area (VMA) composed of p_m bits.

Use the $(16 - p_m)$ bits in VCA as a seed of a predetermined random function f_r to generate a vector of coordinates denoted as $(g'_1, g'_2, \ldots, g'_m)$, where g'_i can be any positive integer, for $i = 1, 2, \ldots, m$ and $m = 2^{p_m - 1} - 1$. Apply the following extraction function as shown in Eq. (6) to the vector of coordinates $(g'_1, g'_2, \ldots, g'_m)$ so as to extract the embedded secret digit s in the $(2m + 1)$-ary notational system.

$$s = f(g'_1, g'_2, \ldots, g'_m) = (g'_1 \times 1 + g'_2 \times 2 + \ldots + g'_m \times m) \bmod (2m + 1). \quad (6)$$

Example 3.5. For a given pair of stego-pixels which is $(126,101)_{10} = (01111\underline{10}, 0110010\underline{1})_2$, $p_m = 3$, $m = 3$, and the vector of coordinates $(g'_1, g'_2, g'_3) = (7,13,9)_{10}$, which is the outcome of a predetermined random function f_r by using the binary stream $(011111\ 0110010)_2$ as the input argument of f_r. For the three bits, i.e., $(\underline{10}\ \underline{1})_2$, of VMA which obviously comes from the given pair of stego-pixels, the most two significant bits, i.e., $(\underline{10})_2$, indicate that $g'_2 \neq g_2$, and $g'_i = g_i$ for $i = 1, 3$. Similarly, since the last significant bit of VMA is $(\underline{1})_2$, it indicates that $g'_2 = g_2 + 1$.The outcome means that the vector of coordinates has been changed from $(7,13,9)_{10}$ to $(7,14,9)_{10}$. After applying Eq. (6) to the vector $(7,14,9)_{10}$, we can extract the secret digit $(s = 6)$.

4. Experimental results

For evaluating the embedding capacity as well as visual stego-image quality, we implemented the most common LSB replacement, Zhan and Wang's method and the proposed method by using C language on the Intel Core 2 Duo 5600 CPU computer with 1GB RAM.

In order to compare the performance of the EMD embedding and the proposed methods, we use the embedding rate defined as $R = S/H \times W$ to calculate the number of secret bits carried by one cover pixel, i.e., bits per pixel (bpp), where S refers to the number of bits of the secret data, and $H \times W$ is the total number of cover pixels.

By Zhang and Wang's method, $S = (H \times W \times \log_2(2n + 1))/n$. Therefore, the embedding rate R of the EMD embedding method is $(\log_2(2n + 1))/n$, where $n \geqq 2$, which has the best embedding rate R of $(\log_2 5)/2$, when n is set to be 2. In our proposed method, $S = (H \times W \times \log_2(2m + 1))/2$ that implies $R = \log_2(2m + 1)/2 = \log_2(p_m - 1)/2$ and the best embedding rate is $\log_2 15/2$, which is almost twice than that of Zhang and Wang's method. Table 1 compares the embedding rate for the EMD embedding and the proposed methods.

The proposed paper keeps $(16 - p_m)$ MSBs of a pixel-pair unchanged and alters p_m LSBs to indicate the virtual modifications on m-dimensional pseudo-random vectors for carrying the secret data. Compared with the LSB replacement, the proposed method uses p_m LSBs to theoretically carry $\log_2(p_m - 1)/2$ secret bits, which is slightly less than the number of secret bits embedded in the LSB replacement. Given a cover pixel-pair, Table 2 compares the embedding capacity for exploiting two and three LSB planes, respectively, to carry the number of secret bits per cover pixel. The last row of Table 2 shows the average secret bits that are exactly embedded into a cover pixel.

Peak signal-to-noise rate (denoted as PSNR) is a common measurement for evaluating the performance of embedding methods. In the experiment, we use PSNR value as a criterion to estimate the stego-image quality. We test the proposed algorithm and Zhang and Wang's method on various grayscale images named "Lena," "Baboon," "Tiffany," "F16," "Pepper," "Goldhill," "Boat," "Babara," and "Zelda." These test images have the same size, that is 512×512.

The EMD embedding method has the maximum embedding rate about 1.16 bits per pixel, meaning that $R = (\log 5)/2$ bits/pixel, and the average PSNR value of stego-images is about 51 dB with 262,144 embedded bits. At the embedding rate $R = (\log 15)/2$ with the embedding capacity of 441,913–463,528 bits, we present the PSNR values of our proposed method for these nine stego-images in Fig. 3. The average PSNR value is about 44.3 dB. As $m = 2^{p_m - 1} - 1$, m is determined by the value of p_m. When the

Table 1
The embedding rate (bpp) comparison between the EMD and the proposed methods

The number of cover pixels for carrying one secret digit	2	3	4
EMD embedding	$\log_2(5)/2$	$\log_2(7)/3$	$\log_2(9)4$
Proposed method with 2LSBs	$\log_2(15)/2$	$\log_2(63)/3$	$\log_2(255)/4$
Proposed method with 3LSBs	$\log_2(63)/2$	$\log_2(511)/3$	$\log_2(4095)/4$

Table 2
Given a cover pixel-pair, the embedding secret bits per cover pixel (bpp) for exploiting two and three LSB planes, respectively

The number of LSB planes used to carry secret bits	2 LSB planes	3 LSB planes
LSB replacement	2	3
Proposed method (theoretical results)	$\log_2(15)/2 = 1.953$	$\log_2(63)/2 = 2.9886$
Proposed method (the average secret bits actually embedded in a cover pixel according to the experimental results)	1.733	2.746

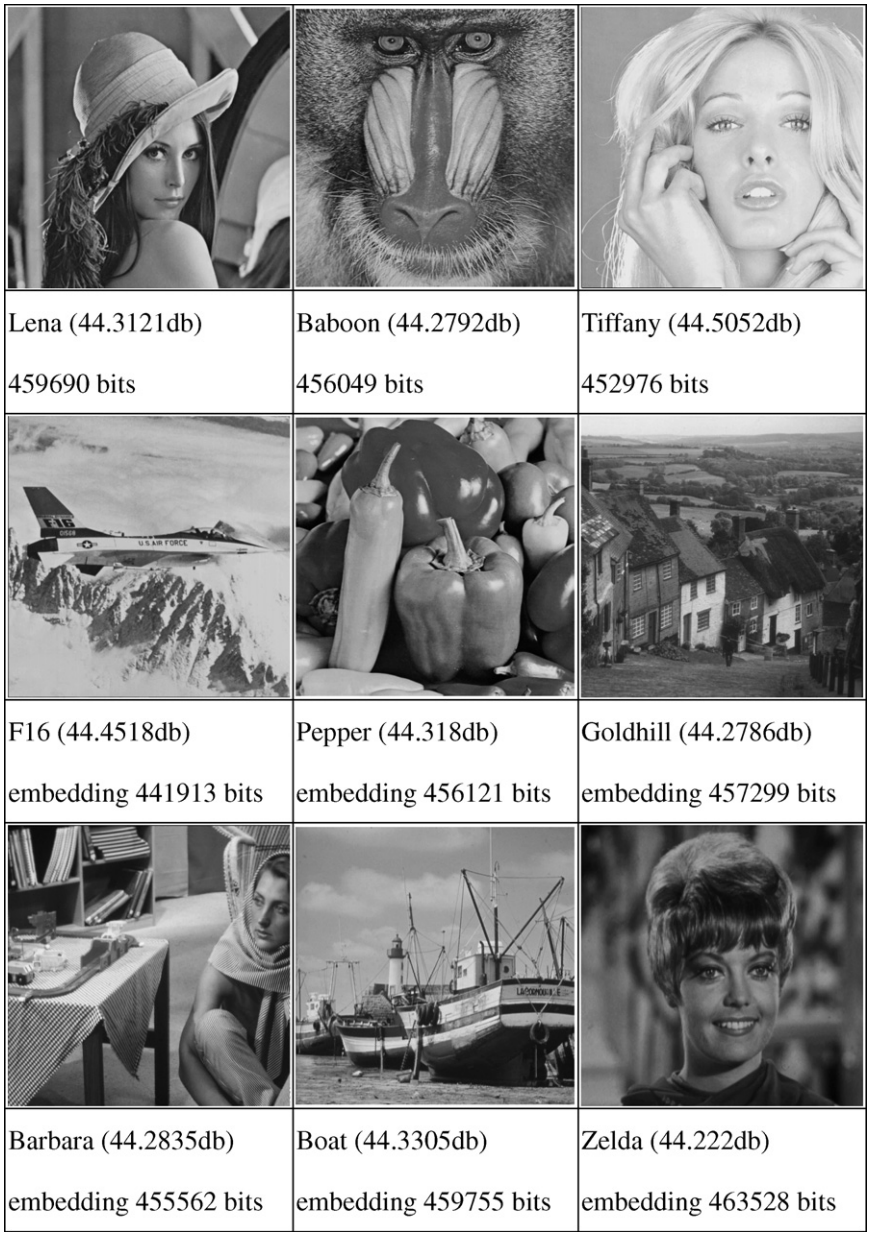

Fig. 3. Results of stego-images with $R = (\log 15)/2$.

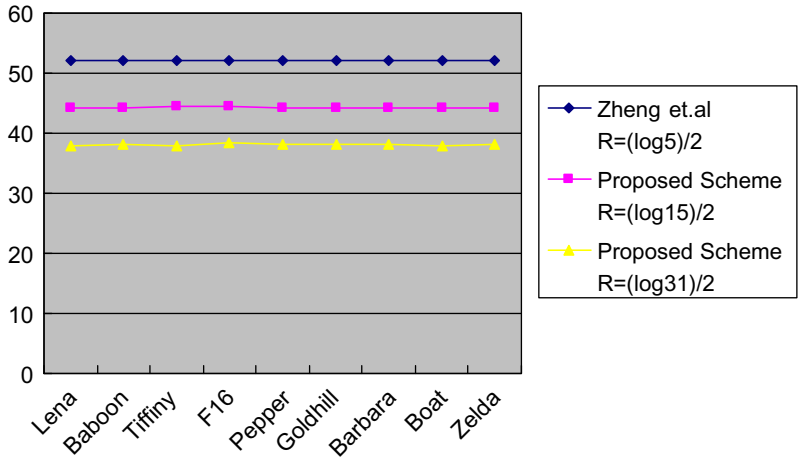

Fig. 4. The PSNR values of proposed method's and Zhang et al.'s method by embedding secret data with 262,144 bits and full embedding.

embedding rate R is $(\log 15)/2$, the PSNR value of stego-image is about 44.3 dB in average; when the embedding rate R is raised to $(\log 31)/2$, the PSNR value will reduce to the average of 38 dB (Fig. 4). Although the PSNR values of stego-image are lower than those of the EMD embedding method, our proposed method has higher embedding rate and more embedded bits as well. In other words, our proposed method can tune down the p_m value to improve the embedding rate. In the proposed method, a trade-off between the embedding capacity and the stego-image quality may be realized by adjusting the p_m value.

The difference image histograms for the detection of LSB replacement and proposed methods are shown in Fig. 5. Fig. 5(a) is the image histogram for the original "Lena" image of size 512×512. Fig. 5(b) and (c) represent the stego-image histograms produced using the LSB replacement and the proposed methods by exploiting two LSB planes to carry secret data. Similarly, Fig. 5(d) and (e) represent the stego-image histograms produced using the LSB replacement and the proposed methods by exploiting three LSB planes to carry secret messages. In terms of the statistical analysis, Fig. 5(b) through (e) illustrate the comparable behaviors for

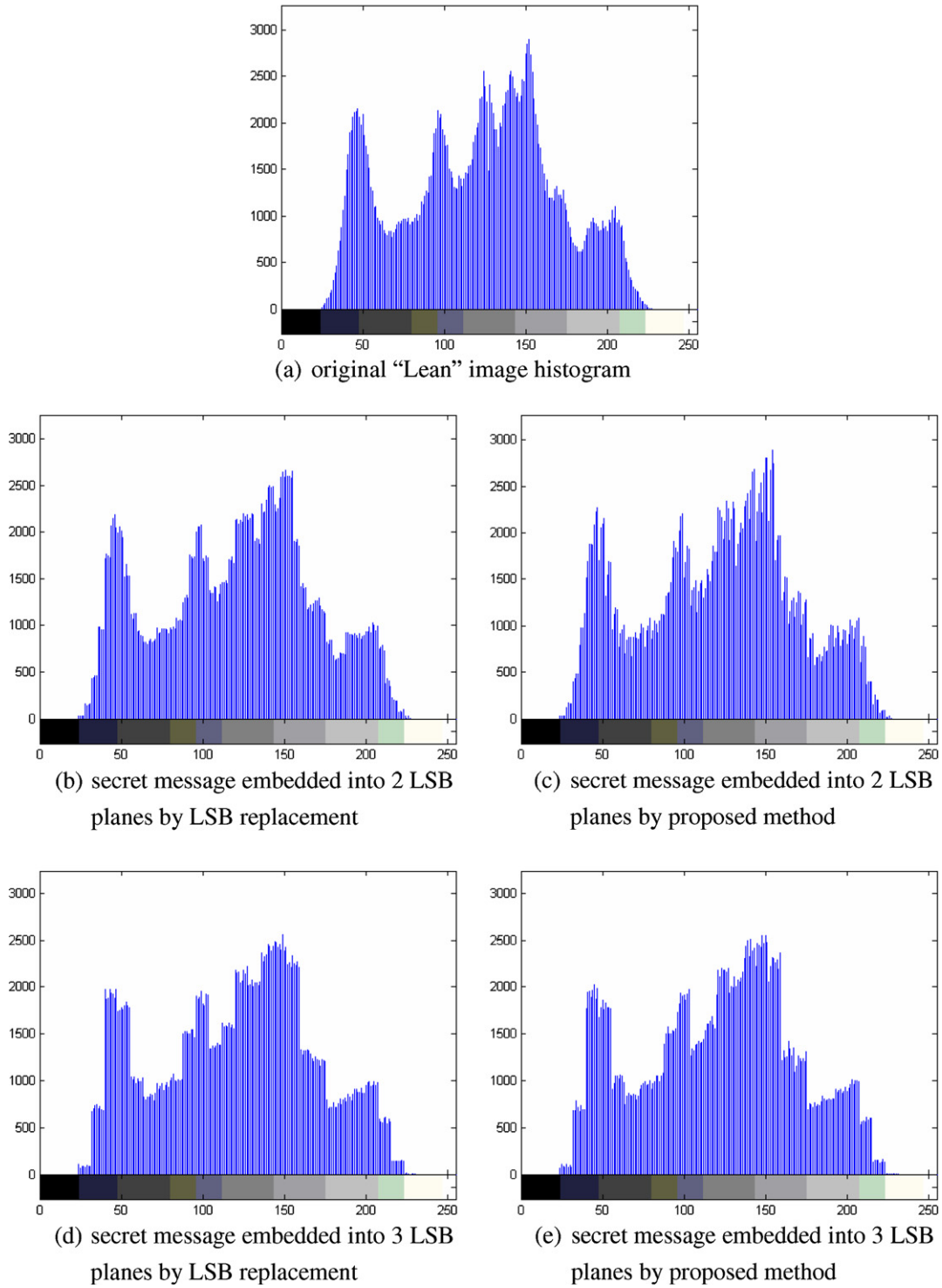

Fig. 5. The difference image histograms for the detection of LSB replacement and proposed methods.

both the LSB replacement and the proposed methods; their stego-image histograms are different from the original one and can be noticeable when the data hiding are performed on three LSB planes. However, the LSB replacement is less secure, because it is easy to sequentially collect secret message in terms of p_m LSB bits, thereby easily leaving recognizable fingerprints. In contrast, the proposed method uses a random generator f_r to project an m-dimensional pseudo-random vector onto a virtual hyperspace firstly and then deals with the virtual modifications to hide secret messages by exploiting the EMD embedding. The random generator f_r as well as the value of m act as secure keys. Without knowing the keys, the attacker has no awareness of the secret message. Therefore, the proposed method can achieve some degree of security.

5. Conclusions

The embedding capacity and the quality of stego-image are two important issues for data hiding methods. Our proposed method has the embedding rate $R = (\log_2(2m + 1))/2$ which is greater than that $R = (\log_2 (2n + 1))/n$ of the EMD embedding method proposed by Zhang et al., when $m > 2$ and $n \geqq 2$. The experimental results show that the proposed method is able to embed more information than the EMD embedding method can do and still keeps good stego-image quality at an acceptable level.

References

[1] Hide and Seek. Available from: <http://www.jjtc.com/Security/stegtools.htm>.
[2] S. Katzenbeisser, F.A.P. Petitcolas, Information Hiding Techniques for Steganography and Digital Watermarking, Artech House, Boston, 2000.
[3] H. Wang, S. Wang, Cyber warfare: steganography vs. steganalysis, Communications of the ACM 47 (10) (2004) 76–82.
[4] S. Dumitrescu, X. Wu, Z. Wang, Detection of LSB steganography via sample pair analysis, IEEE Transactions on Signal Processing 51 (7) (2003) 1995–2007.
[5] A. Ker, Improved detection of LSB steganography in grayscale images, in: J. Fridrich (Ed.), Proceedings of the Sixth International Workshop on Information Hiding, vol. 3200, Springer, Toronto, Canada, May 2004, pp. 97–115.
[6] A. Ker, Quantitative evaluation of pairs and RS steganalysis, in: Edward J. Delp III, Ping W. Wong (Eds.), Proceedings of SPIE – Security, Steganography, and Watermarking of Multimedia Contents VI, SPIE, vol. 5306, California, USA, January 2004, pp. 83–97.
[7] J. Mielikainen, LSB matching revisited, IEEE Signal Processing Letters 13 (5) (2006) 285–287.
[8] C.C. Thien, J.C. Lin, A simple and high-hiding capacity method for hiding digit-by-digit data in images based on modulus function, Pattern Recognition 36 (12) (2003) 2875–2881.
[9] X. Zhang, S. Wang, Efficient steganographic embedding by exploiting modification direction, IEEE Communications Letters 10 (113) (2006) 781–783.
[10] C.K. Chan, L.M. Cheng, Hiding data in images by simple LSB substitution, Pattern Recognition 37 (3) (2004) 469–474.
[11] T. Zhang, X. Ping, A new approach to reliable detection of LSB steganography in natural images, Signal Processing 83 (10) (2003) 2085–2093.