Multimed Tools Appl (2018) 77:5767–5783
DOI 10.1007/s11042-017-4487-2

Minima-maxima preserving data hiding algorithm for absolute moment block truncation coding compressed images

Ngoc-Tu Huynh[1] · **K. Bharanitharan**[2] · **Chin-Chen Chang**[3] · **Yanjun Liu**[3]

Received: 27 July 2016 / Revised: 25 December 2016 / Accepted: 6 February 2017 /
Published online: 17 February 2017

Abstract Preventing secret from being suspicious during transferring over Internet has become an emergent issue in the past decades. Several protecting data methods such as cryptographic techniques, watermarking or steganography techniques, etc. have been proposed to conceal secrets from being discovered. In this paper, we introduce the Minima-Maxima Preserving (MMP) data hiding algorithm for absolute moment block truncation coding (AMBTC) compressed images, which preserves the high and low values in each block to reversibly extract secret and recover host images during the extracting procedure. As a result, image quality is maintained that makes the secret messages hidden and avoids suspicion from attackers. Experimental results show that the scheme has better performance compared to state-of-the-art schemes in terms of visual quality and embedding rate. Besides, with the high embedding rate, the scheme can be widely used in practice.

Keywords Data hiding · Absolute moment block truncation coding (AMBTC) · Compressed image · Embedding rate · Visual quality

✉ Yanjun Liu
yjliu104@gmail.com

Ngoc-Tu Huynh
huynhngoctu@tdt.edu.vn

K. Bharanitharan
Dharan@ieee.org

Chin-Chen Chang
alan3c@gmail.com

[1] Faculty of Information Technology, Ton Duc Thang University, Ho Chi Minh City, Vietnam

[2] Research School of Management, The Australian National University, Canberra, Australia

[3] Department of Information Engineering and Computer Science, Feng Chia University, No. 100 Wenhwa Rd., Seatwen, Taichung 407, Taiwan, Republic of China

1 Introduction

Sharing and transferring digital data over the Internet has been popular during the past decade. In practice, in order to transfer data, the following issues can be considered. Firstly, the most important issue is security since users require their secrets to be transferred safely in an imperceptible way. Secondly, we might be concerned about the accuracy of the methods in which these secrets can be completely and exactly extracted. Finally, the efficiency of these methods allows users to send a large amount of secret information at the same time. Among several highly secure approaches, data hiding is adopted as a simple, secure and efficient way.

Data hiding is the art of concealing a large amount of secret data into host files. Host covers can be text files or multimedia files such as images, audios and videos. Among them, images have been widely used as host covers since they have more space to embed secrets. Data hiding schemes are basically developed in three domains—the spatial domain, the frequency domain and the compression domain. In general, data hiding can be categorized into two groups—reversible data hiding (RDH) [2–5, 12, 14–18, 20, 21, 25, 29, 31–33] and irreversible data hiding (IDH) [6–11, 13, 19, 22, 24, 28, 30, 34]. RDH allows one to embed secret data into an image in such a way that the original image can be reconstructed from the marked image.

One well-known RDH scheme in the spatial domain is the histogram-shifting-based RDH (HSRDH), which was proposed by Ni et al. in 2006 [21]. The main drawback of Ni et al.'s scheme is its extremely low payload, which is not efficient and practical since the demand of transferring data is getting higher and higher in the digital era. Since then, a lot of literature related to histogram modification have been proposed to improve the capacity of this scheme [17, 18, 25]. In [14], Jung et al. proposed an improvement of Ni et al.'s method [21], whereby unlike Ni et al.'s, Jung et al.'s scheme exploited predicted values from a pixel, an edge of image and image's joint noticeable difference (JND) to embed data. Therefore, the scheme achieves better capacity than that of Ni et al.; however, its capacity is still very low. To further improve the embedding payload of this approach, in 2013, Li et al. [17] proposed another RDH scheme by using frequency of difference values of pixel-pairs to form a two-dimension histogram, instead of the conventional one-dimension histogram methods. Although, the scheme significantly enhances the number of embedded bits (from 0.02 bpp of Ni et al. to 0.19 bpp for smooth images), it is not efficient since there are a lot of unused space remaining in an image. Also in 2013, Li et al. [18] proposed a generalized-HSRDH method. In their generalized scheme, the authors figured out that several HSRDHs proposed in the literature are special cases of their generalized construction. Using this scheme, we can embed up to 0.9 bit into per pixel. Up to now, it is the highest capacity for HSRDH-based methods.

In the frequency-domain RDH, the host image is transformed into frequency coefficients via various integer transforms [23] such as discrete Fourier transform (DFT), discrete cosine transform (DCT), and discrete wavelet transform (DWT). The primary property of frequency coefficients is that low frequency coefficients contain more important information whereas in high frequency area information is less significant. Thus, the matrix of frequency coefficients are divided into non-overlapping blocks and the secret data will be embedded in those blocks with low mean values.

While RDH schemes focus on how to recover the original host images, IDH schemes usually aim to construct marked images as natural as possible and avoid large-sized carrier images, but still can carry more secret bits. To reduce the size of host images, there are several compression methods adopted as a preprocessing step before creating marked images [34].

Then, the embedding and the extracting procedures will be processed to generate high quality marked images with high payload. The frequently employed common compression methods are vector quantization/side-match vector quantization (VQ/SMVQ) compression [22], joint photographic experts group (JPEG) compression [24] or block truncation/absolute moment block truncation (BTC/AMBTC) compression [6–8, 26] and so on. The major drawback of most of these data hiding schemes for compression domain is the extremely low embedding capacity. Qin et al. [22] employed image inpainting to embed secret into SMVQ indices. The scheme aims to reduce visual distortion and error diffusion caused by the progressive compression; however, the capacity of the scheme is not more than 0.1 bpp. Guo et al. [7] proposed a data hiding method for BTC, which exploits the concept of secret sharing to enhance the security of the method. The authors used two host images to embed secret data to achieve good visual quality, but embedding rate still needs to be improved. Among these data compression methods, AMBTC is an improvement of BTC in terms of performance and complexity.

In this paper, our goal is to propose a reversible data hiding scheme with high embedding rate while preserving good visual quality. We introduce an algorithm for AMBTC compressed images named Minima-Maxima Preserving algorithm (MMP). In this algorithm, one high-mean value and one low-mean value of each block are kept unchanged to be the information to recover the host image during the extracting procedure. By doing so, our proposed MMP algorithm results in the following advantages, which are: first, it achieves higher payload and better image quality, thus satisfying the requirements of the data hiding method; second, the scheme can reversibly reconstruct the original AMBTC compressed image. The above contributions show the technical merits of our proposed algorithm.

The rest of the paper is organized as follows. In Section 2, we briefly introduce the concept of AMBTC compression method. The MMP algorithm is described in detail in Section 3. Section 4 is our implementation and discussion. Finally, conclusions are made in Section 5.

2 AMBTC compression algorithm

Block truncation coding (BTC) [4] is an efficient lossy compression method, which requires a very low computational complexity. BTC method quantizes the pixels in each block of gray-level image into a two-level bitmap while reserving statistical moments of blocks to maintain an acceptable visual quality of BTC compressed image. To improve the performance of BTC technique, Lema and Mitchell [15] proposed a variant of BTC, called absolute moment block truncation coding (AMBTC). AMBTC preserves the first absolute moment along with the mean. The main difference between BTC and AMBTC is the computation of quantization levels during the encoding phase. With simpler computation, AMBTC is more suitable for real-time applications.

In AMBTC compression, the original image is first divided into non-overlapping blocks. For each block, we compute its mean value and absolute value. After that, two quantization values called low-mean value (a) and high-mean value (b) are calculated. Then, pixels which are lower than the mean value are replaced by a, and pixels, and which are higher than or equal to the mean value are replaced by b. Meanwhile, a bitmap is recorded based on a and b. Figure 1 is an example of the compression (encoding) procedure when the block size of the host image is 4 × 4. The details of AMBTC compression method are described in the following subsections.

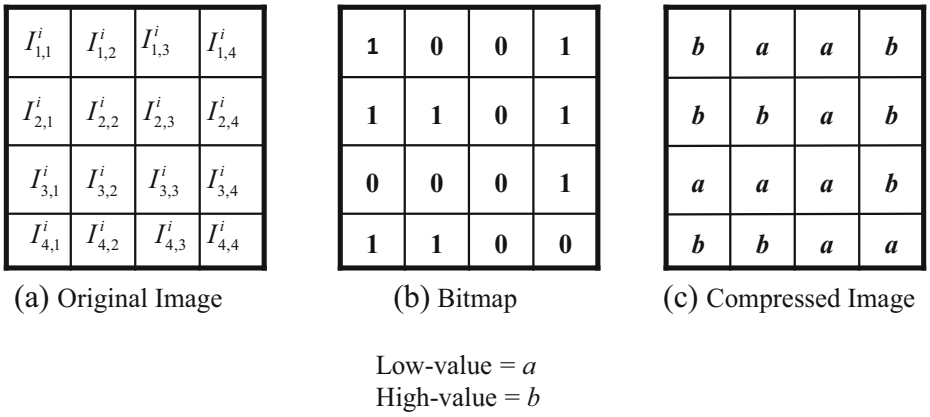


Fig. 1 AMBTC encoding procedure (**a**) Original 4 × 4 Image, (**b**) Corresponding Bitmap, and (**c**) Compressed Image

2.1 Encoding procedure

In Fig. 1, we briefly describe the process of encoding. Figure 1(a) is the original image block with 16 pixels. By applying the AMBTC encoding procedure which is introduced below, we get the bitmap shown in Fig. 1(b) and two values: low-mean value a and high-mean value b. The compressed image is constructed by replacing the value "0" of the bitmap with a and the value "1" of the bitmap with b. The result of the compressed image is shown in Fig. 1(c). Hereafter, we introduce the encoding procedure in details.

Input gray-level image is first divided into non-overlapping blocks sized $M \times M$. We suppose that the number of pixels in each block is $N_b = M \times M$, and $x_1, x_2, ..., x_{N_b}$ denote the pixels in a block. The mean value of each block is calculated by Eq. (1) :

$$\bar{x} = \frac{1}{N_b} \sum_{i=1}^{N_b} x_i. \tag{1}$$

Then, the image block can be compressed by two quantization levels and one bitmap as below:

$$a = \frac{1}{N_b - q} \sum_{x_i < \bar{x}} x_i, \tag{2}$$

$$b = \frac{1}{q} \sum_{x_i \geq \bar{x}} x_i,$$

where q represents the number of pixels whose grey-level are greater than or equal to the mean value$\bar{x}$. Bitmap B of the block represents the pixels, each of which is formed from the grey value by the rule: if the pixel value x_i is greater than or equal to the mean value $\bar{x}$, the corresponding bitmap's element is set to 1; otherwise, it is set to 0.

$$B_i = \begin{cases} 1, & \text{if } x_i \geq \bar{x} \\ 0, & \text{if } x_i < \bar{x} \end{cases}. \tag{3}$$

2.2 Decoding procedure

To decode and reconstruct the block of image, two quantization levels a and b and the bitmap are obtained. All the values "0" of the block are replaced by a, and values "1" are replaced by b:

$$r_i = \begin{cases} a, & \text{if } B_i = 0 \\ b, & \text{if } B_i = 1 \end{cases}. \tag{4}$$

After all the blocks of images are decoded, the original image is decompressed.

3 Proposed minima-maxima preserving (MMP) algorithm

To achieve higher payload and better image quality, in this section, a novel Minima-Maxima Preserving (MMP) Algorithm based on least-significant-bit (LSB) substitution is proposed. Originally, the LSB-substitution method cannot be used to recover the host image since whenever we modify these bits, the receiver might not have any information about the original pixels to recover the host image. To achieve reversibility, the proposed MMP algorithm embeds secret data into AMBTC compressed image by the following way: the first high-mean and low-mean values are kept unchanged for each block of compressed image; for the rest of the pixels in the block, we embed secret data by adaptively replacing one or two LSBs. Therefore, the embedding capacity is significantly enhanced and the marked images have good visual quality.

Figure 2 represents the overall flow of the MMP algorithm that has two main procedures: image compression and secret embedding procedure which is introduced in Subsection 3.1, and secret extraction and host restoration procedure which is introduced in Subsection 3.2, respectively.

3.1 Image compression and secret embedding procedure

Figure 3 briefly describes the process of image compression and secret embedding procedure.

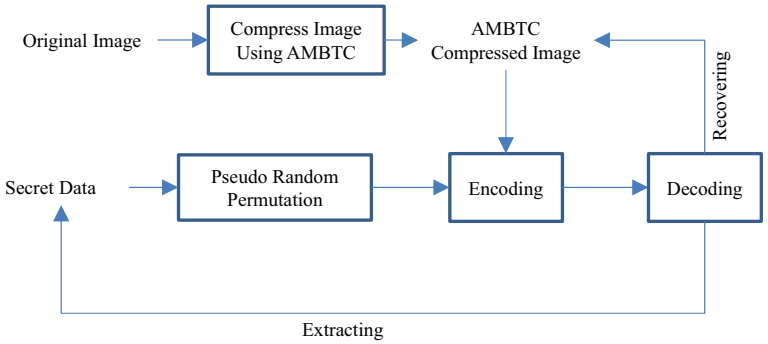


Fig. 2 Overall flow of the proposed algorithm

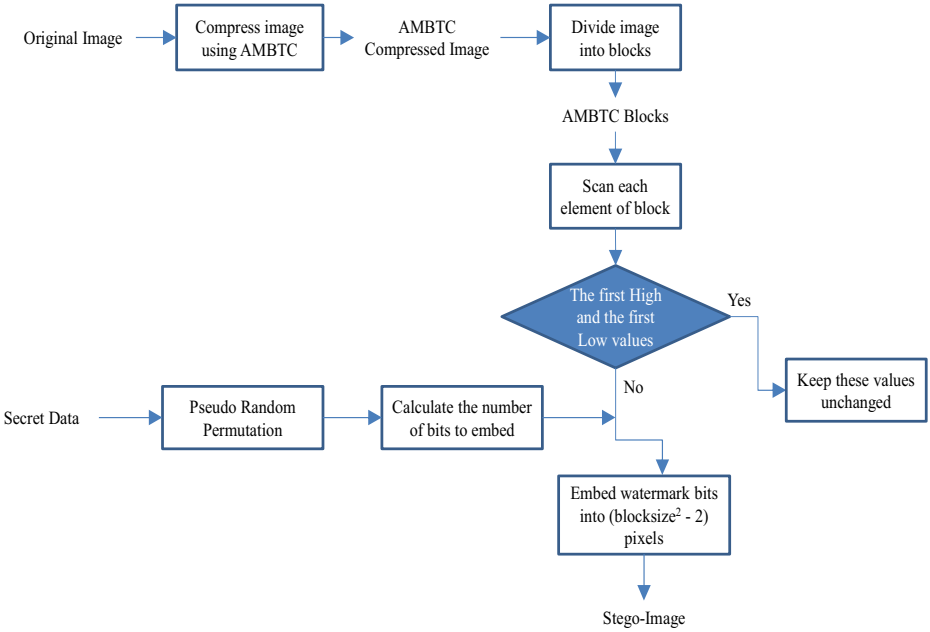


Fig. 3 Flowchart of secret embedding procedure

In our algorithm, we assume that I is the original grayscale image of $H \times W$ pixels represented as

$$I = \{x_{ij} | 0 \le i < H, 0 \le j < W\}, \text{ where } x_{ij} \in \{0, 1, ..., 255\}$$

S is the n-bit secret message such that $S = \{s_i | 0 \le i < n, s_i \in \{0, 1\}\}$. The secret message S is further arranged to be a k-bit virtual image S' which is represented as $S' = \{s_i' | 0 \le i < n', s_i' \in \{0, 1, ..., 2^k - 1\}\}$, where $n' < H \times W$. The secret message S and its corresponding embedded message S' are mapped as follows:

$$s_i' = \sum_{j=0}^{k-1} s_{i \times k + j} \times 2^{k-1-j}. \quad (5)$$

The host image is divided into non-overlapping blocks sized β. Then, we compress this image by employing AMBTC which was introduced in Section 2 to get the AMBTC compressed image C sized $H \times W$:

$$C = c_{ij} | 0 \le i < H, 0 \le j < W, \quad \text{where} \quad c_{ij} \in 0, 1, \ldots, 255 \quad .$$

For each block, the pixels are scanned in raster-scan order and the first high-mean value and low-mean value are recorded and kept unchanged. The message S' will be embedded into k-rightmost LSBs of the remained pixels of the block. The pixel of AMBTC block c_{B_i} storing the k-bit message s_i' is modified to create the marked pixel by using (6):

$$c_{B_i}' = c_{B_i} - c_{B_i} \mod 2^k + s_i'. \quad (6)$$

After all the blocks of image are processed, the marked image is obtained.

The pseudo code of the image compression and secret embedding procedure is demonstrated by Algorithm 1 as follows.

Algorithm 1: **Image Compression and Secret Embedding Procedure**

Input: Host image I such that:

$$I = \{x_{ij} \mid 0 \le i < H, 0 \le j < W\}, \text{ where } x_{ij} \in \{0,1,...,255\}$$

Secret message S: $S = \{s_i \mid 0 \le i < n, s_i \in \{0,1\}\}$

Output: Marked image C'

1. Compress the original host image I by using AMBTC to get a compressed image C.
2. Divide the AMBTC compressed image C into sub-blocks sized β.
3. Adaptive embedding for each block:
   - Keep the first high-mean value and low-mean value unchanged
   - For the remained pixels of the block, we embed secret bits into each pixel by using LSB substitution. There are three cases we need to consider.
     Case 1: The number of secret bits is less than the size of host image; we may use a secret key to choose pixels which are used for embedding.
     Case 2: The number of secret bits is as same as that of host image; we embed the secret bits into each pixel by replacing 1 LSB.
     Case 3: The number of secret bits is larger than that of host image; we embed the secret bits into each pixel by replacing 2 LSBs.
4. Process all the blocks of compressed image to construct the marked image C'.

After receiving the marked image C', to extract the secret and recover the host image, the receiver performs the secret extraction and host restoration procedure which is described in Subsection 3.2.

3.2 Secret extraction and host restoration procedure

In this subsection, we describe the process of extracting phase of MMP algorithm. An overview of secret extraction and host restoration procedure is shown in Fig. 4.

In the extraction procedure, the marked image $C'$is collected. The marked image is first divided into non-overlapping blocks as same as the embedding procedure. The embedded message can be extracted without requiring of the original host image. For each block,

the *k*-LSBs of the selected pixels are extracted by using (7) and combined to reconstruct the original secret message sequence.

$$s_i^{'} = c_{B_i}^{'} \bmod 2^k \tag{7}$$

Since the high-mean value and low-mean value of each block are unmodified, their values can be used to reconstruct the original AMBTC compressed image *C*.

The following Algorithm 2 demonstrates the pseudo code of the extraction and host restoration procedure.

Algorithm 2: Secret Extraction and Host Restoration Procedure

Input: Marked image C' such that:

$$C' = \left\{ c_{ij} \mid 0 \le i < H, 0 \le j < W \right\}, \text{ where } c_{ij} \in \{0,1,...,255\}$$

Output: Secret message S' and original compressed image *C*

1. Obtain the marked image C'.
2. Divide C' into sub-blocks sized β.
3. Extract the high-mean value and low-mean value of each block.
4. For the remained pixels, we recover the embedded message bits s_i' by $s_i' = c_{B_i}' \bmod 2^k$.
5. The original compressed image is reconstructed by replacing the greater pixels with the unchanged high-mean value and the smaller pixels with the unchanged low-mean value.
6. Process all the blocks of the marked image to reconstruct the compressed image *C*.

After all the steps are processed, we exactly obtain the original secret and the reconstructed compressed image.

In the following section, we implement our method to demonstrate that the proposed scheme is efficient and achieves a good performance.

4 Experimental results

In the performed experimental evaluation, we focus on three principal aspects. Firstly, we validate the MMP embedding algorithm to confirm that the high capacity feature is reached. Secondly, we demonstrate that the marked images generated by the MMP algorithm are in good quality. Finally, we evaluate the reconstruction efficiency to illustrate the reversibility of the algorithm. All the experiments are implemented by Matlab on Window 7 OS, platform Intel Core i7, 8GB RAM. Figure 5 is a set of standard grayscale test images which are downloaded from databases [26, 27]. Moreover, in order to objectively evaluate the scheme, we compare our scheme with several schemes which have recently been proposed.

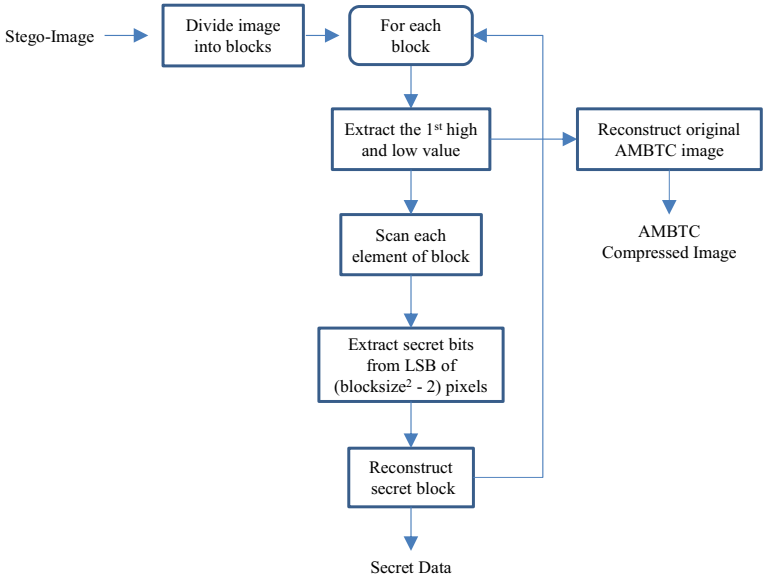


Fig. 4 Flowchart of secret extraction procedure

4.1 Fundamental parameters for validation

We adopt peak-signal-to-noise (PSNR) to measure the image quality generated by the scheme. The PSNR parameter is defined as below:

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{H \times W \times 255^2}{\sum_{i=1}^{H}\sum_{j=1}^{W}\left[\left(O_{i,j}-WI_{i,j}\right)\right]^2}\right), \tag{8}$$

where $H \times W$ is the image size, $O_{i,j}$ and $WI_{i,j}$ are the pixels of an original image and its watermarked image, respectively.

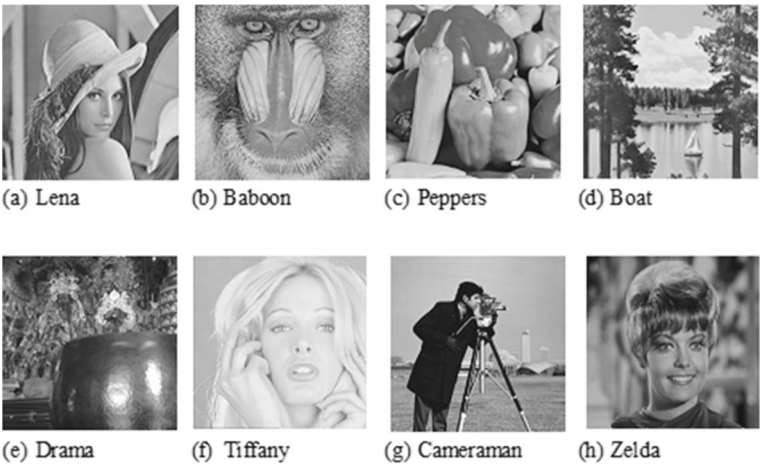


Fig. 5 Host images for testing

Besides, to evaluate the visibility error between the original watermark image and the extracted one using properties of human visual system, we use the Structural Similarity (SSIM) parameter [28], which is defined by (14). According to [28], human visual perception is exceedingly sensitive for extracting structural information from a scene. Therefore, SSIM measurement is a task of three comparisons - luminance, contrast and structure.

To compare the luminance of each image, we first compute its mean intensity:

$$\mu_{wi} = \frac{1}{H \times W} \sum_{i=1}^{H \times W} wi_i \quad , \quad \mu_{ew} = \frac{1}{H \times W} \sum_{i=1}^{H \times W} ew_i, \tag{9}$$

where $H \times W$ is the size of watermark image, wi_i and ew_i are pixel values of the original watermark and extracted watermark images, respectively. The luminance comparisons function $l(wi, ew)$ is a comparison of $\mu_{wi}$and μ_{ew}:

$$l(wi, ew) = \frac{2\mu_{wi}\mu_{ew} + C_1}{\mu_{wi}^2 + \mu_{ew}^2 + C_1}, \tag{10}$$

where $C_1 = (K_1, L)$, $L \in [0,255]$, and constant $K < 1$.

To estimate the contrast of the image, the standard deviation is adopted as (11):

$$\sigma_{wi} = \sqrt{\left(\frac{1}{H \times W-1}\right) \sum_{i=1}^{H \times W-1} (wi_i - \mu_{wi})^2}. \tag{11}$$

The contrast comparisons $c(wi, ew)$is the comparison of μ_{wi} and μ_{ew}:

$$c(wi, ew) = \frac{2\sigma_{wi}\sigma_{ew} + C_2}{\sigma_{wi}^2 + \sigma_{ew}^2 + C_2}. \tag{12}$$

The structure comparison is computed by:

$$s(wi, ew) = \frac{\sigma_{wi,ew} + C_3}{\sigma_{wi}\sigma_{ew} + C_3}. \tag{13}$$

The structural similarity measurement is computed by combining (10), (12) and (13) to yield:

$$SSIM(wi, ew) = [l(wi, ew)]^{\alpha}[c(wi, ew)]^{\beta}[s(wi, ew)]^{\gamma}. \tag{14}$$

Finally, we use the mean SSIM (MSSIM) to evaluate the image quality:

$$MSSIM(wi, ew) = \frac{1}{H \times W} \sum_{j=1}^{H \times W} SSIM(wi, ew). \tag{15}$$

4.2 Results and discussions

The original image is processed block by block during the compression procedure. Next, this compressed image is considered as a host image to carry the secret message. Therefore, the size of block β has a major effect on the quality of marked image. We illustrate our scheme with various sizes of β, where $\beta \in \{4, 8, 16, 32\}$. For each block-size, the highest number of embedded bits is calculated as follows.

– Since the image's block size is $\beta \times \beta$ pixels, the number of blocks is:

$$N = \frac{H \times W}{\beta^2}.$$

– For each block, we keep two values - low-mean value and high-mean value – to recover the original compressed image. Therefore, the number of remained pixels is β^2–2.
– For each pixel of a block, we embed k bits. The embedding capacity of each block is $(\beta^2–2) \times k$.
– Thus, the embedding capacity of the proposed MMP algorithm is defined as below:

$$Capacity = \frac{H \times W}{\beta^2} \times \left(\beta^2–2\right) \times k.$$

Table 1 shows the performance of the MMP algorithm in term of embedding capacity and structural similarity on different test images under different block sizes.

Furthermore, Tables 2, 3 and 4 show that our proposed method can work well on different types of images, i.e. smooth images and complex images. Tables 2, 3 and 4 demonstrate the performance of the MMP algorithm on the image of Lena, Baboon and Cameraman, respectively. Tables 2, 3 and 4 results illustrate that the embedding rate increases when the block-size is larger. Since the larger the block-size is, the lesser the number of un-embedded high and low mean values achieves.

For example, when block-size, $\beta = 4$, the highest embedding rate is 1.75 bpp. This means that we can embed at most $\frac{512\times512}{4^2} \times \left(4^2–2\right) \times 2 = 458752$ bits into a 512 × 512 host image.

The results in Tables 2, 3 and 4 also show that the scheme generates a good visual quality of images, which is higher than 40 dB in average. Moreover, when the block size $\beta = 32$, the SSIM values are significantly decreased. It is obvious that the number of unchanged high and low mean values is smaller when the block size is larger. Therefore, there are more spaces for embedding secrets. Thus, the similarity of the structure might be decreased.

Figure 6 shows the correlation between the visual image quality and embedding rate. In general, both visual quality and embedding rate are trade-off. It means that the more bits we embed in the image, the lower image quality we will get. It can be seen from Fig. 6(a-d) that a good tradeoff between embedding rate and image quality is achieved since even the embedding rate increases significantly, the quality of the marked image decreases slightly.

Table 1 PSNR and SSIM values

Block size	Factor	Drama	Tiffany	Lena	Baboon	Cameraman	Zelda
$\beta = 4$	Capacity (bits)	458,752	458,752	458,752	458,752	458,752	458,752
	PSNR (dB)	39.81	43.54	43.59	43.55	43.58	43.56
	SSIM	0.9993	0.9998	0.9996	0.9997	0.9988	0.9988
$\beta = 8$	Capacity (bits)	507,904	507,904	507,904	507,904	507,904	507,904
	PSNR (dB)	40.19	43.43	43.42	43.41	43.42	43.41
	SSIM	0.9989	0.9999	0.9995	0.9997	0.9969	0.9985
$\beta = 16$	Capacity (bits)	520,192	520,192	520,192	520,192	520,192	520,192
	PSNR (dB)	40.89	43.41	43.37	43.37	43.35	43.33
	SSIM	0.9983	0.9996	0.9980	0.9970	0.9921	0.9944
$\beta = 32$	Capacity (bits)	523,264	523,264	523,264	523,264	523,264	523,264
	PSNR (dB)	40.20	43.47	43.38	43.41	43.32	43.40
	SSIM	0.9948	0.9957	0.9839	0.9996	0.9839	0.9744

Table 2 Performance of the proposed scheme for Lena image

Block size	Factor	Lena						
$\beta = 4$	Capacity (bits)	131,072	196,608	262,144	327,680	393,216	458,752	-
	PSNR (dB)	52.36	51.69	51.15	44.06	43.80	43.59	-
	Embedding rate (bpp)	0.5	0.75	1.00	1.25	1.5	1.75	-
	MSSIM	0.9996	0.9996	0.9996	0.9996	0.9996	0.9996	-
$\beta = 8$	Capacity (bits)	131,072	196,608	262,144	327,680	393,216	458,752	507,904
	PSNR (dB)	52.39	51.79	51.14	44.17	43.79	43.61	43.42
	Embedding rate (bpp)	0.5	0.75	1.00	1.25	1.5	1.75	1.9375
	MSSIM	0.9999	0.9999	0.9999	0.9999	0.9999	0.9997	0.9995
$\beta = 16$	Capacity (bits)	131,072	196,608	262,144	327,680	393,216	458,752	507,904
	PSNR (dB)	52.36	51.66	51.15	43.99	43.72	43.53	43.37
	Embedding rate (bpp)	0.5	0.75	1.00	1.25	1.5	1.75	1.9375
	MSSIM	0.9999	0.9998	0.9996	0.9985	0.9984	0.9982	0.9980
$\beta = 32$	Capacity (bits)	131,072	196,608	262,144	327,680	393,216	458,752	507,904
	PSNR (dB)	52.72	51.86	51.14	44.11	43.83	43.48	43.29
	Embedding rate (bpp)	0.5	0.75	1.00	1.25	1.5	1.75	1.9375
	MSSIM	0.9989	0.9983	0.9979	0.9902	0.9887	0.9877	0.9861

In the following, we evaluate the undetectability performance of the proposed scheme by analyzing that it can withstand the enhancing LSBs attack. The enhancing LSBs attack produces a pattern image in which k LSBs of the marked image are extracted and $(8-k)$ "0" bits are appended to these k LSBs. The pattern image can be used for the detection of LSB substitution because a regular pattern will appear when LSB substitution is used for data hiding. For example, Fig. 7(b) shows an enhancing LSBs attack ($k=2$) on Fig. 7(a) that is a marked image created by LSB substitution. From the regular pattern in Fig. 7(b), the LSB substitution can be easily detected. In contrast, the proposed scheme can resist the enhancing LSBs attack as shown in Fig. 7(c) and (d).

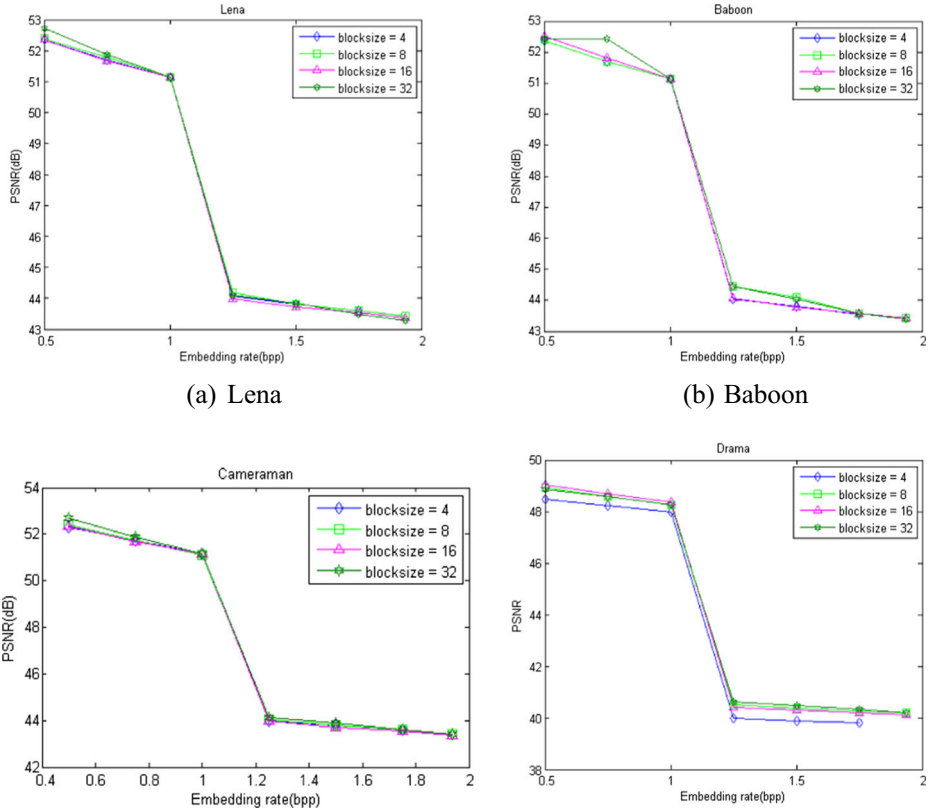
Finally, Table 5 compares the performance of some previous schemes [1, 7, 22, 34] and our proposed scheme under block size $\beta = 4$ in Table 5. Zhang et al.'s method [34], which was proposed in 2013, has a very good quality of the marked images. However, since it is based on histogram modification technique, which embeds secret according to the frequencies of the

Table 3 Performance of the proposed scheme for Baboon image

Block size	Factor	Baboon						
$\beta = 4$	Capacity (bits)	131,072	196,608	262,144	327,680	393,216	458,752	-
	PSNR (dB)	52.38	51.71	51.14	44.04	43.81	43.55	-
	Embedding rate (bpp)	0.5	0.75	1.00	1.25	1.5	1.75	-
	MSSIM	0.9997	0.9997	0.9997	0.9997	0.9997	0.9997	-
$\beta = 8$	Capacity (bits)	131,072	196,608	262,144	327,680	393,216	458,752	507,904
	PSNR(dB)	52.40	51.69	51.15	44.11	43.81	43.56	43.41
	Embedding rate (bpp)	0.5	0.75	1.00	1.25	1.5	1.75	1.9375
	MSSIM	1.0	0.9997	0.9997	0.9997	0.9997	0.9997	0.9997
$\beta = 16$	Capacity (bits)	131,072	196,608	262,144	327,680	393,216	458,752	507,904
	PSNR(dB)	52.53	51.81	51.14	44.11	43.83	43.48	43.29
	Embedding rate (bpp)	0.5	0.75	1.00	1.25	1.5	1.75	1.9375
	MSSIM	0.9999	0.9999	0.9979	0.9902	0.9887	0.9877	0.9861
$\beta = 32$	Capacity (bits)	131,072	196,608	262,144	327,680	393,216	458,752	507,904
	PSNR(dB)	52.44	52.43	51.14	44.46	44.04	43.58	43.4
	Embedding rate (bpp)	0.5	0.75	1.00	1.25	1.5	1.75	1.9375
	MSSIM	0.9998	0.9998	0.9996	0.9975	0.9972	0.9971	0.9969

Table 4 Performance of the proposed scheme for Cameraman image

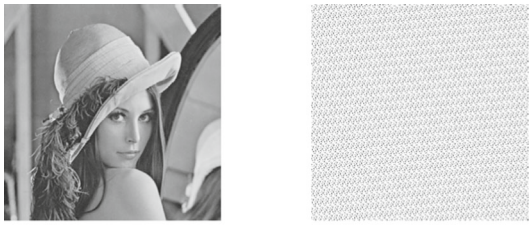
Block size	Factor	Cameraman						
$\beta = 4$	Capacity (bits)	131,072	196,608	262,144	327,680	393,216	458,752	-
	PSNR (dB)	52.30	51.71	51.15	44.00	43.78	43.58	-
	Embedding rate (bpp)	0.5	0.75	1.00	1.25	1.5	1.75	-
	MSSIM	0.9998	0.9998	0.9998	0.9989	0.9989	0.9988	-
$\beta = 8$	Capacity (bits)	131,072	196,608	262,144	327,680	393,216	458,752	507,904
	PSNR (dB)	52.40	51.74	51.13	44.02	43.80	43.59	43.42
	Embedding rate (bpp)	0.5	0.75	1.00	1.25	1.5	1.75	1.9375
	MSSIM	0.9996	0.9996	0.9995	0.9972	0.9972	0.9970	0.9969
$\beta = 16$	Capacity (bits)	131,072	196,608	262,144	327,680	393,216	458,752	507,904
	PSNR (dB)	52.36	51.66	51.15	43.99	43.72	43.53	43.37
	Embedding rate (bpp)	0.5	0.75	1.00	1.25	1.5	1.75	1.9375
	MSSIM	0.9993	0.9990	0.9988	0.9942	0.9935	0.9927	0.9922
$\beta = 32$	Capacity (bits)	131,072	196,608	262,144	327,680	393,216	458,752	507,904
	PSNR (dB)	52.69	51.86	51.14	44.11	43.89	43.62	43.39
	Embedding rate (bpp)	0.5	0.75	1.00	1.25	1.5	1.75	1.9375
	MSSIM	0.9987	0.9979	0.9975	0.9881	0.9863	0.9841	0.9836



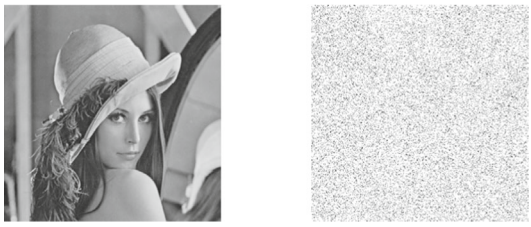
(a) Lena (b) Baboon

(c) Cameraman (d) Drama

Fig. 6 PSNR versus embedding rate of the proposed scheme



(a) Marked image by LSB (b) Enhancing LSBs attack on (a)



(c) Marked image by the proposed scheme (d) Enhancing LSBs attack on (c)

Fig. 7 The enhancing LSBs attack for "Lena"

image sequence instead of pixels themselves, the embedding rate of the scheme is quite low (not more than 1 bpp). Qin et al.'s method [22] is a new idea for data hiding technique since they combine image inpainting technique with data hiding for compression domain. The scheme embeds secret into the VQ/SMVQ compression code, thus, to make the marked image imperceptible, the user should carefully control the compression rate. Therefore, the embedding rate of the scheme cannot be enlarged. Table 5 shows that our scheme achieves better quality and higher embedding rate than that of other schemes.

5 Conclusions

In this paper, a novel reversible data hiding method called MMP algorithm for AMBTC compressed images is proposed. Our MMP algorithm can embed a large amount of secret bits while maintaining a good visual quality of the marked image. Compared to those existing data hiding methods for compressed images, our proposed scheme achieves a good performance in terms of embedding rate, image quality as well as reversibility.

Table 5 Performance comparisons

Method	Secret embedding manner	Embedding capacity (bpp)	Decoded quality (dB)	Image quality
Chang et al. [1]	Bitmap and quantization level	0.22	Lossless	33.00
Zhang et al. [34]	Entropy coder	0.80	Lossless	32.20
Qin et al. [22]	Compression code	0.04	Lossless	31.27
Guo et al. [7]	Bitmap only	1.00	Lossless	39.12
Proposed MMP scheme	Compressed image	1.94	Lossless	43.61

References

1. Chang CC, Lin CY, Fan YH (2008) Lossless data hiding for color images based on block truncation coding. Pattern Recogn 41:2347–2357
2. Coatrieux G, Pan W, Boulahia NC, Cuppens F, Roux C (2013) Reversible watermarking based on invariant image classification and dynamic histogram shifting. IEEE Trans Inf Forensics and Secur 8:111–120
3. Coltuc D (2011) Improved embedding for prediction-based reversible watermarking. IEEE Trans Inf Forensics Secur 6:873–882
4. Delp EJ, Mitchell OR (1979) Image compression using block truncation coding. IEEE Trans Commun 27: 1335–1341
5. Fallahpour M, Megias D, Ghanbari M (2011) Reversible and high-capacity data hiding in medical image. IET Image Process 5:190–197
6. Guo JM, Liu YF (2010) Joint compression/watermarking scheme using majority-parity guidance and halftoning-based block truncation coding. IEEE Trans Image Process 19:2056–2069
7. Guo JM, Liu YF (2012) High capacity data hiding for error-diffused block truncation coding. IEEE Trans Image Process 21:4808–4818
8. Guo JM, Tsai JJ (2011) Data-hiding in halftone images using adaptive noise-balanced error diffusion. IEEE Multimedia 18:48–59
9. Guo JM, Su CC, Liu YF, Lee H, Lee JD (2012) Oriented modulation for watermarking in direct binary search halftone images. IEEE Trans Image Process 21:4117–4126
10. Hamghalam M, Mirzakuchaki S, Akhaee MA (2013) Robust image watermarking using dihedral angle based on maximum-likelihood detector. IET Image Process 7:451–463
11. Hong W, Chen TS (2012) A novel data embedding method using adaptive pixel pair matching. IEEE Trans Inf Forensics 7:176–184
12. Hong W, Chen TS, Wu HY (2012) An improved reversible data hiding in encrypted images using side match. IEEE Signal Process Lett 19:199–202
13. Hou CL, Lu CC, Tsai SC, Tzeng WG (2011) An optimal data hiding scheme with tree-based parity check. IEEE Trans Image Process 20:880–886
14. Jung SW, Ha LT, Ko SJ (2011) A new histogram modification based reversible data hiding algorithm considering the human visual system. IEEE Signal Process Lett 18:95–98
15. Lema MD, Mitchell OR (1984) Absolute moment block truncation coding and its application to color images. IEEE Trans Commun 32:1148–1157
16. Li X, Yang B, Zeng T (2011) Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection. IEEE Trans Image Process 20:3524–3533
17. Li X, Zhang W, Gui X, Yang B (2013a) A novel reversible data hiding scheme based on two-dimensional difference-histogram modification. IEEE Trans Inf Forensics Secur 8:1091–1100
18. Li X, Li B, Yang B, Zeng T (2013b) General framework to histogram-shifting-based reversible data hiding. IEEE Trans Image Process 22:2181–2191
19. Liu YF, Guo JM, Lee JD (2011) Inverse halftoning based on the Bayesian theorem. IEEE Trans Image Process 20:1077–1084
20. Ma K, Zhang W, Zhao X, Yu N, Li F (2013) Reversible data hiding in encrypted images by reserving room before encryption. IEEE Trans Inf Forensics Secur 8:553–562
21. Ni Z, Shi YQ, Ansari N, Su W (2006) Reversible data hiding. IEEE Trans Circuits Syst Video Technol 16: 354–362
22. Qin C, Chang CC, Chiu YP (2014) A novel joint data-hiding and compression scheme based on SMVQ and image inpainting. IEEE Trans Image Process 23:969–978
23. Shi YQ, Li X, Zhang X, Wu H, Ma B (2016) Reversible data hiding: advances in the past two decades. Special Section in IEEE Access: Latest Advances and Emerging Applications of Data Hiding 4:3210–3237
24. Subramanyam AV, Emmanuel S, Kankanhalli MS (2013) Robust watermarking of compressed and encrypted JPEG2000 images. IEEE Trans Multimedia 14:703–716
25. Tai WL, Yeh CM, Chang CC (2009) Reversible data hiding based on histogram modification of pixel differences. IEEE Trans Circuits Syst Video Technol 19:906–920
26. The test image database. (2012) [Online]. Available: http://msp.e.ntust.edu.tw/publicfile/ImageSet.rar
27. The USC-SIPI image database (1997) [Online]. Available: http://sipi.usc.edu/database/
28. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process 13:600–612
29. Wang SY, Li CY, Kuo WC (2013) Reversible data hiding based on two-dimensional prediction errors. IET Image Process 7:805–816
30. Zareian M, Tohidypour HR (2013) Robust quantization index modulation-based approach for image watermarking. IET Image Process 7:432–441

31. Zhang X (2011) Reversible data hiding in encrypted image. IEEE Signal Process Lett 18:255–258
32. Zhang X (2012) Separable reversible data hiding in encrypted image. IEEE Trans Inf Forensics Secur 7: 826–832
33. Zhang W, Chen B, Yu N (2012) Improving various reversible data hiding schemes via optimal codes for binary covers. IEEE Trans Image Process 21:2991–3003
34. Zhang W, Hu X, Li X, Yu N (2013) Recursive histogram modification: establishing equivalency between reversible data hiding and lossless data compression. IEEE Trans Image Process 22:2775–2785

Ngoc-Tu Huynh received the BS degree in mathematics — informatics in 2006 from Danang University, Vietnam, and the MS degree in information engineering and computer science in 2010 from Feng Chia University. Since 2006, she has been a lecturer of Department of Computer Science, College of Information Technology, Danang University, Vietnam. She is currently pursuing her Ph.D in information engineering and computer science, Feng Chia University, Taichung, Taiwan. Her research interests include visual cryptography, watermarking, steganography and image processing.

K. Bharanitharan (S'07–M'09) received the PhD degree in Electrical Engineering from the National Cheng Kung University, Tainan, Taiwan, in 2009. In 2005, he won outstanding international student fellowship award at National Cheng Kung University. He serves as a reviewer for IEEE Transactions on Circuits and Systems for Video Technology, IEEE Transactions on Very Large Scale Integration Systems, IEEE Transactions on Evolutionary Computation, IEEE Signal processing letter, IEEE Transactions on Very Large Scale Integration Systems

since 2009. He has published more than 16 research papers in highly reputed journals and conferences. His research interests include H.264/AVC video coding, HEVC, scalable video coding, image processing, multiview video coding, and associated VLSI architectures. His research works also include Multi-Core reconfigurable systems, Java-based apps development, and dynamic power management for advanced video coding.

Chin-Chen Chang received his Ph.D. degree in computer engineering from National Chiao Tung University. His current title is Chair Professor in Department of Information Engineering and Computer Science, Feng Chia University, from February 2005. He is currently a Fellow of IEEE and a Fellow of IEE, UK. And, since his early years of career development, he consecutively won Outstanding Talent in Information Sciences of the R. O. C., AceR Dragon Award of the Ten Most Outstanding Talents, Outstanding Scholar Award of the R. O. C., Outstanding Engineering Professor Award of the R. O. C., Distinguished Research Awards of National Science Council of the R. O. C., Top Fifteen Scholars in Systems and Software Engineering of the Journal of Systems and Software, and so on. 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.

Yanjun Liu received her Ph.D. degree in 2010, in School of Computer Science and Technology from University of Science and Technology of China (USTC), Hefei, China. She has been an assistant professor serving in Anhui University in China since 2010. She currently serves as a senior research fellow in Feng Chia University in Taiwan. Her specialties include E-Business security and electronic imaging techniques.