An Improved Dual Image-based Reversible Hiding Technique Using LSB Matching

Yu-Lun Wang[1], Jau-Ji Shen[1], Min-Shiang Hwang[2,3]

(Corresponding author: Min-Shiang Hwang)

Department of Management Information Systems, National Chung Hsing University[1]
Department of Computer Science and Information Engineering, Asia University[2]
No. 500, Lioufeng Raod, Wufeng Shiang, Taichung 41354, Taiwan
Department of Medical Research, China Medical University Hospital, China Medical University[3]
No.91, Hsueh-Shih Road, Taichung 40402, Taiwan
(Email: mshwang@asia.edu.tw)

(Received Nov. 11, 2016; revised and accepted Jan. 19, 2017)

Abstract

A dual image technique is used as one of the data hiding method. Dual image copies an image to two same images. Through two images to embed or extract secret data, this technique significantly enhances image quality. A dual image technique is good or bad depending on the merits of its algorithm. This paper proposes a method to improve Lu et al. scheme. They use two pixels as a pair and choose two same images to embed, and then choose both two pixels to continue the procedure. We will depend on circumstances of the second pixel pair after previous embedding, and this case can increase the capacity. The experiment results show that our proposed scheme is effective.

Keywords: Data Hiding; LSB Matching; Reversible Hiding Technique

1 Introduction

With the advance of the technology, the speed of the internet becomes faster and faster, and multimedia spreads more easily. Traditional data encryption is through some mathematical operation to encrypt the plaintext into ciphertext [9]. And then the ciphertext is transferred to the receiver via channels. Hence, the ciphertext usually shows a period of distortion. An unauthorized third party can add a period of nonsensical text, so this will cause that the receiver cannot decrypt correctly. Therefore, a data hiding technique was invented [2, 4, 5].

The sender wants to send an image with embedded secret data. In the process of sending an image, the malicious third party is unable to recognize whether an image has embedded secret data or not. After receive the image, the receiver will use an extracted method to extract the secret data. A data hiding technique is also applicable on intellectual property protection of images; through secret data, it can announce the ownership of the images. However, whether image can completely restore or not becomes an important issue. A reversible data hiding technique was invented [8]. Reversible data hiding means that after extracting secret data, we also can get an original image. This is an important technique specially application to the domain which need an original image, such as medical or military images. A data hiding technique has two important criteria: Capacity and quality (PSNR) [1, 3, 6].

This paper will be described as follow. Section 2 will review Lu et al.'s scheme in detail. Section 3 will introduce out proposed method. Section 4 will show our experiment result and analyze Lu et al.'s scheme. Finally, Section 5 will make a conclusion of this paper.

2 Review of Lu et al.'s Scheme

2.1 Embedding Phase

In Lu et al.'s method [10], they will copy an image into two images to process, X and Y. First, they choose X_i, X_{i+1}, Y_i, Y_{i+1} to embed, four bits in each process. First two bits are embedded in first image X, and next two bits are embedded in the second image Y. They use Equation (1) to embed bits at first pixel in both images and to modify the pixel where in some case conditions are met. Then Equation (2) is used to embed bits at second pixel in both images. They also modify the pixels in Table 1.

$$\begin{cases} LSB(P_{i,j}) = \text{embed bit, don't need to change} \\ LSB(P_{i,j}) \neq \text{embed bit, } P_{i,j} - 1 \end{cases} \tag{1}$$

Table 1: Different cases of the embedding phase

Cases	First pixel = embed bit	Second pixel = embed bit	First Pixel change	Second pixel change
1	Yes	Yes	+1	0
2	Yes	No	-1	0
3	No	Yes	0	1
4	No	No	0	0

Table 2: Analysis of the LSB matching method in simultaneously hiding two pairs

Cases	**The Pixel value modification statuses**				**Pixel restoration statuses**	
	$X^1_{i,j}$	$X^1_{i,j+1}$	$Y^2_{i,j}$	$Y^2_{i,j+1}$	$P_{i,j}$	$P_{i,j+1}$
1	0	0	-1	0	V	
2	0	1	0	1		V
3	0	1	-1	0	V	
4	-1	0	0	0	V	
5	-1	0	0	1	V	
6	-1	0	-1	0	V	
7	1	0	1	0	V	

$$\begin{cases} LSB(\lfloor \frac{P_{i,j}}{2} \rfloor + P_{i,j+1}) = \text{embed bit}, \\ \qquad\qquad \text{don't need to change} \\ LSB(\lfloor \frac{P_{i,j}}{2} \rfloor + P_{i,j+1}) \neq \text{embed bit}, \\ \quad \text{if first pixel doesn't change, } P_{i,j+1}+1, \\ \quad \text{if first pixel changed, } P_{i,j}+1 \end{cases} \tag{2}$$

In some cases, we need to use a special rule to modify the pixels. These cases are shown in Tables 2, 3, and 4.

Table 3: Modification rule table: Pixel value modification statuses

Rules/Cases	$X^1_{i,j}$	$X^1_{i,j+1}$	$Y^2_{i,j}$	$Y^2_{i,j+1}$
1	0	0	-1	0
2	0	1	0	1
3	0	1	-1	0
4	-1	0	0	0
5	-1	0	0	1
6	-1	0	-1	0
7	0	1	0	0

Table 4: Modification rule table: The final modified camouflage pixel values

Rules/Cases	$X'_{i,j}$	$X'_{i,j+1}$	$X''_{i,j}$	$X''_{i,j+1}$
1	2	1	-1	0
2	0	1	0	-1
3	2	0	-1	0
4	-1	0	2	1
5	-1	0	2	0
6	-1	2	1	-1
7	-1	-1	1	2

Assume that we have to embed secret data in four pixels, 100, 97, 91, 110, respectively. And the secret data is 10011111. First, we calculate that LSB (100) of the first pixel of the image X is 0, and the secret data is 1. It is obviously not equal, so we use Equation (1) to get the modified pixel 99. Then we move on the next pixel of the image X and use Equation (2) to calculate whether LSB $(99/2+97)$ is equal to the second secret bit 0 or not. The result of this step is equal, so we get the result of the modified pixel is 99 and 97. We keep going to the first pixel of the image Y and calculate LSB (100). Then we use Equation (1) to get the modified pixel, 100, and calculate next pixel $Y_{i,j+1}$, by Equation (2). The result of the $Y_{i,j+1}$ is 97. After the procedure we get the modified pixels: 99, 97, 100, and 97, respectively. To check with Table 3, we find that the pixel changing rules are -1, 0, 0, 0, respectively. It conforms to case 4, so we need to further modify $X_{i,j}$, $X_{i,j+1}$, $Y_{i,j}$, $Y_{i,j+1}$, respectively. Finally, we get 99, 97, 102, 98 pixels that represent $X'_{i,j}$, $X'_{i,j+1}$, $Y'_{i,j}$, $Y'_{i,j+1}$, respectively. And then choose next pixel pair to do the embed procedure again, the next pixel pair is 91, 110. We still use Equation (1) to embed secret data in the $X_{i,j}$ and get the modified pixel, 91. Use Equation (2) to embed secret data in the $X_{i,j+1}$ and get the modified pixel, 110. And move on the pixel $Y_{i,j}$, $Y_{i,j+1}$ of the image Y, and, finally, we get the modified pixels are 91, 100, 91, 100, respectively. The pixel changing rule is 0, 0, 0, 0, which doesn't conform to any case in Tables 3 and 4, so we don't need to change the modified pixel. After finishing the embedding phase, send the modified stego-images X and Y to the receiver.

2.2 Extraction Phase

In the extraction phase, we use Equation (3) to extract the first pixel of the pixel pair and Equation (4) to extract the second pixel of the pixel pair. In the recover process, we can use Equation (5) to recover the original cover image.

$$LSB(P_{i,j}) = \text{secret data} \qquad (3)$$

$$LSB(\lfloor\frac{P_{i,j}}{2}\rfloor + P_{i,j+1}) = \text{secret data} \qquad (4)$$

$$\lfloor\frac{X_{i,j}+Y_{i,j}}{2}\rfloor = \text{original pixel} \qquad (5)$$

Assume that we get the modified images X and Y from the sender, we have to extract the secret data and recover the original images. First, we extract first pixel pair of the image X' and Y', 99, 97 and 102, 98, respectively. Then we use Equation (3) to extract first pixel, and LSB (99) is 1. Move on the second pixel of the pixel pair and use Equation (4) to extract the second pixel, and LSB $(99/2+97)$ is 0. After extracting the pixel of the image X', we continue to extract the pixel of the image Y', so we still use Equation (3) to extract the first pixel of the pixel pair of the image Y'. LSB (102) is 0 and Equation (4) is used to extract the second pixel of the pixel pair, and LSB (102/2+98) is 1. After extracting process, we can extract secret data 1001 and move on to the next pixel pairs, 91, 110 and 91, 110. We use Equation (3) to extract the first pixel of the pixel pair, and LSB (91) is 1; and then Equation (4) is used to extract second pixel of the pixel pair, and LSB $(91/2+110)$ is 1. The extraction result of the pixel pair of the image X' is 11; because the pixel pair of the image X' and the pixel pair of the image Y' are the same, we can extract same secret data 11. The extraction result of the second pixel pair is 1111, so we can correctly extract the secret data just like what the sender embeds in the stego-image. After extraction phase is finished, we can use Equation (5)) to recover the original image. The first pixel in both stego-images X' and Y' are 99 and 102, so we use Equation (5) to recover the first pixel, and $(99+102)/2$ is 100; and after finishing the recover procedure, we can get four pixels as 100, 97, 91, 110.

3 The Proposed Method

In our proposed scheme, the idea of Lu et al.'s method is not always the case, so we will determine whether we use the second pixel of the pixel pair as the first pixel for next embedding phase or not. The condition is as follows: After embedded in two pixels over, we use Equation (6) to decide whether we will use the second pixel or not.

$$|X_{i,j+1} - Y_{i,j+1}| \qquad (6)$$

In the embedding phase, we use Equation (1) to embed secret data in the first pixel of the pixel pair and use Equation (2) to embed secret data in the second pixel of the pixel pair. After finishing the embedding process on the current pixel pair, we use Equation (6) to determine whether we will use the second pixel as the first pixel of the next embedding process or not. If the result of Equation (6) is not equal to zero, we use a new pixel pair to do next embedding process. When the embedding phase is finished, the extraction phase will start after the sender sends the stego-image and the receiver wants to get the correct secret data.

In the beginning, we choose the first pixel pair to extract the secret data, and then we use Equation (3) to extract the first pixel of the pixel pair, and use Equation (4) to extract the second pixel of the pixel pair. After extracting the secret bits correctly, we use Equation (6) to decide whether we will use the second pixel to do next extracting process or not. If the result of Equation (6) is less than 3 or equal to 0, we determine to use the second pixel as the first pixel of the pixel pair of the next extraction process. Otherwise, we use a new pixel pair for next extraction process. And before starting extraction process, we need to recover the second pixel of the pixel pair first; if not, we may not correctly extract the secret bit. Assume that we have four pixels to embed, 100, 97, 91, 110, respectively, and the secret bits is 010111111100. First, we use Equation (1) to embed a secret pixel in the first pixel of image X, and the result of the LSB (100) is 0, which is equal to the secret pixel which we want to embed. Then move on the second pixel of the image X, and use Equation (2) to embed a secret bit; the result of the LSB $(100/2+97)$ is 1, which is equal to the secret bit 1. Then after finishing the first process, we get the modified pixels are 100, 97, 100, 97, respectively. We use Equation (6) to determine which pixel will be used in the next embedding process; since 97 - 97 = 0, we use 97 and 91 to do next embedding process; the result of the LSB (97) is 1, and the result of the LSB $(97/2+91)$ is 1.

After finishing this embedding process, we use Equation (6) again to decide next embedding process pixels; since $91-91$ is 0, we use 91, 110 to do next embedding process. The result of the LSB (91) is 1, and the result of the LSB $(91/2+110)$ is 1. Then move on the pixel pair of the image Y; LSB (91) is 1 and not equal to the secret bit 0, so the modified pixel is 90. The second pixel LSB $(90/2+110)$ is 1, and is not equal to the secret bit 0, so the modified pixels of image Y is 92, 110, respectively. Finally, we send the modified pixels: 100, 97, 91, 110 and 100, 97, 92, 110 to the receiver.

After the receiver receives the stego-image, we can start the extraction and recovery phases. First we take the first and second pixels of the images X and Y as the pixel pair and use the recover result of the second pixel to extract the secret bit. Starting from image X, the result of the LSB (100) is 0, and the result of the LSB $(100/2+97)$ is 1. And moving on the image Y, the result of the LSB (100) is 0, and the result of the LSB $(100/2+97)$ is 1. After finishing the extraction process, we use Equation (6) to determine the pixel pair of the next extraction process. The result of the Equation (6) is 0, so we use 97, 91 and 97, 92 to do next extraction process. The result of LSB

Table 5: Analysis of the improved LSB matching method in simultaneously hiding two pairs

Cases	**The Pixel value modification statuses**				**Pixel restoration statuses**	
	$O^1_{i,j}$	$O^1_{i,j+1}$	$O^2_{i,j}$	$O^2_{i,j+1}$	$X_{i,j}$	$X_{i,j+1}$
1	0	0	-1	0	V	V
2	0	1	0	1		V
3	0	1	-1	0	V	V
4	-1	0	0	0	V	V
5	-1	0	0	1	V	V
6	-1	0	-1	0	V	V
7	1	0	1	0	V	V
8	0	0	0	1		V
9	1	0	0	1		V
10	0	1	1	0		V
11	0	1	0	0		V

(97) is 1, and the result of LSB $(97/2 + 91)$ is 1; and the result of the image Y LSB (97) is 1, and the result of LSB $(97/2 + 91)$ is 1. Then we use Equation (6) to calculate 92 - 91 is 1; since it is less than 3 we determine that the pixel pairs of the next extraction are 91, 110 and 92, 110. The result of the LSB (91) is 1, and the result of the LSB $(91/2+110)$ is 1. The result of the pixel pair of the image Y LSB (92) is 0, and the result of LSB $(92/2 + 110)$ is 0. Finally, we extract the secret bits 010111111100 and use Equation (5) to correctly recover the original image. We also propose a special case as shown in Tables 5, 6, and 7.

Table 6: Modification rule table: Pixel value modification statuses

Rule/Case	$O^1_{i,j}$	$O^1_{i,j+1}$	$O^2_{i,j}$	$O^2_{i,j+1}$
1	0	0	-1	0
2	0	1	0	1
3	0	1	-1	0
4	-1	0	0	0
5	-1	0	0	1
6	-1	0	-1	0
7	1	0	1	0
8	0	0	0	1
9	1	0	0	1
10	0	1	1	0
11	0	1	0	0

Table 7: Modification rule table: The final modified camouflage pixel values

Rule/Case	$X'_{i,j}$	$X'_{i,j+1}$	$X''_{i,j}$	$X''_{i,j+1}$
1	2	3	-1	-2
2	0	3	0	-3
3	2	-1	-2	2
4	-1	-2	2	3
5	-1	2	2	-2
6	-1	4	1	-3
7	-1	-3	1	4
8	0	-2	0	3
9	1	-2	0	3
10	0	3	1	-2
11	0	3	0	-2

4 Experiments Result and Analysis

There are two criteria in data hiding area, quality and capacity. We use the grayscale images in the experiment. The source of grayscale images is in the Waterloo Greyscale Set 2 database[1]. The images are the TIF format standard images with 512×512 pixels. We use a peak signal-to-noise ratio (PSNR) to quantify image quality as follows:

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[P_{i,j} - X_{i,j}]^2$$

$$PSNR = 10 \times \log_{10}[\frac{255^2}{MSE}]$$

The PSNR value is the higher the better, and the capacity is the bigger the better, where m and n are the images sizes. We use Matlab 8.5.0.197613 (R2015a) and assume that the secret bit which wants to be embedded in the cover image is all of one. The results in Table 8 show that the capacity of the proposed method is better than the original method, and the PSNR is worse than the original method [10].

[1] http://links.uwaterloo.ca/Repository.html

Table 8: The image and total hidden capacity comparison table

SChemes	Images	Lena	Mandrill	Pepper	Barbara	Boat	Goldhill	Zelda	Washsat
Lu et al. [4]	PSNR(1)	49.13	47.95	49.11	49.14	49	49.17	49.14	49.13
	PSNR(2)	49.12	49.15	49.08	49.11	49.07	49.09	49.09	49.09
	Capacity	524,288	522,996	524,192	524,288	524,208	524,288	524,288	524,276
Proposed Method	PSNR(1)	40.97	40.94	40.99	40.98	40.96	40.98	41.02	41.23
	PSNR(2)	41.30	41.34	41.23	41.20	41.56	41.24	41.03	41.22
	Capacity	617088	618977	608877	610372	632797	613288	599905	614372

5 Conclusion

We proposed an improved method to have a better capacity. The idea of the proposed method is to utilize the current embedded second pixel as the first pixel of next time embedding. However, the quality of the image slightly decreases. How to get both quality and capacity better is the feature work.

Acknowledgments

This research was partially supported by the Ministry Of Science and Technology, Taiwan (ROC), under contract no.: MOST 104-2221-E-468-004 and MOST 105-2410-H-468-009.

References

[1] L. C. Huang, T. H. Feng, M. S. Hwang, "A new lossless embedding techniques based on HDWT," *IETE Technical Review*, vol. 34, no. 1, pp. 40–47, 2017.

[2] B. Jana, "Dual image based reversible data hiding scheme using weighted matrix," *International Journal of Electronics and Information Engineering*, vol. 5, no. 1, pp. 6–19, 2016.

[3] F. Li, Q. Mao, C. C. Chang, "A reversible data hiding scheme based on IWT and the sudoku method," *International Journal of Network Security*, vol. 18, no. 3, pp. 410–419, May 2016.

[4] T. C. Lu, C. Y. Tseng, K. M. Deng, "Reversible data hiding using local edge sensing prediction methods and adaptive thresholds," *Signal Processing*, vol. 104, pp. 152–166, Nov. 2014.

[5] T. C. Lu, C. Y. Tseng, J. H. Wu, "Dual imaging-based reversible hiding technique using LSB matching," *Signal Processing*, vol. 108, pp. 77–89, Mar. 2015.

[6] S. Manoharan, D. RajKumar, "Pixel value differencing method based on CMYK colour model," *International Journal of Electronics and Information Engineering*, vol. 5, no. 1, pp. 37–46, 2016.

[7] J. Mielikainen, "LSB matching revisited," *IEEE Signal Processing*, vol. 13, no. 5, pp. 285–287, May 2006.

[8] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Transactions on Circuits System Video Technology*, vol. 13, no. 8, pp. 890–896, Aug. 2003.

[9] G. Tuychiev, "The encryption algorithm GOST28147-89-PES16-2 and GOST28147-89-RFWKPES16-2," *International Journal of Electronics and Information Engineering*, vol. 6, no. 1, pp. 1–11, 2017.

[10] S. Zhang, T. Gao, L. Yang, "A reversible data hiding scheme based on histogram modification in integer DWT domain for BTC compressed images," *International Journal of Network Security*, vol. 18, no. 4, pp. 718–727, July 2016.

Biography

Yu-Lun Wang study in the Department of Management Information Systems, Chung Hsing University.

Jau-Ji Shen received his Ph.D. degree from National Taiwan University in 1988. His research interests include digital image, software engineering, information security, and data base technique. His work experiences include the Director of National Formosa University Library and the Associate Dean of Management School in Chaoyang University of Technology. Now, he is a professor in the Department of Management Information Systems, National Chung Hsing University.

Min-Shiang Hwang received the Ph.D. degree in computer and information science from the National Chiao Tung University, Taiwan in 1995. Dr. Hwang was the Chairman of the Department of Information Management, Chaoyang University of Technology (CYUT), Taiwan, during 1999-2003. He was a distinguished professor and Chairman of the Department of Management Information Systems, NCHU, during 2003-2011. He obtained the 1997, 1998, 1999, 2000, and 2001 Excellent Research Award of National Science Council (Taiwan). His current research interests include information security, cryptography, image compression, and mobile computing. Dr. Hwang has published over 200+ articles on the above research fields in international journals.