A High Payload Steganography Scheme for Color Images Based on BTC and Hybrid Strategy

Hon-Hang Chang[1] Yung-Chen Chou[2] Chiung-Chen Tseng[2] Timothy K. Shih[1]

[1] Department of Computer Science and Information Engineering, National Central University,
Taoyuan 32001, Taiwan, ROC
{sicachang, timothykshih}@gmail.com

[2] Department of Computer Science and Information Engineering, Asia University,
Taichung 41354, Taiwan, ROC
{yungchen, ilxgaau}@gmail.com

Received 12 July 2014; Revised 10 February 2015; 17 May 2015

Abstract. Block Truncation Coding (BTC) is one of the popular compression techniques in the image data hiding because its low computation cost and easy to implement. For a color image, there exist three pairs of high mean and low mean with three bitmaps. Common bitmap is a good way for saving the size of compression code. Chang *et al*. presented a data hiding method to conceal secret data into color BTC compression code by rearranging high mean and low mean encoding sequences. The proposed method is attempting to embed more secret data into the color BTC compression code. In this paper, a novel method is proposed to encrypt the amount of the element '1'. If the amount is even number, then the data is implied to a secret bit '0', otherwise, will be implied to a secret bit '1'. Furthermore, the Less Significant Bit (LSB) of all high mean and low mean pairs can also be used to embed the secret data. Actually, in the proposed method, each block conceals at least 10 secret bits.

Keywords: Block Truncation Coding (BTC), data hiding, color image, common bitmap

1 Introduction

As the development of computer networks and information techniques, the confidential data delivery via Internet has also increased dramatically. In order to protect confidential data delivered over public computer networks, both the Crypto technique and the Steganography technique, are the most useful methods to maintain the security of confidential data delivery. Crypto technique will encrypt the confidential data with meaningless (also called "cipher text") and seem to be random noise data. For instance, the confidential data might always be failed in the delivery process, if the transmission was detected by network administrators.

Steganography technique is another one of the confidential data delivery. The main idea is to use multimedia for carrying the encrypted confidential data. Then the stego media is sent to receiver over a public computer network. Since the stego media is very similar to the original, unexpected user will not easily notice stego media. Here, the covered media can be text, audio, video, and image.

Digital image is one of the most popular multimedia in the daily life. In this paper, we focus on data hiding in a digital image. Data hiding technique in a digital image can be briefly classified into three categories namely, spatial domain, frequency domain, and compression domain. The spatial domain data hiding technique directly adjusts cover all image's pixels for implying confidential data [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12]. Spatial domain data hiding technique is easy to implement with less computation cost compared with frequency domain and compression domain techniques.

Frequency domain method is to convert a cover image into frequency coefficients by using transformation technique (e.g., Digital Wavelet Transformation, Digital Fourier Transformation, Digital Cosine Transformation, etc.). Then, the confidential data is concealed into the specified frequency coefficients. After that, the modified coefficients are transformed back to spatial domain as a stego image [13] [14] [15]. The stego image generated in frequency domain has better visual quality than a stego image generated in spatial domain, that is, the stego image generated by frequency domain method is very similar to its original image. The high computation cost is a drawback of the frequency domain methods.

Digital image compression provides a good solution for saving the cost of storage and transmission. Because the tiny distortions caused by data embedding in an image are difficult to discover by human eyes. Lossy compression technique provides high compression rate to significantly reduce the size of image storage requirements (e.g., Vector Quantization [16], Side-match vector quantization [17], Block Truncation Coding [16]). Contrary, lossless compression technique is focused on maintaining the visual quality of decompression image than the compression rate. The compression domain data hiding method tries to modify the compression process according to data hiding rules [19] [20] [21]. Hence, the compression domain data hiding method does not only reduce the size of cover image but also carries confidential data.

The visual quality of the stego image and embedding capacity are the two most important factors for evaluating an image data embedding method. A high visual quality stego image cannot be easier detected by an unexpected user than a low visual quality image. The embedding capacity is the total bits of confidential data that can be embedded in the cover image. A large embedding capacity represents that the image carries more confidential data.

Traditional color BTC is usually applied for red, green, and blue planes. Then, the three bitmaps and three high mean and low mean pairs play the compression code. Use of a common bitmap in color BTC can significantly improve the performance of compression rate. Chang *et al.* proposed a common bitmap generation method using Genetic algorithm [20]. Chou *et al.* presented a low computation cost common bitmap generation method [22]. In this paper, a compression domain data hiding method is presented. The proposed method adopts lossy image compression technique for achieving high embedding capacity.

The remaining sections are organized as follows. Relevant background information is described in Section 2 and the proposed method is detailed in Section 3. Section 4 summarizes the experimental results for evaluating the proposed method's performance. Finally, conclusions are made in Section 5.

2 Related Works

2.1 Block Truncation Coding for Color Image

For a gray level image, 128 bits are needed for a block of 4×4. But only 32 bits are needed in the same block size after processing with BTC. Bits needed in original color image and BTC compressed color image are 384 and 96 respectively. Due to the low complexity and low compression rate, BTC can be used in the embedding phase for color image. The processing of the BTC is as follows.

First, divide the color image into non-overlapping blocks sized $n \times m$ for the Red, Green, and Blue plane, respectively. Mean value per block in three dimensions can be calculated as,

$$\overline{B}_i^{\{R,G,B\}} = \frac{1}{n \times m} \sum_{p=0}^{n-1} \sum_{q=0}^{m-1} b_i^{\{R,G,B\}}(p,q), \quad \textbf{(1)}$$

where, $\{R, G, B\}$ represents the Red, Green and Blue planes, n and m represent the height and width of the block, B_i represents the i-th block in the color image.

Then, all of pixels in B_i is classified into two groups and denoted as $G_H^{\{R,G,B\}}$ and $G_L^{\{R,G,B\}}$. The $G_H^{\{R,G,B\}}$ group is composed of the pixels value greater than $\overline{B}_i^{\{R,G,B\}}$, and $G_H^{\{R,G,B\}}$ group is composed of the pixels value greater than $\overline{B}_i^{\{R,G,B\}}$. After that, the high mean (i.e., denoted $HM_{\{R,G,B\}}$) and low mean (i.e., denoted $LM_{\{R,G,B\}}$) of block can be gained by calculating the mean value of group $G_H^{\{R,G,B\}}$ and $G_L^{\{R,G,B\}}$, respectively. Thus, every block contains (HM_R, LM_R), (HM_G, LM_G), and (HM_B, LM_B).

2.2 Generating the Common Bitmap by the Error Function

In the traditional color BTC method, each block needs three bitmaps for the image decompression. For saving the bitmap requirement, Chang *et al.* created a common bitmap generation to use a common bitmap instead of three bitmaps. The common bitmap is adopted for Red, Green, and Blue plane to reconstruct the image in the same time. Chang *et al.* adopted Genetic Algorithm (GA) to help generating a suitable common bitmap [20]. Further, Chou *et al.* [22] proposed a common bitmap generating scheme using the difference function instead of GA algorithm for common bitmap generation. Chou *et al.*'s method is to test a pixel adopts HM and LM at the same time, then the differences between original pixel value and adapted HM and LM were calculated. For every pixel, if the minimal difference value is adopted HM value, then the corresponding indicator in the common bitmap will set to '1', otherwise the indicator is set to '0' (i.e., using LM to reconstruct the decompressed image). The common bitmap is generated after all of blocks have been generated the common bitmap. Obviously, Chou *et al.*'s method has lower computation cost than Chang *et al.*'s method.

2.3 Chang et al.'s Color BTC Data Embedding [20]

Chang et al. proposed a compression domain data embedding method to conceal secret data into color image. First, the color image is divided into non-overlapping blocks. Then, for every block, a common bitmap is generated with three pairs of high mean (*HM*) and low mean (*LM*) corresponding to red, green and blue plane, respectively. Here, three pairs of *HM* and *LM* are denoted as (HM_R, LM_R), (HM_G, LM_G), and (HM_B, LM_B). Normally, the compression code of color BTC is composed by $CBM \| HM_R \| LM_R \| HM_G \| LM_G \| HM_B \| LM_B$. Chang et al. modified the compression rule to imply the secret message, such as $HM_R \| LM_R$ implies secret bit '0' and $LM_R \| HM_R$ implies secret bit '1'. Table 1 summarized all of cases for secret data embedding. Chang et al.'s method can easily embed three secret bits in a block.

Table 1. Chang *et al.*'s data embedding rules

case	secret	Compression code
1	000	$CBM \| HM_R \| LM_R \| HM_G \| LM_G \| HM_B \| LM_B$
2	001	$CBM \| HM_R \| LM_R \| HM_G \| LM_G \| \mathbf{LM_B} \| \mathbf{HM_B}$
3	010	$CBM \| HM_R \| LM_R \| \mathbf{LM_G} \| \mathbf{HM_G} \| HM_B \| LM_B$
4	011	$CBM \| HM_R \| LM_R \| \mathbf{LM_G} \| \mathbf{HM_G} \| \mathbf{LM_B} \| \mathbf{HM_B}$
5	100	$CBM \| \mathbf{LM_R} \| \mathbf{HM_R} \| HM_G \| LM_G \| HM_B \| LM_B$
6	101	$CBM \| \mathbf{LM_R} \| \mathbf{HM_R} \| HM_G \| LM_G \| \mathbf{LM_B} \| \mathbf{HM_B}$
7	110	$CBM \| \mathbf{LM_R} \| \mathbf{HM_R} \| \mathbf{LM_G} \| \mathbf{HM_G} \| HM_B \| LM_B$
8	111	$CBM \| \mathbf{LM_R} \| \mathbf{HM_R} \| \mathbf{LM_G} \| \mathbf{HM_G} \| \mathbf{LM_B} \| \mathbf{HM_B}$

3 The Proposed Scheme

In order to improve the performance of embedding capacity, we compose several embedding strategies to conceal secret data into color image as many as possible. The key steps of the proposed embedding procedure are detailed in following subsections.

3.1 Data Embedding Phase

Because color BTC can significantly reduce the size of color image, a new compression code generation is proposed for secret data delivering. For easy description the proposed data embedding, let a cover image $I = \{p_{i,j} \mid i = 1, 2, \ldots, H; j = 1, 2, \ldots, W\}$, $p_{i,j} = (r, g, b)$, and $\{r, g, b\} \in \{0, 1, \ldots, 255\}$ where H and W represent the height and width of image, and r, g, and b are corresponded to red, green, and blue color for a pixel. To consider the compression performance, a common bitmap generation technique [22] also adopted to generate the common bitmap. For example, (x_R^k, x_G^k, x_B^k) represents the original pixel set {72, 64, 63} in position (2, 3) of the block B_k, and $\{HM_R=80, HM_G=70, HM_B=65\}_k$ and $\{LM_R=60, LM_G=65, LM_B=61\}_k$ are high mean and low mean in this block B_k respectively. Then, to decide the comment bit '0' or '1' which can produce smaller errors $ER^0= |72-60|+|64-65|+|63-60|=16$ and $ER^1= |72-80|+|64-70|+|63-63|=14$. So the common bit map CBM_k in position (2, 3) of the block B_k can use 3 high mean {80, 70, 60} to represent the original pixel, and common bit in position of this common bitmap will be 1, the remain comment bits are generated in the same procedure. Let *CBM*, HM_i, and LM_i represent the common bitmap and high mean and low mean of a block.

First, the image is divided into non-overlapping blocks (B_i), sized $n \times m$ and represented by $I = \{B_i \mid i = 0, 1, \ldots, N_B-1\}$, where N_B represents the number of blocks. For image block B_i, three pairs of *HM* and *LM* are generated by the color BTC method. For further reduction size of compression code, the common bit map CBM can also be generated by color BTC. After that, three parts are used to conceal secret data namely: the 1 count of *CBM*; the LSB substitution for high mean and low mean pair; and the order of high mean and low mean arrangement.

For embedding secret data into *CBM*, if the count of '1' in *CBM* is an even value then it can be used to imply the secret bit '0'. Contrary, if the count of '1' in *CBM* is an odd value then it implies the secret bit '1'. In case of the '1' count value is different from the secret bit, then to flip one value in the *CBM*. To consider the visual quality of the stego image, the victim value in *CBM* is selected by choosing the smallest distortion between original pixel value and the corresponding high mean value and low mean value.

For embedding secret data into high mean and low mean by using LSB substitution, the value of *HM* and *LM* is adjusted for implying secret data according to the proposed adjusting rules. *HM* and *LM* represent the high mean and low mean in each block using BTC (refer section 2.1). For example in the color image, there are three pairs of *HM* and *LM* respectively.

We found that, in case of the block stated in a smooth area, the blocks may be have the same value of *HM* and *LM*. In this case, Chang *et al.*'s method will not be used to conceal secret data. In the proposed data hiding method, the LSB bits of *HM* and *LM* values are substituted with secret data. Also, we try to eliminate the case of *HM* = *LM* and adopting Chang *et al.*'s *HM* and *LM* order arrangement hiding technique to embed secret data. Fig. 1 illustrates the proposed high mean and low mean adjustment rules for concealing secret data "00", "01", "10", and "11", respectively. The goal of embedded rule is try to make the high mean or low mean change to even (embed secret '0') or odd (embed secret '1'). So, in secret extracted phase, we only need to verify the *HM* or *LM* is even or odd to extract the secret '0' or '1' respectively. For instance, if the high mean *HM* (or low mean *LM*) is even and the following secret bit is '1', the *HM* (or low mean *LM*) must be changed to odd using LSB; otherwise, to maintain original or change another even to embed secret '0'.

Fig. 1(a) illustrates the case of *HM* = *LM*. Normally, the high mean adjustment is to increase the value of *HM*, and the low mean adjustment is to decrease the value of *LM*. After that, Chang *et al.*'s *HM* and *LM* sequence arrangement is adopted to embed secret data.

The key steps of the proposed data embedding method are summarized as follows:

Embedding Procedure:

Input: Secret data *S* and cover image *I*

Output: Compressed code

Step 1: Divide the cover image *I* into non-overlapping blocks $\{B_i \mid i=0, 1, \ldots, N_B-1\}$ sized $n \times m$.

Step 2: Generate the common bitmap *CBM* and three pairs of high mean and low mean values by Color BTC (i.e., (HM_R, LM_R), (HM_G, LM_G), and (HM_B, LM_B)).

Step 3: Count the number of '1' in *CBM* for B_i and denoted *m*.

Step 4: If mod(m, 2) != secret bit, then to flip an element value in *CBM*.

Step 5: Adjust the high mean and low mean values to imply two bits secret bits data according to the embedding rule shown in Fig. 1.

Step 6: Rearrange the sequence of high mean and low mean values to embed three secret bits using Chang et al.'s method [20].

Step 7: Repeat Step 2 to Step 6 until all of secret data were embedded into compression code.

3.2 Data Extraction Phase

The workflow of decoding phase for each block B_k is showing in Fig. 2. The data extraction is an inverse works from embedding phase. After the receiver received the stego compression code, the secret data can be extracted by applying the proposed data extraction procedure. First, take $n \times m$ bits to reconstruct the *CBM*. If there exit the even number of '1' in *CBM* then the secret bit is '0', otherwise the secret bit is '1'. After that, take the following compression code to reconstruct three pairs of mean values. For every mean value pair, if the first value is greater than the second value then extract secret bit '0', otherwise the secret bit is '1' and swap those two mean values from *LM*||*HM* to *HM*||*LM*. Then, for each mean value, if the mean value is an even number then extract secret '0', otherwise extract secret '1'. The decompression image can be reconstructed by filling the mean values to the pixels according to *CBM*. The key steps of the proposed secret data extraction and image decompression are summarized as follows:

Extracting Procedure:

Input: Color BTC compression code

Output: Reconstructed image I' and the secret data *S*

Step 1: Take $n \times m$ bits from the compression code to rebuild the *CBM*.

Step 2: If the number of '1' in *CBM* is an even amount then extract secret bit '0', otherwise extract secret bit '1'.

Step 3: Take three pairs of the mean values from compression code.

Step 4: For every mean value pair, if the first value is greater than the second value, then extract secret '0', otherwise, extract secret '1' and swap those two mean values.

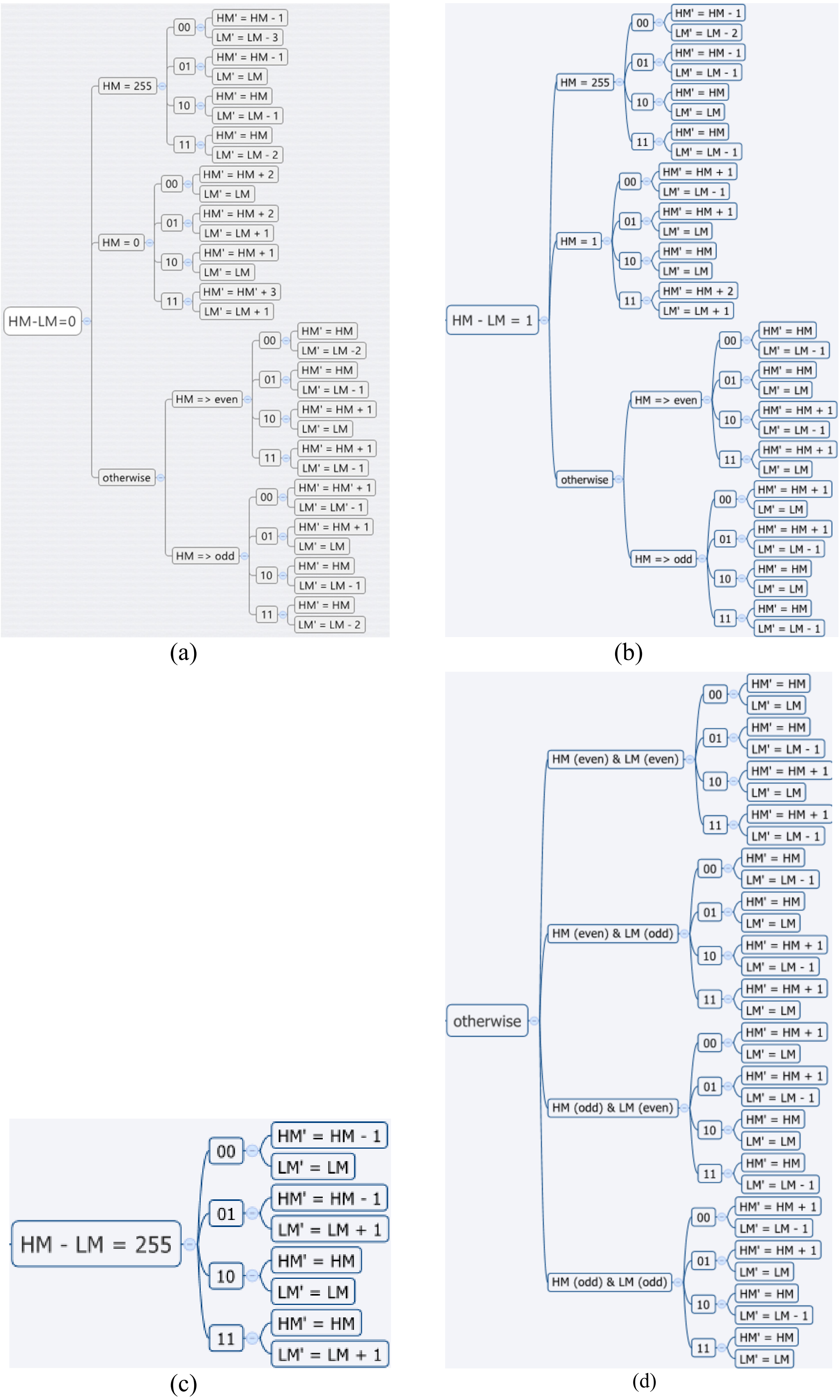


Fig. 1. The rules of *HM* and *LM* adjustment for embed two secret message

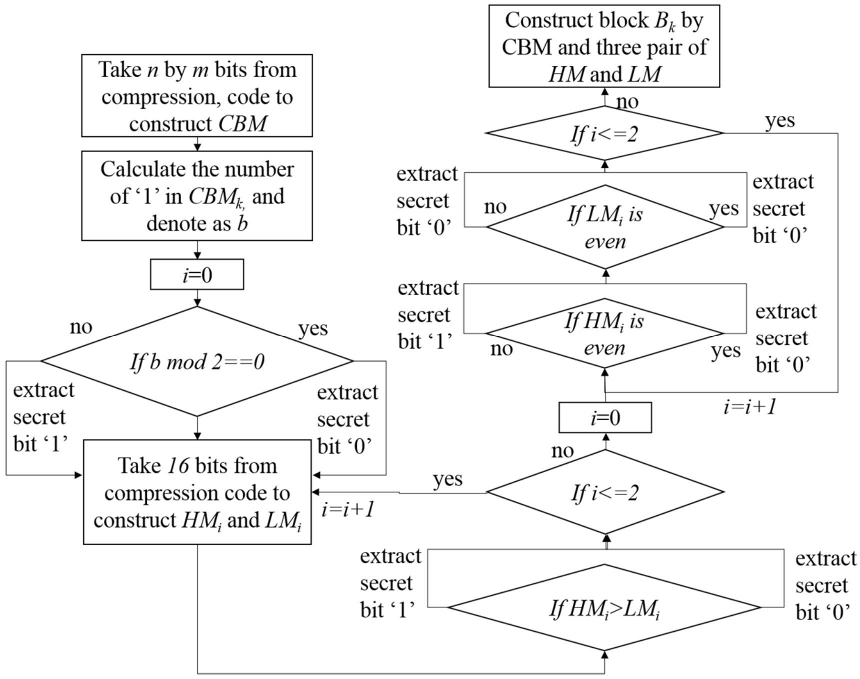


Fig. 2. The work flow of decoding phase for each block B_k

Step 5: For every mean value, if the mean value is even number then extract secret bit '0', otherwise extract secret bit '1'.

Step 6: Repeat Step 1 to Step 5 until all compression codes was checked.

Step 7: Rearrange the extracted secret data.

4 The Experimental Results

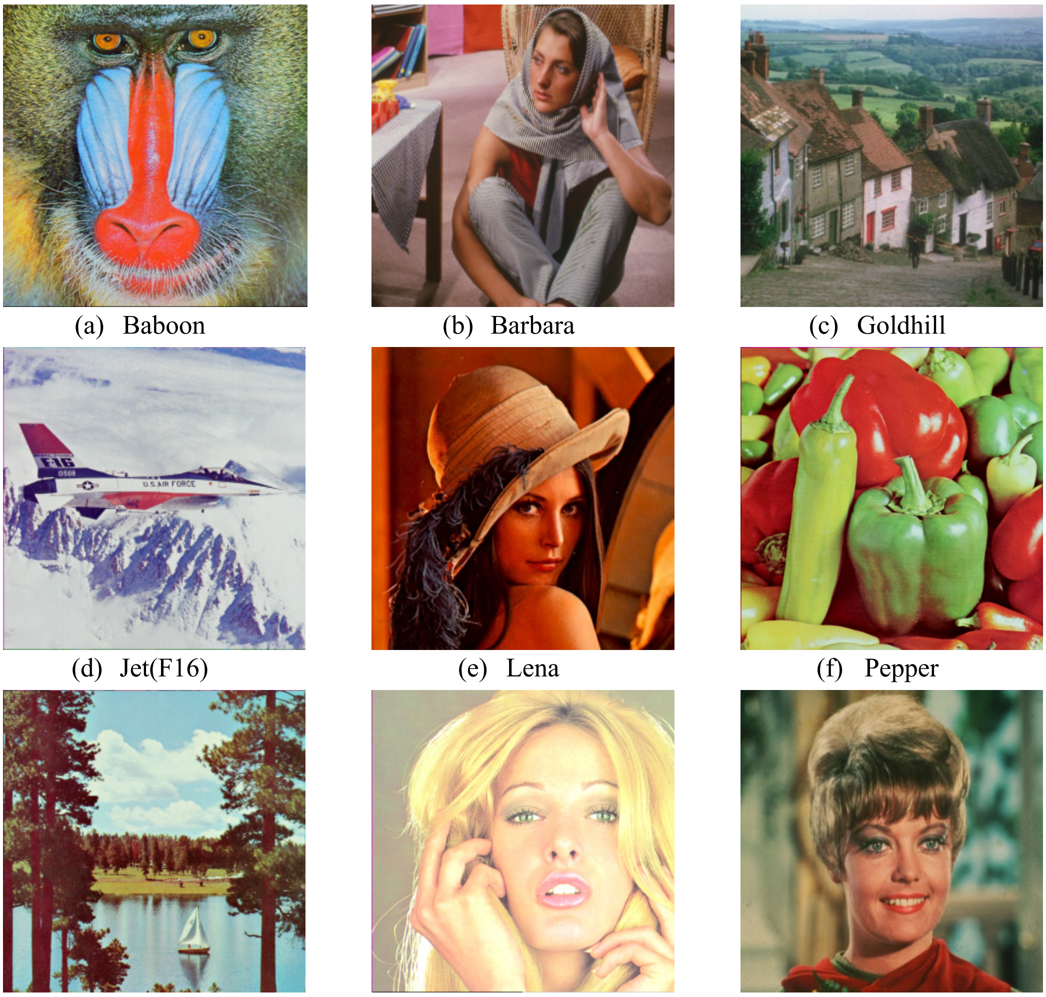
In order to evaluate the performance of the proposed method, we implemented the proposed method and Chang *et al.*'s method [20] by using Octave software. Nine common used test images were used in our simulation (i.e., referring to Fig. 3). The test images include the complex content and smooth content for testing the performance of the proposed method in terms of visual quality. The visual quality of stego image and the embedding payload are two most important factors for evaluating the performance of data hiding method. A stego image with good visual quality means the stego can be hidden from unexpected user. Thus, the secret data was succeed in delivering to the receiver. Using human eyes' to evaluate the visual quality is a subject way, because it can be caused by personal favorite. In this paper, we adopted peak-signal-to-noise-ratio (PSNR) to measure the visual quality. PSNR is an objective measurement which is defined as follows,

$$PSNR_{\text{color}} = \frac{(PSNR_{\text{R}}+PSNR_G+PSNR_B)}{3}, \quad (2)$$

$$PSNR_{\{\text{R,G,B}\}} = 10 \times \log_{10} \frac{255^2}{MSE_{\{R,G,B\}}}, \quad (3)$$

$$MSE_{\{\text{R,G,B}\}} = \frac{1}{W \times H} \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} \left(I_{i,j}^{\{R,G,B\}} - \widehat{I_{i,j}^{\{R,G,B\}}} \right)^2, \quad (4)$$

where H and W represent the height and width of cover image I. $MSE_{\{R,\ G,\ B\}}$ represents that the mean square error in Red, Green, and Blue plane, respectively. A large value of PSNR means that the stego image is most similar to the original cover image. Contrary, a small value of PNSR indicates that the stego image is dissimilar to the corresponding cover image. Generally, a user is hard to distinguish the distortion on the image when PSNR is greater than 30 dB.



(a) Baboon (b) Barbara (c) Goldhill

(d) Jet(F16) (e) Lena (f) Pepper

(g) Sailboat (h) Tiffany (i) Zelda

Fig. 3. The test images sized 512 × 512 pixels

Fig. 3 shows the visual quality comparison of the proposed method in different block size setting. Obviously, the block size of 2 × 2 have the better visual quality of the stego image in comparison to the block size sets of 4 × 4 and 8 × 8, since the larger block, the more pixels are adapted to the distortion caused by HM or LM. However, in worst case of block size 8×8, the visual quality of stego images generated by the proposed method still greater than 27 dB.

In the image Baboon with complex image content (Fig. 3(a)), each block has lost number of complex discrete pixels. Hence most of block of pixels cannot represent by only one pair of *HM* or *LM* that cause more serious distortion. The image Tiffany (h) and Zelda (i) cause less distortion, because most of blocks contain similar pixels. The image Tiffany is represented with similar color in the part of hand, hair, face, and background. Also, the image Zelda which contains large number of blocks with similar color in the wide range of background are blurred. That is the reason for less distortion when using BTC-based method in these two images. Hence, the smooth image content with less distortion, resulted in higher PSNR value.

On the other hand, the embedding payload means that the largest payload for concealing secret data into a cover image. A large payload data hiding method can reduce the transmission rounds for secret data delivery. The embedding payload is defined as,

$$\text{payload} = \frac{\|S_e\|}{H \times W} \text{ bpp}, \quad \textbf{(5)}$$

where, $\|S_e\|$ represents the total bits of secret data embedded into the cover image.

The proposed method is a compression domain data hiding method. The size of compression code is another important factor for our proposed method design. Thus, the compression rate is adopted to evaluate the performance of image compression. The compression rate is defined as,

$$c = \frac{\|Comp\|}{\|I\|}, \quad (6)$$

where, ||*Comp*|| represents the size of compression code and ||*I*|| represents that the size of image.

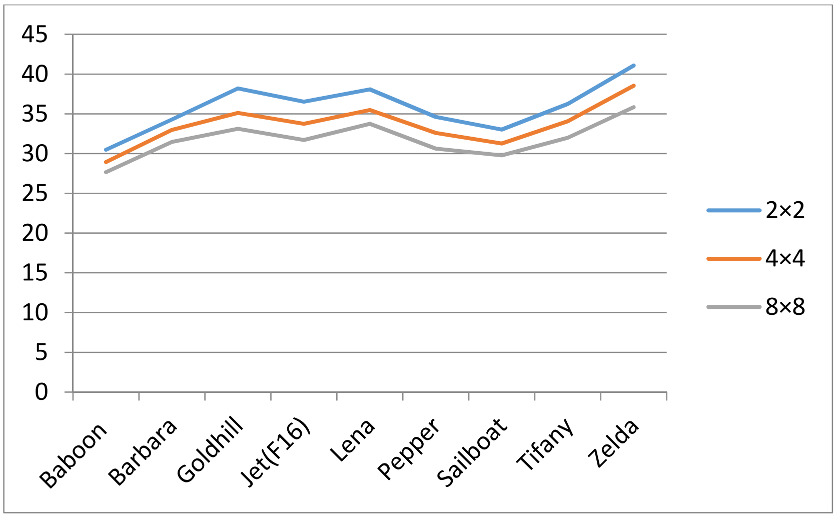


Fig. 4. The visual quality comparison of the proposed method in different block size

Table 2 summarizes the comparison result of color BTC compression code and the embedding payload in different block size.

Comparing to Chang et al.'s method [20], our method achieve higher payload as more as 6 secret bits using different order of 3 pairs of high mean and low mean in each block. Also, all blocks are embeddable in the proposed scheme in comparison to the Chang et al.'s method which cannot embed any secret if all pixels are the same in the current block. So, extra 9 secret bits can be embedded in this case. As displayed in right side of the Table 2, the payload is lower in some smooth images (ex: Tiffany and Sailboat), since these two images contain more blocks with similar pixel values that cannot embed any secret. As all blocks are embeddable with same embedded rule, our compression rate and payload are fixed.

As some cases of Chang et al.'s method cannot embed the secret using the order high mean and low mean, only 40 bits can represent the block ($CBM \| LM_R \| LM_G \| LM_B$). Whereas in our method these blocks are not wasted and can be used 64 bits to embed 9 secret bits ($CBM \| HM_R' \| LM_R' \| HM_G' \| LM_G' \| HM_B' \| LM_B'$). Here, *HM*'and *LM*'are the modified high mean and low mean by the proposed embedded rule (refer Fig. 1). Because small number of this case, only the compression rate is larger than Chang et al.'s method with tiny margin. Therefore, our method can treat this few blocks as un-embeddable blocks to perform the same compression rate with Chang et al.'s method.

We found that, a small block size setting will get higher embedding payload but the compression size is large than the original image size. On the other hands, the larger block size provides a better compression rate outcome, but results in worse performance, in terms of embedding payload. Thus, we conclude that, a suitable block size is 4 × 4. Furthermore, the proposed method provided better performance compared with Chang *et al.*'s method, in terms of embedding payload. As all high mean and low mean are modified in the proposed scheme for secret embedding, the PSNR values are lower than Chang et al.'s method, but maintains slight distortion than Chang et al.'s method.

In future work, we plan to consider more possible of arrangements for three pairs of high mean and low mean to conceal more secrets, while maintaining acceptable visual quality. Or, our embedding scheme (refer Fig. 1) can be considered as a tool to improve BTC-based embedding method to perform higher payload.

Table 2. The comparison of compression code and the embedding payload

	The proposed method						Chang et al.'s method [20]					
	2×2		4×4		8×8		2×2		4×4		8×8	
	c	payload	c	payload	c	payload	c	payload	c	payload	c	payload
Baboon	1.63	2.50	0.50	0.63	0.22	0.16	1.62	0.75	0.50	0.19	0.22	0.05
Barbara	1.63	2.50	0.50	0.63	0.22	0.16	1.62	0.75	0.50	0.19	0.22	0.05
Goldhill	1.63	2.50	0.50	0.63	0.22	0.16	1.62	0.75	0.50	0.19	0.22	0.05
Jet(F16)	1.63	2.50	0.50	0.63	0.22	0.16	1.61	0.74	0.50	0.19	0.22	0.05
Lena	1.63	2.50	0.50	0.63	0.22	0.16	1.62	0.75	0.50	0.19	0.22	0.05
Pepper	1.63	2.50	0.50	0.63	0.22	0.16	1.58	0.72	0.50	0.19	0.22	0.05
Sailboat	1.63	2.50	0.50	0.63	0.22	0.16	1.51	0.68	0.50	0.19	0.29	0.05
Tiffany	1.63	2.50	0.50	0.63	0.22	0.16	1.50	0.68	0.50	0.19	0.22	0.05
Zelda	1.63	2.50	0.50	0.63	0.22	0.16	1.62	0.75	0.50	0.19	0.22	0.05

5 Conclusions

Color BTC compression is an easy method to implement image compression. The experimental results show that the Color BTC provides good compression performance. Because the BTC method needs to record high mean and low mean values for an image block. The proposed method designed high mean and low mean value adjustment to imply the secret message. Thus, the proposed method significantly increases the embedding capacity. Also the visual quality of the stego image generated by the proposed method maintain the good visual quality outcome. That means the proposed method has higher chance to hide the unexpected user for achieving the goal of secret data delivery.

References

[1] J. Tian, "Reversible Data Embedding Using a Difference Expansion," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 13, pp. 890-896, 2003.

[2] A. Alattar, "Reversible Watermark Using the Difference Expansion of a Generalized Integer Transform," *IEEE Transactions on Multimedia*, Vol. 13, pp. 1147-1156, 2004.

[3] Y. Ni, N. Shi, W. Su, "Reversible Data Hiding," *IEEE Transactions on Circuits and Systems of Video Technology*, Vol. 16, pp. 354-362, 2006.

[4] C.C. Chang, P. Tsai, M.H. Lin, "An Adaptive Steganographic Scheme for Color Images," *Fundamenta Informaticae*, Vol. 62, pp. 275-289, 2004.

[5] D. Thodi, J. Rodriguez, "Expansion Embedding Techniques for Reversible Watermarking," *IEEE Transactions on Image Processing*, Vol. 16, pp. 721-730, 2007.

[6] C.C. Lin, W.L. Tai, C. Chang, "Multilevel Reversible Data Hiding Based on Histogram Modification of Difference Images," *Pattern Recognition*, Vol. 41, No. 12, pp. 3582-3591, 2008.

[7] H.J. Kim, V. Sachnev, Y.Q. Shi, J. Nam, H.G. Choo, "A Novel Difference Expansion Transform for Reversible Data Embedding," *IEEE Transactions on Transaction on Information Forensics Security*, Vol. 3, No. 3, pp. 456-465, 2008.

[8] V. Sachnev, H.J. Kim, J. Nam, S. Suresh, Y.Q. Shi, "Reversible Watermarking Algorithm Using Sorting and Prediction," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 19, pp. 989-999, 2009.

[9] W.L. Tai, C.M. Yeh, C.C. Chang, "Reversible data hiding based on histogram modification of pixel differences," *IEEE Transaction on Circuits and Systems for Video Technology*, Vol. 19, pp. 906-910, 2009.

[10] L. Li, B. Luo, Q. Li, X. Fang, "A Color Images Steganography Method by Multiple Embedding Strategy Based on Sobel Operator," in *Proceedings of International Conference on Multimedia Information Networking and Security (MINES '09)*, pp. 118-121, 2009.

[11] L. Luo, Z. Chen, M. Chen, X. Zeng, Z. Xiong, "Reversible Image Watermarking Using Interpolation Technique," *IEEE Transactions on Information Forensics Security*, Vol. 5, pp. 187-193, 2010.

[12] W.J. Yang, K.L. Chung, H.Y.M. Liao, "Efficient Reversible Data Hiding for Color Filter Array Images," *Information Sciences*, Vol. 190, pp. 208-226, 2012.

[13] C.C. Chang, C.C. Lin, C.S. Tseng, W.L. Tai, "Reversible Hiding in DCT-based Compressed Images," *Information Sciences*, Vol. 177, pp. 2768-2786, 2007.

[14] C.C. Chang, T.D. Kieu, W.C. Wu, "A Lossless Data Embedding Technique by Joint Neighboring Coding," *Pattern Recognition*, Vol. 42, pp. 1597-1603, 2009.

[15] L. Kamstra, H.J.A.M. Heijmans, "Reversible Data Embedding Into Images Using Wavelet Techniques and Sorting," *IEEE Transactions on Image Processing*, Vol. 14, pp. 2082-2090, 2005.

[16] A. Gersho, R.M. Gray, "Vector Quantization," *IEEE ASSP Magazine*, Vol. 1, no. 2, pp. 4-29, 1984.

[17] T. Kim, "Side Match and Overlap Match Vector Quantizer for Images," *IEEE Transactions on Image Professing*, Vol. 1, no. 2, pp. 170-185, 1992.

[18] E.J. Delp, O.R. Roger, "Image Compression Using Block Truncation Coding," *IEEE Transactions on Communication*, Vol. 27, no. 9, pp. 1335-1341, 1979.

[19] C.C. Chan, C.Y. Lin, "Reversible Steganographic Method Using SMVQ Approach Based on Declustering," *Information Sciences*, Vol. 177, pp. 1796-1805, 2007.

[20] C.C. Chang, C.Y. Lin and Y.H. Fan, "Lossless Data Hiding for Color Images Based on Block Truncation Coding," *Pattern Recognition*, Vol. 41, No. 7, pp. 2347-2357, 2008.

[21] Y. Zhang, S.Z. Guo, Z.M. Lu and H. Luo, "Reversible Data Hiding for BTC-Compressed Images Based on Lossless Coding of Mean Tables," *IEICE Transactions on Communications*, Vol. 2, pp. 624-631, 2013.

[22] Y.C. Chou and H.H. Chang, "A Data Hiding Scheme for Color Image Using BTC Compression Technique," in *Proceedings of 9th IEEE International Conference on Cognitive Informatics (ICCI)*, IEEE Press, pp. 845-850, 2010.