2011 Seventh International Conference on Intelligent Information Hiding and Multimedia Signal Processing

Reversible Data Hiding by Reduplicated Exploiting Modification Direction Method

Chin-Feng Lee*
Department of Information Management
Chaoyang University of Technology
Taichung, 41349, Taiwan, R.O.C.
e-mail: lcf@cyut.edu.tw

Kai-Chin Chen
Department of Information Management
Chaoyang University of Technology
Taichung, 41349, Taiwan, R.O.C.
e-mail: s9914614@cyut.edu.tw

***Abstract*—Zhang *et al.* in 2006 proposed a data hiding technique called Exploiting Modification Direction (EMD) with good capacity and high image quality. Related follow-up improvements are proposed by other scholars to raise the change between pixel values thus increasing the capacity and without reversibility. This paper utilizes the high image quality feature of EMD to develop a reversible data hiding method while maintaining high capacity, acceptable image quality, and reversibility.**

Keywords-Data Hiding; Reversible Data Hiding; EMD

I. INTRODUCTION

Data hiding technique conceals the secret message into cover image, where the image embedded with secret message is called stego-image. Then this stego-image is being transmitted to prevent the third party from modifying, intercepting, and tampering, thus protecting the data.

There are two major research areas in data hiding techniques: irreversible data hiding and reversible data hiding. Irreversible data hiding technique cannot recover images back to cover images even after the receiver retrieved the embedded secret message. Such technique holds an extremely high capacity but it destroys images [2, 4, 6, 7, 10]. As for reversible data hiding technique, stego-images can be restored back to the original images after retrieving the embedded secret data with a lower capacity than irreversible method [1, 3, 5, 8, 9].

From the above methods, this paper utilizes the high image quality feature of EMD to develop a reversible data hiding technique, with the goal of achieving high capacity, acceptable image quality, and reversibility. The earliest EMD data hiding technique is proposed by Zhang *et al.* [10] in 2006. A group of n neighboring pixels in cover image are used with modulus operation to embed a secret digial value in n-ary notational system while achieving good capacity and high image quality.

In 2008, Lee *et al.* [6] proposed a method to embed two secret digits in a group of cover pixels combining with LSB technique in order to increase embedding capacity; however, reversibility is still not achieved. Such method is called R-EMD. In this paper, we utilize R-EMD data hiding technique and image expansion to embed and restore images thus achieves reversibility while maintaining high embedding capacity.

The remainder of the paper is organized as follows. in Section 2, we will focus on discussion of Zhang *et al.*'s proposed method based on EMD data hiding technique and Lee *et al.*'s proposed R-EMD data hiding technique. Our proposed method is introduced in Section 3. Section 4 contains the experimental results and conclusion is given in Section 5.

II. THE PROPOSED SCHEME

Lee *et al.*'s R-EMD method divides cover image into several pixel-groups and each pixel-group has n pixel values denoted as $(p_1, p_2, ..., p_n)$, and then uses calculation of modulus function to embed secret message into pixel-group twice. Only certain of pixel values among n cover pixel values can be modified while secret data are embedding by modulus function. The changes in pixel values are not noticeable, hence high capacity and good image quality are maintained as advantages. However, this method does not possess reversibility.

This paper utilizes features of data hiding technique from R-EMD and image expansion to perform data hiding so that images are restored while achieving reversibility. Also, the same group of cover image pixel value $(p_1, p_2, ..., p_n)$ is embedded twice with secret messages to increase effective capacity. Figure 1 represents the schematic flow chart for the proposed scheme. The notations used are described as follows:

(1) $b_i^8\ b_i^7\ ...\ b_i^1$ ：The binary expression of a cover pixel p_i.

(2) $p_i^{(1)}$ ：A medium value after a secret digit in a $(2n+1)$-ary notational system has been embedded into pixel p_i.

(3) $p_i^{(2)}$ ：A stego-pixel after a secret digit in a $(2n+1)$-ary notational system has been embedded into pixel $p_i^{(1)}$.

(4) $p_{i,4}^{(3)}$ ：A stego-pixel after the difference value between $p_i^{(2)}$ and p_i has been hidden in pixel $p_{i,4}$.

(5) g_i ：A decimal value from the first 7 bits of p_i, i.e., the value of $\lceil (p_i+1)/2 \rceil$.

(6) $g_i^{(1)}$, $g_i^{(2)}$ ：A decimal value after a secret digit in a $(2n+1)$-ary notational system has been embedded into g_i, $g_i^{(1)}$ respectively.

Section A introduces image interpolation. R-EMD is applied on images is images extended by image interpolation. Details of message embedding and extracting methods are introduced in Sections B and C respectively.

978-0-7695-4517-2/11 $26.00 © 2011 IEEE
DOI 10.1109/IIHMSP.2011.29

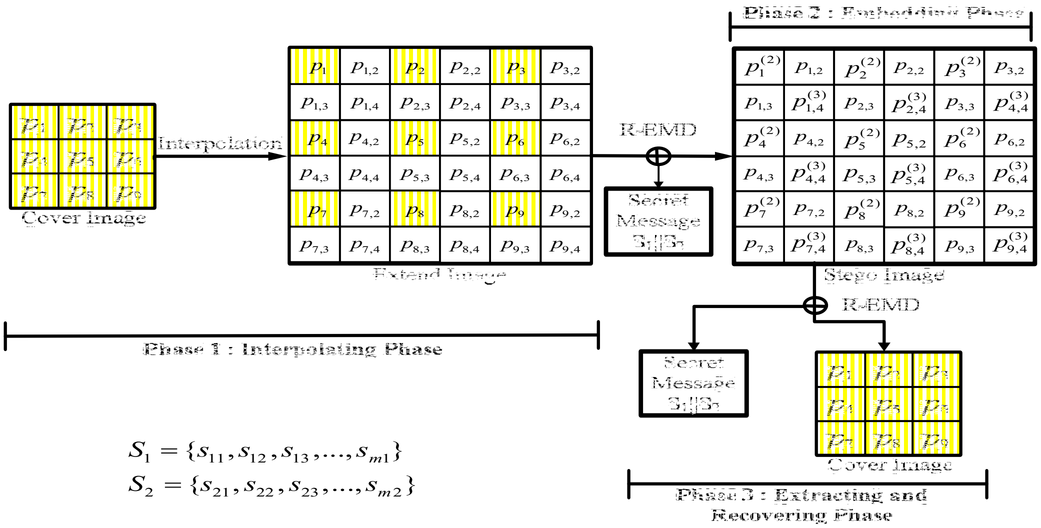

Figure 1. The proposed schematic flow chart

A. Interpolating Phase

The main goal is to achieve image recovery while maintaining reversibility. The steps are described in details as follows:

First, the image expansion is done on cover image by method of interpolation. For example, a 3×3 image is expanded to the size of 6×6 as shown in Phase 1 of Figure 1. As for calculation, p_1 is taken as an example to demonstrate how to combine its two neighboring pixels p_2, and p_4 to calculate three predicted pixels $(p_{1,2}, p_{1,3}, p_{1,4})$. The calculations are $p_{1,2} = \lfloor (p_1 + p_2)/2 \rfloor$, $p_{1,3} = \lfloor (p_1 + p_4)/2 \rfloor$, $p_{1,4} = \lfloor (p_{1,2} + p_{1,3})/2 \rfloor$, and so on. $(p_2, p_3, \ldots, p_9)$ can be completed as shown in Phase 1 of Figure 1.

B. Embedding Phase

First, take a cover image of $M\times N$ into $(M\times N)/n$ groups of n pixel values $(p_1, p_2, \ldots, p_n)$. The detailed embedding steps are as follows:

Step 1: Convert cover pixel value p_i into binary as $(p_i)_{10} = (b_i^8\ b_i^7\ b_i^6\ b_i^5\ b_i^4\ b_i^3\ b_i^2\ b_i^1)_2$.

Step 2: Divided p_i into two parts G and R . $G = (g_1, g_2, \ldots, g_n)$ where each $(g_i)_{10} = (b_i^8\ b_i^7\ b_i^6\ b_i^5\ b_i^4\ b_i^3\ b_i^2\ )_2$ and $R = \left\{ b_i^1 \middle| i = 1, 2, \ldots, n \right\}$.

Step 3: Apply EMD method on the group $G = (g_1, g_2, \ldots, g_n)$ as follows.

$$F = f(g_1, g_2, \ldots, g_n) = (\sum_{i=1}^{n} g_i \times i) \bmod (2n+1) \tag{1}$$

$$D = (s - F) \bmod (2n+1) \tag{2}$$

A certain g_i among $(g_1, g_2, \ldots, g_n)$ is adjusted to embed the secret digit s_1 in a $(2n+1)$ -ary notational system according to Equation (3).

$$g_i^{(1)} = \begin{cases} g_i, & \text{if } s_1 = F_1 \\ g_D + 1, & \text{if } s_1 \neq F_1 \text{ and } D \leq n \\ g_{(2n+1)-D} - 1, & \text{if } s_1 \neq F_1 \text{ and } D > n \end{cases} \tag{3}$$

Step 4: Through the third step, the first embedding of secret digit is finished. To increase capacity, the second embedding of s_2 in a $(2n+1)$-ary notational system will be carried out. Similarly, use Equation (1) and plug $(g_1^{(1)}, g_2^{(1)}, \ldots, g_n^{(1)})$ in function f to get a function value F_2 . Then, use Equation (2) to adjust a certain $g_i^{(1)}$ among $(g_1^{(1)}, g_2^{(1)}, \ldots, g_n^{(1)})$, and make other adjustment in certain $g_i^{(1)}$ among $(g_1^{(1)}, g_2^{(1)}, \ldots, g_n^{(1)})$ according Equation (4). The adjusted value is represented as $(g_1^{(2)}, g_2^{(2)}, \ldots, g_n^{(2)})$.

$$g_i^{(2)} = \begin{cases} g_i^{(1)}, & \text{if } s_2 = F_2 \\ g_D^{(1)} + 1, & \text{if } s_2 \neq F_2 \text{ and } D \leq n \\ g_{(2n+1)-D}^{(1)} - 1, & \text{if } s_2 \neq F_2 \text{ and } D > n \end{cases} \tag{4}$$

Now use the n bit of $R' = (b_1^{1'}, b_2^{1'}, \ldots, b_i^{1'}) = (b_1^1, b_2^1, \ldots, b_i^1)$ in least significant bit of n pixel value of medium group $(p_1^{(1)}, p_2^{(1)}, \ldots, p_n^{(1)})$ to indicate either $g_i^{(1)}+1$ or $g_i^{(1)}-1$ will be used as an indicator bit for retrieving the second secret digit. Here are two cases to explain the meaning of the least significant bit $b_i^{1''}$ in medium pixel value $p_i^{(2)}$.

Case 1: R is adjusted to be the set of indicator bits, where each bit is set as "0" , i.e., $b_i^{1''}=0$, for $i=1, 2, \ldots, n$, and $g_i^{(2)} = g_i^{(1)}$, for $i=1, 2, \ldots, n$.

Case 2： R is adjusted to be the set of indicator bits, where there is only one bit "0" or "1" in R. (Case 2-1) When $b_i^{1''}=0$ and $b_j^{1''}=1$, for $j=1, 2, ..., n$ and $j \neq i$, $g_i^{(2)} = g_i^{(1)} - 1$, and $g_j^{(2)} = g_j^{(1)}$. (Case 2-2) When $b_i^{1''}=1$ and $b_j^{1''}=0$, for $j=1, 2, ..., n$ and $j \neq i$, $g_i^{(2)} = g_i^{(1)} + 1$ and $g_j^{(2)} = g_j^{(1)}$.

Step 5: Follow Step 1 through Step 3, then calculate $p_i^{(2)} = g_i^{(2)} * 2 + b_i^{1''}$ for $i=1, 2, ..., n$. An embedded medium image is obtained.

Step 6: For the medium image after embedding, use Equation (5) to calculate the difference dp between the cover pixel value and its corresponding medium pixel value. To achieve the goal of image recovery, the difference value dp we are kept in the medium pixel value by Equation (6) where $p_{i,4} = \lfloor (p_{i,2} + p_{i,3})/2 \rfloor$.

$$dp = p_i - p_i^{(2)} \tag{5}$$

$$p_{i,4}^{(3)} = p_{i,4} + dp \tag{6}$$

Step 7: Repeat Step 5 and all difference value dp is embedded in the medium image and a stego-image is obtained. Then we can send it to the receiver, as shown in Phase 2 of Figure 1.

C. *Extracting and Recovering Phase*

As the receiver has the stego-image, steps for retrieving secret message and recovering image are shown as follows:

Step 1: Take stego-image and divide it into groups of n pixel value as. Then, divide the binary representation of $(p_1^{(2)}, p_2^{(2)}, ..., p_n^{(2)})$ into two parts, $G^{(2)} = (g_1^{(2)}, g_2^{(2)}, ..., g_n^{(2)})$ and $R'' = (b_1^{1''}, b_2^{1''}, ..., b_i^{1''})$.

Step 2: Use the group $G^{(2)}$ as an input of the function f in Equation (1) to retrieve the second secret digit s_2. Then use the indicator bit R'' to restore $G^{(2)}$ to $G^{(1)}$.

Step 3: After restoring $(g_1^{(1)}, g_2^{(1)}, ..., g_n^{(1)})$, substitute $g_i^{(1)}$ into Equation (1) and calculate to retrieve the first secret digit s_1.

Step 4: Finally, use Equation (7) to retrieve the embedded difference dp, and then substitute the difference value dp and the stego pixel value $p_i^{(2)}$ into Equation (8) to obtain the cover pixel value p_i. The cover image is restored after the steps, as shown in Phase 3 of Figure 1.

$$dp = p_{i,4}^{(3)} - \lfloor (p_{i,2} + p_{i,3}) / 2 \rfloor \tag{7}$$

$$p_i = p_i^{(2)} + dp \tag{8}$$

The embedding and extracting procedures will be explained with following examples. First take Figure 2 as an example for demonstrating embedding process of secret message. A 3×3 cover image is expanded by interpolation into the corresponding 6×6 extend image as shown in Figure 3. Assume that $n = 3$, the original cover image is grouped with every three pixel value. Thus, the first pixel group is (150, 158, 153). First convert each pixel value into binary and divide it into two parts, the first seven bits g_i and the least significant bit. So, $g_1 = (1001011)_2 = 75$, $g_2 = (1001111)_2 = 79$, and $g_3 = (1001100)_2 = 76$. Then we will carry out the first embedding of secret digit $s_1 = 3$ in 7-ary notational system as two steps. Step 1: substitute G=(75, 79, 76) into Equation (1) to get $F = 6$ while using both Equation (2) and Equation (3) to determine which g_i need to be adjusted. Step 2: Knowing that $D = 4$ which satisfies $D > n$, $g_{(2n+1)-D} - 1 = g_3 - 1 = 76 - 1 = 75$. According to $G^{(1)}$ = (75, 79, 75) and R'=(0, 0, 1), pixel group (75*2+0, 79*2+0, 75*2+1)= (150, 158, 151) are obtained.

After that the embedding of first digit, the second secret digit $s_2 = 5$ in 7-ary notational system can be hidden in the same manner by substituting pixel group (150, 158, 151) into Equation (1) to get $F_2 = 3$, while using Equation (2) and Equation (4) to determine which $g_i^{(1)}$ need to be adjusted. Knowing that $D = 2$ where $D \leq n$, $g_2^{(2)} = g_2^{(1)} + 1 = 79 + 1 = 80$. Besides, $R'' = (0, 1, 0)$ is then obtained as shown in TABLE I. An embedded medium pixel group (75*2+0, 80*2+1, 75*2+0) = (150, 161, 150) will be obtained. After calculating other cover image pixel in order, an embedded medium image is obtained, as shown in Figure 4.

To recover the image, Equation (5) calculates nine difference values which are (150-150, 158-161, 153-150, 154-154, 161-163, 165-162, 159-156, 157-155, 160-160) = (0, -3, 3, 0, -2, 3, 3, 2, 0). Then Equation (6) is used to embed the difference values into the medium pixels under the lower right position for every 2×2 block. After that, the stego-pixels at the the lower right position for every 2×2 block are (153, 154, 159, 156, 159, 166, 161, 159, and 160) and a stego-image is obtained as shown in Figure 5.

Now we will demonstrate the process of retrieving secret message and restoring image. Take Figure 5 as an example; first divide the stego-image into groups of three pixel values. The first group of stego-pixel value is (150, 161, 150). Take pixel values and convert it into binary. Then, divide it into two parts $G^{(2)}$ and R'', take the first seven bit representing as $g_1^{(2)} = (1001011)_2 = 75$, $g_2^{(2)} = (1010000)_2 = 80$, $g_3^{(2)} = (1001011)_2 = 75$. Retrieve $s_2 = 5$ after calculating by Equation (1), according to the indicator bit $R'' = (0, 1, 0)$ which belongs to Case 2-2, as shown in TABLE Ⅰ, $g_2^{(1)} = g_2^{(2)} - 1 = 80 - 1 = 79$.

Then take the restored $G^{(1)}$ = (75, 79, 75) after substituting into Equation (1) to retrieve secret digit $s_1 = 3$. After that, retrieve all difference values, and from Equations (7) and (8), we get the cover image as shown in Figure 2.

150	158	153
154	161	165
159	157	160

Figure 2. Cover image

150	154	158	155	153	153
152	153	159	157	159	156
154	157	161	163	165	165
156	156	159	161	162	163
159	158	157	158	160	160
159	158	157	157	160	160

Figure 3. Cover image after expansion (extend image)

150	154	161	155	150	153
152	153	159	157	159	156
154	157	163	163	162	165
156	156	159	161	162	163
156	158	155	158	160	160
159	158	157	157	160	160

Figure 4. Medium image after embedding

8	154	161	155	150	153
152	153	159	154	159	159
154	157	163	163	162	165
156	156	159	159	162	166
156	158	155	158	160	160
159	161	157	159	160	160

Figure 5. Stego-image

TABLE I. DEFINITIONS OF THE LAST BIT OF PIXEL VALUE IN PIXEL GROUP BY $n = 3$.

R''	Value modifications	
(0, 1, 1)	$g_1^{(2)} = g_1^{(1)} - 1$; $g_2^{(2)} = g_2^{(1)}$; $g_3^{(2)} = g_3^{(1)}$	(Case 2-1)
(1, 0, 0)	$g_1^{(2)} = g_1^{(1)} + 1$; $g_2^{(2)} = g_2^{(1)}$; $g_3^{(2)} = g_3^{(1)}$	(Case 2-2)
(1, 0, 1)	$g_1^{(2)} = g_1^{(1)}$; $g_2^{(2)} = g_2^{(1)} - 1$; $g_3^{(2)} = g_3^{(1)}$	(Case 2-1)
(0, 1, 0)	$g_1^{(2)} = g_1^{(1)}$; $g_2^{(2)} = g_2^{(1)} + 1$; $g_3^{(2)} = g_3^{(1)}$	(Case 2-2)
(1, 1, 0)	$g_1^{(2)} = g_1^{(1)}$; $g_2^{(2)} = g_2^{(1)}$; $g_3^{(2)} = g_3^{(1)} - 1$	(Case 2-1)
(0, 0, 1)	$g_1^{(2)} = g_1^{(1)}$; $g_2^{(2)} = g_2^{(1)}$; $g_3^{(2)} = g_3^{(1)} + 1$	(Case 2-2)
(0, 0, 0)	$g_1^{(2)} = g_1^{(1)}$; $g_2^{(2)} = g_2^{(1)}$; $g_3^{(2)} = g_3^{(1)}$	(Case 1)

III. EXPERIMENTAL RESULTS

There are four grayscale images with size of 256×256 taken as test images: Lena, Pepper, Jet, and Baboon. The performance efficiency of proposed scheme will be analyzed based on embedding rate and peak signal to noise ratio (PSNR).

The experimental results are performed to compare the embedding rate (bits) and image quality (PSNR) using Kuo *et al.*'s scheme and ours. For all images, our proposed method has the embedding rate $(2\log_2 7)/3$, which is about 1.87 bpp with average PSNR of 33.32 dB. From the TABLE Ⅱ, the average maximum embedding rate conducted by Kuo *et al.*'s scheme is 1.03 bpp. So our proposed scheme outperforms the Kuo *et al.*'s scheme because the gain is about 0.84 bpp. In other words, one cover pixel of our proposed scheme can carry 0.84 secret bits more than Kuo *et al.*'s scheme does. Although the average PSNR of our method is lower than that of Kuo *et al.*'s, the embedding rate of our method is higher. The average PSNR values of our proposed scheme is 33.32dB which is lower than that of Kuo *et al.*'s with the average PSNR values of 34.89 dB, but the stego-images still keep in acceptable image qualities and are imperceptible to human eyes. Therefore, our proposed scheme not only can maintain high embedding capacity but also preserve acceptable image quality, and reversibility.

TABLE II. COMPARISONS OF CAPACITIES AND PSNRS

Test Image	Kuo *et al.*'s Scheme[3]		Our Scheme	
	Bit_rate	PSNR	Bit_rate	PSNR
Lena	0.97	35.70	1.87	34.92
Pepper	0.78	37.84	1.87	37.51
Jet	0.81	35.93	1.87	30.31
Baboon	1.53	30.09	1.87	30.55
Average	1.03	34.89	1.87	33.32

IV. CONCLUSIONS

In this paper, reduplicated embedding of secret message can be done with a satisfying embedding rate around 1.87 bpp. Exploiting image expansion, R-EMD data hiding technique originally without reversibility can achieve reversibility as a result. The experimental results indicated that the capacity can be successfully increased with image quality at around 33.32 dB on average.

REFERENCES

[1] A. M. Alattar, "Reversible Watermark Using the Difference Expansion of a Generalized Integer Transform," *IEEE Transactions on Image Processing,* vol. 13, no. 8, pp. 1147-1156, August 2004.

[2] H. J. Kim, C. Kim, Y. Choi, S. Wang, and X. Zhang, "Improved modification direction methods," *Computers & Mathematics with Applications*, vol. 60, pp. 319-325, July 2010.

[3] W. C. Kuo and W. T. Hsu, "Reversible Data Hiding Scheme based on EMD Method," *2010 Taiwan Academic Network Conference* (in Chinese version).

[4] C. F. Lee and H. L. Chen, "A Novel Data Hiding Scheme Based on Modulus Function," *Journal of Systems and Software*, vol. 83, no. 5, pp. 832-843, May 2010.

[5] C. F. Lee, H. L. Chen, and H. K. Tso, "Embedding Capacity Raising in Reversible Data Hiding Based on Prediction of Difference Expansion," *Journal of Systems and Software*, vol. 83, no. 10, pp. 1864-1872, October 2010.

[6] C. F. Lee, L. Y. Chuang, and C. C. Chang, "Hiding Information Employing Reduplicating Embedding," *Proceedings of 2008 IEEE Asia-Pacific Services Computing Conference,* pp. 825-828, Dec. 2008.

[7] J. Mielikainen, "LSB Matching Revisited," *IEEE Signal Processing Letters,* vol. 13, no. 5, pp. 285-287, May 2006.

[8] Z. Ni, Y. Q. Shi, N. Ansari, and W. Su, "Reversible Data Hiding," *Proceedings of IEEE Transactions on Circuits and Systems for Video Technology*, vol. 2, pp. II-912-II-915, March 2003.

[9] J. Tian, "Reversible Data Embedding Using a Difference Expansion," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 8, pp. 890-896, August 2003.

[10] X. Zhang and S. Wang, "Efficient Steganographic Embedding by Exploiting Modification Direction," *IEEE Communications Letters*, vol. 10, pp. 781-783, Nov.2006.