Improved Steganographic Embedding Exploiting Modification Direction in Multimedia Communications

Cheonshik Kim[1], Dongkyoo Shin[1], Dongil Shin[1], and Xinpeng Zhang[2]

[1] Dept. of Computer Engineering, Sejong University 98 Gunja-Dong,
Gwangjin-Gu, Seoul, 143-747, Korea
mipsan@paran.com, {shindk,dshin}@sejong.ac.kr
[2] School of Communication and Information Engineering, Shanghai University,
Shanghai 200072, China
xzhang@shu.edu.cn

Abstract. Steganography provides secure communications over the Internet with a cover image. However, it is difficult to transfer many messages with small-sized images. We have improved EMD (Exploiting Modification Direction), proposed by Zhang and Wang, to solve this problem. In this paper, we have developed a $(2^{n+2}\text{-}1)$-*ary* scheme. Our scheme shows a higher embedding rate, $R=\log_2(2^{n+2}-1)/n$, which is greater than that of the EMD scheme, because the EMD scheme embedding rate is $\mathrm{R}=\log_2(2n+1)/n$, for $n>=2$. The experimental results show that our scheme is able to embed twice as many secret bits in multimedia communications compared to the existing EMD embedding method. Our method has low complexity and achieves higher embedding performance with good perceptual quality against the earlier arts. An experiment verified our proposed data hiding method in multimedia communications.

Keywords: Steganography, data hiding, EMD, secret communications, multimedia communications.

1 Introduction

Almost every one of us has heard a friend complaining that his email account has been hacked. Alternatively, it may have happened to you. The truth is that exposure to hacking of you yahoo messenger or any other kind of email supplier account has become quite a problem to users. Hacking MSN password accounts is no longer the realm of professionals. Thanks to the widespread use of the internet, any cyber-terrorist can learn the required tricks to master the art of hacking email passwords. He only needs to make a basic search with keywords, such as hacking yahoo passwords, msn hacking programs, hacking yahoo mail, hotmail hacking programs or even something as simple as hotmail hacking guide. All of that is out there, ready to be learned.

Thus, researchers have investigated steganography for personal safety communications with various multimedia, such as audio, text and images. Steganography is the art and science of hiding communications.

J.J. Park et al. (Eds.): STA 2011, CCIS 186, pp. 130–138, 2011.

A steganographic system thus embeds hidden content in unremarkable cover media, so as not to arouse an eavesdropper's suspicion. Sender and receivers agree on a steganographic protocol system to communicate. This needs both parties need to know how a message is encoded in the cover medium. For example, Alice creates a new image with messages. Alice sends an image to Bob. When Bob receives the image, he uses the shared secret message. However, statistical analysis may reveal the presence of a hidden message in cover medium. Thus, the cover image with a message should be able to resist such an attack to safely communicate.

But, it is not difficult to avoid detectable traces, as image quality is in inverse proportion to the modification of an image. Therefore, to increase the security of an image in multimedia communications, the amount of hidden data or modification of an image needs to be reduced [1], [2]. There are two main issues [3], [4], [5] in steganography: First, creating stego images must be imperceptible to protect secret messages without detection or extraction. In order to preserve, an image with secret image require high quality of an image. A second important issue is embedding capacity. There are a few schemes to provide high capacity in an image. According to implementation of steganography, there are two categories, which are spatial domain and frequency domain method [6].

First, secret hidden messages are hidden in the spatial domain of a host image. This method is embedded into the pixels of the host image directly. Usually, this can be hidden by replacing the rightmost four the least significant bits per pixel in LSB substitution schemes [7], [8], [9]. Second, the discrete cosines transform (DCT) [10] or the discrete wavelet transform (DWT) [11] is used to transform pixel values. Zhang and Wang [12] proposed steganography to embed a secret message in an image in a $(2n+1)$-*ary* notation system. They use a $(2n+1)$-*ary* notational system, where n is the numbering of the pixel to carry one digit. The embedding rate of the secret message in an image is $R = (\log_2(2n+1)) = n$.

That is, it is about 1.161 bpp (bits per pixels) when $n = 2$. In Chin-Feng Lee's [13] method the capacity steadily increases as m increased and the *dB* value decreases as m increases. This method used a virtual hyper cube to hide a message in an image; this was created by a random number. In fact, it is impossible to change virtual pixels. Thus, the modification value must be stored in a pair-pixel. This paper introduces a new method improving Zhang and Wang's embedding method (called EMD), especially in embedding capacity with high image quality, it is about 46 *dB*.

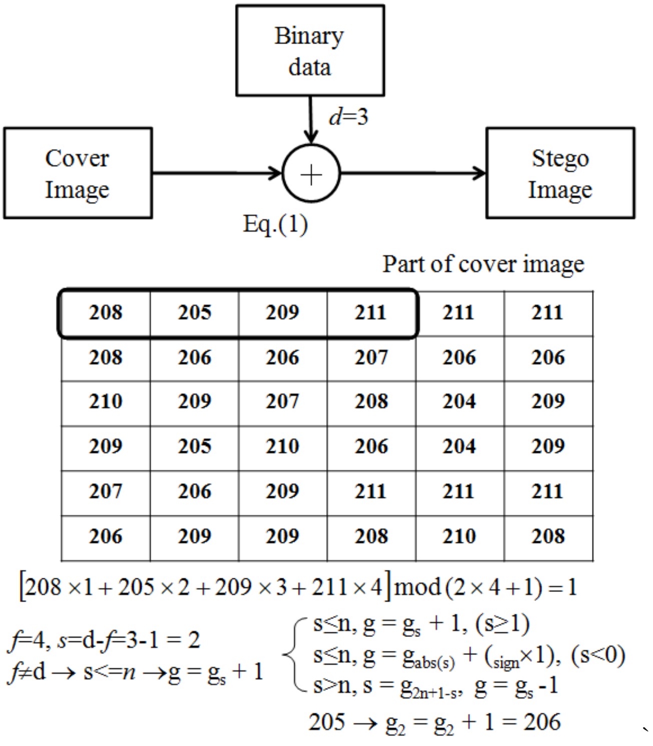
We use a $(2^{n+2}-1)$-*ary* notational system in our proposed method that encodes a stream of bits with a cover pixel. The embedding rate of our method is double that of the EMD method, when $n = 2$. Moreover, the quality of the stego image is very high and there is no problem in security. Therefore, our proposed scheme is a novel steganography scheme in which the image's quality and capacity are balanced to increase capacity in a cover image.

2 EMD Embedding Method

EMD [12] is a novel method to hide data in an image with schemes of modification directions. This method carried a over 3 bits with n cover pixels, when $n \geq 2$. $(g_1, g_2, \ldots, g_n)$ is a group of pixels. Using this method, n pixels are used to carry one secret digit in the (2*n*+1)-*ary* notational system. Especially, in this method only one pixel is incremented or decremented by 1 in a group of pixels. A vector $(g_1, g_2, \ldots, g_n)$ in *n*-dimensional space is indicated with its ***f*** value, which is calculated by the Eq. (1):

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

No modification is needed if a secret digit *d* equals the extraction function of the original pixel-group. Secret data: $d \neq f$, calculate $\boldsymbol{s} = \mathrm{d} - f \bmod (2n+1)$. If ***s*** is no more than *n*, increase the value of g_s by 1, otherwise, decrease the value of g2*n* + 1 - *s* by 1.



208	205	209	211	211	211
208	206	206	207	206	206
210	209	207	208	204	209
209	205	210	206	204	209
207	206	209	211	211	211
206	209	209	208	210	208

$$[208 \times 1 + 205 \times 2 + 209 \times 3 + 211 \times 4] \mathrm{mod}(2 \times 4 + 1) = 1$$

$$f\text{=}4,\ s\text{=d-}f\text{=}3\text{-}1 = 2$$

$$f \neq \mathrm{d} \rightarrow \mathrm{s} <= n \rightarrow \mathrm{g} = \mathrm{g_s} + 1$$

$$\begin{cases} \mathrm{s} \leq \mathrm{n},\ \mathrm{g} = \mathrm{g_s} + 1,\ (\mathrm{s} \geq 1) \\ \mathrm{s} \leq \mathrm{n},\ \mathrm{g} = \mathrm{g_{abs(s)}} + (_{\mathrm{sign}} \times 1),\ (\mathrm{s} < 0) \\ \mathrm{s} > \mathrm{n},\ \mathrm{s} = \mathrm{g_{2n+1-s}},\ \mathrm{g} = \mathrm{g_s} \text{-}1 \end{cases}$$

$$205 \rightarrow \mathrm{g_2} = \mathrm{g_2} + 1 = 206$$

Fig. 1. EMD Encoding procedure

Example 1: [208 205 209 211], *n* = 4, *f* = 1. Let *d* = 3. Since ***s*** = 2, an encoder would increase the gray value of the first pixel by 1 to produce the stego-pixels [208 206 209 211]. At the decoder, *f* = 3. The decoder can extract secret message 3. The detailed explanation can be seen in Fig.1.

Example 2: [137 139 141 140], *n* = 4, *f* = 3. Let *d* = 0. Since s = 6, an encoder would decrease the gray value of the third pixel by 1 to produce the stego-pixels [137 139 140 140]. At the decoder, *f* = 0. The decoder can extract secret message 0.

In fact, the EMD embedding scheme provides a very good stego image quality with the PSNR value greater than 52 *dB* and the embedding capacity is 1.16 bpp when n = 2. However, it is not appropriate to send a big message to a receiver. This requires that the capacity of an image is increased.

3 The Proposed Scheme

In this section, we shall present the proposed steganography scheme based on a $(2^{n+2}-1)$-*ary* notational system in a group of pixels. Our scheme is composed of the embedding and extracting procedure, described below. This paper proposes a novel steganographic embedding method that fully exploits the modification directions. In this method, modifications in different directions are used to represent distinctive secret data, leading to a higher embedding efficiency.

3.1 The Embedding Procedure

In the EMD method, just one bit is increased or decreased in a group of pixels. Therefore, we have an idea from those viewpoints that it is possible to increase or decrease by a maximum of four bits in a group of pixels with a high stego image. We formulated the concept of Eq.2 to hide the more secret bit than that of the EMD method. b_i uses embedding and extracting secret data in function ***f*** as a weighted sum modulo $(2^{n+2}-1)$. n is a group of pixels. In the case of n=2, b_i is [1, 3, 6, 9, 12,…, (n-1)×3]. Table 1 is used for encoding in our proposed $(2^{n+2}-1)$-*ary* system. ***s*** uses encoding to make $d = f$, and it is composed of three types. If $d \geq f$, calculate $d - f$, and if d < f and $n > 2$, calculate $(2^{n+2} - 1)-|d-f|$, and if $d < f$ and $n = 2$, calculate $(2^{n+2} - 1)-|d-f|$. After finding the value ***s*** from equation (4), apply equation (5) to a group of pixels.

$$f(g_1, g_2, ..., g_n) = \left[\sum_{i=1}^{n}(g_i \bullet b_i)\right] \text{mod } (2^{n+2}-1) \tag{2}$$

where,

$$[b_1, b_2, ..., b_n] = \{[1, 3, 6, 9, ..., (n-1)\times 3, \ n \geq 2 \tag{3}$$

and

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

$$g = g + g_s \tag{5}$$

Example 3: We will explain the embedding procedure with an example. For a group of pixels, vector g = [10 5 3]. If the extraction function value is ***f*** and the value to hide is ***d***, we need to find a suitable ***g*** to make $f = d$. For example, let n be 3. The numbers generated by the basis vector [1 3 6] are shown in Table 2.

Step1: It is necessary to calculate the ***f*** value with a three pair of g and basis vector [1 3 6]. ***f*** is 13 as Eq.2.

Step2: If you want to hide decimal digit *5* into *g*, what you need to do first is to compute the value g_s with Eq.4. After calculation, the result is -8. In this case, ***d*** is less than or equal ***f***. i.e., we calculate s= $(2^{n+2}-1)-|d-f|$. That is, ***s*** is 7.

Step3: Look up *s*=7 from Table 1 using the index and find the row of 7, which is [1 0 1] starting with the first and last value of the pixel value vector that should be increased by 1. Therefore, g_s is [1 0 1]. It is referenced by ***s***. So, the result of a group of pixels becomes [11 5 4].

Table 1 is used to reference ***s*** encode and decode using Eq.2. We need to expand the EMD method.

Table 1. A basis vector [1, 3, 6] when *n* is 3

index	***1***	***3***	***6***
0	0	0	0
1	1	0	0
2	-1	1	0
3	0	1	0
4	1	1	0
5	-1	0	1
6	0	0	1
7	1	0	1
8	-1	1	1
9	0	1	1
10	1	1	1
11	2	1	1
12	0	0	2
13	1	0	2
14	-1	1	2
15	0	1	2
16	0	-1	-2
17	1	-1	-2
18	-1	0	-2
19	0	0	-2
20	-2	-1	-1
21	-1	-1	-1
22	0	-1	-1
23	1	-1	-1
24	-1	0	-1
25	0	0	-1
26	1	0	-1
27	-1	-1	0
28	0	-1	0
29	1	-1	0
30	-1	0	0

3.2 The Extracting Procedure

The decoding procedure is very simple, because you have only to know the ***f*** value as in Eq.2 with a group of pixels. That is, ***f*** is an embed bit in a pixel-group.

Example 4: Consider a pixel-group [253 203 250] with *n* = 3 and ***f*** = 7. We assume a secret digit is 4, ***s*** is 12. The ***s*** value can be calculated by Eq.4. The next step is to reference ***s*** from Table 1, and apply referenced values to a group of pixel value vectors g = [253 203 252]. The decoding process is very simple. That is, we apply Eq.2 to a group of pixels, g = [254 203 252]. Therefore, we will find ***f*** = 4, and digit 4 is extracted successfully from the group of pixels [253 203 252].

Step1: First, we must calculate f with Eq.2 and the basis vector [1 3 6]. In this case, we can extract f = 7 in the (2^{n+2}-1)-*ary* system.

Step2: For a secret digit 4, we need to find the ***s*** value from Eq.4. It is easy to extract the ***s*** value from (2^{n+2}-1)-|d-f|, that is, ***s*** = 12.

Step3: Look up Table 1 and find ***s*** from the row, which is [0 0 2] stating that the last value of the pixel value vector should be increased by 2. Therefore, the changed pixel value vector becomes [253 203 252]. In the case of an overflow or underflow, (g_1, g_2, g_3) has to be adjusted to the appropriate values. The rules are in Eq.6 and Eq.7

$$if\ \ g_s > 255, \qquad g = g_s - (2^{n+2} - 1) \tag{6}$$

$$if\ \ g_s < 0, \qquad g = g_s + (2^{n+2} - 1) \tag{7}$$

Step4: A receiver is needed to calculate f with Eq.2 to find the message from the stego-image. No extra calculation is needed, so it is a very simple algorithm. In this case, ***f*** is 4.

4 Experimental Results

The purpose of our scheme is to embed data into a cover image for secret communications. This requires the proposed method to have high capacity and good image quality. We experimented with some grayscale image and compared them to other methods, such as EMD [12] and Lee's method [13], to show the performance of the proposed method. The secret data used in this experiment use the MATLAB function rand () to generate a pseudo random number, rounded to (2^{n+2}-1)-*ary* number to generate a payload. Then, we apply the EMD [12] and proposed scheme to test images, including the University of Southern California [17].

In general, the evaluation of data hiding performance depends on the visual quality of the stego image and data hiding capacity. Hence, data hiding capacity is a crucial issue to consider in evaluating the performance of data hiding methodologies. Data hiding capacity is defined as the amount of data that can be hidden under the cover image. The capacity of EMD is $R = \log_2(2n+1)/n$, where n is a group of pixel numbers and n ≥ 2, which has the best embedding rate R of $(\log_2(5))/2$=1.161, when n is set to be 2.

In the proposed method, R = $\log_2(2^{n+2}\text{-}1)/n$, where n = 2, which has also the best embedding rate R of $\log_2(15)$= 1.9534, which is about two times than that of the EMD method. Fig.2 compares the embedding rate for EMD [12] embedding and the proposed methods. The simulation reveals the capacity of our scheme is better than that of the EMD scheme.

Lee's method [13] is to use a virtual hypercube as a random number to increase capacity in an image. This method only stores g_{2n} +1 - s value in a pair pixel. Thus, a group of pixels do not modify from Eq.1. p_m is a local range in a pair of pixels, where *s* is stored. When p_m is 3, the capacity of data hiding is less than that of EMD. This is why we do not compare it to the proposed method.

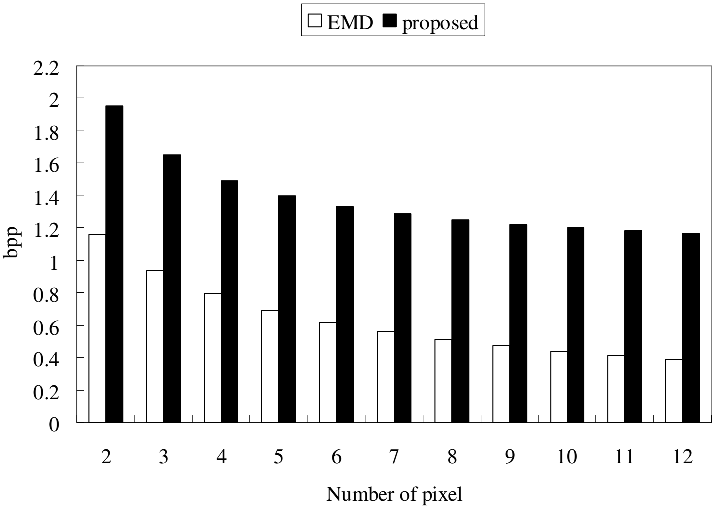


Fig. 2. Comparison of capacity rate between proposed and EMD

We experiment with nine grayscale images, such as Lena, Baboon, Elaine, Airplane, Pepper, Goldhill, Barbara, Boat and Zela, as cover-images. Their size is 512×512 pixels.

In our proposed method, the embedding rate R = $(\log_2(15))/2=1.9534$. This is a greater payload than that of EMD. In the real experiment, we hide 516,823 bits in the cover image and the proposed method is higher capacity than that of Lee's method [13]. Furthermore, our method is better than Lee's [13] method, because it does not always flipping four bits in a group of pixels. Sometimes, there is no flipping bit in a group of pixels. Conversely, Lee's method always flips three bits, when p_m is three. Thus, this method is a not a complete method to use in the aspect of optimization. In the case of less optimization, it can be possible to allow a statistical analysis attack. Therefore, Lee's method [13] does not conceal the secret message in a stego image from the steganographic technique [14]. The average PSNR value of our proposed method is about 48.2797 *dB*. In fact, our method is not as good quality as that of EMD. However, 48.2797 *dB* are a high quality image, so it can resist steganalysis detection using a human visual system. Thus, it is acceptable to use a carrier for secret communications. Table 2 shows the comparison PSNR between our proposed method and Lee's method when n = 3. As you can see Table 2, our proposed method is very good PSNR against that of Lee's method.

Table 2. The PSNR of nine test cover images when $n = 3$

Test Image	*Proposed Method*	*Lee's method [13]*
Lena	48.2765	46.5777
Baboon	48.2768	46.5613
Tiffany	48.2877	46.5953
Airplane	48.2887	46.5626
Pepper	48.3869	48.2887
Goldhill	48.2805	46.5894
Barbara	48.2738	46.5681
Boat	48.2749	46.5371
Zelda	48.2717	46.5827

5 Conclusion

Secret communications over the Internet have become an issue of communication fields, because it is not easy to protect a secret message from hackers and attackers. Steganography provided a secure safety channel for communications. Thus, researchers developed various stegnography schemes. However, there are a few methods to provide high capacity of secret messages and good quality of an image. Therefore, we improved a steganography of EMD [12] scheme. Our proposed scheme provided a $\log_2(2^{n+2}-1)$-*ary* system, so that our method has better capacity than that of EMD [12]. Lee's method [13] has lower image quality than our method.

Acknowledgement

This work was supported by the Industrial Strategic technology development program (10011346, Development of Intelligent Human Interface Terminal and Context-Awareness Gateway) funded by the Ministry of Knowledge Economy (MKE, Korea)

References

1. Wang, H., Wang, S.: Cyber warfare:steganography vs. steganalysis. Communication of the ACM 47(10), 76–82 (2004)
2. Goljan, M., Hogea, D., Soukal, D.: Quantitative Steganalysis of Digital Images: Estimating the Secret Message Length. ACM Multimedia Systems Journal, Special Issue on Multimedia Security 9(3), 288–302 (2003)
3. Katzenbeisser, S., Petitcolas, F.A.P. (eds.): Information Hiding Techniques for Steganography and Digital Watermarking. Artech House Books, Boston (2000) ISBN I-58053-35-4

4. Westfeld, A.: F5-A steganographic algorithm. In: Moskowitz, I.S. (ed.) IH 2001. LNCS, vol. 2137, pp. 289–302. Springer, Heidelberg (2001)
5. Provos, N., Honeyman, P.: Hide and Seek: An Introduction to Steganography. IEEE Security and Privacy 1(3), 32–44 (2003)
6. Yua, Y.-H., Changa, C.-C., Hub, Y.-C.: Hiding secret data in images via predictive coding. Pattern Recognition 38(5), 691–705 (2005)
7. Mielikainen, J.: LSB matching revisited. IEEE Signal Processing Letters 13(5), 285–287 (2006)
8. Chan, C.-K., Cheng, L.M.: Hiding data in images by simple LSB substitution. Pattern Recognition 37(3), 469–474 (2004)
9. Lin, I.-C., Lin, Y.-B., Wang, C.-M.: Hiding data in spatial domain images with distortion tolerance. Computer Standards and Interfaces 31(2), 458–464 (2009)
10. Chang, C.C., Chen, T.S., Chung, L.Z.: A steganographic method based upon JPEG and quantization table modification. Information Sciences-Informatics and Computer Science 141(1-2), 123–138 (2002)
11. Spauldinga, J., Noda, H., Shirazib, M.N., Kawaguchia, E.: BPCS steganography using EZW lossy compressed images. Pattern Recognition Letters 23(13), 1579–1587 (2002)
12. Zhang, X., Wang, S.: Efficient Steganographic Embedding by Exploiting Modification Direction. IEEE Communications Letters 10(11), 781–783 (2006)
13. Lee, C.-F., Chang, C.-C., Wang, K.-H.: An improvement of EMD embedding method for large payloads by pixel segmentation strategy. Image and Vision Computing 26(12), 1670–1676 (2008)
14. Fridrich, J., Goljan, M., Du, R.: Detecting LSB steganography in color, and gray-scale images. IEEE Trans. Multimedia 8, 22–28 (2001)
15. Westfe3d, A., Pfitzmann, A.: Attacks on Steganographic Syste. In: Proc. 3rd Information Hiding Workshops, Dresden, Germany, September 28-October 1, pp. 61–75 (1999)
16. Goljan, M., Soukal, D.: Higher-Order Statistical Steganalysis of Palette Images. In: Proc. SPIE, Electronic Imaging, Security, Steganography, Watermarking of Multimedia Contents V, Santa Clara, California, pp. 178–190 (2003)
17. http://sipi.usc.edu/database/