Signal Processing 90 (2010) 2954–2964

Contents lists available at ScienceDirect

Signal Processing

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

An improved section-wise exploiting modification direction method

Jianjun Wang [a,*], Yiting Sun [a], Huan Xu [a], Kangkang Chen [a], Hyoung Joong Kim [b], Sang-Hyun Joo [c]

[a] Department of Electronic Engineering, Fudan University, Shanghai 200433, China
[b] CIST, Korea University, Seoul 136-701, Republic of Korea
[c] ETRI, Daejon 305-700, Republic of Korea

ARTICLE INFO

Article history:
Received 28 September 2009
Received in revised form
22 April 2010
Accepted 23 April 2010
Available online 6 May 2010

Keywords:
Steganography
Exploiting modification direction
Chaos

ABSTRACT

In this paper, a novel section-wise exploiting modification direction (EMD) method is proposed. By using section-wise strategy, the proposed method combines several pixel groups of the cover image together to indicate adjusting modification directions. In each group, the data hiding is performed by the EMD method. Theoretically, we have proved that the modification directions of our section-wise approach is far from the EMD method. The experimental results show that the section-wise strategy can improve the embedding efficiency and the visual quality further than the EMD method, and reduce the possibility of detection.

1. Introduction

In recent years, the Internet has offered great convenience in the transmission of a large amount of data. In order to ensure the security of the data transmission over the Internet, data encryption and information hiding are two widely used techniques [1–3]. Data encryption is a technique of protecting data from illicit access by transforming original secret data into meaningless form, which can arouse the attention of interceptors. Nevertheless, information hiding differs from data encryption in that it embeds the secret data into a meaningful host medium to distract the attention of observers.

Steganographic schemes [4,5] and watermarking methods [6,7] are two main branches of information hiding. In this paper we just focus on steganography. Steganographic schemes hide secret information in cover carriers, so that the existence of secret information is undetectable. Actually, steganography may be used to protect personal privacy, business activity, and national security through a covert channel. On the other hand, it may be misused for crime. For example, terrorists, criminals, and other hostile entities may use steganography to conceal the planning and coordination of their illicit activities. Thus, it raises the concerns of those who wish to prevent such unlawful communications. The technology devoted to defeating steganography is known as steganalysis. The prime goal of steganalysis is to detect the presence of steganography [8]. Because of this reason, steganography pays more attention to the visual quality, the statistical imperceptibility, the capacity of embedded data, and the resistance against detection. There are many kinds of digital media which can be used as cover carriers in steganogarphic scheme, such as text, image, audio, and video. Because of insensitivity of human visual system, digital images have been widely used as cover carriers in most steganographic schemes [9].

Steganography in image can be performed in both spatial domain and transform domain. As it is difficult to extract robust features of stego image in spatial domain, steganography in spatial domain is still researched widely, despite its robustness is much awful. In recent years, many researchers are more enthusiastic to improve the embedding efficiency and decrease the possibility of detection. Least-significant-bit (LSB) matching is the

* Corresponding author. Tel.: +86 21 65642142; fax: +86 21 55664041.
E-mail address: wangjj@fudan.edu.cn (J. Wang).

0165-1684/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.sigpro.2010.04.022

conventional efficient steganography method, and it is proved much more difficult to detect than simple LSB replacement. In 2006, Mielikainen [10] developed a pair-wise LSB matching method to improve the embedding efficiency. Zhang and Wang [11] also proposed the exploiting modification direction (EMD) method, which fully exploits the modification directions, so that it leads to a higher embedding efficiency. The EMD method achieves higher embedding efficiency than many other existing spatial domain steganographic methods [12,13]. Recently, many researchers have proposed different kinds of schemes to improve the EMD method [14–29]. In these methods, some combine two or more different codes together to improve the efficiency, or perform double-layer embedding, or disobey the rule of the EMD that allows only one pixel to be changed by another in each group of pixels [23–27]. Others also use some optimization methods to further improve the imperceptibility [28]. In 2008, Lee et al. [29] improved the EMD method for large payloads, which rearranges the two pixels in a group into the vector of coordinate area (VCA) and the vector of modification area (VMA). Although the payloads are enlarged, there exist 3 bits changed out of 16-bit of two pixels in a group, which is far worse than only 1 bit changed out of 16-bit of two pixels in the EMD method. In other words, the large payloads are obtained at the cost of the visual quality of the stego image.

In this paper, we will also propose an efficient improvement to the EMD method. Different from other existing improved algorithms, our proposed method neither uses double-layer embedding nor combines several codes in one, but optimizes the EMD with section-wise strategy, which is similar as the segmentation management of memory in computer architecture. The experimental results show that the proposed method not only leads to higher embedding efficiency than the EMD, but also gains better visual quality and lowers the possibility of detection.

The rest of this paper is organized as follows. Section 2 introduces the EMD method, and other improved EMD methods. The proposed method is described in Section 3. In Section 4, experimental results are given to show the performance of the proposed method compared with the EMD. Finally, the conclusions are presented in Section 5.

2. Related work

In this section, the EMD method and two improved EMD methods, which were proposed in 2009, are introduced briefly. They are Chao et al.'s method [21] and Jung and Yoo's method [22].

2.1. The EMD method

The exploiting modification direction (EMD) steganographic scheme is proposed by Zhang and Wang [11]. A data hider first converts the secret message into a sequence of $(2n+1)$-ary secret digits, so that each secret digit falls into the region $[0,2n]$. Then all the pixels of the host image are permutated randomly and partitioned into groups, making each group consist of n pixels. The grayscale values of the n pixels in a group are denoted as $g_1,g_2,\ldots,g_n$, and the $(2n+1)$-ary secret digit as d. The embedding function f as a weighted sum modulo $(2n+1)$ is calculated as

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

They proved that the vector $[g_1, g_2, \ldots, g_n]$ in the n-dimension space may be represented by a unit hyper-cube. The f values of the hyper-cube and its $2n$ neighbors are mutually different. It means that each embedding direction is different from others, i.e., each embedding direction is unique. Thus, the embedded $(2n+1)$-ary secret digits can be extracted correctly.

After calculating the value of f, we compare its value with the value of the secret digit d. If f is equal to d, no modification is needed in this group. Otherwise, we calculate their difference value $s=|d-f|\bmod(2n+1)$. If s is not greater than n, then g_s is increased by 1. If s is greater than n, then g_{2n+1-s} is decreased by 1. If the adjusted pixel value is outside the region [0,255], then the corresponding group will be readjusted, so that any pixel value is within the range. The extraction procedure is quite simple: the value of f of stego pixel group is the hidden secret $(2n+1)$-ary digit. The EMD method provides high PSNR value. At most one pixel in the cover pixel group is modified. It can also achieve high embedding efficiency because it uses n pixels in a group to represent $(2n+1)$ different modification directions.

2.2. Chao et al.'s method

Chao et al. [21] proposed an improved EMD method by diamond encoding. The proposed method can hide more secret data than the EMD method while keeping the stego-image quality degradation imperceptible.

In this method, the neighborhood set $D_k(i,j)$ of four pixel values x, y, i, and j is constructed by

$$D_k(i,j)=\{(x,y)\quad |i-x|+|j-y|\le k\}. \quad (2)$$

where k is a positive integer. The neighborhood set D_k contains all the vectors (x,y) with the distance to vector (i,j) smaller than k. The number of elements of the set D_k, l, is calculated by $l=2k^2+2k+1$, and each member in D_k is called neighboring vector of (i,j). One can use parameter k to calculate the value of l and obtain the embedding position. Chao et al.'s diamond encoding method uses a diamond function f to compute the diamond characteristic value (DCV) in embedding and extraction procedures. The DCV of two pixel values i and j is calculated by

$$f(i,j)=((2k+1)\times i+j)\ \mathrm{mod}\, l. \quad (3)$$

where l is the number of elements of the set D_k. From the definition of the DCV, it is easy to find that the DCV of the vector (i,j), the member of D_k, belongs to $\{0,1,2,\ldots,l-1\}$, and any two DCVs of vectors in $D_k(i,j)$ are distinct. M_k is denoted as the embedded digit, which belongs to $\{0,1,2,\ldots,l-1\}$. For secret data embedding, one can replace the DCV of the vector (i,j) with the embedded secret digit. Therefore, the modulus distance between $f(i,j)$ and D_k is $P_k=f(i,j)-M_k \bmod l$. For each k, one can

design a distance pattern P_k to search which neighboring pixel owns the modulus distance P_k. Then, the vector (i,j) is replaced with the neighboring vector (i',j') by P_k. The vector (i',j') the member of $D_k(i,j)$ and the DCV of (i',j') equals to the embedded secret digit M_k. The vector (i',j') can be used to extract the correct secret digit by

$$f(i',j') = ((2k+1)(i'+j'))\mathrm{mod}l. \quad (4)$$

A detailed description of Chao et al.'s method can be found in [21].

2.3. Jung and Yoo's method

Jung and Yoo proposed an improved EMD method [22]. In their method, one cover pixel can carry each secret digit in a $(2n+1)$-ary notational system. The proposed method achieves two times the capacity of the EMD method.

The embedding procedure is as follows.

For each cover pixel value, g_i, the function value f is calculated by

$$f = (g_i + x)\mathrm{mod}(2n+1). \quad (5)$$

where $|x| \le n$. If the value of a pixel, g_i, belongs to set $\{0, 1\}$ or $\{254, 255\}$ for each case, then x is chosen from set $\{0,1,2, \ldots, 2n\}$ or $\{-2n, -2n+1, -2n+2, \ldots, -2, -1, 0\}$.

A stego pixel value, g_i', is obtained by

$$g_i' = g_i + x. \quad (6)$$

where the selected value of x satisfies the condition $f=d$, where d is an n-ary secret digit.

The extracting procedure is as follows.

The extraction method is very easy. A secret digit d can be obtained by

$$d = g_i'\mathrm{mod}(2n+1). \quad (7)$$

A detailed description of Jung's method can be found in [22].

3. The proposed method

The success of the EMD method is that it uses only n pixels to represent $(2n+1)$ different directions. The embedding efficiency and rate can be calculated as follows:

$$E = (2n+1)\log_2 \frac{(2n+1)}{2n}. \quad (8)$$

$$R = \log_2 \frac{(2n+1)}{n}. \quad (9)$$

where n is the number of pixels in a cover pixel group.

It is clear that a larger n leads to a smaller embedding rate according to Eq. (9). In other words, the EMD method has its maximum embedding rate when $n=2$. The equation of the embedding efficiency E describes the ratio between the number of embedded bits and the distortion energy caused by the embedding procedure. It means that one can achieve higher embedding efficiency E by representing more bits with fewer changes. In the EMD method, Zhang and Wang assume that the redundancy rate is zero when converting binary stream into $(2n+1)$-ary digits and the cutoff length is large enough. The cutoff length indicates the number of binary digits, which are converted to $(2n+1)$-ary digits. They show an example in which the cutoff length is 4, i.e. 4 bits are chosen from the binary secret message and converted to $(2n+1)$-ary digits at a time, and then the following 4 bits, and so on. If the cutoff length is large enough or even infinite, the redundancy does not exist when converting the binary stream into $(2n+1)$-ary stream. However, it is not practical. In image steganography, the length of the secret message is always finite, and it is far away from large enough to satisfy the assumption. Moreover, the procedure of converting binary stream into $(2n+1)$-ary digits become much complex with the growth of the length of binary stream. The conversion of a $2L$-length binary stream is much more complex than an L-length one. It involves much more complex computation including divisions and multiplications. Thus, in practice, a definite cutoff length is always required. For convenience, a short cutoff length is often used, for example, the cutoff length is 4 in Zhang and Wang's method [11].

In our proposed method, we do not change the cutoff length. Instead, we utilize the combination of two or more pixel groups to represent more embedding directions. The main difference between the EMD method and our method is as follows: first, the pixel groups in the EMD are independent of each other, each n-pixel group represents a segment of binary stream with the cutoff length; while in our method, two or more pixels groups are treated as a section to represent more embedding directions than these groups can do independently. Second, the total number of pixels in a section, n, can be converted into a $(2n+1)$-ary digit. This $(2n+1)$-ary digit is used to define an amount of SP or DP variation, i.e., the value of SP or DP is taken from 0 to $2n$. While the secure data is not converted into $(2n+1)$-ary digits. Using the section-wise strategy, no greater cutoff length is needed; thus, it avoids the decomposition of quite large numbers when achieving longer cutoff length. In this way, the proposed method can reach higher embedding efficiency and better performance than the EMD method.

3.1. Permutation

In steganography, permutation of the cover image and secret message is an important step. It will not only enhance the security performance of steganography, but also break the correlation of neighboring pixels to reach better statistical characteristics. A variety of traditional permutation methods have been utilized by many researchers. Nowadays, with higher requirements for security and statistical balance in steganography, many researchers are now focusing on the chaos-based or chaos-like permutation and encryption [30–32], because the chaos-based system manifests much better non-linear and random characteristics.

According to the chaos theory, the behavior of certain dynamical systems, whose state evolves with time, may exhibit dynamics that are highly sensitive to initial conditions. As a result of this sensitivity, which manifests itself as an exponential growth of perturbations in the

initial conditions, the behavior of chaotic systems appears to be random. Given the same initial value, exactly the same chaotic sequence can be generated. Chaos-like or chaos-based image transform has been proved to work better than conventional mapping method such as affine cipher and Arnold transform [30]. In our proposed method, we use one-dimensional logistic chaotic sequence to permute the pixel order.

One-dimensional logistic chaotic sequence is defined as follows:

$$x_{n+1} = 1-\mu x_n^2. \tag{10}$$

It is proved that when $\mu > 1.40{,}115$, the cycle of the sequence is infinite and the sequence will enter the chaos status [32]. In order to use the chaotic sequence to permute cover image and secret message, we first calculate the first $L+k$ elements of the sequence, where L is the length of the stream that needs to be converted and k the redundant elements. Then we abandon the first k elements and sort the remainder L elements. For example, we need a permutation sequence of length $L=1000$. Let the initial value x_0 be 0.1, the parameter μ be 1.5, and the redundant length $k=100$. From Eq. (10), we can get the chaotic sequence as $[0.1, 0.9850, -0.4553, 0.6890, \ldots]$. After calculating the first $L+k=1000+100$ elements of the chaotic sequence, we abandon the first 100 elements and sort the *remainder* 1000 elements. The order of the 1000 elements decides the permutation order. The order of the sorted elements is taken as the permutation key. The initial value x_0, μ, k, and the permutation key are needed when the receiver recovers the correct order.

3.2. Data embedding

In the EMD method, the cover pixels are divided into n-pixel groups, and a $(2n+1)$-ary secret digit is embedded into each group. Unlike the EMD, in our proposed method, we first divide the cover images into pixel sections. The number of pixels in each section can either be equal or not. Next, the section is divided into two groups, a selective group and a descriptive one. We call it 2-level section. A selector pointer (SP) and a descriptor pointer (DP) are assigned to a selective group and a descriptive one according to its level, respectively. The total number of pixels in the selective group or the descriptive group, n, can be converted into a $(2n+1)$-ary digit. This $(2n+1)$-ary digit is used to define an amount of SP or DP variation whose value varies from 0 to $2n$. Once the values of SP and DP are decided, we can build a table of SP by DP. We call it selector and descriptor table. In this table, the values of all items are the different binary streams which are decided by the cutoff length, L, i.e., each item is the secret message to be hidden. At the same time, each SP points to each row in the table, and each DP also points to each column in the table. Note that each row represents the modification directions of the selective group, and each column also represents the modification directions of the descriptive group. Therefore, a pair (SP, DP) can decide a binary stream to be embedded, and vice versa. Say, assume a section has n pixels, r is the number of pixels in the selective group, $n-r$ is the number of pixels in the descriptive group. To build the selector and descriptor table, the following conditions must be satisfied:

1) $n \geq 4$, it means that each group must have two pixels at least;
2) $(2r+1)(2(n-r)+1) \geq 2^L$, it denots that the table must have enough items. Because the table must contain all items which are 2^L different binary streams.

In order to compare the modification directions of the section-wise approach with the EMD method, theoretically, we assume two cover pixel groups, and each of them consists of n pixels. Because the modification directions depend on the secret data to be embedded. We can regard it as a problem of all permutations in mathmatics. In the EMD method, each group can represent $2n+1$ different modification directions. All the permutations of two group are $(2n+1)^2$, i.e., the different modification directions of the EMD method is $(2n+1)^2$. However, in our section-wise approach, selective group has n pixels, it can represent $(2n+1)$ different modification directions. Note that SP points to $2n+1$ rows, i.e., each row has $(2n+1)$ different modification directions. All the permutations of rows are $(2n+1)^{(2n+1)}$, similarly, the all permutations of columns is $(2n+1)^{(2n+1)}$; thus, the selector and descriptor table can represent $(2n+1)^{2(2n+1)}$ different modification directions. In this sense, we have proved that the modification directions of our section-wise approach is far from the EMD method.

As mentioned above, in the EMD, a cutoff length is always finite when converting the binary stream into a series of $(2n+1)$-ary digits. A binary number of the length L needs $\lceil \log_{2n+1} 2^L \rceil$ $(2n+1)$-ary digits to represent it. Thus, in the data embedding procedure, an appropriate cutoff length L needs to be decided first. In the section-wise method, cutoff length is decided by calculating the maximum directions that this kind of section can represent. The purpose is to make the redundancy as small as possible between the binary number and the $(2n+1)$-ary number.

For example, as $L=4$, a 4-bit binary number can represent $2^4=16$ different modification directions. In order to represent the 4-bit binary stream, two 5-ary digits are needed in the EMD. Each digit will be embedded into an n-pixel group. Thus, it needs 4 pixels, of which 2 pixels will be changed at most. However, in the section-wise approach, to embed the 4-bit binary number, we take 2-level strategy to build a selective group and a descriptive group, each group has 2 pixels. From the definition of the selector and descriptor table, we can know that the value of SP is taken from 0 to 4, so does DP. Thus, the selector and descriptor table could represent 25 different modification directions, which is greater than 16. The proposed method still needs 2 out of 4 pixel changes to represent the 4-bit binary number. It seems that the pixel change rates are almost the same. However, with increase in the cutoff length, our proposed method will work more efficiently than the EMD. Let the cutoff length $L=5$. The 5-bit binary number has $2^5=32$ different modification directions. In the EMD, this 5-bit binary

number is converted to a 5-ary number with 3 digits, so it needs 3 pixel groups to embed the 3 digits, and each pixel group has 2 pixels. It means 3 out of 6 pixels will be changed at most. In the section-wise strategy, if the value of SP is taken from 0 to 4 and DP from 0 to 6, the selector and descriptor table could represent 35 different modification directions. This time, in order to embed the 5-bit number, only 2 out of 5 pixels will be changed at most. So it works more efficiently than the EMD method.

The assumption we make here is that the cutoff length is finite when converting the binary stream into $(2n+1)$ notional system. Actually it is always finite in practice. When the binary bits are converted to $(2n+1)$-ary digits, there always exist redundancy. We can easily find the combination method of 2-level selectors and descriptors to reduce such redundancy. In this case, the redundancy can be decreased as little as possible so that the average numbers of the modification pixels in a group can be fewer. However, if the secret message is already in $(2n+1)$-ary notional system, and no convert is need, Zhang's EMD method is indubitable the best, there is no need to use our section-wise strategy.

After the selector and descriptor table is built, the embedding procedure is very easy. According to the data to be embedded, using table generated above, get the corresponding SP and DP, and then perform the EMD embedding procedure using Eq. (1). This process goes on and on until all secret data are embedded into cover image.

The entire embedding procedure of our proposed method is concluded as follows:

Step 1: Reshape the cover image and the secret message into streams.

Step 2: Determine the acceptable cutoff length L.

Step 3: Permute the cover image and the secret message by using chaotic sequence.

Step 4: Divide the pixel groups into 2-level of section. Assign the appropriate pixels to SP and DP, so that the whole section can represent more modification directions than the L-bit binary streams can do. The dividing method can be various and can be treated as part of the secret key.

Step 5: Construct selector and descriptor table in terms of SP and DP. In 2-level situation, SP and DP indicate the row and the column of the table, respectively. The cross point of each row and each column indicates an unique embedding direction.

Step 6: According to the data to be embedded, using table generated by step 5, get the corresponding SP and DP, and then perform the EMD embedding procedure using Eq. (1). This process goes on and on until all secret data are embedded into cover image.

Step 7: Reshape the stego stream to the original size and order of the cover image.

The flowchart of embedding procedure is shown in Fig. 1. An example will demonstrate our proposed method in detail as follows.

First, we convert the cover image and the secret message into streams by scanning the image from top to bottom, and left to right. Assume that the cover image is a 512×512 8-bit grayscale image and the secret message is a 512×512 binary image. Both the converted cover image

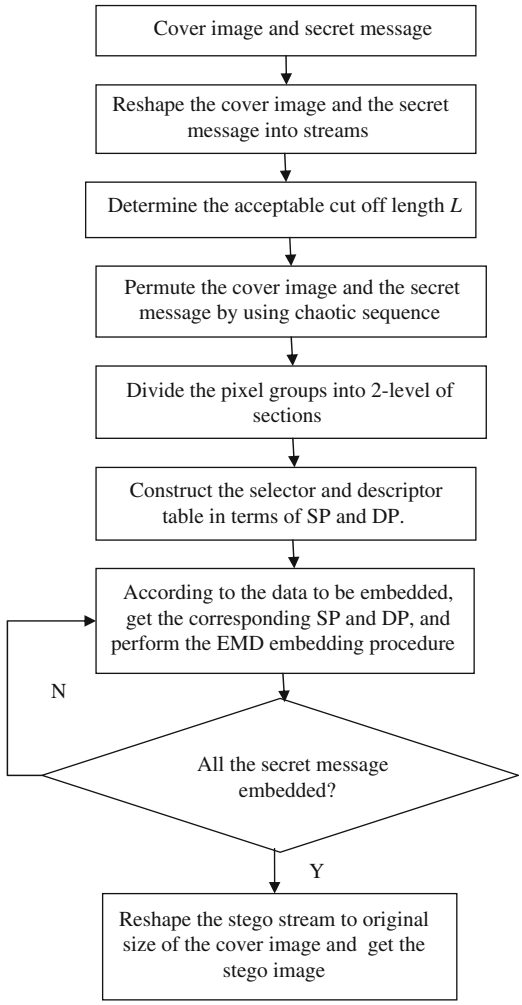

Fig. 1. Flowchart of data embedding procedure.

stream and secret message stream come to the length of 262,144. Then we determine a chaotic sequence with the initial values $x_0=0.1$ and $\mu=1.5$. Let the redundant length be 100. We calculate the first 262,244 elements of the sequence. Skipping the first 100 elements, we sort the following 262,144 elements and record their new orders. According to the new order, we rearrange the cover image stream. The secret message can be processed with the same procedure. The initial value x_0, μ, and the news orders are used as encryption keys and recorded for data extraction.

For convenience, let the cutoff length L be 5. Each time we embed 5 bits taken from the secret message stream into cover image stream. We choose SP=2 and DP=3. Thus, the selector SP has 5 different values and the descriptor DP has 7 different values. 35 different items can be filled in the 5×7 table in some kind of order which is decided by the data hider. For simplicity, the 32 different items, which 5 bits binary streams can represent are numbered in ascending way and arranged in the table from left to right, and top to bottom. As the total items of the table is 35, 3 positions are left empty. Each item in the

	DP=0	DP=1	DP=2	DP=3	DP=4	DP=5	DP=6
SP=0	00000	00001	00010	00011	00100	00101	00110
SP=1	00111	01000	01001	01010	01011	01100	01101
SP=2	01110	01111	10000	10001	10010	10011	10100
SP=3	10101	10110	10111	11000	11001	11010	11011
SP=4	11100	11101	11110	11111	Empty	Empty	Empty

Fig. 2. 5×7 selector and descriptor table.

table can be decided by choosing a unique SP and DP. The SP is numbered from 0 to 4, and DP is numbered from 0 to 6. The arrangement of the selector and descriptor table is shown in Fig. 2.

For SP=0, the secret message from 00000 to 00110 is arranged to DP from 0 to 6.
For SP=1, the secret message from 00111 to 01101 is arranged to DP from 0 to 6.
For SP=2, the secret message from 01110 to 10100 is arranged to DP from 0 to 6.
For SP=3, the secret message from 10101 to 11011 is arranged to DP from 0 to 6.
For SP=4, the secret message from 11100 to 11111 is arranged to DP from 0 to 3, and the other three available positions will not be used.

For example, if a section has 5 pixels (166,167,167,169,168), we can divide the section into a selective group (166,167), and a descriptive group (167,169,168). The 5-bit binary streams is (01001). In this example, SP points to the selective group (166,167), and DP points to descriptive group (167,169,168). The value of SP is on [0,4] and DP is on [0,6]. In the embedding procedure, by looking up the 5-bit secret message according to Fig. 2, we can get SP=1 and DP=2. We embed 1 into the selector group, and 2 into the descriptor group by the EMD method. We get the pixels of two stego group, which are (167,167) and (168,169,168). After embedding all the secret bits into the cover image, we convert the stego stream back to the deciphered order using the same chaotic sequence. Then the stego stream is reshaped back to 512×512 binary image and then we get the stego image.

3.3. Data extraction

The data extraction procedure is quite simple. The receiver should own the keys to decipher the chaotic sequence and to distinguish the selector groups and the descriptor groups. For each group, calculate the value from Eq. (1), where the result is the hidden digits in the group. Then look up the cross points of the table for SP and DP to extract the embedded number. After deciphering the extracted stream by using the same chaotic sequence, the receiver can get the original secret message.

4. Experimental results

To demonstrate the performance of the proposed method, several experimental results are given in this section. The histogram analysis of pixel difference between the stego-image and the original image, imperceptibility test, and anti-detection test are performed. The image databases we use include the standard image library and the UCID uncompressed image database [33]. Standard images, such Lena, Baboon, and Plane, are 512×512 grayscale images. The images from the UCID database are 1000 512×384 uncompressed images. The secret message we use is the logo of Fudan University. Before embedding, we first convert the UCID images into grayscale ones. The secret image is converted into binary image and extended to the appropriate size according to the different cover image sizes.

4.1. Histogram analysis

In order to realize a fair comparison among EMD method, Jung's method, Chao et al.'s method, and the proposed method, we embed the same message into the same cover image using these four methods. What we are interested in is to compare the amount of modifications introduced by the four methods.

Fig. 3 shows the cover images and secret data in our test. Fig. 4 is the histograms of pixel difference between cover images and stego images, with 100% payload, while Fig. 5 with 50% payload. The less modification made to the cover image, the less difference between the stego and the cover images will be, i.e., the probability of zeros appearing in the histogram will be maximum. From Figs. 4 and 5, one can easily find that the proposed method has the highest bar at the value of 0, which means the amount of modifications introduced is minimum among the four methods, i.e. the proposed method introduces fewer embedding changes to the cover image than others.

4.2. Imperceptibility test

In this test, we use standard images as a cover image and adjust the size of the secret image so that the stego images have 100% and 50% payload, respectively. With different payloads, we test the visual quality of the stego image in terms of PSNR and SSIM [34,35]. From Tables 1 and 2, we can find that the value of PSNR and SSIM

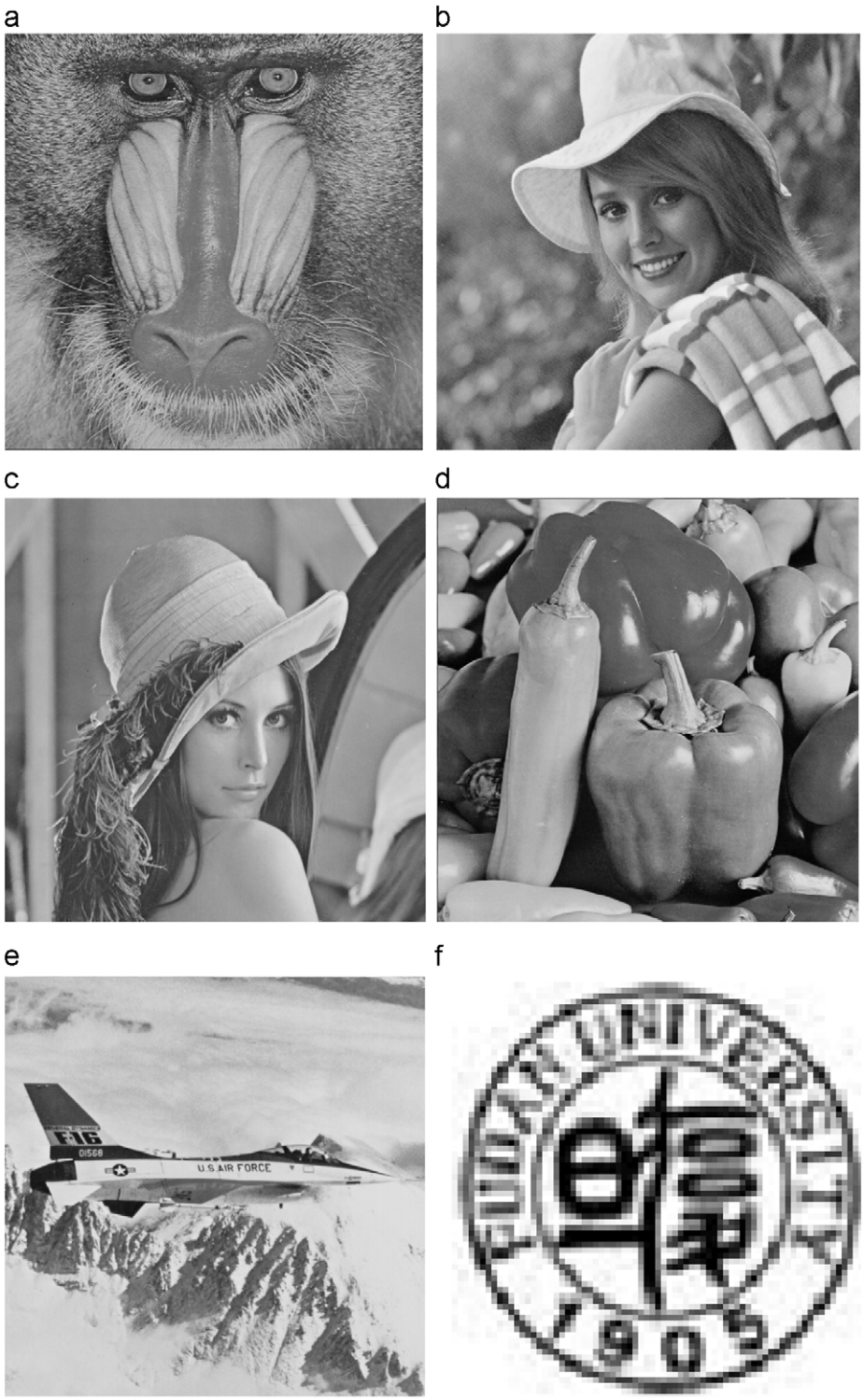

Fig. 3. The images in our experiments: (a)–(e) cover images (f) Fudan Logo.

obtained by the proposed method is bigger than one obtained by Jung's method, Chao et al.'s method, and the EMD method. So the visual quality of the stego images using our proposed method is better than Jung and Yoo's method, Chao et al.'s method, and the EMD method.

4.3. Probability of detection test

Anti-detection performance is a very important aspect when evaluating the performance of steganography. Like the LSB matching and Mielikainen's pair-wise LSB

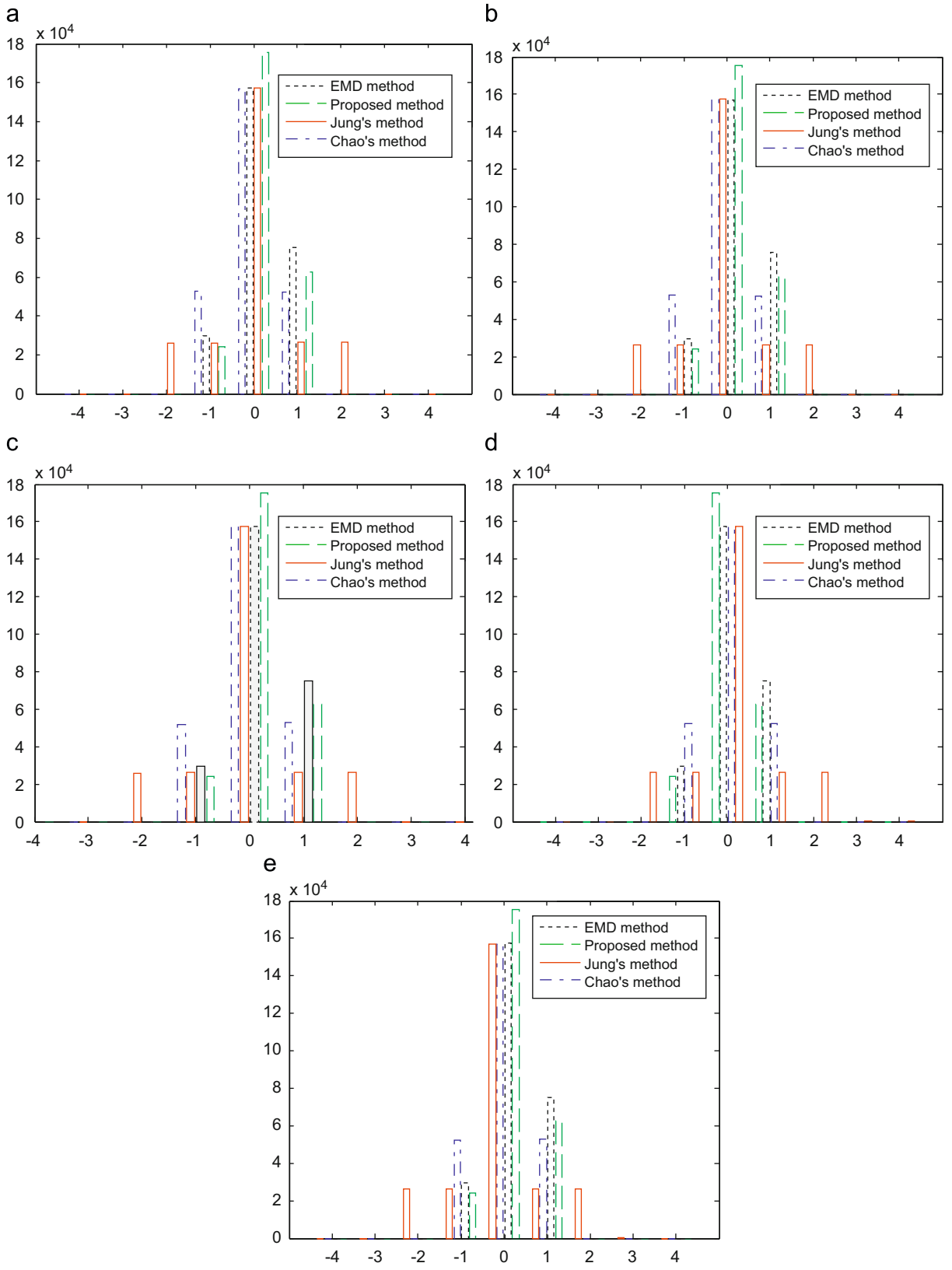

Fig. 4. Histogram of pixel difference between the cover and stego-images with 100% payload. (a) Histogram of pixel difference between cover and stego-Baboon with 100% payload. (b) Histogram of pixel difference between cover and stego-Elaine with 100% payload. (c) Histogram of pixel difference between cover and stego-Lena with 100% payload. (d) Histogram of pixel difference between cover and stego-Peppers with 100% payload. (e) Histogram of pixel difference between cover and stego-Plane with 100% payload.

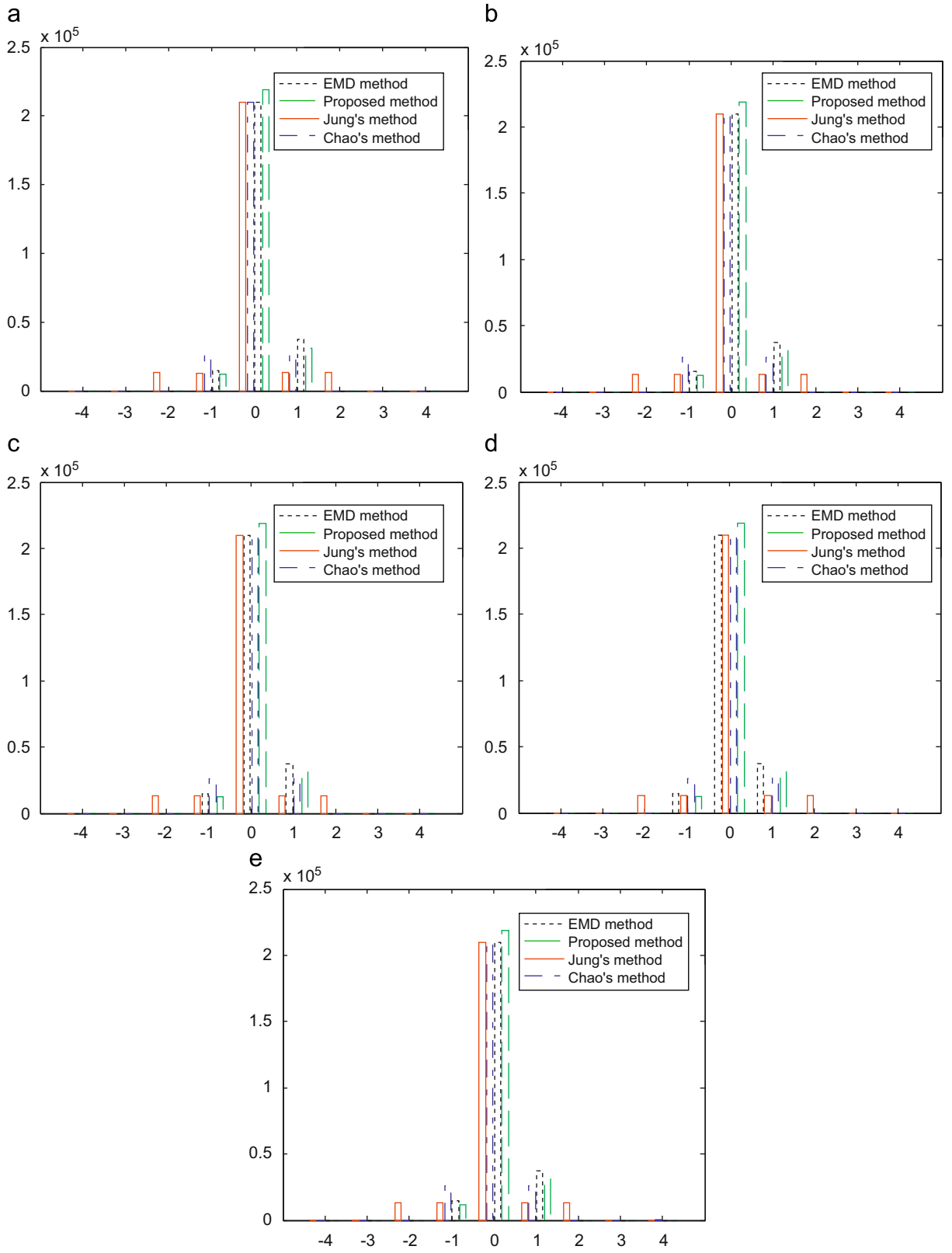

Fig. 5. Histogram of pixel difference between the cover and stego-images with 50% payload. (a) Histogram of pixel difference between cover and stego-Baboon with 50% payload. (b) Histogram of pixel difference between cover and stego-Elaine with 50% payload. (c) Histogram of pixel difference between cover and stego-Lena with 50% payload. (d) Histogram of pixel difference between cover and stego-Peppers with 50% payload. (e) Histogram of pixel difference between cover and stego-Plane with 50% payload.

Table 1
Visual quality comparison with 100% payload.

	Jung's method		Chao et al.'s method		EMD		Proposed method	
	PSNR(db)	SSIM	PSNR(db)	SSIM	PSNR(db)	SSIM	PSNR(db)	SSIM
Baboon	48.1232	0.9975	52.0955	0.9988	52.1153	0.9989	52.9396	0.9990
Elaine	48.1235	0.9938	52.1051	0.9976	52.1020	0.9978	52.9250	0.9981
Lena	48.1347	0.9904	52.1121	0.9965	52.1150	0.9971	52.9228	0.9975
Peppers	48.1482	0.9927	52.1101	0.9970	52.1008	0.9972	52.9332	0.9976
Plane	48.1348	0.9897	52.1055	0.9964	52.1146	0.9967	52.9305	0.9972

Table 2
Visual quality comparison with 50% payload.

	Jung's method		Chao et al.'s method		EMD		Proposed method	
	PSNR(db)	SSIM	PSNR(db)	SSIM	PSNR(db)	SSIM	PSNR(db)	SSIM
Baboon	51.1464	0.9994	55.1022	0.9995	55.1196	0.9994	55.9527	0.9995
Elaine	51.1473	0.9966	55.1230	0.9988	55.1036	0.9988	55.9282	0.9990
Lena	51.1494	0.9947	55.1120	0.9980	55.1183	0.9985	55.9384	0.9987
Peppers	51.1641	0.9961	55.1202	0.9985	55.1120	0.9985	55.9283	0.9988
Plane	51.1289	0.9944	55.1185	0.9979	55.1255	0.9983	55.9364	0.9986

matching method, the EMD method is very difficult to detect. One of the best detectors against these kinds of steganography is based on the center of mass (COM) of the histogram characteristic function (HCF) [36]. On that basis, Andrew D. Ker proposed an improved calibrated adjacency HCF-COM detector. Thus, in our test, we use this calibrated adjacency HCF-COM detector to test the stego images from the stego image database with 100% and 50% payloads, respectively.

In Fig. 6, we give receiver operating characteristic (ROC) curves on the UCID image databases when the payload is 100%. The curves show how the probabilities of detection and false positive vary as the detection threshold is adjusted. In Fig. 7, the payloads are reduced to 50%.

Figs. 6 and 7 show that at the same probability of false positive our method has lower probability of detection than other methods. This fact also shows that fewer modification introduced by steganographic scheme to the cover image is less detectable because it is less likely to disturb the statistics of the cover to trigger detection.

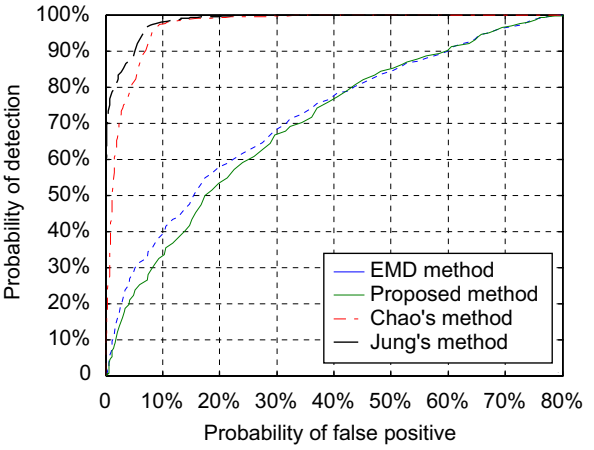

Fig. 6. ROC curves with a 100% payload.

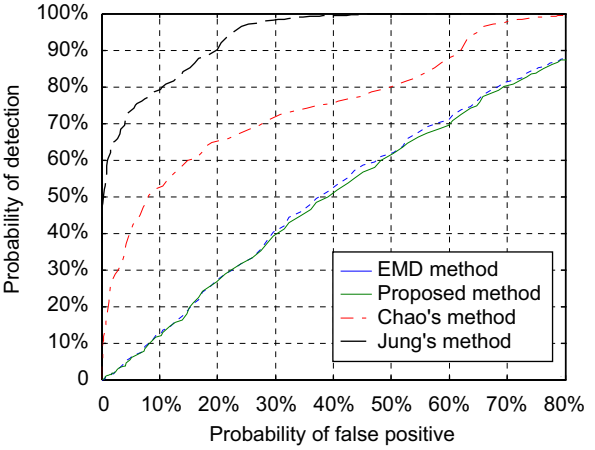

Fig. 7. ROC curves with a 50% payload.

5. Conclusions

In this paper, we introduced a section-wise strategy to improve the embedding efficiency of the EMD method. The acceptable assumption is that the cutoff length is always finite and not very large when converting the binary message into $(2n+1)$-ary digits. By choosing the appropriate combination of selectors and descriptors, our method can represent more modification directions with less pixel changes than the EMD method. By carrying out this strategy, the visual quality of the stego images is enhanced, and the probability of detection is decreased at the same time. Unlike other improvements, we neither sacrifice the visual quality of the stego images, combine different codes, nor perform multi-layer embedding.

Experimental results have shown the performance of the proposed method. It is notable that our experiments were carried out in spatial domain. In future work we could apply the proposed method to transform domain.

References

[1] H.J. Highland, Data encryption: a non-mathematical approach, Computers & Security 16 (5) (1997) 369–386.
[2] W. Stallings, in: Cryptography and Network Security: Principles and Practice, 3rd ed., Pearson Education, New Jersey, 2003.
[3] F.A.P. Petitcolas, R.J. Anderson, M.G. Kuhn, Information hiding—a survey, Proceedings of the IEEE 87 (7) (1999) 1062–1078.
[4] C.K. Chan, L.M. Cheng, Hiding data in images by simple LSB substitution, Pattern Recognition 37 (3) (2004) 469–474.
[5] R.Z. Wang, C.F. Lin, J.C. Lin, Image hiding by optimal LSB substitution and genetic algorithm, Pattern Recognition 34 (3) (2001) 671–683.
[6] S.J. Xiang, H.J. Kim, Jw Huang, Audio watermarking robust against time-scale modification and MP3 compression, Signal Processing 88 (10) (2008) 2372–2387.
[7] C. Deng, X. Gao, X. Li, D. Tao, A local Tchebichef moments-based robust image watermarking, Signal Processing 89 (8) (2009) 1531–1539.
[8] X.Y. Luo, D.S. Wang, P. Wang, F.L. Liu, A review on blind detection for image steganography, Signal Processing 88 (9) (2008) 2138–2157.
[9] Cheddad Abbas, Joan Condell, Curranand Kevin, Mc Kevitt Paul, Digital image steganography: survey and analysis of current methods, Signal Processing 90 (3) (2010) 727–752.
[10] J. Mielikainen, LSB matching revisited, IEEE Signal Processing Letters 13 (5) (2006) 285–287.
[11] X.P. Zhang, S.Z. Wang, Efficient steganographic embedding by exploiting modification direction, IEEE Communications Letters 10 (11) (2006) 781–783.
[12] D.C. Wu, W.H. Tsai, A steganographic method for images by pixel-value differencing, Pattern Recognition Letters 24 (9-10) (2003) 1613–1626.
[13] H.C. Wu, N.I. Wu, C.S. Tsai, M.S. Hwang, Image steganographic scheme based on pixel-value differencing and LSB replacement methods, IEE Proceedings of Vision, Image and Signal Processing 152 (5) (2005) 611–615.
[14] X.P. Zhang, W.M. Zhang, S.Z. Wang, Efficient double-layered steganographic embedding, Electronics Letters 43 (8) (2007) 482–483.
[15] W.M. Zhang, S.Z. Wang, X.P. Zhang, Improving embedding efficiency of covering codes for applications in steganography, IEEE Communications Letters 11 (8) (2007) 680–682.
[16] X.P. Zhang, W.M. Zhang, S.Z. Wang, Integrated encoding with high efficiency for digital steganography, Electronics Letters 43 (22) (2007).
[17] C.C. Chang, W.L. Tai, K.N. Chen, Improvements of EMD embedding for large payloads, International Conference on Intelligent Information Hiding and Multimedia Signal Processing 2007 (IIHMSP 2007), vol. 1, 2007, pp. 473–476.
[18] C.F. Lee, Y.R. Wang, C.C. Chang, A. Steganographic Method with high embedding capacity by improving exploiting modification direction, International Conference on Intelligent Information Hiding and Multimedia Signal Processing 2007 (IIHMSP 2007), vol. 1, 2007, pp. 497–500.
[19] W.M. Zhang, X.P. Zhang, S.Z. Wang, A. Double Layered, ''Plus–Minus One'' data embedding scheme, IEEE Signal Processing Letters 14 (11) (2007) 848–851.
[20] J.Y., Byun, K.H. Jung, K.Y. Yoo, Improved data hiding method by exploiting modification direction, 2008 International Symposium on Ubiquitous Multimedia Computing.
[21] R.M. Chao, H.C. Wu, C.C. Lee, Y.P. Chu, A novel image data hiding scheme with diamond encoding, EURASIP Journal on Information Security, Volume 2009, Article ID 658047, doi:10.1155/2009/658047.
[22] K.H. Jung, K.Y. Yoo, Improved exploiting modification direction method by modulus operation, International Journal of Signal Processing, Image Processing and Pattern 2 (1) (2009) 79–87.
[23] P. Sur, J. Goel, A. Mukhopadhyay, Spatial domain steganographic scheme for reducing embedding noise, International Symposium on Communications, Control and Signal Processing 2008 (ISCCSP 2008), pp. 1024–1028.
[24] H.M. Sun, K.H. Wang, C.C. Liang, Y.S. Kao, A LSB substitution compatible steganography, TENCON 2007, 2007 IEEE Region 10 Conference, pp. 1–3.
[25] C.F. Lee, H.L. Chen, High-capacity data hiding using virtual window partition, International Conference on Intelligent Information Hiding and Multimedia Signal Processing 2007 (IIHMSP 2007), vol. 1, 2007, pp. 315–318.
[26] C.M. Wang, N.I. Wu, C.S. Tsai, M.S. Hwang, A high quality steganographic method with pixel-value differencing and modulus function, Journal of Systems and Software 81 (1) (2008) 150–158.
[27] C.C. Lin, N.L. Hsueh, A lossless data hiding scheme based on three-pixel block differences, Pattern Recognition 41 (4) (2008) 1415–1425.
[28] C.C. Chang, C.F. Lee, L.Y. Chuang, Using dynamic programming strategy to find an optimal solution to exploiting modification direction embedding method, International Conference on Intelligent Information Hiding and Multimedia Signal Processing 2007 (IIHMSP 2007), vol. 1, 2007, pp. 489–492.
[29] 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.
[30] D.X. Qi, J.C. Zou, X.Y. Han, A new class of scrambling transformation and its application in the image information covering, Science in China, Series E 43 (3) (2000) 304–312.
[31] J. Cheng, J.I. Guo, A new chaotic key-based design for image encryption and decryption, IEEE International Symposium on Circuits and Systems, 2000 (ISCAS 2000, Geneva), vol. 4, 2000, pp. 49–52.
[32] H. Zhang, X.F. Wang, Z.H. Li, D.H. Liu, Y.C. Lin, A new image encryption algorithm based on chaos system, IEEE International Conference on Robotics, Intelligent Systems and Signal Processing 2003, vol. 2, 2003, pp. 778–782.
[33] G. Schaefer, M. Stich (2004) UCID—an uncompressed colour image database, in: Proceedings of SPIE, Storage and Retrieval Methods and Applications for Multimedia 2004, San Jose, USA, pp. 472–480. http://vision.cs.aston.ac.uk/datasets/UCID/ucid.html.
[34] W. Zhou, A.C. Bovik, A universal image quality index, IEEE Signal Processing Letters 9 (3) (2002) 81–84.
[35] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Transactions on Image Processing 13 (4) (2004) 600–612.
[36] A.D. Ker, Steganalysis of LSB matching in grayscale images, IEEE Signal Processing Letters 12 (6) (2005) 441–444.