A High Payload Steganographic Scheme for Compressed Images with Hamming Code

Junlan Bai[1], Chin-Chen Chang[2]
(Corresponding author: Chin-Chen Chang)
School of Electronic Engineering, University of Electronic Science and Technology of China[1]
No. 2006 Xiyuan Ave, West Hi-Tech Zone, Chengdu, 611731, China
Department of Information Engineering and Computer Science, Feng Chia University[2]
No. 100 Wenhwa Rd., Seatwen, Taichung 40724, Taiwan
(Email: alan3c@gmail.com)
(Received Aug. 24, 2015; revised and accepted Nov. 27, 2015 & Jan. 3, 2016)

Abstract

Data hiding schemes in the compressed domain have attracted more attention since the compressed image format is one of the most frequently transmitted formats over the Internet. Specifically, among various compression algorithms, Absolute Moment Block Truncation Coding (AMBTC) is a good choice due to its extremely low complexity and acceptable distortion. In this study, we propose a novel steganography using matrix embedding with Hamming code to insert secret message into AMBTC compressed bit-stream. Experimental results demonstrate that our scheme outperforms the other four existing BTC-based data hiding approaches in terms of embedding capacity, bit rate, and hiding efficiency.

Keywords: AMBTC compression, data hiding, hamming code

1 Introduction

With the rapid development of the contemporary society, new devices and powerful software make the world more digitalized and informationized. The wireless network offer ubiquitous channels to deliver and exchange various kinds of multimedia information, which becomes indispensable for humans' daily life. However, the potential brought about by freely uploading, downloading and manipulating data can't be fully realized without the protection of privacy. Data hiding technique [1, 14] was proposed to embed confidential data undetectably and imperceptibly in digital content.

Data hiding can be applied to three domains, which are spatial, frequency and compression domains of the cover images. Herein, the difference is the domain where the embedding happens. Compared with much research in the spatial domain and frequency domain data hiding schemes, data hiding techniques in the compression domain have not been fully explored. Currently, the compressed image format is frequently used transmission format over the Internet for reducing the storage space and transmission bandwidth. Since enormous digitalized data transmitted through the open channels occupies a relatively large part of bandwidth, compression techniques such as vector quantization (VQ) [10, 17, 18, 19, 20], JPEG [15], block truncation coding (BTC) [6], and Absolute Moment Block Truncation Coding (AMBTC) [11], are favored by many scholars. BTC was firstly proposed by Delp and Mitchell in 1979 [6]. It is a well-documented spatial image compression algorithm with low complexity compared to other compression techniques, such as JPEG [15]. BTC has been applied to graphics, digital video and colour digital imagery. Lema and Mitchell presented an AMBTC technique [11] to further increase the compression rate. In recent years, diverse data hiding schemes [2, 3, 4, 7, 8, 13, 21] based on the compression domain have been developed by academic researchers.

In 2002, Jo and Kim introduced a watermarking scheme [8] based on vector quantisation (VQ). It improved the degree of spreading watermark information by partitioning the codebook into three groups. An optimal vector quantizer codebook design contributes to increasing the steganographic image quality. However, it still offers worse visual quality of the stego image compared to hiding in BTC-compressed image. For the sake of improving the quality, Chuang and Chang [3] presented a scheme of embedding a majority of secret message into the smooth block of BTC-compressed image and the rest of secret data into the complex block in 2006. Although their technique achieved a relatively satisfactory quality, the hiding ability of each compressed block was less than one secret bit, which resulted in a rather low payload. In 2010, Chen et al. proposed an embedding algorithm [2] which exploited the relationship between the low- and high-means to judge whether to conceal confidential information into the corresponding compressed block or not. In [2], each

block can carry one bit or explore bits in the bitmap sometimes to embed extra secret data. In order to further enlarge the payload, Sun et al. presented a data hiding scheme [13] by losslessly embedding secret data in both the low- and high-mean tables in 2013. In 2015, Chang et al. proposed a data hiding scheme [4] for BTC compressed image by means of dynamic programming strategy. The dynamic programming strategy was employed to find a best bijective mapping relationship by using LSB replacement. In 2012, Kim et al. proposed a secret sharing scheme for hiding a watermark in two image shadows made from AMBTC based on Hamming Code [9]. Inspired by their method, we employ (7, 4) Hamming code to embed secret information in a digital image with higher embedding capacity than the existing data hiding schemes [2, 3, 4, 7, 8, 13, 21] based on the compression domain while remaining an acceptable visual quality.

The rest of the paper is organized as follows. Section 2 introduces the related techniques used in this paper. Our proposed scheme is presented in Section 3. The experimental results are illustrated in Section 4. Finally, the conclusions of the paper are stated in Section 5.

2 Preliminaries

This section will review the Absolute Moment Block Truncation Coding (AMBTC) algorithm, (7, 4) Hamming code and the matrix embedding.

2.1 Absolute Moment Block Truncation Coding (AMBTC)

In order to speed up the transmission process and save the bandwidth, Lema and Mitchell presented an AMBTC technique [11] for compressing grayscale and colour images in 1984. It outperforms BTC in computational complexity and mean squared error (MSE). AMBTC requires preserving two moments, mean and the first absolute central moment of image blocks, respectively. This algorithm starts off by partitioning the image into non-overlapping blocks with the size of $n \times n$. Denote $k = n \times n$ as the total number of pixels of the block. For each segmented block, the block mean value η is computed as follows:

$$\eta = \frac{1}{k}\sum_{i=1}^{k} x_i,$$

where x_i denotes the gray-level intensity of the pixel i in a block. Since AMBTC is a one-bit quantizer, η is utilized as a threshold for binarizing all the pixels in each block into two clusters with p and $k-p$ pixels, individually. If the intensity of the pixel is lower than η, it falls into the first cluster C_0. Otherwise, it falls into the second cluster C_1. The result is bitmap which is employed to record the distributions of the two quantization levels. The low- and high-mean used to represent a block is organized as follows:

$$a = \frac{1}{p}\sum_{x_i<\eta} x_i, \quad (1)$$

$$b = \frac{1}{k-p}\sum_{x_i\geq\eta} x_i, \quad (2)$$

where the two variables a and b are the quantization levels. Finally, the reconstructed AMBTC compression image is expressed as follows:

$$y_{i,j} = \begin{cases} a, & \text{if } h_{i,j}=0, \\ b, & \text{if } h_{i,j}=1, \end{cases} \quad h_{i,j} = \begin{cases} 0, & \text{if } x_{i,j}=C_0, \\ 1, & \text{if } x_{i,j}=C_1, \end{cases}$$

where $h_{i,j}$ is the element in the bitmap BM and $y_{i,j}$ denotes the pixel in the compressed AMBTC image. From the above equations, we can see each compressed block will generate a trio (a, b, BM), two quantization levels and one bitmap. An example of the AMBTC scheme is described in Figure 1. Denote that Figure 1(a) is the original image block containing 4×4 pixels. Firstly, the mean value η is computed as 138. Then the bitmap BM is achieved based on AMBTC algorithm and the result is illustrated in Figure 1(b). According to Equations (1) and (2), we can obtain the two quantization level values of a and b as 135 and 166, respectively. Finally, the generated compressed trio (135,166, 0011001000001101) is transmitted to the receiver. Herein, the receiver decodes this trio and reconstructs the image block shown in Figure 1(c).

123	104	158	192
137	119	188	129
133	101	121	130
141	168	115	149

(a) Original image block

0	0	1	1
0	0	1	0
0	0	0	0
1	1	0	1

(b) AMBTC bitmap

135	135	166	166
135	135	166	135
135	135	135	135
166	166	135	166

(c) Reconstructed image block

Figure 1: An example of AMBTC algorithm procedure

2.2 The (7, 4) Hamming Code

Hamming code [16] is widely used and favored by researchers due to its perfect structure and ability to detect and correct one error in a block of bits. It is a kind of block error correction linear codes. The (7, 4) Hamming code is used in this paper to operate bit modification. It encodes four original message bits $k = (c_6, c_5, c_4, c_3)$ by adding three extra parity check bits $r = (c_2, c_1, c_0)$ to form the transmitted codeword of length 7, which is ready for transmission.

In Hamming code, there are two important matrices: Generator matrix G and parity check matrix H, which are essential for encoding and decoding. At the encoding side, the codeword is obtained by multiplying the 4 data bits $k = (c_6, c_5, c_4, c_3)$ with the generator matrix

$G_{4\times7}$, and then applying modulo of 2. The resulting 7-bit codeword will be sent to the receiving side through a noisy channel. At the receiving side, the receiver reads the codeword, multiplies parity check matrix $H_{3\times7}$ with it and takes modulo of 2 again to check whether there is any error. The result is a syndrome vector of three bits. If the syndrome is '000', there is no error. If a single bit error occurs, the value is not equal to '000'. Then, we need to find which column of H is identical to the syndrome. For example, if the syndrome is '011' and the fourth column in H equals '011', flip the fourth bit of the received codeword and finally obtain the correct original data bits by ignoring the last three bits.

2.3 The Matrix Embedding with Hamming Code

Matrix embedding was originally introduced by Crandall [5] in 1998 to achieve high embedding efficiency, i.e., to reduce steganographic modifications to the cover images. The Hamming code is initially applied to matrix embedding, in which $n-k$ secret bits are inserted into n cover pixels by an $[n, k]$ Hamming code, of which the parity check matrix is H and $n = 2^k - 1$. Mao [12] designed a fast algorithm for matrix embedding in 2014. In the matrix embedding, an embedding group is defined as the cover vector $x = (x_1, x_2, ..., x_n)$ and the secret bits $m = (m_1, m_2, ..., m_{n-k})$. Change the positions of the columns in parity check matrix H to sort all the columns in ascending order in advance. Then, the mathematical model of embedding process is expressed as follows:

$$y = \mathrm{Emd}(x, m) = x + F(m - \mathrm{H}x),$$

where y is the received stego vector and $F(\cdot)$ is a function that transforms $(m - \mathrm{H}x)$ into a decimal value i and sets the i-th bit to '1' while other bits of the vector are set to '0'. At the receiving side, the receiver can easily extract the embedded message by:

$$\begin{aligned} m' &= \mathrm{Ext}(y) \\ &= \mathrm{H}(x + F(m - \mathrm{H}x)) \\ &= \mathrm{H}x + m - \mathrm{H}x = m, \end{aligned} \tag{3}$$

3 Proposed Scheme

In this section, the proposed scheme will be thoroughly presented. First, AMBTC algorithm is performed on the original grayscale cover image to obtain the compressed data that can be represented by a low-mean, a high mean, and a bitmap. Then, the secret message is concealed into the AMBTC compressed trio (a, b, BM). The advantage is to achieve a higher payload compared to other data hiding schemes performed on the compression domain. There are two phases for embedding in our proposed scheme. We firstly perform using the (7, 4) Hamming code on the low- and high-mean values of each compressed trio and exploit the relationship between them to hide one more secret bit. Then, we embed secret message into the bitmaps. The reason why we choose (7, 4) Hamming code is based on two considerations. On the one hand, we can embed three secret bits into seven cover bits. On the other hand, only one bit in the cover vector will be flipped after embedding, which means the modification to the cover vector is reduced to the minimum.

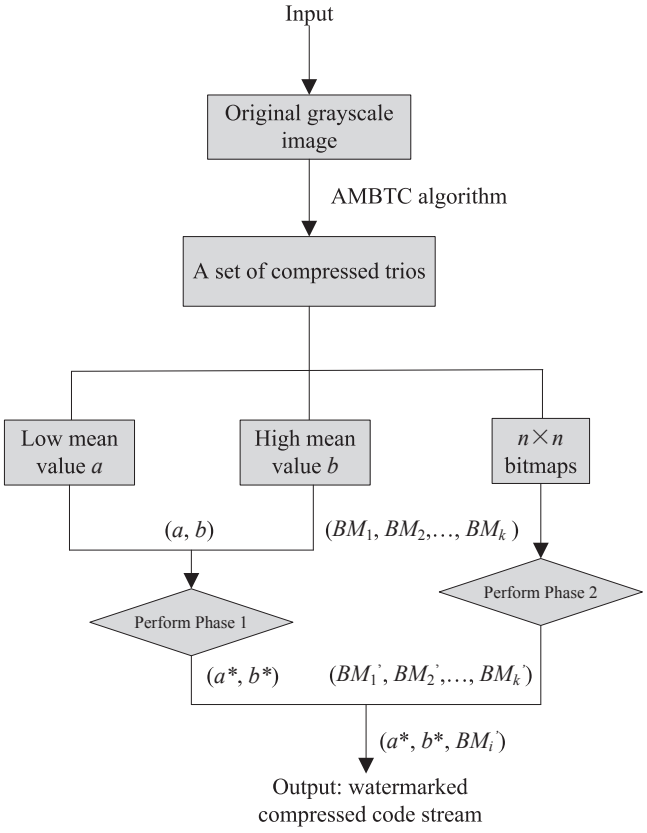

Figure 2: The flowchart of the embedding phase

3.1 Embedding Phase

After the original grayscale image is partitioned into a set of $n \times n$ blocks for AMBTC compression, each block undergoes the same process, thereby the description below towards a single block processing is adopted for simplicity. Normally, n equals 4. The flowchart of the embedding procedure which contains two phases is illustrated in Figure 2. A string of secret message is generated by a random number generator in advance.

The embedding phase 1: Embed secret message into two quantization levels.

Step 1. Extract four LSBs of low mean value a and three LSBs of high mean value b to constitute the column vector $x = (x_1, x_2, x_3, x_4, x_5, x_6, x_7)$ and read three secret message bits denoted as column vector $m = (m_1, m_2, m_3)$. Both of them are composed of the embedding group $\{x, m\}$.

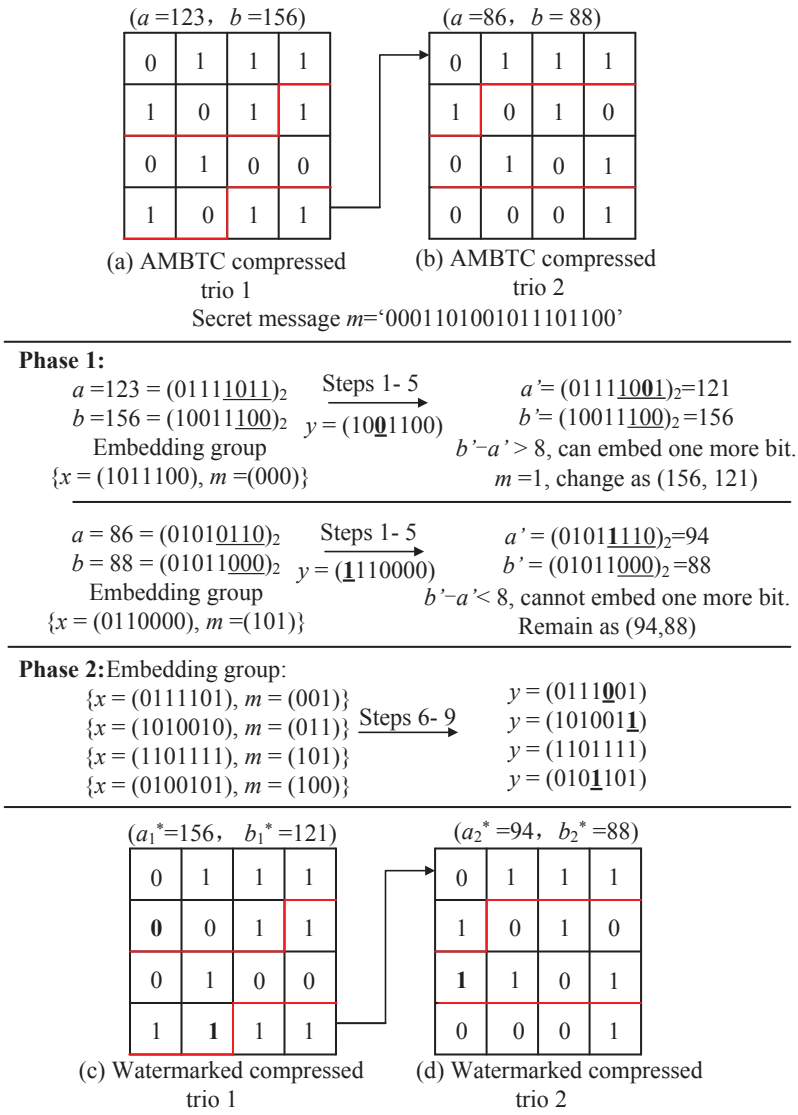

Figure 3: Example of embedding procedure of the proposed scheme

Step 2. Construct the parity check matrix H of (7, 4) Hamming code. Change the positions of the columns in H to ensure they array in ascending order in decimal form.

Step 3. Compute syndrome $s = m - \mathrm{H}x$, and transform s into a decimal number i. Change the i-th bit of vector x to obtain the stego vector $y = (y_1, y_2, y_3, y_4, y_5, y_6, y_7)$. Each time, three secret bits are embedded into seven bits.

Step 4. Replace the four LSBs of low mean value a with (y_1, y_2, y_3, y_4) and three LSBs of high mean value b with (y_5, y_6, y_7) to obtain newly generated quantization pair (a, b).

Step 5. Compare the difference value between a' and b' to see whether they can be applied to the second embedding. If $b' - a' \leq 8$, this pair belongs to the unembeddable category. If not, we can embed one more secret bit through interchange operation. If the corresponding embedded bit is '1', exchange the two quantization levels, which becomes (b', a'). Otherwise, leave the location of (b', a') remain the same. Denote the final result of the quantization pair as (a^*, b^*).

It is notable that we set a threshold as eight to determine whether it is an embeddable group or not, because after operating Steps 1-4, the rangeability of a and b brought by matrix embedding may be eight and four respectively and only one of them will be changed. If a' is larger than b', it may suffer from confusion in the data extraction phase, because we cannot figure out whether the result is obtained by matrix embedding or interchange operation. In order to avoid ambiguity, we set a threshold to distinguish between these two cases. After all the quantization level pairs are processed, we conduct embedding in the bitmaps.

The embedding phase 2: Embed secret message into bitmaps.

Step 6. Concatenate each bitmap as (BM_1, BM_2,..., BM_k) to form a sequence of bitmaps.

Step 7. Sequentially take seven bits from the bitmaps and denote them as cover vector x. Then read three secret message bits and denote them as vector m each time.

Step 8. Perform the matrix embedding procedure which is the same as in Steps 2-3 in Phase 1 until the entire bit-stream in the bitmaps is successively processed.

Step 9. Update the bitmaps (BM'_1, BM'_2,..., BM'_k) and combine them with the corresponding quantization levels to obtain the final compressed trios (a^*, b^*, BM'_i).

After the whole embedding process is done, we successfully conceal the secret message into the compressed trios.

An embedding example is given in Figure 3 to demonstrate the embedding procedure of the proposed scheme. Denote that two compressed trios are (123, 156, 0111101101001011) and (86, 88, 0111101001010001), respectively. Assume that the secret message m is '0001101001011101100'. The parity check matrix H used in this example is shown in Equation (4), in which all the columns in H have been put in ascending order in decimal form.

$$\mathrm{H} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \tag{4}$$

Firstly, we embed three secret bits into the first quantization level pair (123, 156) according to the operations in Steps 1-4 of Phase 1. Then the difference value between a' and b' is calculated to check whether it is suitable for hiding one more bit. In this case, the pair belongs to the embeddable category. Since the secret bit is '1', their location will be interchanged. Then the second quantization level pair is operated in the similar way as the first pair. When embedding is finished according to Phase 1, we will hide secret message into bitmaps. The final results are illustrated in Figures 3(c) and (d), where the bold one indicates the alteration.

3.2 Extracting Phase

In the extracting phase, the secret message is extracted from a received watermarked AMBTC compressed code stream by using our designed rules. The flowchart of the extraction procedure is depicted in Figure 4. At the receiving side, the receiver firstly extracts the secret message hidden in Phase 1 from the two quantization level pairs by comparing their values. The absolute difference between a and b is computed to check whether it has been applied the interchange operation. If the absolute difference is larger than eight, we can ascertain that one more bit has been embedded into each quantization level pair and the secret bit is obtained by the inverse way as that of in the embedding phase. Otherwise, no secret bit is embedded by interchange operation. Then the rest secret data can be easily achieved by Equation (3) from the bitmaps. Finally, all bits are concatenated to exactly recover the hidden secret message.

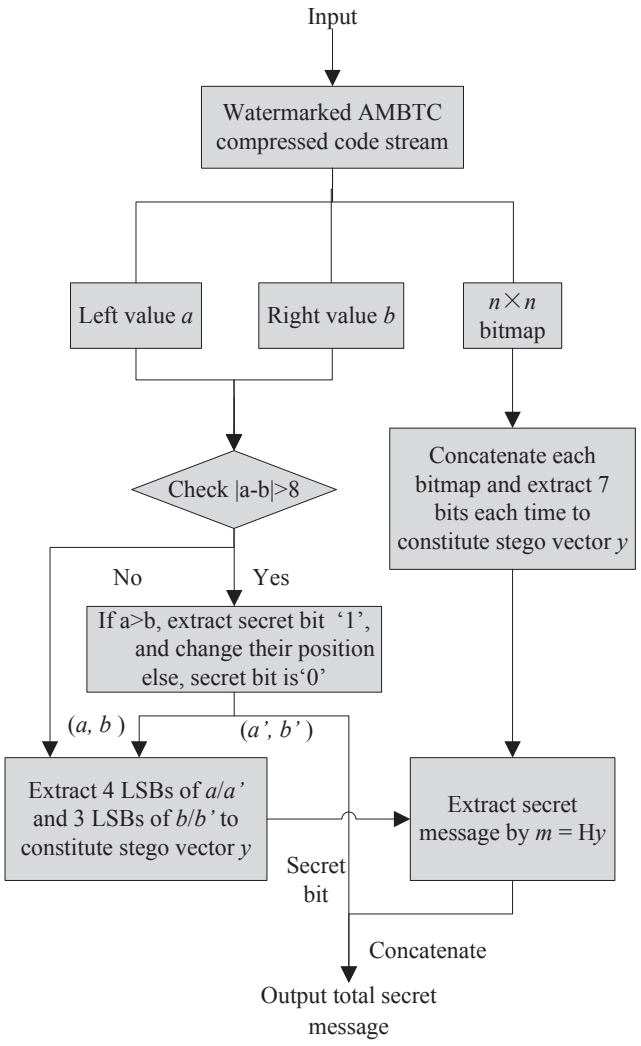

Figure 4: The flowchart of the extracting phase

4 Experimental Results

In this section, we conduct some experiments and discuss the results to demonstrate the superiority of our proposed scheme. Figure 5 lists the six grayscale test images used in our experiments, where they are 'Lena', 'Zelda', 'Elain', 'Jet', 'Boat', 'Goldhill', respectively. The block size used in the AMBTC compression is set as 4×4 in our experiment. We employ four parameters to measure the performance of the proposed scheme, where they are peak signal-to-noise ratio *PSNR*, hiding capacity *HC*, bit rate *BR*, and hiding efficiency *HE*. As we all know that

Table 1: The comparative results of the proposed scheme with other schemes

Scheme	Criteria	Lena	Zelda	Elain	Jet	Boat	Goldhill	Average
AMBTC	PSNR (dB)	33.23	36.74	33.83	33.25	31.76	32.87	33.61
	BR (bpp)	2.00	2.00	2.00	2.00	2.00	2.00	2.00
Proposed scheme	PSNR (dB)	28.92	32.25	29.41	28.11	27.56	28.74	29.17
	BR (bpp)	2.00	2.00	2.00	2.00	2.00	2.00	2.00
	HC(bits)	169250	168480	172479	168388	170103	172771	170245
Chang et al's scheme [4]	PSNR (dB)	30.71	34.26	31.26	30.99	31.02	30.49	31.45
	BR (bpp)	2.00	2.00	2.00	2.00	2.00	2.00	2.00
	HC(bits)	114688	114688	114688	114688	114688	114688	114688
Sun et al's scheme [13]	PSNR (dB)	Cannot be reconstructed by the watermarked/stego code stream						
	BR (bpp)	2.06	2.06	2.06	2.06	2.06	2.06	2.06
	HC(bits)	64008	64008	64008	64008	64008	64008	64008
Hong et al's scheme [7]	PSNR (dB)	33.23	36.74	33.83	33.25	31.76	32.87	33.61
	BR (bpp)	2.00	2.00	2.00	2.00	2.00	2.00	2.00
	HC(bits)	16384	16384	16384	16384	16384	16384	16384
Chuang et al's scheme [3]	PSNR (dB)	30.45	33.98	31.07	30.52	29.84	30.52	31.06
	BR (bpp)	2.00	2.00	2.00	2.00	2.00	2.00	2.00
	HC(bits)	13051	13169	13248	12637	11989	12486	12763

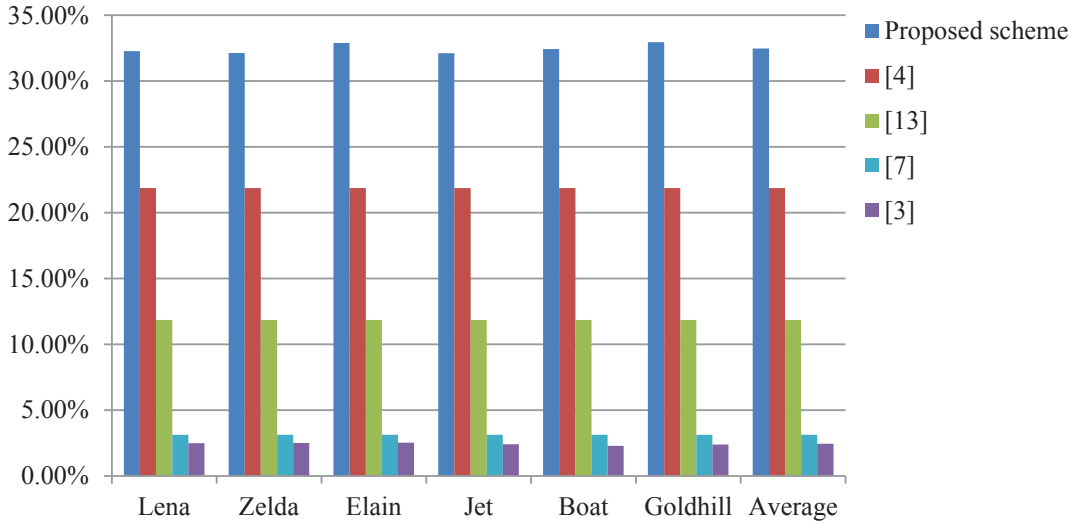

Figure 6: Comparisons of hiding efficiency between the proposed scheme and other schemes

peak signal-to-noise *PSNR* is used to evaluate the visual quality of the watermarked AMBTC compressed images. With respect to the value of *PSNR*, it is defined as shown in Equation (5), and MSE (mean square error) is calculated by Equation (6). Bit rate *BR* measures the number of bits required to represent one pixel, which is computed by Equation (7) and *CS* means the length of the output code stream. Hiding capacity *HC* is used to measure the number of embedded secret bits. Hiding efficiency *HE* is defined by Equation (8), which represents the percentage

(a) Lena (b) Zelda (c) Elain

(d) Jet (e) Boat (f) Goldhill

Figure 5: Test standard images

of *HC* and *CS*.

$$PSNR = 10\log_{10}(\frac{255^2}{MSE})(\text{dB}), \quad (5)$$

$$MSE = \sum_{i=1}^{H}\sum_{j=1}^{W}(x_{i,j} - x'_{i,j})^2/(L \times W), \quad (6)$$

$$BR = \frac{|CS|}{L \times W}(bpp), \quad (7)$$

$$HE = \frac{HC}{|CS|}, \quad (8)$$

where $x_{i,j}$ and $x'_{i,j}$ are the values of pixels in the original grayscale image and the stego image, respectively.

We compare our scheme with the other four existing schemes [3, 4, 7, 13]. Table 1 shows the comparative results for different test images between the proposed scheme and other schemes under different criteria. With respect to the hiding capacity, we observe that the average hiding capacity of Chuang and Chang's scheme is the lowest among these schemes while the proposed scheme achieves the highest. The highest capacity reaches 172771 bits which is more than half of that of Chang et al.'s scheme. Although the payload is significantly increased with the cost of visual quality, the average *PSNR* of the proposed scheme is still greater than 29dB which is acceptable by human visual system (HVS). The visual quality of the proposed scheme decreases as the complexity of the texture increases, because the difference between low- and high-mean values is mostly bigger in complex images than that of in smooth images. The bit rate of the proposed scheme is the same as that of AMBTC algorithm. Compared with Sun et al.'s scheme, extra bits are introduced to be included in the watermarked code stream, which leads to a higher bit rate than the other four schemes. Furthermore, it cannot be constructed by the stego code stream since it contains extra data. Apparently, the proposed scheme outperforms other schemes in terms of hiding efficiency *HE* mainly due to its high embedding capacity. Figure 6 illustrates the comparative results of hiding capacity *HE* of the proposed scheme and the other schemes. It is notable that the *HE* of Chang et al.'s scheme is double that of Sun et al.'s scheme. However, the embedding efficiency of our scheme is 10% higher than that of Chang et al.'s scheme.

5 Conclusions

In this paper, a novel data hiding scheme based on (7, 4) Hamming code for AMBTC compressed images is presented. It achieves high embedding capacity while preserving an acceptable visual quality when compared with other existing BTC-based schemes. We not only hide secret message into two quantization levels but also embed secret bits into the bitmaps. For each embedding group consisting of three secret bits and seven cover bits, only one bite in the cover vector will be flipped and three secret bits are successfully embedded. Moreover, to further increase hiding capacity, we apply the interchange operation to the quantization pair to hide one extra bit. The experimental results confirm that the proposed scheme outperforms other existing schemes in terms of embedding capacity which can be used to the high-payload-needed application.

References

[1] P. K. Amin, N. Liu, K. P. Subbalakshmi, "Statistical attack resilient data hiding," *International Journal of Network Security*, vol. 5, no. 1, pp. 112–120, 2007.

[2] J. Chen, W. Hong, T. S. Chen, and C. W. Shiu, "Steganography for BTC compressed images using no distortion technique," *The Imaging Science Journal*, vol. 58, no. 4, pp. 177–185, 2010.

[3] J. C. Chuang and C. C. Chang, "Using a simple and fast image compression algorithm to hide secret information," *International Journal of Computers and Applications*, vol. 28, no. 4, pp. 329–333, 2006.

[4] C. C. Chang, Y. J. Liu and S. T. Nguyen, "A novel data hiding scheme for Block Truncation Coding - compressed images using dynamic programming strategy," in *Proceedings of 6th International Graphic and Image Processing*, pp. 94431Q–94431Q, 2015.

[5] R. Crandall, "Some notes on steganography," 1998. (`http://www.dia.unisa.it/~ads/corso-security/www/CORSO-0203/steganografia/LINKS\%20LOCALI/matrix-encoding.pdf`)

[6] E. J. Delp and O. R. Mitchell, "Image compression using block truncation coding," *IEEE Transactions on Communications*, vol. 27, no. 9, pp. 1335–1342, 1979.

[7] W. Hong, T. S. Chen and C. W. Shiu, "Lossless steganography for AMBTC compressed images," in *Proceedings of 1st International Congress on Image and Signal Processing*, vol. 2, pp. 13–17, Sanya, China, 2008.

[8] M. Jo, H. D. Kim, "A digital image watermarking scheme based on vector quantization," *IEICE Transactions on Information and System*, vol. E85-D, no. 6, pp. 1054–1056, 2002.
[9] C. Kim, et al. "A(2, 2) secret sharing scheme based on hamming code and AMBTC," in *Intelligent Information and Database Systems*, LNCS 7198, pp. 129–139, 2012.
[10] Y. Linde, A. Buzo, R. M. Gray, "An algorithm for vector quantizer design," *IEEE Transactions on Communications*, vol. 28, no. 1, pp. 84–95, 1980.
[11] M. Lema and O. R. Mitchell, "Absolute moment block truncation coding and its application to color images," *IEEE Transactions on Communications*, vol. 32, no. 10, pp. 1148–1157, 1984.
[12] Q. Mao, "A fast algorithm for matrix embedding steganography," *Digital Image Processing*, vol. 25, no. 4, pp. 248–254, 2014.
[13] W. Sun, et al. "High performance reversible data hiding for block truncation coding compressed image," *Signal, Image and Video Processing*, vol. 7, no. 2, pp. 297–306, 2013.
[14] N. I. Wu, C. M. Wang, M. S. Hwang, "Data Hiding: Current Status and Key Issues," *International Journal of Network Security*, vol. 4, no. 1, pp. 1–9, 2007.
[15] G. K. Wallace, "The JPEG still picture compression standard," *Communications of the ACM*, vol. 34, no. 4, pp. 30–44, 1991.
[16] F. J. M. Williams and N. J. Sloane, "The Theory of Error-Correcting Codes," *Elsevier*, 1977.
[17] C. Zhu, L.H. Li, T.J. Wang, Z.Y. He, "Partial-distortion-weighted fuzzy competitive learning algorithm for vector quantization," *Electronics Letters*, vol. 30, no. 6, pp. 505–506, Mar. 1994.
[18] C. Zhu and L. M. Po, "Partial distortion sensitive competitive learning algorithm for optimal codebook design," *Electronics Letters*, vol. 32, no. 19, pp. 1757–1758, 1996.
[19] C. Zhu, L.M. Po, "Minimax partial distortion competitive learning for optimal codebook design", *IEEE Transactions on Image Processing*, vol. 7, no. 10, pp. 1400–1409, 1998.
[20] C. Zhu and Y. Hua, "Image vector quantization with minimax L_∞ distortion," *IEEE Signal Processing Letters*, vol. 6, no. 2, pp. 25–27, 1999.
[21] 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, 2016.

Junlan Bai received the B.S. degree in electronic engineering from University of Electronic Science and Technology of China, China, in 2013. She is currently pursuing the M.S. degree at the same university. Her research interests include information hiding and digital image processing.

Chin-Chen Chang received his Ph.D. degree in computer engineering from National Chiao Tung University. His first degree is Bachelor of Science in Applied Mathematics and master degree is Master of Science in computer and decision sciences. Both were awarded in National Tsing Hua University. Dr. Chang served in National Chung Cheng University from 1989 to 2005. His current title is Chair Professor in Department of Information Engineering and Computer Science, Feng Chia University, from Feb. 2005. Prior to joining Feng Chia University, Professor Chang was an associate professor in Chiao Tung University, professor in National Chung Hsing University, chair professor in National Chung Cheng University. He had also been Visiting Researcher and Visiting Scientist to Tokyo University and Kyoto University, Japan. During his service in Chung Cheng, Professor Chang served as Chairman of the Institute of Computer Science and Information Engineering, Dean of College of Engineering, Provost and then Acting President of Chung Cheng University and Director of Advisory Office in Ministry of Education, Taiwan. Professor Chang has won many research awards and honorary positions by and in prestigious organizations both nationally and internationally. He is currently a Fellow of IEEE and a Fellow of IEE, UK. On numerous occasions, he was invited to serve as Visiting Professor, Chair Professor, Honorary Professor, Honorary Director, Honorary Chairman, Distinguished Alumnus, Distinguished Researcher, Research Fellow by universities and research institutes. His current research interests include database design, computer cryptography, image compression and data structures.