

Robust Image Hiding Method

Yu-An Ho¹, Yung-Kuan Chan², Chwei-Shyong Tsai², Yen-Ping Chu¹

Institute of Computer Science, National Chung Hsing University,
No. 250, Kuokuang Rd., Taichung, Taiwan, R.O.C.¹

Department of Management Information Systems, National Chung Hsing University,
No. 250, Kuokuang Rd., Taichung, Taiwan, R.O.C.²

Abstract

Image hiding is hiding a secret image embedding to a cover image. This paper intends to propose an image hiding method. First, a secret image is compressed by BST. Meanwhile, dividing the compressed data into different ranks depends on the significance. Then, the cover image is transformed into frequency domain data by discrete wavelet transformation. The dominant part of compressed data is hidden to the lower frequency sub-band which isn't easy to destroy; then hide the trivial data into the higher frequency sub-band. Finally, when the dominant part of the compressed secret image data is damaged, the error data will be modified by the hamming code method. In addition, when the trivial data of the compressed data of the secret image is damaged, it would affect the data relating to the block only. The proposed method has 2.25 bits/pixels of the hiding capacity. The image quality of stego-image and extracted secret image are measured in terms of peak-signal-to-noise ratio and discusses robustness of noise and compression.

Keywords: Image hiding, Data hiding, Stego-image, Secret-image, LSB

1. Introduction

With the rapid development of the Internet and multimedia technologies, one can easily conceal a large amount of information in various kinds of digital media. Image hiding embeds a secret image in another image which is usually called a cover image and the hiding procedure is named embedding. After hiding the secret image, the cover image becomes a stego-image [1, 4, 6]. The purpose of image hiding is to embed the secret image from unauthorized people seeking it out.

In recent years, a lot of image hiding technologies [2, 7] have been proposed already. Thien et al. [2] submitted a high hiding-capacity method. It could embed by digit by digit on real-time and reach a higher visual quality than LSB to avoid artificial edges which is caused by LSB. Then, Chan et al. [3] also submitted Optimal Pixel Adjustment Process, OPAP. The concept is according to the method which was submitted by Wang et al. [7]. This

method adjusts the last 4 bits of the image to do an optimal pixel process and computes simply. These technologies mostly only focus on the first four requirements, but omit the requirement of robustness. As a result, this paper proposes a BST-based robust image hiding method (BBRIHM) with robustness.

When the dominant compression data of the secret image is damaged, the BBRIHM makes use of the Hamming code technology [8] to detect them and attempts to recover them. If the subordinate compression data of the secret image is changed, the BBRIHM will also try to make only the block related to this data in the secret image be affected.

2. Data Embedding

The BBRIHM embeds a secret image I_s in a cover image I_c . Here I_s and I_c are all gray-level scale images. In order to reduce the data size of I_s , the BBRIHM adopts BST to encode I_s and classifies the compression data in accordance with their importance. After that, it transforms I_c into a frequency domain image I_f with DWT, and then hides the dominant data of the compression data of I_s in lower frequency sub-band of I_f and the trivial data in the higher frequency sub-band.

2.1. Base Switching Transformation

Generally, the colors of most neighboring pixels in an image are very close. When the pixel colors of a certain image block are all similar, one can use only the minimal pixel color and the differences between the minimal color and all pixel colors in the block to describe the content of this block. Since the color differences among the pixels in the block are all minute, only a few bits are enough to describe each color difference. As a result, it can save a great amount of memory space. We call it base switching transformation method (BST) [9].

The BBRIHM first partitions I_s into small blocks of 3×3 pixels. We named them secret blocks. Suppose C_M and C_m are the maximal and minimal pixel colors in a certain secret block B . It is enough to record the color difference of each pixel in B with only $\lceil \log_2(C_M - C_m) \rceil$ bits. In the BST, if the minimal pixel color C_m has been changed, all the pixel colors of the whole block will also be altered after decompressing. Therefore, C_m is the dominant

data of the compression data.

2.2. Hamming Code Appending

Hamming codes [8] can correct single bit errors or detect two bit errors in a unit data. A unit data is called a code word. The Hamming rule is expressed by the following inequality:

$$2^p \geq p+c+1,$$

where p is the number of parity bits and c is the number of data bits. Such a code is said to be a $(p+c, c)$ Hamming code.

Since C_m is an eight-bit data, $c = 8$. Let $p=4$; $2^4 \geq (8+4)+1=13$. Let $b_1b_2...b_8$ be the eight bits of C_m . p is number of parity bits which needs adding to C_m . It means that four extra parity bits $p_1p_2p_3p_4$ are used to detect whether there is something wrong with $b_1b_2...b_8$. We call $C_h = p_1p_2p_3p_4b_1b_2...b_8$ Hamming code data bits.

Table 1: The corresponding relation of bits

$p_1p_2p_3p_4b_1b_2...b_8$ and $e_1e_2e_3e_4$																		
<div><div><div>$e_1e_2e_3e_4$</div><div>error</div><div>equation</div><div>bit</div></div></div>	$e_1e_2e_3e_4$	0	1	2	4	8	3	5	6	9	10	12	7	11	13	14	15	
	No error	p_1	p_2	p_3	p_4	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	-	-	-		
	e_1	0	1	0	0	0	0	0	0	1	1	1	0	1	1	1		
	e_2	0	0	1	0	0	0	1	1	0	0	1	1	0	1	1		
	e_3	0	0	0	1	0	1	0	1	0	1	0	1	1	0	1		
e_4	0	0	0	0	1	1	1	0	1	0	0	1	1	1	0			

The BBRIHM makes the dominant data bits corresponding to the 4 equations e_1, e_2, e_3 , and e_4 . Table 1 illustrates their corresponding relations. Equation e_i on Table 1 corresponds to the following equation (i), where \oplus is the bitwise EXCLUSIVE OR operation.

$$\begin{cases} b_4 \oplus b_5 \oplus b_6 \oplus b_8 \oplus p_1 = 0 \dots\dots\dots(1) \\ b_2 \oplus b_3 \oplus b_6 \oplus b_7 \oplus p_2 = 0 \dots\dots\dots(2) \\ b_1 \oplus b_3 \oplus b_5 \oplus b_7 \oplus b_8 \oplus p_3 = 0 \dots\dots\dots(3) \\ b_1 \oplus b_2 \oplus b_4 \oplus b_7 \oplus b_8 \oplus p_4 = 0 \dots\dots\dots(4) \end{cases}$$

2.3. Data Hiding

When reconstructing I_s , if the compression data of B has already been destroyed, the BBRIHM will make use of the neighboring blocks of B to estimate the pixel colors of B . To guarantee that there are some correct neighboring blocks of B able to estimate the pixel colors of B , the BBRIHM classifies all secret blocks of I_s into classes 1 and 2 shown in Fig. 2.

1	2	1	2
2	1	2	1
1	2	1	2
2	1	2	1

Fig. 2: Two classes of secret blocks

Discrete wavelet transform (DWT) technique [5] can transform a spatial domain image into a frequency domain image. The DWT coefficients on lower frequency bands are more insensitive to the lossy image processing operations. Therefore, the BBRIHM transforms I_c into a frequency domain image I_f , and hides the dominant and subordinate data of secret blocks in the lower frequency DWT coefficients and the trivial data in the higher frequency DWT coefficients of I_f .

The BBRIHM hides the dominant data $D_h = p_1p_2p_3p_4b_1b_2...b_8$ and the subordinate data $S = s_1s_2s_3$ of the compression data of a secret block B in LL and the trivial data D_t in other frequency bands. To prevent the secret image from unauthorized people figuring out any messages about the secret image, the BBRIHM takes a private key K as the seed of a random number generator G which will be used to specify the coefficients veiling the dominant and subordinate data of the compression data of a secret block.

To embed the compression data of a secret block B , the BBRIHM employs the random number generator G to randomly selects 15 DWT coefficients $C_1^{LL}, C_2^{LL}, \dots, C_{15}^{LL}$ from LL to hide the D_h and S of B . Here if B is in class 1, the 4-th rightmost bits of the 15 coefficients must not have concealed any secret data; otherwise, the 3-th rightmost bits of the 15 coefficients must not have embedded any secret data. Thereafter, the 15 data bits of D_h and S respectively substitute for the 4-th rightmost bits of the 15 coefficients if B is in class 1; otherwise, they replace the 3-th rightmost bits of the 15 coefficients. Since the 4-th rightmost bits of the 15 coefficients is more insensitive to the lossy image operations than the 3-th rightmost bits, the D_h and S of a secret block in class 1 is more robust than those in class 2 in resisting the damage caused by lossy image operation.

Each of the 15 DWT coefficients $C_1^{LL}, C_2^{LL}, \dots, C_{15}^{LL}$ corresponds to three DWT coefficients respectively in sub-bands LH, HL , and HH . C_i^{LL} is related to C_i^{LH}, C_i^{HL} , and C_i^{HH} which have the same location with C_i^{LL} in LL, LH, HL , and HH respectively.

Let $|D_t|$ be the size of D_t . If $|D_t| \leq 45$, the BBRIHM embeds the whole D_t in the 45 DWT coefficients; otherwise, it only takes the leftmost 45 bits of D_t out to be covered with the 45 DWT coefficients, and abandons the rest of D_t . The BBRIHM hides the 15 leftmost bits of D_t in the 15 corresponding DWT coefficients in $C_1^{LH}, C_2^{LH}, \dots, C_{15}^{LH}$, the 16-th to 32-th leftmost bits in $C_1^{HL}, C_2^{HL}, \dots, C_{15}^{HL}$, and the 33-th to 45-th leftmost bits in $C_1^{HH}, C_2^{HH}, \dots, C_{15}^{HH}$. If B is in class 1, the BBRIHM cloaks D_t in the second rightmost bits of

the 45 DWT coefficients; else, in the rightmost bit of the DWT coefficients.

3. Data Extracting

To extract the secret image I_s from the stego-image, the BBRIHM uses the same private key K as the seed of the random number generator G , and transforms the stego-image with spatial format into the image I_{st} with frequent format by DWT. During extracting the secret block B , the BBRIHM finds out C_1^{LL} , C_2^{LL} , ..., C_{15}^{LL} from LL via G , which ships D_h and S of B . If B is in class 1, D_h and S are covered with the 4-th rightmost bits of the 15 coefficients; otherwise, with the 3-th rightmost bits of the 15 coefficients. After searching out C_1^{LL} , C_2^{LL} , ..., C_{15}^{LL} , thereupon, C_1^{LH} , C_2^{LH} , ..., C_{15}^{LH} , C_1^{HL} , C_2^{HL} , ..., C_{15}^{HL} , C_1^{HH} , C_2^{HH} , ..., C_{15}^{HH} are also decided. If $|D_t| \leq 45$, D_t can be obtained from C_1^{LH} , C_2^{LH} , ..., C_{15}^{LH} , C_1^{HL} , C_2^{HL} , ..., C_{15}^{HL} , C_1^{HH} , C_2^{HH} , ..., C_{15}^{HH} ; else all the binary digits after the 45-th leftmost bits of D_t are given 0-bits.

For the BBRIHM, if the compression data of one secret block is changed, only the content of the block may be influenced. D_m is the dominant data of the block; suppose it is destroyed, every pixel color in this block is also converted. Therefore, the BBRIHM will try to re-estimate the block if D_h which is blemished and may make D_m undependable. Provided that one among the equations e_1 , e_2 , e_3 , and e_4 is 1, some bits in D_h have been changed; then the BBRIHM intends to revise D_m by side-match method [10].

Let $p_1 p_2 \dots p_9$ be the 9 pixels of B , and $u_1 u_2 u_3 d_1 d_2 d_3 l_1 l_2 l_3 r_1 r_2 r_3$ be the neighbors of the top, bottom, left and right pixels of B . Fig. 2 shows the neighborhood of B . When an error detected via Hamming code, the BBRIHM revises the secret block by side match method. The method estimates $p_1 p_2 \dots p_9$ by: $p_1 = (u_1 + l_1)/2$, $p_2 = u_2$, $p_3 = (u_3 + r_1)/2$, $p_4 = l_2$, $p_5 = r_2$, $p_6 = (l_3 + d_1)/2$, $p_7 = d_2$, $p_8 = (r_3 + d_3)/2$, $p_9 = \sum_{i=1}^8 p_i / 8$.

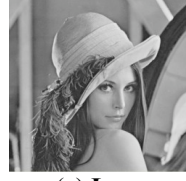
	u_1	u_2	u_3	
l_1	p_1	p_2	p_3	r_1
l_2	p_4	p_5	p_6	r_2
l_3	p_7	p_8	p_9	r_3
	d_1	d_2	d_3	

Fig. 2: The neighborhood of a block in a secret image

4. Experience

The purpose of this section is to investigate the performances of the BBRIHM by experiments. These experiments take image Lena as the cover image and image Airplane as the secret image. Fig. 3

demonstrates both images Lena and Airplane respectively with 512×512 and 384×192 pixels. PSNR (peak of the signal-to-noise) is often used to measure the similarity between the two images.



(a) Lena



(b) Airplane

Fig. 3: The testing images



(a) Lena



(b) Airplane

Fig. 4: The stego-image and secret images in experiment 1

In the first experiment, Airplane is hidden in Lena by the BBRIHM method. The experimental result demonstrates that the PSNR between the cover image Lena and stego-image is 31.37 dB, and the data hiding rate is 2.25 bits/pixel. After processing Optimal Pixel Adjustment Process [2, 3], PSNR can increase to 33.88 dB. Fig. 4 shows the stego-image and extracted secret image. Table 2 shows PSNRs of stego-image of simple LSB substitution (LSB), Optimal Pixel Adjustment Process (OPAP) [2, 3] and BST-based robust image hiding method (BBRIHM). It can find that the stego-image quality of BBRIHM is between LSB and OPAP.

Table 2: PSNRs of stego-image (dB)

	LSB	OPAP	BBRIHM
PSNR	32.08	34.44	33.88

Deal with the software through PhotoImpact 8, by putting noises into in stego-images in the second experiment. While dealing with this, the value that we establish the parameter variance respectively is 1%, 2% and 3%. As to the original cover image (Fig. 3 (a)), PSNRs is respectively 33.77, 33.36 and 32.75 dB; and stego-image in Fig. 4 (a), PSNRs is respectively 49.62, 42.91 and 39.16 dB.

Later, extract the hidden data from the stego-image which has already joined noises and rebuild the secret image. This experiment revises the mistake of D_{min} with HH and HS methods separately, too. Table 3 shows the image quality (PSNR) of these rebuilt secret images. Results of columns HH and HS are the PSNR revised with HH and HS methods.

Table 3: PSNRs of the secret image of reconstruction of the second experiment (dB)

Variance	LSB	OPAP	HH	HS
1%	14.68	14.62	15.53	17.53
2%	11.20	11.21	12.21	14.15
3%	9.78	9.79	10.65	12.58

The HS correction method can offer the revision result better than HH correction method. Because Hamming code can only revise the data that only contains a wrong bit correctly, whether or measure only with wrong bit under two correctly. Table 4 proves the results that revise with HH correction of D_{\min} of blocks in class 1 and class 2. It shows that the data was damaged more seriously; revise the D_{\min} of blocks with HH correction method would get the worse revision results. Especially D_{\min} of blocks in class 2 is hidden in the relatively apt damaged place, if we revised the mistake of D_{\min} of blocks in class 2 with HH correction method, we get this result instead.

Table 4: The result of HH correction method, to stego-image including noisy

Variance	1%	2%	3%
No. of error bits in D_m of the blocks in class 1 without error correction	1310	2853	4485
No. of error bits in D_m of the blocks in class 1 with HH correction	934	2560	4433
No. of error bits in D_m of the blocks in class 2 without error correction	2680	5884	9011
No. of error bits in D_m of the blocks in class 2 with HH correction	2343	5965	9314

Deal with JPEG 2000 compression method with PhotoImpact 8 to compress with lossy compression with stego-image. The parameter quality was established as 97, 98, ..., 100 respectively. This experiment comes to having wrong D_{\min} to revise with HH and HS method separately too. Table 5 demonstrates the result of this experiment. The result in columns HH and HS are the rebuilt secret image quality that was revised with HH and HS method separately. CR is the compression result after JPEG 2000 compression method compressing the stego-image. The following definition is shown:

$$CR = \frac{\text{The data size of the original image}}{\text{The data size of compressed image}}$$

Table 5: Demonstrates the result of this experiment

Quality	CR	PSNR (dB)			
		LSB	OPA	HH	HS
100	2.17	14.82	14.50	15.89	17.94
99	3.51	11.49	11.03	11.82	13.98
98	3.66	10.46	10.64	11.76	13.76
97	8.35	9.63	8.97	9.13	10.50

5. Conclusions

The demand of robustness in image hiding filed is not requested as strongly as it is in watermarking filed. As a result, image hiding method usually neglects the basic demand of robustness. Because of adverse circumstances, when the stego-image is transmitted to the network it would often cause the data transmitted to damage. Also in order to reduce the data transmission time, it is common to compress the image which is going to transmit with lossy compression before transmitting. As a result, this report proposes image hiding method of BBRIHM, this method not only can meet the traditional basic demands of image hiding, but also can resist noise and certain damage in the secret image that is caused by lossy compression. And this method can be offered to 2.25 / the high data hiding rate of pixel.

6. References

- [1] C. C. Chang, T. S. Chen and L. Z. Chung, "A Steganographic Method Based upon JPEG and Quantization Table Modification," *Information Sciences*, Vol. 141, No. 1, 2002, pp. 123-138.
- [2] C. C. Thien and J. C. Lin, "A Simple and High-Hiding Capacity Method for Hiding Digit-by-Digit data in images based on Modulus function," *Pattern Recognition*, Vol. 36, No. 13, 2003, pp. 2875-2881.
- [3] C. K. Chan, L. M. Cheng, "Hiding data in image by simple LSB substitution", *pattern recognition*, Vol. 37, No. 3, 2004, pp. 469-474.
- [4] J. B. Feng, I. C. Lin, C. S. Tsai, and Y. P. Chu, "Reversible Watermarking: Current Status and Key Issues," *International Journal of Network Security*, Vol. 2, No. 3, 2006, pp. 161-170.
- [5] J. Wang, L. Ji, "A Region and Data Hiding Based Error Concealment Scheme for Images," *IEEE Transformations on Consumer Electronics*, Vol. 47, No. 2, 2001, pp. 257-262.
- [6] N. F. Johnson, and S. Jajodia, "Steganography: Seeing the Unseen," *IEEE Computer*, February 1998, pp. 26-34.
- [7] R. Z. Wang, C. F. Lin and J. C. Lin, "Image Hiding by Optimal LSB substitution and genetic algorithm," *Pattern Recognition*, Vol. 34, No. 3, 2001, pp. 671-683.
- [8] R. W. Hamming, "Error Detecting and Error Correcting Codes," *Bell System Tech. j.*, Vol. 29, April 1950, pp. 147-160.
- [9] T. J. Chuang and J. C. Lin, "New approach to image encryption," *Journal of Electronic Imaging*, Vol. 7, No. 2, April 1998, pp. 350-356.
- [10] T. Kim, "Side Match and Overlap Match Vector Quantizers for Images," *IEEE Transactions on Image Processing*, Vol. 1, April 1992, pp. 170-185.