Computers and Mathematics with Applications 60 (2010) 319–325

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

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

Improved modification direction methods

H.J. Kim[a,*], C. Kim[b], Y. Choi[a], S. Wang[c], X. Zhang[c]

[a] Department of Information Management and Security, Korea University, Seoul, 136-701, Republic of Korea
[b] Department of Computer Science and Engineering, Sejong University, Seoul 143-747, Republic of Korea
[c] School of Communication and Information Engineering, Shanghai University, Shanghai 200072, China

ARTICLE INFO

Keywords:
Data hiding
Number theory
Steganography

ABSTRACT

The original exploiting modification direction (EMD) method proposed by Zhang and Wang is a novel data hiding technique which can achieve large embedding capacity with less distortion. The original EMD method can hide $(2n + 1)$-ary numbers by modifying at most one least-significant bit (LSB) of n pixel values. The proposed methods in this paper, 2-EMD and EMD-2, modify at most two pixels of the LSB values. Efficiency of the proposed methods is shown theoretically and through experiments. The 2-EMD and EMD-2 can hide even larger numbers than the EMD with similar distortion under the same conditions. This paper shows that the EMD-2 is much better than the EMD, and slightly better than 2-EMD when n is 3, 4 and 5. The way to generate basis vector can be used for the generalization of the n-EMD and EMD-n where $n > 1$.

1. Introduction

Two important issues of data hiding techniques are preserving good image quality and increasing the embedding capacity altogether at the same time. However, this is an irreconcilable requirement. If we try to reduce image distortion, we have to sacrifice the embedding capacity. If we increase embedding capacity, image quality gets worse. The first generation approach has modified the least-significant bit (LSB) values. This simple method is called LSB replacement technique. This method can embed as many bits as the total number of pixels. However, statistical analysis based on the chi-square test using neighboring pixel value pairs can detect the presence of a hidden message [1]. Two values whose binary representations differ only in the LSB level are called a pair of values. For example, two consecutive numbers – 70 (i.e., 01000110 in binary representation) and 71 (i.e., 01000111 in binary format) – are a pair of values. In general, the frequencies of two neighboring values are statistically rarely equal in number. However, after the LSB replacement embedding, observation of Westfeld and Pfitzmann [1] conclude that most of their frequencies are getting closer. If the message to be hidden is really random, the frequencies of the pairs become nearly equal after embedding message due to its true randomness. If the message is not random, the pair of values may be normal and does not give us any hint. However, practitioners do not want to hide plain text. They believe that hiding plain text is more dangerous than cipher text. As a conclusion, the chi-square test is an effective technique against LSB modification methods.

Therefore, reducing the embedding capacity becomes an alternative solution to reduce image degradation. Tseng et al. [2] hide as many as $\lfloor \log_2(mn + 1) \rfloor$ bits of data in an $m \times n$ binary image block by changing at most two bits in the block. Matrix encoding technique in F5 algorithm [3] changes at most one LSB value to embed k bits into p pixels where $p = 2^k - 1$. Thus, this encoding technique uses a $(1, p, k)$ Hamming code. Modified matrix encoding technique [4] uses a $(2, p, k)$ Hamming

[*] Corresponding author. Tel.: +82 2 3290 4895.
E-mail addresses: khj-@korea.ac.kr (H.J. Kim), mipsan@paran.com (C. Kim), ciechoi@korea.ac.kr (Y. Choi), shuowang@shu.edu.cn (S. Wang), xzhang@shu.edu.cn (X. Zhang).

0898-1221/$ – see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2010.01.006

Table 1
Numbers generated by a basis vector [1, 2] when n is 2 in the EMD.

	1	2
0	0	0
1	1	0
2	0	1
3	0	−1
4	−1	0

code to modify at most two pixels. This modified matrix encoding technique allows more degree of freedom than the original matrix encoding technique.

Zhang and Wang [5] has proposed a novel data hiding technique to transform the binary secret data into a stream of secret digits using a $(2n+1)$-ary notational system. Their embedding method called exploiting modification direction (EMD) uses n cover pixels to carry one secret digit in the $(2n+1)$-ary number system. The maximum possible error of the modified pixel is ± 1 because their scheme changes only one LSB value. The pixel segmentation system proposed by Lee et al. [6] can hide more large numbers by modifying two pixel values. However, image quality gets considerably degraded and worse than 8 dB.

Two new EMD methods proposed in this paper, 2-EMD and EMD-2, are very simple to implement. The embedding capacity of these methods is larger than the pixel segmentation method, and much larger than the EMD. However, the average image quality of the EMD-2 is around 52 dB which is 8 dB higher than the pixel segmentation method, but similar to the EMD and 2-EMD. The efficiency of the 2-EMD and EMD-2 is compared with the EMD under the same condition.

2. EMD embedding method

The EMD method proposed by Zhang and Wang [5] is a novel method for hiding data. Each secret digit in a $(2n+1)$-ary notational system is carried by n cover pixels, where $n \geq 2$, and at most one pixel value is increased or decreased by 1 in the EMD method. A group of pixel values is represented as a vector G, where $G_n = [g_1, g_2, \ldots, g_n]$. A vector $[g_1, g_2, \ldots, g_n]$ in an n-dimensional space is mapped to a value f, which is computed by Eq. (1) as a weighted sum modulo $(2n+1)$:

$$f(g_1, g_2, \ldots, g_n) = \left[\sum_{i=1}^{n}(g_i \cdot i)\right] \bmod (2n+1). \tag{1}$$

No modification is needed if a secret digit d equals the extraction function f of the original pixel group. When the secret data d is not equal to f, we calculate $s = d - f \bmod (2n+1)$. If s is not larger than n, we increase the value of g_s by 1; otherwise, we decrease the value of g_{2n+1-s} by 1. Eq. (1) can be represented as an inner product between an image pixel value vector G_n and a basis vector $B_n = [1, 2, \ldots, n]$ such as

$$f(g_1, g_2, \ldots, g_n) = G_n \cdot B_n^T \bmod (2n+1). \tag{2}$$

The basis vector for EMD can be easily derived since only one pixel value is changed. Consider a case where n is 2. The basis vector B_2 is given as [1, 2]. Note that the number 0 can be generated by nullifying the basis vector (see Table 1) such as $0 = (0) \cdot 1 + (0) \cdot 2$. The number 1 is generated by setting the associated element 1 in the basis vector (i.e., b_1) by 1 and also nullifying the basis vector element 2 (i.e., b_2) such as $1 = (1) \cdot 1 + (0) \cdot 2$. The coefficients for the basis vector for the number 1, C_1, is [1, 0]. On the other hand, 3 can be generated by resetting the first element by 0 and setting the second element by -1 and taking modulus 5 based on Eq. (2). Thus, C_3 is $[0, -1]$. In other words, $(0) \cdot 1 + (-1) \cdot 2$ is -2, but $[(-2) \bmod 5]$ becomes 3. It is obvious that all five numbers from 0 to 4 can be generated according to Eq. (2) by the linear combination of the basis elements with their associated coefficients. Note that the five numbers are uniquely decided since we can choose 1 or -1 only once for each case.

3. The proposed EMD-2 scheme

In this section, we shall present our data hiding scheme based on $(2w+1)$-ary notational system in a group of pixels. The proposed data hiding scheme is composed of the embedding and extracting procedures, which are described below. This paper proposes a novel steganographic embedding method, EMD-2, that fully exploits the modification directions by allowing at most two pixels to be modified. In this method, modifications in different directions are used to represent different secret data, leading to a higher embedding efficiency. For the EMD-2 embedding method, the basis vector should be generated first.

3.1. Basis vector

The EMD-2 allows at most two pixels to be modified. Since one more pixel value is changed compared with the EMD, that changes at most one pixel value, the numbers generated by the new basis vector should be larger than that of the EMD. The

Table 2
Numbers generated by a basis vector [1, 3] when n is 2 in the EMD-2.

	1	3
0	0	0
1	1	0
2	−1	1
3	0	1
4	1	1
5	−1	−1
6	0	−1
7	1	−1
8	−1	0

Table 3
Numbers generated by a basis vector [1, 2, 6] when n is 3 in the EMD-2.

	1	2	6
0	0	0	0
1	1	0	0
2	0	1	0
3	1	1	0
4	0	−1	1
5	−1	0	1
6	0	0	1
7	1	0	1
8	0	1	1
9	0	−1	−1
10	−1	0	−1
11	0	0	−1
12	1	0	−1
13	0	1	−1
14	−1	−1	0
15	0	−1	0
16	−1	0	0

best choice of the basis vector with $n = 2$ is given as $B_2 = [1, 3]$ in the EMD-2. Note that Table 2 shows only nine numbers generated from 0 to 8. The number 2 is generated by a linear combination of 1 and 3 as $2 = (-1) \cdot 1 + (1) \cdot 3$. Negative numbers are turned into positive numbers by the modulus operation.

The best choice of the basis vector with $n = 3$ is $B_3 = [1, 2, 6]$ in the EMD-2. This basis vector can generate numbers from 0 to 16. The maximum positive number, 8, is generated as $8 = (0) \cdot 1 + (1) \cdot 2 + (1) \cdot 6$ by a linear combination. Thus, we can have eight positive numbers, also eight negative numbers, and the number 0 as a zero element. Thus, 17-ary numbers can be generated (see Table 3). It is easy to show that B_4 is [1, 2, 6, 11] for 27-ary numbers, B_5 is [1, 2, 6, 11, 16] for 37-ary numbers, and so on. Therefore, as a generalization, B_n becomes $[1, 2, 6, \ldots, 6 + 5(n - 3)]$ where $n > 2$.

3.2. Embedding procedure

In the EMD method, at most one bit is increased or decreased in a group of pixel values. Similarly, it is possible to increase or decrease two pixel values in a group. In order to hide one more secret bit than the EMD method, we need to formulate Eq. (3), whose role is similar to Eq. (1). Each basis element b_i is used as an input to the extraction function f as weighted sum modulo $(2w + 1)$. The value of w is determined as follows. When n is 2, w is 4; when n is larger than 2, it is formulated as $8 + 5(n - 3)$. Tables 2 and 3 are used to encode in the proposed $(2w + 1)$-ary system. Eqs. (3), (4) and (5) provide the basic structures for the EMD-2. The extraction function, basis elements, and the modulus value are the keys of the new embedding procedure.

$$f(g_1, g_2, \ldots, g_n) = \left[\sum_{i=1}^{n} (g_i \cdot b_i)\right] \bmod (2w + 1), \tag{3}$$

where

$$[b_1, b_2, \ldots, b_n] = \begin{cases} [1, 3] & n = 2 \\ [1, 2, 6, 11, 16, 21, \ldots, 6 + 5(n - 3)] & n > 2 \end{cases} \tag{4}$$

and

$$w = \begin{cases} 4 & n = 2 \\ 8 + 5(n - 3) & n > 2. \end{cases} \tag{5}$$

The EMD and EMD-2 methods are equivalent to a syndrome coding technique such as Hamming code or BCH code. The value s is computed by Eq. (6).

$$s = \begin{cases} d - f & \text{if } d \geq f \\ (2w + 1) - |d - f| & \text{if } d < f \text{ and } n > 2 \\ 4 - |d - f| & \text{if } d < f \text{ and } n = 2. \end{cases} \tag{6}$$

The modified pixel value vector G'_n can be computed by Eq. (7).

$$G'_n = G_n + C_s, \tag{7}$$

where C_s is the coefficient vector for the number s associated with the basis vector B_n. Let d be the data to be hidden. How to use the value s for data embedding is summarized by the following steps.

Step 1: Calculate f value with G_n and the associated basis vector B_n according to Eq. (3).
Step 2: Compute the value s according to Eq. (6).
Step 3: Find the associated coefficient vector C_s and compte the modified pixel value vector G'_n as $G'_n = G + C_s$ according to Eq. (7).

The following example illustrates how this embedding procedure works.

Example 1. Assume that the host signal vector G_3 is given as [10 5 3]. Since the dimension of the vector n is 3, we can use Eqs. (3)–(5) to compute f value or Table 3. As a result, the associated basis vector B_3 is given as [1 2 6] shown in Table 3.

Step 1: Calculate f value with G_3 and the associated basis vector [1 2 6]. In this case, f value is 4 according to Eq. (3). Inner product of [10 5 3] and [1 2 6] is 38, and 38 mod 17 is 4.
Step 2: Since d is larger than f (i.e., $5 > 4$), the value of s is 1 according to Eq. (6).
Step 3: Look up the Table 3 and find the row for the number 1 which is [1 0 0]. In other words, its coefficient vector C_1 is [1 0 0]. Thus, according to Eq. (7), the modified pixel value vector becomes [11 5 3].

3.3. *The extracting procedure*

In order to extract the secret message from the stego image, we need to calculate the f value according to Eq. (3) using a group of modified pixel values. That is, f is just the embedded message in the stego image. The embedding procedure is organized to ensure that f is just the embedded message. Thus, the extraction procedure is very simple. The encoder allows the decoder to recover the hidden message just by computing the value f.

4. The proposed 2-EMD scheme

The 2-EMD scheme embeds data using two consecutive EMDs. Thus, basic properties of the 2-EMD are the same as EMD. Of course, there are slight differences between EMD and 2-EMD. In case of EMD, we can hide two $(2n + 1)$-ary numbers consecutively by flipping at most one pixel value among n pixel values. On the other hand, we can hide one $((2n + 1)^2)$-ary number by flipping at most one pixel value among each n pixel values in case of 2-EMD. Since the EMD can hide one $(2n + 1)$-ary number into n pixel values, two consecutive EMDs, (e.g., 2-EMD), can hide one $((2n + 1)^2)$-ary number into $2n$ pixel values. In this case, we need a simple number representation computation for 2-EMD. First EMD can hide a $2n = 1$-ary number N_1, and second EMD can hide another $2n + 1$-ary number N_2. In this case, the number N for 2-EMD can be computed as $N = (2n + 2)N_1 + N_2$, where $0 \leq N_1, N_2 \leq 2n + 1$ and $0 \leq N \leq (2n + 1)^2 - 1$. For convenience's sake, we consider 2-EMD with two blocks, both having n pixel values.

5. Experimental results

The purpose of our scheme is to embed secret data into a cover image such that the stego image still maintains a good image quality. We apply the 2-EMD and EMD-2 methods to various standard gray scale test images.

Table 4 shows the result of the 2-EMD and EMD-2. Each EMD can change at most one pixel value. Thus, up to two pixel values can be changed by applying the EMD twice for 2-EMD. In the first stage, the EMD is applied to a block of n_1 pixel values. Next, the same EMD is applied to a block of n_2 pixel values. Those blocks are not overlapping each other. On the other hand, the EMD-2 is applied to a block of n pixel values, where $n = n_1 + n_2$. Therefore, EMD-2 can change up to two pixel values in a block of n pixel values. As a result, both methods can modify up to two pixel values in a block.

For example, EMD-2 with 4 pixels can hide a 27-ary number, while 2- EMD with two pixels each hide a 25-ary number. One EMD with two pixels can hide a 5-ary number. Therefore, 2-EMD can hide a 25-ary number. When n is small, EMD-2 is much more efficient than 2-EMD. If we can find a better basis, the EMD-2 can be improved further.

Table 4
Maximum numbers to be hidden using 2-EMD and EMD-2.

2-EMD			EMD-2	
n_1	n_2	Number system	n where $n = n_1 + n_2$	Number system
1	1	9-ary (3 × 3)	2	9-ary
1	2	15-ary (3 × 3)	3	17-ary
2	2	25-ary (5 × 5)	4	27-ary
2	3	35-ary (5 × 7)	5	37-ary
3	3	49-ary (7 × 7)	6	47-ary
3	4	63-ary (7 × 9)	7	57-ary
4	4	81-ary (9 × 9)	8	67-ary

In general, the evaluation of data hiding performance depends on the visual quality of stego image and data hiding capacity. Data hiding capacity is defined as the amount of data that can be hidden under the data hiding mechanism. Zhang and Wang's EMD method [5] achieves the embedding rate R given as follows:

$$R = \log_2((2n+1)/n). \tag{8}$$

Therefore, the best embedding rate R of the EMD is achieved when n is 2 since R is a monotonically decreasing function. Its best embedding rate is $\log_2(3/1)$. It means that the optimal embedding rate is achieved when only one pixel is used. The embedding rate of EMD-2, R_2, is given as follows:

$$R_2 = \log_2((2w+1)/n). \tag{9}$$

Note that R in Eq. (8) converges to 0. On the other hand, the embedding rate function R_2 of n is a monotonically increasing function, which converges to $\log_2(10)$. When w is 4 or n is 2 in Eq. (5), the EMD-2 achieves the embedding rate as $\log_2(9/2)$. The embedding rate of 2-EMD, ${}_2R$, is given as follows:

$${}_2R = \log_2((2n_1+1)(2n_2+1)/n). \tag{10}$$

Note that ${}_2R$ in Eq. (10) converges to ∞.

Now, assume that we modify up to two LSB values among four pixels. In this case, the EMD-2 achieves the embedding bit rate $\log_2(27/4)$ or 2.75. If 2-EMD for each group of two pixels can achieve the rate $\log_2(25/4)$ or 2.64 (see Table 4). Therefore, the embedding rate of the EMD-2 is 1.04 (i.e., 2.75/2.64) times larger than 2-EMD. If we try to modify up to two LSB values using a group of five pixels, the EMD-2 can hide a 37-ary number, while the 2-EMD can hide a 35-ary number. Thus, EMD-2 still hides a larger number. However, if we try to use a group of six pixel values, the EMD-2 can hide a 47-ary number, while the 2-EMD can hide a 49-ary number. Thus, it is obvious that the EMD-2 is better than 2-EMD when the number of pixels is less than six.

Nine gray level images, Lena, Baboon, Tiffany, Pepper, Goldhill, Barbara, Boat, Zelda, and Bridge, each of 512 × 512 and 256 × 256 pixels, are used as the cover images for image quality comparisons. The peak signal to noise ratio (PSNR) is used in this paper to evaluate the image quality. The PSNR of a gray level image is defined as follows:

$$\mathrm{PSNR} = 10 \cdot \log_{10} \frac{255^2}{\mathrm{MSE}}\ \mathrm{dB}. \tag{11}$$

The mean squared error (MSE) for an $N \times N$ gray scale image is defined as follows:

$$\mathrm{MSE} = \left(\frac{1}{N}\right)^2 \sum_{i=1}^{N}\sum_{j=1}^{N}(x_{ij} - \bar{x}_{ij})^2. \tag{12}$$

Here x_{ij} denotes the original pixel value, and $\bar{x}_{ij}$ denotes the modified pixel value at position (i, j). Fig. 1 shows that image quality is very good and average PSNR values of these images are 52.03 dB when n is 4.

Fig. 2 shows the histograms of the original Lena image and the EMD-2 method. One of the concerns is the fear of the chi-square test that is used to see if the pair of pixels show any evidence of the presence of hidden data. Fig. 2 shows that no clues about data hiding can be found. Compared with the original Lena histogram, minor differences are inevitable. Therefore, a blind test may hardly find such differences. This histogram shows the frequency of pixel values between 50 and 100 only to highlight the differences visually and clearly. Left- and right-hand side bars are histograms of the original image and its modified image using EMD-2, respectively.

The pixel segmentation method [6] can be represented in a different way. This method is reformulated as follows. Assume that two pixels are allowed to be modified up to two bit-depth. This case is equivalent to modifying up to four LSBs of four pixel values. This approach uses three bits to represent the magnitude of the number to be hidden, and one bit for the sign of the number. Thus, this system can hide 15-ary numbers. Apparently, this 15-ary number is larger than a 9-ary number using the EMD. However, the EMD modifies at most one LSB value. Thus, comparison of the EMD with the pixel segmentation approach is not fair. Instead of spreading the bit-planes, the pixel segmentation method stacks the bit-planes to increase the bit rate such as $\log_2(15/2)$ or 2.91. However, maximum expected error is ±3 because of the two bits of depth changes. If

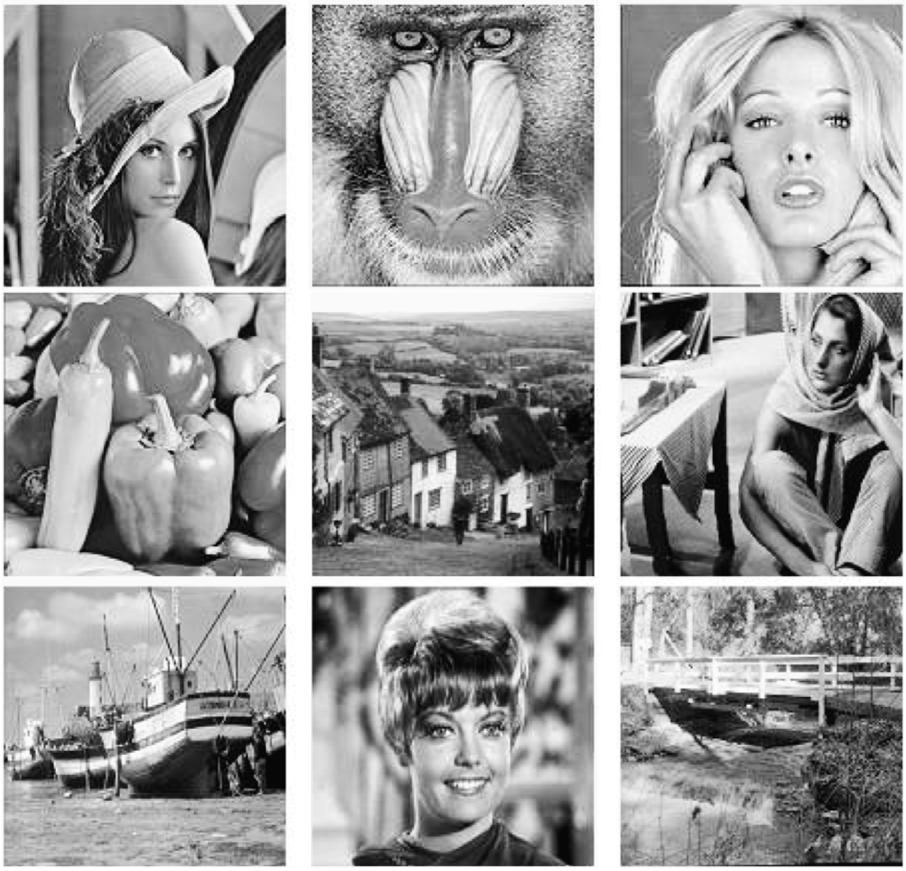

Fig. 1. Images after data embedding using EMD-2 using Lena, Baboon, Tiffany, Pepper, Goldhill, Barbara, Boat, Zelda, and Bridge images.

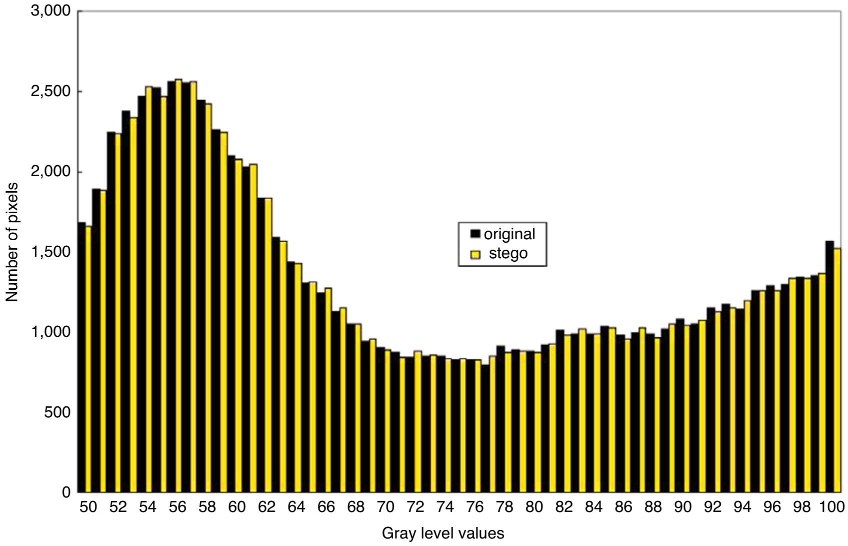

Fig. 2. The histogram of the original Lena image (left-hand side bars) and its modified images using the EMD-2 (right-hand side bars).

we spread the bit-depths over four pixel values, the bit rate is equivalent to $\log_2(15)/4$, which is still larger than $\log_2(9)/4$, the rate of the EMD, but worse than that of 2-EMD with $\log_2(25)/4$ and much worse than that of EMD-2 with $\log_2(27)/4$.

Fig. 3 shows that three Lena images look almost the same. Because both 2-EMD, and the EMD-2 modify small number of LSB values, image quality is very high after data hiding and easy to pass steganalysis. Zhang and Wang [5] already prove that

Fig. 3. Original (Left), EMD stegoed (Center), and EMD-2 stegoed Lena images.

the EMD is robust against steganalysis. 2-EMD is exactly the same as the EMD in mechanism and the EMD-2 is a variant of the EMD. Therefore, we can say that EMD-2 is as secure as the EMD and 2-EMD.

One remarkable fact found in this paper is the embedding rate of 2-EMD and EMD-2 compared with EMD. In case of EMD, the embedding rate *R* is monotonically decreasing and a maximum is obtained when only one pixel is used. Using one pixel in EMD is better than LSB (least-significant bit) modification because the embedding rate of the former is 1 (i.e., $\log_2(2/1)$) while that of the latter is 1.585 (i.e., $\log_2(3/1)$). The embedding rate functions of 2-EMD and EMD-2 are monotonically increasing. An optimal embedding rate is achieved with EMD-2 for up to five pixels and with 2-EMD for more than five pixels.

6. Conclusion

In this paper, we proposed an image hiding scheme based on $\log_2(2(5n-7)+1)$-ary number system, where n is larger than 2. The proposed scheme can be used to embed and extract secret data in an image systematically. The minimum bit rate is $\log_2(9/2)$ or 2.17 and the maximum rate is $\log_2(10)$ or 3.32. This approach is better than the EMD [5] method in terms of the embedding bit rate when n is any positive number. In addition, we proposed a variant of EMD, the 2-EMD, which is the same as the EMD. The 2-EMD is better than EMD-2 when the number of pixels used is 3, 4, and 5. The EMD-2 is always better than the pixel segmentation method [6] for all n. The image quality of 2-EMD is the same as that of EMD, and is slightly better than EMD-2, but much better than the pixel segmentation method. Theoretically and experimentally these facts have been demonstrated in this paper. By comparing 2-EMD and EMD-2, it is obvious that it is crucial to select a good basis. The way to generate a basis vector can be used for the generalization of the proposed method for EMD-n, where $n > 2$. n-EMD with $n > 1$ also deserves attention.

Acknowledgements

This work was in part supported by the CT R&D program of MCST/KOCCA [2-09-1205-001-10987-11-001, Development of Broadcasting Content Distribution Service Framework Technology using National Standard Content Identification], the IT R&D program of MKE/KEIT [2008-F-036-02, Development of Anonymity-based u-knowledge Security Technology], and the CT R&D program of MCST/KOCCA [2-09-1205-001-10987-11-001, Development of Broadcasting Content Distribution Service Framework Technology Using National Standard Content Identification], and the CTRC program.

References

[1] A. Westfeld, A. Pfitzmann, Attacks on steganographic systems, Lecture Notes in Computer Science 1768 (1999) 61–76.

[2] Y.-C. Tseng, Y.-Y. Chen, H.-K. Pan, A secure data hiding scheme for binary images, IEEE Transactions on Communications 50 (8) (2002) 1227–1231.

[3] A. Westfeld, F5: A steganographic algorithm, Lecture Notes in Computer Science 2137 (2001) 289–302.

[4] Y. Kim, Z. Duric, D. Richards, Modified matrix encoding technique for minimal distortion steganography, Lecture Notes in Computer Science 4437 (2008) 167–176.

[5] X. Zhang, S. Wang, Efficient steganographic embedding by exploiting modification direction, IEEE Communications Letters 10 (11) (2006) 781–783.

[6] C.-F. Lee, C.-C. Chang, K.-H. Wang, An improvement of EMD embedding method for large payloads by pixel segmentation strategy, Image and Vision Computing 26 (12) (2008) 1670–1676.