

An efficient reversible data hiding method for AMBTC compressed images

Wien Hong^{1,2} · Yuan-Bo Ma²  · Hung-Che Wu³ ·
Tung-Shou Chen⁴

Received: 12 February 2016 / Revised: 10 September 2016 / Accepted: 4 October 2016 /
Published online: 31 October 2016
© Springer Science+Business Media New York 2016

Abstract Analyzing multimedia data in mobile devices is often constrained by limited computing capacity and power storage. Therefore, more and more studies are trying to investigate methods with algorithm efficiency. Sun et al. proposed a low computing cost reversible data hiding method for absolute moment block truncation coding (AMBTC) images with excellent embedding performance. Their method predicts quantization values and uses encrypted data bits, division information, and prediction errors to construct the stego codes. This method successfully embeds data while providing a comparable bit-rate; however, it does not fully exploit the correlation of neighboring pixels and division of prediction error for better embedding. Therefore, the payload and bit-rate are penalized because the embedding performance directly depends on the prediction accuracy and division efficiency. In this paper, we use median edge detection predictor to better predict the quantization values. We also employ an alternative prediction technique to increase the prediction accuracy by narrowing the range of prediction values. Besides, an efficient centralized error diversion technique is proposed to further decrease the bit-rate. The experimental results show that the proposed method offers 8 % higher payload with 5 % lower bit-rate on average if compared to Sun et al.'s method and has better embedding performance than prior related works.

Keywords Reversible data hiding · AMBTC · Prediction · Low computing cost

✉ Yuan-Bo Ma
mayb1314@foxmail.com

¹ School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, China

² Department of Electronic Communication and Software Engineering, Nanfang College of Sun Yat-Sen University, Guangzhou, China

³ Department of Business Administration, Nanfang College of Sun Yat-Sen University, Guangzhou, China

⁴ Department of Computer Science and Information Engineering, National Taichung University of Science and Technology, Taichung, Taiwan

1 Introduction

Information security is a relatively new research area and has been widely investigated in multiple fields such as cloud service [6, 31], networking [9, 23], and multimedia applications [17, 29, 30]. As one of the emerging research area, data hiding in social multimedia is in a constant progress toward a new era. Nowadays, data hiding technology is able to embed data into the vast majority of digital media, such as images, audios and videos. Among them, images are widely used as cover images since they are easily available. After embedding, the image containing secret data is called the stego image. According to whether the original image can be reconstructed, data hiding methods are classified into lossy [10, 11, 15, 20, 22] and reversible [13, 19, 21, 24, 27]. For lossy methods, the cover image is permanently destroyed after embedding so that the original image is unable to recover. However, in some special applications such as medical and military usages, people need to restore the cover image while hiding the secret data. Therefore, the reversible data hiding (RDH) methods have been widely investigated and fast developed.

RDH methods can be applied to images in spatial or compressed domain. Spatial domain RDH methods directly modify image pixels in the spatial domain to insert data. Methods of this type are easily implemented and offer a relatively high payload and quality. Therefore, researchers have widely paid attention to investigating RDH methods of this type. In 2003, Tian [27] proposed an efficient RDH method based on difference expansion for digital images. Later in 2006, Ni et al. [21] advanced a histogram based RDH method. Data is embedded into peak points by shifting the pixels of histogram. Many other RDH methods [13, 19, 24] have been proposed to extend Ni et al.'s and Tian's works and have better embedding performance. On the other hand, compressed domain RDH methods embed data into an image's compressed format. As the compressed image is the most common form in mobile social networks, investigating RDH methods for compressed image have become an important issue. As a result, the RDH methods for compressed images, such as joint photographic experts group (JPEG) [1], vector quantization (VQ) [2], and block truncation coding (BTC) [3, 12, 18, 25, 33], are proposed recently to fulfill the requirements in this field. The RDH methods for JPEG compressed images embed data by modifying the coefficients in the frequency domain, while RDH methods for VQ compressed images embed data by modifying the index table or rearranging the corresponding codebook. RDH methods for compressed images of these two types often require significant amount of computing cost.

Similar to JPEG and VQ compressed images, BTC is also a block-based compression method of which the compressed code of block i is represented by a trio (q_i^H, q_i^L, B_i) , where q_i^H and q_i^L are a pair of quantization values, and B_i is a bitmap. This method was first presented by Delp and Mitchell [5] in 1979 and later Lema and Mitchell [16] proposed the absolute mean block truncation coding (AMBTC) method to improve the performance of BTC. Unlike VQ methods using machine learning techniques [4, 7, 8, 28, 32, 34] to obtain the codebook, AMBTC requires insignificant computing cost and can be easily implemented. Therefore, it is suitable for low bandwidth communication channels and low power image processing applications, such as personal digital assistant (PDA), field-programmable gate array (FPGA) and portable image signal processor. Due to the vast demands for embedding data into low computing cost compressed images, a number of lossy and reversible data hiding methods for AMBTC codes have been proposed.

For lossy data hiding method, AMBTC-compressed code is usually classified as smooth or complex blocks by a predefined threshold. In methods proposed by Ou et al. [22] in 2015, more secret data bits are embedded into smooth blocks and two quantization levels are

recalculated to minimize the distortion. In 2016, Huang et al. [15] proposed a hybrid method to embed more secrets into the complex blocks with large variation of the blocks. At the same year, Malik et al. [20] modified the AMBTC technique by providing two bit plane with four quantization levels to improve image quality and payload. Although the aforementioned methods achieve good visual quality and payload, they cause permanent distortion to the original AMBTC codes. To fulfill the requirement of reversibility of the AMBTC codes, some RDH methods for AMBTC are proposed [3, 12, 18, 25, 33] for diverse applications.

According to the format of AMBTC stego codes, RDH methods for AMBTC are grouped into two types. In type I method, the AMBTC stego codes that mimic the traditional AMBTC codes are created as output. That is, the stego codes can be decoded via a standard AMBTC decoder and transformed into a stego image directly. Chen et al. [3] proposed a simple but efficient RDH for AMBTC codes of this type. Their method embeds data bits by exchanging the quantization value q_i^H and q_i^L to achieve the reversibility of data hiding. Inspired by Chen et al.'s method, some advanced methods with enhanced capacity and image quality are proposed [12, 18]. In 2015, Lin et al. [18] proposed a type I method which secret data is embedded into selected AMBTC block using different combinations of the mean value and the standard deviation. Although methods of this type create formal AMBTC stego codes, they suffer from low embedding capacity and high bit-rate.

The type II RDH methods for AMBTC codes represent the AMBTC stego codes with specific coding structures [25, 33]. Besides bitmap B_i , additional information such as encrypted data bits, division information, and prediction errors are added to a binary code stream to form the stego codes. Methods of this type cannot be directly decoded by a standard AMBTC decoder but only with the decoders designed for the corresponding methods. Nevertheless, with the help of the special structure of stego codes, type II methods often offer higher payload and smaller stego codes than those of type I methods. This means that more secret data can be embedded while saving the storage space.

In 2013, Sun et al. [25] proposed a type II RDH for AMBTC method which combines the joint neighboring coding (JNC) technique for better embedding performance. JNC was firstly proposed by Chang et al. [2] to improve the embedding capacity. In [2], a key-selected neighbor is chosen to predict each to-be-encoded element, and the error between the selected neighbor and the to-be-encoded element is calculated. Once the error is obtained, it is represented by different numbers of bits according to a pre-defined classification rule. Since the to-be-encoded element is highly correlated to its neighbors, the prediction error tends to be small. As a result, the codes can be represented with fewer bits and they thus provide extra rooms for data insertion. Sun et al. exploit the JNC technique to propose a RDH method for AMBTC codes. In their method, secret data are embedded into pairs of quantization values. To embed data, the key-selected neighbor of the to-be-encoded quantization value is chosen to predict this quantization value, and later the prediction error is calculated. The prediction error is then encoded based on a pre-defined error division. Finally, the encrypted data bits, division information and encoded prediction error are concatenated to generate final compressed stego codes.

In type II method, a stego code of low bit-rate is crucial. The bit-rate can be reduced either by using a better predictor to reduce the prediction errors, or by employing a better classification rule to efficiently encode the prediction errors. However, the JNC technique adopted in Sun et al.'s method unnecessarily increases the bit-rate because the selection of prediction value is equivalent to a random selection. A random selected prediction value often causes larger prediction errors, and subsequently increases the bit-rate. Moreover, the prediction error

classification rule used in Sun et al.’s method may cause more bits to encode because they use m bits to record the prediction errors falling in the range $[-(2^m - 1), 2^m - 1]$. However, the most frequently occurred prediction error is zero, and thus a prediction error of zero should be represented by using fewer bits than other less occurred prediction errors.

To address the aforementioned problems, this paper proposes three techniques to perform Type II RDH for AMBTC codes to improve Sun et al.’s method. The main contribution of this paper is to effectively reduce the size of the stego code, as summarized as follows:

- 1) We adopt the median edge detection (MED) predictor instead of JNC to predict the quantization values. Since the neighboring quantization values are highly correlated and MED always chooses either the best or the second-best prediction values from three neighboring candidates, MED provides a better prediction result than that of JNC used in Sun et al.’s method.
- 2) We observe that prediction value of the higher quantization value cannot be smaller than its lower counterpart and vice versa. Therefore, we propose an alternative prediction (AP) technique to narrow the range of prediction values and thus the prediction errors can be further reduced.
- 3) Because zero is the most frequent occurred prediction error, we further propose a centralization error division (CED) technique to modify the classification rule to categorize zero prediction errors into a new division and require no bits to record.

The rest of this paper is organized as follows. Section 2 introduces the AMBTC coding method, Sun et al.’s method, and MED predictor. Section 3 illustrates the proposed method and discusses the basis of how it works. In Section 4, we conduct several experiments to validate the applicability of the proposed method. The concluding remarks are given in Section 5.

2 Related work

In this section, AMBTC coding method is briefly introduced, followed by Sun et al.’s RDH method for AMBTC codes and MED predictor.

2.1 AMBTC coding method

AMBTC is a block-based lossy compression technique. The concept of AMBTC is to preserve the mean and first absolute central moment of each block in images. In the encoding procedure, an image I is partitioned into a set of non-overlapping blocks $\{\Phi_i\}_{i=1}^N$ of size $v \times v$, where N is the total number of blocks. Let $\{\Phi_{i,j}\}_{j=1}^{v \times v}$ be the j -th pixel in block Φ_i , and $\bar{\Phi}_i$ be the block mean. Two quantization values, i.e., higher quantization value q_i^H and lower quantization value q_i^L of block Φ_i , can be calculated as

$$q_i^H = \frac{1}{y_i} \sum_{\Phi_{i,j} \geq \bar{\Phi}_i} \Phi_{i,j}, \tag{1}$$

$$q_i^L = \frac{1}{v \times v - y_i} \sum_{\Phi_{i,j} < \bar{\Phi}_i} \Phi_{i,j}, \tag{2}$$

where y_i denotes the number of pixels having a value larger than or equal to $\bar{\Phi}_i$. The compressed code for block Φ_i is then represented by the trio (q_i^H, q_i^L, B_i) , where $B_i = \{B_{i,j}\}_{j=1}^{v \times v} \in (0, 1)$ is the bitmap for recording the quantized result. The j -th bit in B_i can be obtained by using the following equation:

$$B_{i,j} = \begin{cases} 1 & \Phi_{i,j} \geq \bar{\Phi}_i, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

Each block is processed in the same manner, and the AMBTC compressed codes $\{q_i^H, q_i^L, B_i\}_{i=1}^N$ are obtained. To decode compressed codes $\{q_i^H, q_i^L, B_i\}_{i=1}^N$, each trio (q_i^H, q_i^L, B_i) is investigated. Let $\tilde{\Phi}_i$ be the de-compressed block of the i -th trio, and $\tilde{\Phi}_{i,j}$ be the j -th pixel in $\tilde{\Phi}_i$. If $B_{i,j} = 0$, we set $\tilde{\Phi}_{i,j} = q_i^L$. On the other hand, if $B_{i,j} = 1$, we set $\tilde{\Phi}_{i,j} = q_i^H$. After decoding all the compressed codes, the de-compressed image $\tilde{I} = \{\tilde{\Phi}_i\}_{i=1}^N$ can be obtained.

2.2 Sun et al.’s method

In 2013, Sun et al. [25] proposed a type II RDH method for AMBTC codes using JNC. This method selects a prediction value according to the to-be-embedded bits and obtains the prediction errors for data embedment. To embed data, Sun et al.’s method classifies the quantization values into referential and predictable groups. The quantization values in the first row, the first column, and the last column are chosen as the referential quantization values for recovering the predictable ones. Other quantization values are predictable, where two bits secret data can be carried by each quantization value. Since the embedment in higher and lower quantization values share the same procedures, we only take higher ones for illustration.

Let $\{q_i^H, q_i^L, B_i\}_{i=1}^N$ be the AMBTC compressed codes and q_i^H be the higher quantization value to be encoded. One of the four neighboring quantization values of q_i^H is used to predict the value of q_i^H . The locations of these four neighboring quantization values are represented by binary indices $(00)_2$, $(01)_2$, $(10)_2$ and $(11)_2$, respectively, as shown in Fig. 1. To select which neighbor should be used to predict q_i^H , two secret bits $s_{i,1}, s_{i,2}$ are extracted from secret data S

Fig. 1 The Neighbors of

$(01)_2$	$(10)_2$	$(11)_2$
$(00)_2$	q_i^H	

and are xor-ed with two key-generated random bits $r_{i,1}r_{i,2}$ to obtain the xor-ed results $s'_{i,1}s'_{i,2} = s_{i,1}s_{i,2} \oplus r_{i,1}r_{i,2}$, where \oplus is the xor operator. The selected quantization value for predicting q_i^H , denoted by p_i^H , is the neighboring quantization value whose binary indices are $s_{i,1}s_{i,2}$.

Secret bits $s_{i,1}s_{i,2}$ are embedded by inserting $s'_{i,1}s'_{i,2}$ into the codes of q_i^H . To encode q_i^H , the prediction error $e_i^H = q_i^H - p_i^H$ is calculated. e_i^H is then represented by 8-bit or m -bit binary codes according to the equations given below:

$$(e_i^H)_2 = \begin{cases} (e_i^H, m)_2, & 0 \leq e_i^H \leq 2^m - 1, \\ (-e_i^H, m)_2, & -(2^m - 1) \leq e_i^H < 0, \\ (e_i^H, 8)_2, & e_i^H > 2^m - 1, \\ (-e_i^H, 8)_2, & e_i^H < -(2^m - 1), \end{cases} \tag{4}$$

where $(e_i^H, x)_2$ denotes the x -bit binary code of e_i^H . Eq. (4) partitions the range of e_i^H into four divisions and uses two bits $d_{i,1}d_{i,2}$ to record the division information. The assignment of the two-bit division information is given in Fig. 2. The xor-ed secret bits $s'_{i,1}s'_{i,2}$, division information $d_{i,1}d_{i,2}$, and the prediction error $(e_i^H)_2$, are concatenated. Therefore, the final codes $C_i^H = s'_{i,1}s'_{i,2} || d_{i,1}d_{i,2} || (e_i^H)_2$ for q_i^H can be obtained, where $||$ is the concatenation operator. The block size and the length of secret data are used as the side information to extract the secret data and recover the AMBTC code.

To extract the embedded secret bits and recover q_i^H from C_i^H , the key-generated random bits $r_{i,1}r_{i,2}$ are obtained. Extract the first two bits $s'_{i,1}s'_{i,2}$ from C_i^H , and xor $s'_{i,1}s'_{i,2}$ with $r_{i,1}r_{i,2}$, the secret bits $s_{i,1}s_{i,2}$ are obtained. According to $s_{i,1}s_{i,2}$, the prediction value p_i^H that is used to predict q_i^H can also be determined from one of the four neighboring quantization values of q_i^H which have been recovered previously. Extract the next two bits $d_{i,1}d_{i,2}$, we have the division information and then extract $(e_i^H)_2$ from C_i^H . Convert $(e_i^H)_2$ into decimal form, q_i^H can be recovered by using the equation $q_i^H = e_i^H + p_i^H$.

Here we use a simple example to illustrate the encoding and decoding procedures of Sun et al.'s method. Let $q_i^H = 176$ be the higher quantization value to be encoded, and $m = 4$. Figure 3 shows the arrangement of q_i^H and its four neighbors with pre-defined location $(00)_2$, $(01)_2$, $(10)_2$ and $(11)_2$, respectively. Let the two key-generated random bits $r_{i,1}r_{i,2}$ be 01_2 and two secret bits $s_{i,1}s_{i,2}$ be 10_2 . The xor-ed bits can be calculated as $s'_{i,1}s'_{i,2} = 10_2 \oplus 01_2 = 11_2$. Therefore, the location of prediction value p_i^H is $s'_{i,1}s'_{i,2} = 11_2$. According to the location defined in Fig. 1, the upper-right neighboring quantization value located at $(11)_2$ is selected as the prediction value $p_i^H = 189$. The prediction error can then be calculated by $e_i^H = q_i^H - p_i^H = 176 - 189 = -13$. Because $-(2^m - 1) \leq e_i^H < 0$, the division information is $d_{i,1}d_{i,2} = 10_2$. Using Eq. (4), the 4-bit binary prediction error is $(e_i^H)_2 = 1101_2$. Concatenate $s'_{i,1}s'_{i,2}$, $d_{i,1}d_{i,2}$, and $(e_i^H)_2$, the final codes $C_i^H = s'_{i,1}s'_{i,2} || d_{i,1}d_{i,2} || (e_i^H)_2 = 11 \ 10 \ 1101_2$ are obtained.

To recover q_i^H and extract the embedded bits from $C_i^H = 11101101_2$, the first two bits $s'_{i,1}s'_{i,2} = 11_2$ are read. The embedded secret bits $s_{i,1}s_{i,2} = 10_2$ can be obtained by taking the

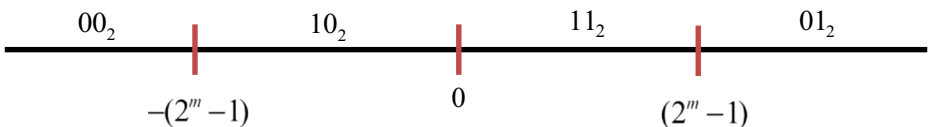


Fig. 2 Assignment of the division information

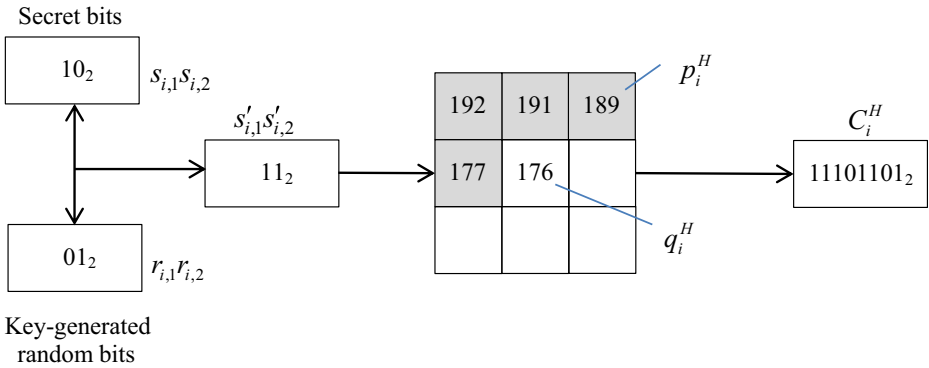


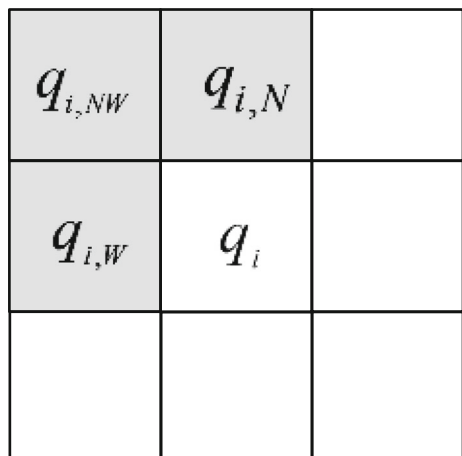
Fig. 3 An example of Sun et al.'s method

xor operation on $s'_{i,1}s'_{i,2} = 11_2$ and the key-generated random bits $r_{i,1}r_{i,2} = 01_2$. Since $s'_{i,1}s'_{i,2} = 11_2$, the prediction value $p_i^H = 189$ can be located. By reading the next two bits $d_{i,1}d_{i,2} = 10_2$ from C_i^H , the division information is known and the prediction error is recorded in the next four bits. Therefore, $(e_i^H)_2 = 1101_2$ and the prediction error is negative. The quantization value q_i^H can then be recovered by $q_i^H = e_i^H + p_i^H = -13 + 189 = 176$.

2.3 Median Edge Detection (MED) predictor

The MED predictor [14] is an edge detection method which is initially used in image compression and computer vision. The essence of the MED predictor is to detect the most change in intensity by extracting the boundary between the visited pixel and its neighbors in images. The MED predictor provides a more accurate result with insignificant computing cost. As a result, the proposed method adopts the MED predictor to predict quantization values. Let q_i be the visited higher or lower quantization value. The three neighboring quantization values of q_i , denoted by $q_{i,N}$, $q_{i,NW}$ and $q_{i,W}$ respectively, are showed in Fig. 4.

Fig. 4 The context of the MED predictor



The MED predictor predicts q_i by using its neighboring quantization values to obtain the prediction value p_i , as given in the following equation:

$$p_i = \begin{cases} \min(q_{i,N}, q_{i,W}), & q_{i,NW} \geq \max(q_{i,N}, q_{i,W}), \\ \max(q_{i,N}, q_{i,W}), & q_{i,NW} \leq \min(q_{i,N}, q_{i,W}), \\ q_{i,N} + q_{i,W} - q_{i,NW}, & \text{otherwise.} \end{cases} \quad (5)$$

3 Proposed method

In contrast to type I method, Sun et al.'s method belongs to type II and offers a significant higher embedding capacity with lower bit-rate. However, there are some issues which can be re-considered to improve the embedding performance. Firstly, the prediction value in Sun et al.'s method is randomly selected from four neighboring quantization values. The random selection comes out a relatively inaccurate prediction value, which directly leads to a larger error and thus it requires more compressed codes. Secondly, the prediction error is encoded by assigning different bits based on divisions shown in Fig. 2. Note that Sun et al.'s method uses m bits to record the prediction errors within the range $[-(2^m - 1), 2^m - 1]$. However, a prediction error of zero is the most frequently occurred value and is unnecessary to be represented by m bits. A better division of errors should take this property into account in order to minimize the bit-rate.

We propose a novel type II RDH method for AMBTC codes which advances new techniques to improve Sun et al.'s method. First of all, in the proposed method, the prediction value is chosen by median edge detection (MED) predictor instead of random selection to achieve a better result. Secondly, we propose an alternative prediction (AP) technique to further narrow the range of prediction errors. As for an AMBTC code (q_i^H, q_i^L, B_i) , the equation $q_i^H \geq q_i^L$ always holds. Therefore, the minimal prediction value p_i^H of q_i^H should be q_i^L , and the maximal prediction value p_i^L of q_i^L should be q_i^H . By restricting the minimum value of p_i^H or the maximum value of p_i^L , the AP technique successfully narrows the prediction errors so as to reduce the bit-rate. Thirdly, the centralized error division (CED) technique is used to better classify the errors into divisions. The CED technique classifies the prediction error of zero into a new division and requires none of the bits to record in this division. Therefore, the bit-rate can be further decreased.

3.1 Alternative prediction technique

The AMBTC codes for each block have a pair of quantization values with the numerical order $q_i^H \geq q_i^L$. Therefore, the minimal prediction value p_i^H of q_i^H can be restricted to q_i^L . As a result, if $p_i^H < q_i^L$, we may simply replace p_i^H by an alternative prediction value $\hat{p}_i^H = q_i^L$ to narrow the range of prediction errors. Similarly, if $p_i^L > q_i^H$, an alternative prediction value $\hat{p}_i^L = q_i^H$ can be set. We term the replacement of prediction values by alternative values as the alternative prediction (AP) technique. However, the prediction of q_i^H using the AP technique requires the information of q_i^L and vice versa. Therefore, the AP technique can only be used to predict either q_i^H or q_i^L . For the security consideration, we use a key to randomly select which quantization

value should be predicted using the AP technique. If q_i^H is selected to be alternatively predicted, the prediction values of q_i^H and q_i^L can be recalculated by

$$\begin{cases} \hat{p}_i^H = \max(q_i^L, p_i^H), \\ \hat{p}_i^L = p_i^L. \end{cases} \tag{6}$$

On the other hand, if q_i^L is selected to be predicted, we use the following equation to recalculate the prediction values of q_i^H and q_i^L :

$$\begin{cases} \hat{p}_i^H = p_i^H, \\ \hat{p}_i^L = \min(q_i^H, p_i^L). \end{cases} \tag{7}$$

Note that, when only the MED predictor is applied, the range of prediction error e_i for a given 8-bit quantization value is $[-255, 255]$. With the proposed AP technique, the range of prediction errors can be reduced to $|e_i| \leq q_i^H - q_i^L$ and thus the bit-rate can be further reduced with only a small amount of calculation.

In the decoding procedure, the prediction values p_i^H and p_i^L are obtained using the MED predictor. According to the given key, the decoder knows whether q_i^H or q_i^L is selected for prediction at the encoding stage, and thus Eq. (6) or (7) can be applied to obtain the prediction value \hat{p}_i^H and \hat{p}_i^L .

3.2 Centralized Error Division (CED) technique

In type II RDH method for AMBTC codes, the division arrangement of prediction errors affects the bit-rate significantly. A well-designed error division should consider the occurrence of individual prediction errors, and use as fewer bits as possible to represents the most occurred error. Let $e_i = q_i - \hat{p}_i$ be the prediction error between the to-be-encoded quantization value q_i and the prediction value \hat{p}_i obtained by the AP technique. In the proposed method, we divide the range of prediction error e_i into four divisions and use 0, m or 8 bits to encode e_i . If $e_i = 0$, none of the bits are required to encode e_i . If $-(2^m - 1) \leq e_i < 0$ or $0 < e_i \leq 2^m - 1$, $|e_i|$ is encoded with m bits. However, if $|e_i| > 2^m - 1$, since e_i itself needs 9 bits to represent but q_i only requires 8 bits, we thus encode q_i using 8-bit binary representation of e_i . The rules of encoding the prediction error e_i are given in the following equation:

$$(e_i)_2 = \begin{cases} \text{empty}, & e_i = 0, \\ \left(|e_i|, m \right)_2, & -(2^m - 1) \leq e_i < 0, \\ (e_i, m)_2, & 0 < e_i \leq 2^m - 1, \\ (q_i, 8)_2, & |e_i| > 2^m - 1. \end{cases} \tag{8}$$

To record the division information which is required in the decoding stage, two bits $d_{i,1}d_{i,2}$ are assigned to represent the divisions used in CED, as shown in Fig. 5.

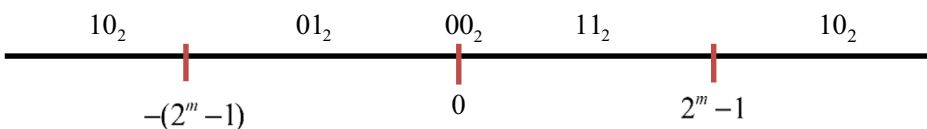


Fig. 5 Assignment of division information $d_{i,1}d_{i,2}$ for CED

In the decoding procedure, 2-bit division information $d_{i,1}d_{i,2}$ is extracted and x -bit prediction error $(e_i)_2$, where $x = 0, m, \text{ or } 8$ can be determined. The original quantization value q_i can then be recovered by applying the following equation:

$$q_i = \begin{cases} \hat{p}_i, & d_{i,1}d_{i,2} = 00_2. \\ \hat{p}_i - |e_i|, & d_{i,1}d_{i,2} = 01_2. \\ \hat{p}_i + |e_i|, & d_{i,1}d_{i,2} = 11_2. \\ e_i, & d_{i,1}d_{i,2} = 10_2. \end{cases} \tag{9}$$

3.3 Embedding procedures

To embed secret data S , image I is partitioned into $N = N_R \times N_C$ blocks of size $v \times v$, where N_R and N_C are the number of rows and columns of the partitioned blocks, respectively. Let $\{q_i^H, q_i^L, B_i\}_{i=1}^N$ be the AMBTC compressed codes of I and C be the final AMBTC stego codes. All the quantization values in the first column and the first row are pre-saved with eight bits as referential quantization values, and they do not insert any data. The rest quantization values are predictable and two secret bits can be inserted into each quantization value. The detailed embedding procedures are listed as follows.

Input: AMBTC compressed codes $\{q_i^H, q_i^L, B_i\}_{i=1}^N$, secret data S , keys k_1, k_2 and a pre-defined integer m .

Output: The AMBTC stego codes C .

- Step 1 Scan the quantization values $\{q_i^H\}_{i=1}^N$ and $\{q_i^L\}_{i=1}^N$ using the raster scan order. Use eight bits each to encode referential quantization values. Let R be the encoded results.
- Step 2 Scan the predictable quantization values and use the MED predictor to predict q_i^H and q_i^L . Use the key k_1 to randomly select whether q_i^H or q_i^L should be predicted using the AP technique, as described in Section 3.1. Let \hat{p}_i^H and \hat{p}_i^L be the predicted results.
- Step 3 Generate four random bits $r_{i,1}r_{i,2}r_{i,3}r_{i,4}$ using the key k_2 , and read four secret bits $s_{i,1}s_{i,2}s_{i,3}s_{i,4}$ from S . Perform the bitwise xor operation on $r_{i,1}r_{i,2}r_{i,3}r_{i,4}$ and $s_{i,1}s_{i,2}s_{i,3}s_{i,4}$ to obtain the xor-ed bits $s'_{i,1}s'_{i,2}s'_{i,3}s'_{i,4}$.
- Step 4 Calculate the prediction errors $e_i^H = q_i^H - \hat{p}_i^H$ and $e_i^L = q_i^L - \hat{p}_i^L$. According to e_i^H and e_i^L , the division information $d_{i,1}d_{i,2}$ and $d_{i,3}d_{i,4}$ is obtained, respectively. The x -bit prediction errors $(e_i^H)_2$ and $(e_i^L)_2$ can then be calculated, as described in Section 3.2.
- Step 5 Concatenate the xor-ed secret bits $s'_{i,1}s'_{i,2}$, division information $d_{i,1}d_{i,2}$, and prediction error $(e_i^H)_2$, we obtain the codes $C_i^H = s_{i,1}s'_{i,2}||d_{i,1}d_{i,2}||(e_i^H)_2$ for q_i^H . Similarly, the codes $C_i^L = s'_{i,3}s'_{i,4}||d_{i,3}d_{i,4}||(e_i^L)_2$ for q_i^L can be obtained.
- Step 6 Repeat Steps 2-5, until all the secret data are embedded and the quantization values are processed. Concatenate R , the codes of bitmap $\{B_i\}_{i=1}^N$, and $\{C_i^H\}_{i=1}^N, \{C_i^L\}_{i=1}^N$, the final AMBTC stego codes C are obtained.

To extract the embedded secret data and recover the original AMBTC compressed code, we have to know N_R, N_C and v so that R and $\{B_i\}_{i=1}^N$ can be accurately recovered. As a result, in addition to k_1 and k_2 , the parameters N_R, N_C and v are also taken as the keys and have to be transferred to the receiver side via a secret channel.

3.4 Decoding and extraction procedures

Once the receiver has the AMBTC stego codes C and the required keys, the embedded secret data S can be extracted, and the original AMBTC codes $\{q_i^H, q_i^L, B_i\}_{i=1}^N$ can be covered. The detailed procedures are listed as follows.

Inputs: AMBTC stego codes C , keys k_1, k_2, N_R, N_C, v , and integer m .

Outputs: AMBTC codes $\{q_i^H, q_i^L, B_i\}_{i=1}^N$ and secret data S .

- Step 1 Read the first $2 \times (N_R \times N_C - 1) \times 8$ bits from C and reconstruct all the referential quantization values R , which should be located on the first row and the first column of the quantization array. Read the next $\{B_i\}_{i=1}^N \times 16$ bits to reconstruct the bitmap $\{B_i\}_{i=1}^N$, and then read the remaining to obtain $\{C_i^H\}_{i=1}^N$ and $\{C_i^L\}_{i=1}^N$, respectively.
- Step 2 Extract two bits $s'_{i,1}s'_{i,2}$ from C_i^H and two bits $s'_{i,3}s'_{i,4}$ from C_i^L . Use the key k_2 to generate four random bits $r_{i,1}r_{i,2}r_{i,3}r_{i,4}$. The embedded four secret bits $s_{i,1}s_{i,2}s_{i,3}s_{i,4}$ can be obtained by performing the xor operation on $s'_{i,1}s'_{i,2}s'_{i,3}s'_{i,4}$ and $r_{i,1}r_{i,2}r_{i,3}r_{i,4}$.
- Step 3 Calculate prediction values p_i^H and p_i^L using the MED predictor. According to the key k_1 , the alternative prediction values \hat{p}_i^H and \hat{p}_i^L are obtained.
- Step 4 Read the next two bits $d_{i,1}d_{i,2}$ from C_i^H . Based on $d_{i,1}d_{i,2}$, read next x bits from C_i^H to obtain $(e_i^H)_2$, where x could be 0, m , or 8. Convert $(e_i^H)_2$ into its decimal form to obtain $|e_i^H|$. Similarly, read the next two bits $d_{i,3}d_{i,4}$ from C_i^L , and $|e_i^L|$ can then be obtained with similar calculation.
- Step 5 Use Eq. (9) to recover the original predictable quantization values q_i^H and q_i^L with the aids of $d_{i,1}d_{i,2}$ and $d_{i,3}d_{i,4}$.
- Step 6 Repeat Steps 2-5 until all the secret data S are extracted and all the original quantization values $\{q_i^H\}_{i=1}^N$ and $\{q_i^L\}_{i=1}^N$ are recovered.

Once all the codes $\{q_i^H\}_{i=1}^N$, $\{q_i^L\}_{i=1}^N$, and $\{B_i\}_{i=1}^N$ are recovered, the AMBTC compressed image \tilde{I} can be constructed, as described in Section 2.1.

3.5 A simple example

In this sub-section, we use a pair of predictable quantization values as an example to illustrate the proposed method. For simplicity, the processing of referential quantization values is not illustrated in this example. Let $q_i^H = 176$ and $q_i^L = 172$ be the to-be-encoded predictable quantization values, as shown in Fig. 6. Suppose $s_{i,1}s_{i,2}s_{i,3}s_{i,4} = 1000_2$, $r_{i,1}r_{i,2}r_{i,3}r_{i,4} = 0101_2$ and $m = 4$. The four xor-ed secret bits $s'_{i,1}s'_{i,2}s'_{i,3}s'_{i,4}$ can be calculated by $s_{i,1}s_{i,2}s_{i,3}s_{i,4} \oplus r_{i,1}r_{i,2}r_{i,3}r_{i,4} = 1101_2$. According to Eq. (5), the prediction values $p_i^H = 176$ and $p_i^L = 177$ are calculated. Suppose the key k_1 selects the AP technique to be applied on q_i^L , and thus we use Eq. (7) to obtain the final prediction values $\hat{p}_i^H = p_i^H = 176$ and $\hat{p}_i^L = \min(q_i^H, p_i^L) = \min(176, 177) = 176$. The prediction errors $e_i^H = q_i^H - \hat{p}_i^H = 176 - 176 = 0$ and $e_i^L = q_i^L - \hat{p}_i^L = 172 - 176 = -4$ are obtained. To encode q_i^H , since prediction error is $e_i^H = 0$, we have the division information $d_{i,1}d_{i,2} = 00_2$. According to Eq. (8), none of the bits are required to encode e_i^H . The codes $C_i^H = 1100_2$ for q_i^H can be obtained by concatenating $s'_{i,1}s'_{i,2}$, $d_{i,1}d_{i,2}$ and $(e_i^H)_2$. To encode q_i^L , the division information is $d_{i,3}d_{i,4} = 01_2$ since $-(2^m - 1) \leq e_i^L < 0$. Therefore, $|e_i^L|$ will be encoded by four bits $(e_i^L)_2 = 0100_2$. Finally, the code for q_i^L is $C_i^L = s'_{i,3}s'_{i,4}||d_{i,3}d_{i,4}||(e_i^L)_2 = 0101\ 0100_2$. The detailed encoding procedures are shown in Fig. 6.

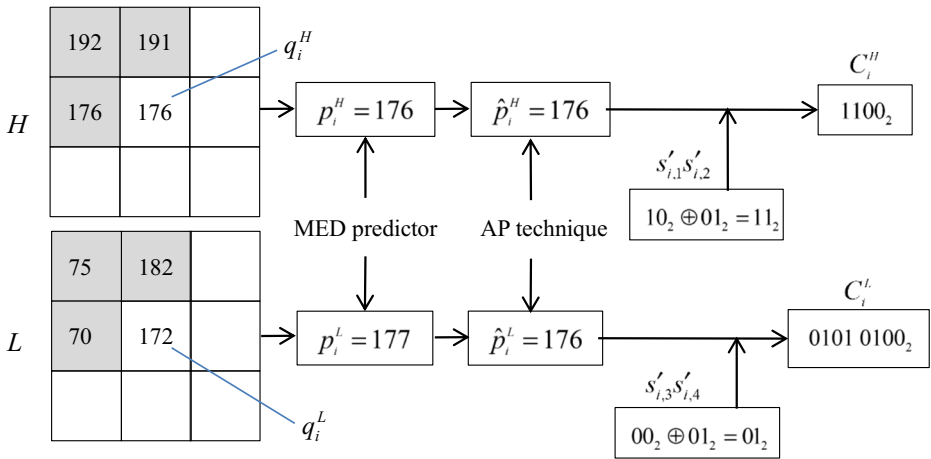


Fig. 6 The encoding example of the proposed method

To extract the embedded secret data and recover q_i^H and q_i^L , read the first two bits $s'_{i,1}s'_{i,2} = 11_2$ from C_i^H and two bits $s'_{i,3}s'_{i,4} = 01_2$ from C_i^L . The bits $s'_{i,1}s'_{i,2}s'_{i,3}s'_{i,4}$ are xor-ed with $r_{i,1}r_{i,2}r_{i,3}r_{i,4} = 0101_2$, and the secret data $s_{i,1}s_{i,2}s_{i,3}s_{i,4} = 1000_2$ are extracted. The prediction values $p_i^H = 176$ and $p_i^L = 177$ can then be calculated by using the MED predictor. The alternative prediction values $\hat{p}_i^H = 176$ and $\hat{p}_i^L = 176$ are recalculated according to Eq. (7) under the guidance of the key k_1 . By reading the next two bits $d_{i,1}d_{i,2} = 00_2$ from C_i^H , we know the prediction error is 0. Therefore, according to Eq. (9), $q_i^H = \hat{p}_i^H = 176$. Read the next two bits $d_{i,3}d_{i,4} = 01_2$ from C_i^L , we know the next four bits are prediction error $(e_i^L)_2 = 0100_2$, and its decimal value is four. Using Eq. (9), the quantization value $q_i^L = 176 - 4 = 172$ can be recovered. The detailed decoding procedures are shown in Fig. 7.

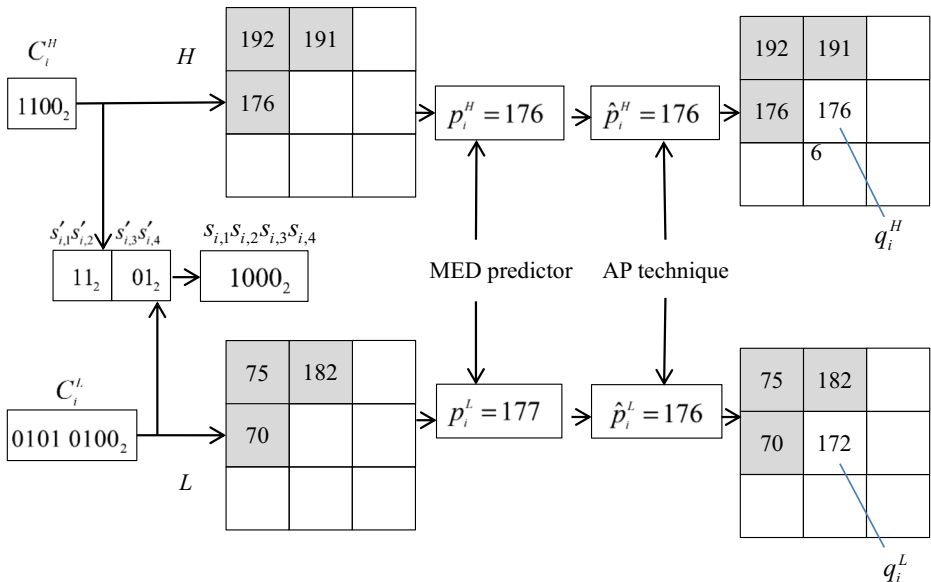


Fig. 7 The decoding example of the proposed method

4 Experimental results

In this section, we perform several experiments to demonstrate the performance of the proposed method and compare the results with other related works. Six standard 8-bit grayscale images of size 512×512 , including Lena, Jet, Peppers, Baboon, Tiffany, and House, are used as cover images, as shown in Fig. 8. All images are obtained from the SIPI image database [26]. In the experiment, a block size of 4×4 is used to generate the AMBTC codes.

In the experiment, the peak signal-to-noise ratio (PSNR) is used to measure the image quality of the AMBTC compressed image. In general, the higher the PSNR, the closer the image is to its original. We also compare the embedding performance of the proposed method, including payload, bit-rate and embedding efficiency (EF) with other related methods. The payload is used to evaluate the total bits embedded into the AMBTC codes, while the bit-rate is applied to measure the number of bits required to record a pixel (bpp) of a compressed image. A low bit-rate compressed image requires less storage space, which is an important issue regarding applications such as memory limited storage devices or streaming data on the internet. The embedding efficiency, as its name implies, measures the performance of data embedment and is defined by

$$EF = \frac{\text{Payload}}{\text{Size of AMBTC stego codes}} \quad (10)$$

An embedding method with higher embedding efficiency indicates that the method offers larger payload under the same size of stego codes.



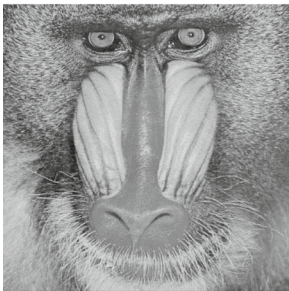
(a) Lena



(b) Jet



(c) Peppers



(d) Baboon



(e) Tiffany



(f) House

Fig. 8 Six test images used in the experiment

4.1 Performance of the proposed method

To see how the integer parameter m affects the embedding performance of the proposed method, we plot a graph of bit-rate (in bpp) versus m for the six test images in Fig. 9. Since $m = 8$ is the worst condition which cannot reflect any improvements, we only plot the range $1 \leq m \leq 7$. In the experiments, all the images are fully embedded with the same size of secret data (64,316 bits).

Figure 9 reveals that for all the images, $m = 3$ or 4 has the smallest bit-rate. This implies that proper assignment of the parameter m results in a smaller size of compressed codes. If m is too small, more prediction errors will be represented with eight bits and thus the bit-rate is increased. If m is too big, a considerable number of small prediction errors are forced to be encoded by m bits unnecessarily, and therefore the bit-rate is also increased. It should be noted that the bit-rate for a standard AMBTC compressed image of a block sized 4×4 is two bpp. However, the proposed method for smoother images such as Jet, Lena, Peppers and Tiffany offers a bit-rate which is lower than two bpp, even additional two bits are inserted into each quantization value (which is equivalent to the embedment of four bits into each AMBTC block). The results indicate that the proposed method not only embeds data into the AMBTC codes but offers a more comparable bit-rate than that of the original AMBTC codes. Besides, the bit-rate of the Baboon image is the largest due to the inaccurate prediction because the Baboon image contains the richest texture.

4.2 Bit-rate comparisons of the proposed and Sun et al.'s methods

To examine how the proposed three techniques (MED, AP, and CED) affect the bit-rate, we apply these techniques to Sun et al.'s method [25] to see how each technique influences the performance individually. When these three techniques are all applied together, we name the proposed method MAC, which is the acronym of MED, AP, and CED. The results are shown in Fig. 10a–f, respectively. In the legend, “[25] + MED” denotes Sun et al.'s prediction method

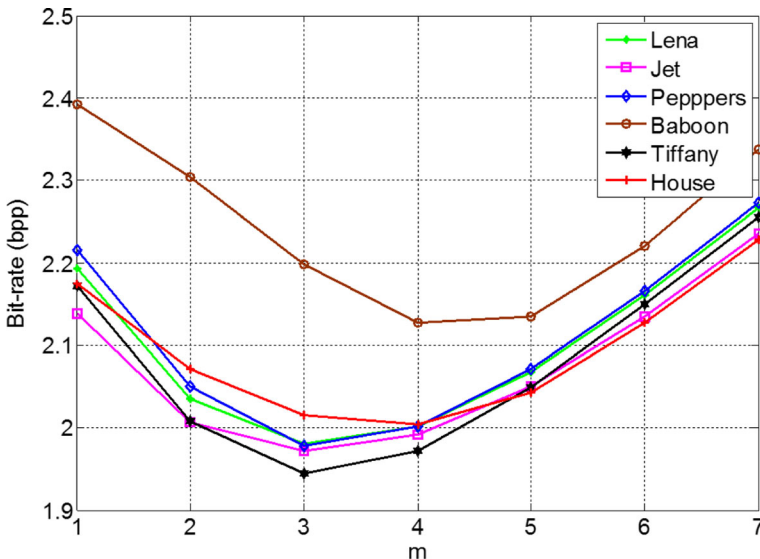
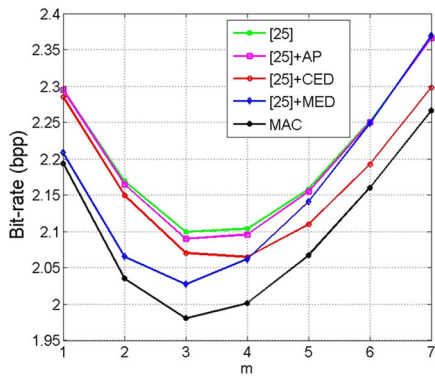
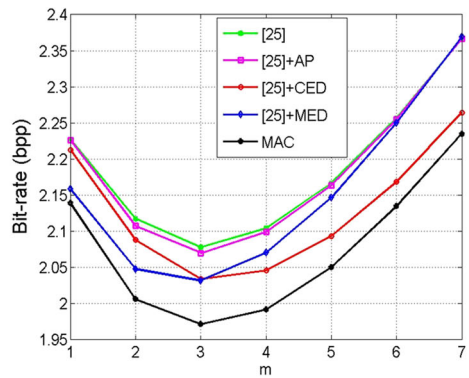


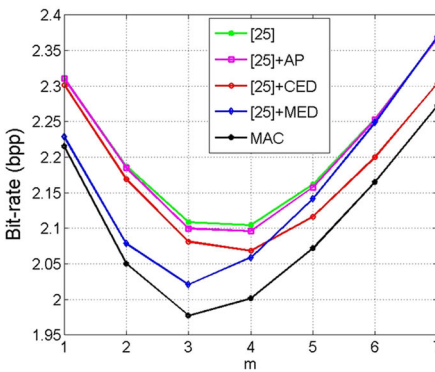
Fig. 9 The bit-rate comparison for six test images



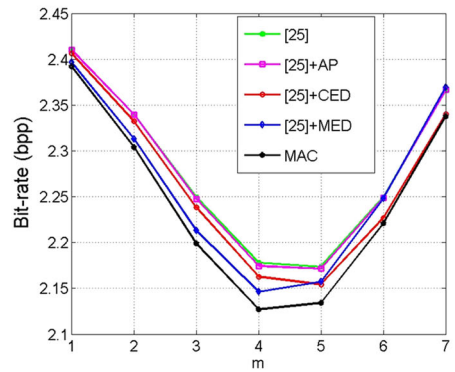
(a) Lena



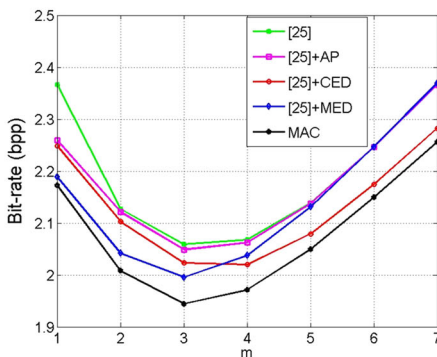
(b) Jet



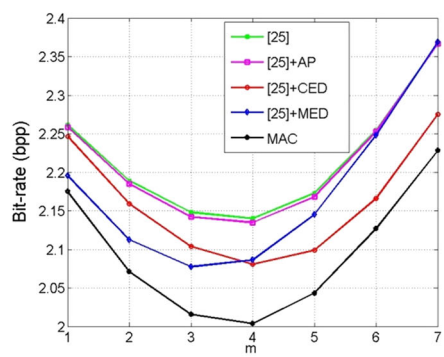
(c) Peppers



(d) Baboon



(e) Tiffany



(f) House

Fig. 10 Bit-rate comparisons using different techniques

is replaced by the proposed MED technique, while “[25] + AP” implies the AP technique is added with Sun et al.’s prediction method. “[25] + CED” represents Sun et al.’s error division technique which is replaced by the proposed CED technique.

Figure 10a–f show the contribution of each technique to the reduction of bit-rate when they are applied to Sun et al.'s method independently. We take the Lena image as an example. In Sun et al.'s method, the bit-rate (2.10 bpp) is the lowest when $m = 3$. When the AP technique is applied to Sun et al.'s method, the bit-rate is slightly reduced to 2.09 bpp. This is because some of the larger prediction errors are confined to smaller ones and thus fewer bits are required to record these errors. When the division technique used in Sun et al.'s method is replaced by the CED technique, bit-rate is further reduced to 2.07 bpp. The result implies that the proposed CED technique efficiently classifies the errors into divisions and more prediction errors are recorded using fewer bits. When the MED predictor is employed instead of Sun et al.'s prediction method, the bit-rate is reduced to 2.03 bpp. The results show that the MED predictor improves the prediction accuracy to a large extent. Finally, when all the MED, AP, and CED techniques are applied, the bit-rate is significantly reduced from 2.10 bpp to 1.98 bpp.

It should be noticed that though the above analyses focus on $m = 3$, the proposed MAC method with other m values also performs better than Sun et al.'s method. Moreover, the tests on other images also show similar results. The results indicate that the proposed MAC method indeed successfully reduces the bit-rate while providing a slightly higher payload than that of Sun et al.'s method, as will be discussed in the next sub-section.

4.3 Performance comparison with related works

In this section, we compare the proposed method with two related type II works, including Zhang et al.'s [33] and Sun et al.'s methods [25]. We also intentionally compare a type I method - Chen et al.'s method [3] for reference. In Chen et al.'s method, the embedment of additional secret bits in bitmap is implemented to achieve the highest embedding capacity, as described in [3]. In Zhang et al.'s method, four neighboring quantization values are employed as an embedding unit to achieve the best embedding result. In Sun et al.'s and the proposed MAC methods, the parameter m is properly selected such that the bit-rate is the smallest. We compare the four benchmarks, including PSNR, payload, bit-rate, and embedding efficiency EF for each method. The results are shown in Table 1.

Table 1 shows that the PSNR between the original image and AMBTC compressed image varies from one image to another. In general, a complex image has a lower PSNR while a smoother one is higher. However, for each test image, all the RDH methods for the AMBTC codes offer the same PSNR as the AMBTC compressed image since the original AMBTC codes can be completely recovered. Chen et al.'s method has significantly lower payload and bit-rate since this method belongs to type I. Zhang et al.'s method provides lower bit-rate than other methods, but the payload of the MAC and Sun et al.'s methods is two times more than that of Zhang et al.'s method. The increased payload is attributed to the fact that the MAC and Sun et al.'s methods carry 2-bit data in each quantization value while Zhang et al.'s method carries only one bit. Apparently, in contrast to Sun et al.'s method, the MAC method provides better embedding performance because the bit-rate is successfully reduced while keeping the payload slightly higher. In addition, the MAC method has a great improvement in EF. Since Chen et al.'s method only embeds one-bit secret data into a pair of quantization values, the EF is the lowest. When comparing with other two type II methods, the MAC method also has remarkable improvements. The EF is approximately two times higher than that of Zhang et al.'s method, and is around 8 % higher on average than that of Sun et al.'s method.

Table 1 Performance comparisons of the proposed method and other related works

Performance	Methods	Lena	Jet	Peppers	Baboon	Tiffany	House
PSNR (dB)	AMBTC	33.2	31.95	33.39	26.91	30.36	30.87
	Chen et al.	33.2	31.95	33.39	26.91	30.36	30.87
	Zhang et al.	33.2	31.95	33.39	26.91	30.36	30.87
	Sun et al.	33.2	31.95	33.39	26.91	30.36	30.87
	MAC	33.2	31.95	33.39	26.90	30.35	30.87
Bit-rate (bpp)	AMBTC	2.00	2.00	2.00	2.00	2.00	2.00
	Chen et al.	2.00	2.00	2.00	2.00	2.00	2.00
	Zhang et al.	1.91	1.89	1.91	2.03	1.89	1.90
	Sun et al.	2.10	2.10	2.10	2.18	2.07	2.14
	MAC	2.00	1.99	2.00	2.13	1.97	2.00
Payload (bits)	Chen et al.	16,384	16,384	16,384	16,384	16,384	16,384
	Zhang et al.	32,768	32,768	32,768	32,768	32,768	32,768
	Sun et al.	64,008	64,008	64,008	64,008	64,008	64,008
	MAC	64,516	64,516	64,516	64,516	64,516	64,516
EF	Chen et al.	0.0417	0.0417	0.0417	0.0417	0.0417	0.0417
	Zhang et al.	0.065	0.066	0.065	0.064	0.067	0.065
	Sun et al.	0.116	0.116	0.116	0.112	0.118	0.114
	MAC	0.123	0.124	0.123	0.116	0.125	0.123

5 Conclusions

In this paper, we propose a type II RDH method for the AMBTC compressed code. In general, the stego codes of type I methods can be decoded by a standard AMBTC decoder, but the embedding capacity is low. Sun et al.'s type II method has good embedding performance; however, some issues can be further reconsidered. We propose three techniques to improve Sun et al.'s method by using better prediction techniques and more efficient error divisions. The proposed MED and AP predictors better predict the quantization values and thus the accuracy of prediction is significantly improved. The proposed CED technique divides prediction errors into divisions more efficiently. Therefore, fewer bits are required to encode the prediction errors. The experiment results show that the proposed MAC method offers a comparable embedding capacity while providing a lower bit-rate and higher efficiency than Sun et al.'s method and other related works.

References

1. Chang CC, Chen TS, Chung LZ (2002) A steganographic method based upon JPEG and quantization table modification. *Inform Sci* 141(1-2):123–128
2. Chang CC, Kieu TD, Wu WC (2009) A lossless data embedding technique by joint neighboring coding. *Pattern Recogn* 42(7):1597–1630
3. Chen J, Hong W, Chen TS, Shiu CW (2010) Steganography for BTC compressed images using no distortion technique. *Imag Sci J* 58(4):177–185

4. Chen B, Shu H, Coatrieux G, Chen SX, Coatrieux JL (2015) Color image analysis by quaternion-type moments. *J Math Imag Vision* 51(1):124–144
5. Delp EJ, Mitchell OR (1979) Image compression using block truncation coding. *IEEE Trans Commun* 27(9): 1335–1342
6. Fu Z, Sun X, Liu Q, Zhou L, Shu J (2015) Achieving efficient cloud search services: multi-keyword ranked search over encrypted cloud data supporting parallel computing. *IEICE Trans Commun* E98-B(1):190–200
7. Gu B, Sheng VS, Tay KY, Romano W, Li S (2015) Incremental support vector learning for ordinal regression. *IEEE Trans Neural Netw Learn Syst* 26(7):1403–1416
8. Gu B, Sheng VS, Wang Z, Ho D, Osman S, Li S (2015) Incremental learning for v-support vector regression. *Neural Netw* 67:140–150
9. Guo P, Wang J, Xue HG, Chang SK, Kim JU (2014) A variable threshold-value authentication architecture for wireless mesh networks. *J Internet Technol* 15(6):929–936
10. Hong W (2013) Adaptive image data hiding in edges using patched reference table and pair-wise embedding technique. *Inform Sci* 221(1):473–489
11. Hong W, Chen TS (2012) A novel data embedding method using adaptive pixel pair matching. *IEEE Trans Inform Forensics Sec* 7(1):176–184
12. Hong W, Chen TS, Shiu CW, Wu MC (2011) Lossless data embedding in BTC codes based on prediction and histogram shifting. *Appl Mech Mater* 65:182–185
13. Hong W, Chen TS, Chen J (2015) Reversible data hiding using Delaunay triangulation and selective embedment. *Inf Sci* 308(1):140–154
14. Hong W, Chen TS, Shiu CW (2009) Reversible data hiding for high quality images using modification of prediction errors. *J Syst Software* 82:1833–1842
15. Huang YH, Chang CC, Chen YH (2016) Hybrid secret hiding schemes based on absolute moment block truncation coding. *Multimed Tools Appl*. doi:10.1007/s11042-015-3208-y
16. Lema M, Mitchell OR (1984) Absolute moment block truncation coding and its application to color images. *IEEE Trans Commun* 32(10):1148–1157
17. Li J, Li X, Yang B, Sun X (2015) Segmentation-based image copy-move forgery detection scheme. *IEEE Trans Inform Forensics Sec* 10(3):507–518
18. Lin CC, Liu XL, Tai WL, Yuan SM (2015) A novel reversible data hiding scheme based on AMBTC compression technique. *Multimed Tools Appl* 74(11):3823–3842
19. Lou CH, Hu CH (2012) LSB steganographic method based on reversible histogram transformation function for resisting statistical steganalysis. *Inf Sci* 188(4):346–358
20. Malik A, Sikka G, Verma HK (2016) A high payload data hiding scheme based on modified AMBTC technique. Doi: 10.1007/s11042-016-3815-2
21. Ni ZC, Shi YQ, Ansari N, Su W (2006) Reversible data hiding. *IEEE Trans Circ Syst Video Technol* 16(3): 354–362
22. Ou D, Sun W (2015) High payload image steganography with minimum distortion based on absolute moment block truncation coding. *Multimed Tools Appl* 74(21):9117–9139
23. Shen J, Tan H, Wang J, Wang J, Lee S (2015) A novel routing protocol providing good transmission reliability in underwater sensor networks. *J Internet Technol* 16(1):171–178
24. Shiu CW, Chen YC, Hong W (2015) Encrypted image-based reversible data hiding with public key cryptography from difference expansion. *Signal Process Image Commun* 39:226–233
25. Sun W, Lu ZM, Wen YC, Yu FX, Shen RJ (2013) High performance reversible data hiding for block truncation coding compressed images. *SIVIP* 7(2):297–306
26. The USC-SIPI Image Database. <http://sipi.usc.edu/database>
27. Tian J (2003) Reversible data embedding using a difference expansion. *IEEE Trans Circ Syst Video Technol* 13(8):890–896
28. Wen X, Shao L, Xue Y, Fang W (2015) A rapid learning algorithm for vehicle classification. *Inf Sci* 295(1): 395–406
29. Xia Z, Wang X, Sun X, Liu Q, Xiong N (2014) Steganalysis of LSB matching using differences between nonadjacent pixels. *Multimed Tools Appl*. doi:10.1007/s11042-014-2381-8
30. Xia Z, Wang X, Sun X, Wang B (2013) Steganalysis of least significant bit matching using multi-order differences. *Security Commun Netw* 7(8):1283–1291
31. Xia Z, Wang X, Sun X, Wang Q (2016) A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data. *IEEE Trans Parallel Distrib Syst* 27(2):340–352

32. Yuan C, Sun X, Rui L (2016) Fingerprint liveness detection based on multi-scale LPQ and PCA. *Chin Commun* 13(7):60–65
33. Zhang Y, Guo SZ, Lu ZM, Luo H (2013) Reversible data hiding for BTC-compressed images based on lossless coding of mean tables. *IEICE Trans Commun E96.B(2):624–631*
34. Zheng Y, Jeon B, Xu D, Wu QMJ, Zhang H (2015) Image segmentation by generalized hierarchical fuzzy C-means algorithm. *J Intell Fuzzy Syst* 28(2):961–973



Wien Hong received his M.S. and Ph.D. degree from the State University of New York at Buffalo, USA in 1994 and 1997, respectively. He is currently a professor in the Department of Electronic Communication and Software Engineering at Nanfang College of Sun Yat-Sen University. He is also a distinguished researcher at Nanjing University of Information Science and Technology. His research interests include steganography, watermarking and image compression.



Yuan-Bo Ma received her M.S. degree in IC Design from Hong Kong University of Science and Technology, Hong Kong in 2011, and B.S. degree in Electronic Engineering from Sun Yat-Sen University at Guangdong, China in 2010. She is currently a full-time teacher in the Department of Electronic Communication and Software Engineering at Nanfang College of Sun Yat-Sen University, Guangdong, China. Her research interests include information security, and digital signal processing.



Hung-Che Wu received his B.A. from Western Michigan University, USA, in 1999, M.S. from the University of Wisconsin-Stout, USA, in 2001, and Ph.D. from Lincoln University, New Zealand, in 2010. He is currently an associate professor in the Department of Business Administration at Nanfang College of Sun Yat-Sen University, China. His research interests involve hospitality security and travel database technology.



Tung-Shou Chen received his B.S. and Ph.D. degrees from National Chiao Tung University in 1986 and 1992, respectively, both in Computer Science and Information Engineering. Currently, he is a professor and head of the Department of Computer Science and Information Engineering at National Taichung University of Science and Technology, Taichung, Taiwan. His current research interests include data mining, image cryptosystems, steganography, image compression and bioinformatics.