

Article

Efficient Reversible Data Hiding Scheme for AMBTC-Compressed Images

Chia-Chen Lin ^{1,*}, Thai-Son Nguyen ², Chin-Chen Chang ^{3,4} and Wen-Chi Chang ³

¹ Department of Computer Science and Information Engineering, National of Chin-Yi University of Technology, Taichung 41170, Taiwan

² Department of Information Technology, Tra Vinh University, Tra Vinh 940000, Vietnam; thaison@tvu.edu.vn

³ Department of Information Engineering and Computer Science, Feng Chia University, Taichung 40724, Taiwan; ccc@cs.ccu.edu.tw (C.-C.C.); M0692719@o365.fcu.edu.tw (W.-C.C.)

⁴ Department of Computer Science and Information Engineering, Asia University, Taichung 41354, Taiwan

* Correspondence: ally.cclin@ncut.edu.tw

Abstract: Reversible data hiding has attracted significant attention from researchers because it can extract an embedded secret message correctly and recover a cover image without distortion. In this paper, a novel, efficient reversible data hiding scheme is proposed for absolute moment block truncation code (AMBTC) compressed images. The proposed scheme is based on the high correlation of neighboring values in two mean tables of AMBTC-compressed images to further losslessly encode these values and create free space for containing a secret message. Experimental results demonstrated that the proposed scheme obtained a high embedding capacity and guaranteed the same PSNRs as the traditional AMBTC algorithm. In addition, the proposed scheme achieved a higher embedding capacity and higher efficiency rate than those of some previous schemes while maintaining an acceptable bit rate.

Keywords: data hiding; high efficiency; AMBTC; privacy protection; reversibility

check for
updates

Citation: Lin, C.-C.; Nguyen, T.-S.; Chang, C.-C.; Chang, W.-C. Efficient Reversible Data Hiding Scheme for AMBTC-Compressed Images. *Appl. Sci.* **2021**, *11*, 6741. <https://doi.org/10.3390/app11156741>

Academic Editor: Byung-Gyu Kim

Received: 26 May 2021

Accepted: 9 July 2021

Published: 22 July 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Data hiding is an important technique by which a secret message is protected by being embedded into a cover image. Data hiding can be classified into two types of schemes, i.e., irreversible and reversible. One essential property of irreversible data hiding schemes is high embedding capacity [1,2]. However, they distort the cover image significantly. Moreover, this visual quality cannot be reconstructed after secret message extraction. Therefore, this type cannot be used in some special applications, i.e., military and medicine, in which a cover image is required to be restored to its original version. To solve this problem, reversible data hiding schemes [3–7] are a promising solution for these applications because the schemes can recover the original version of cover images without any distortion after the secret message was extracted.

Recently, many data hiding schemes for compressed images have been proposed to save the bandwidth space of the embedding image. This is because the size of the compressed image is much smaller than that of the original image, regardless of whether the secret message has been embedded. Some compression techniques, i.e., JPEG [8,9], vector quantization (VQ) [10–12], and block truncation coding (BTC) [13–20], are applied for data hiding to achieve high embedding capacity while maintaining a low bit rate.

In the literature, various data hiding schemes [13–20] have been proposed for BTC-compressed images. In 2004, Lin and Chang [13] introduced a data hiding scheme based on LSB replacement for BTC-compressed images. This scheme obtained a small distortion of a stego image. To improve the visual quality of the stego image, in 2006, Chuang and Chang [14] used smooth regions of the bitmap table in a BTC-compressed image to contain a secret message. However, this scheme is irreversible, meaning that the original BTC-compressed image cannot be recovered after the secret message was extracted. In 2008,

Chang et al. [15] proposed a reversible data hiding scheme based on the order of mean values to embed a secret message into BTC-compressed images. Moreover, in 2008, Hong et al. [16] proposed a new reversible data hiding scheme for BTC-compressed images that toggled or preserved the BTC bitmap table to hide the secret bits. In addition, in Hong et al.'s scheme, the relationship between the high mean value and low mean value is used for embedding data. However, when two mean values are the same, the extracted bits will contain errors. To overcome the weakness in Hong et al.'s scheme, Chen et al. [17] proposed an improved reversible data hiding scheme for BTC-compressed images. In this scheme, the difference value of the high mean value and low mean value is calculated and used to determine whether only one secret bit is embedded into the block or more secret bits are embedded into the bitmap table of the block by toggling. In 2011, Li et al. [18] proposed a reversible data hiding scheme for BTC-compressed images by using bitmap flipping and histogram shifting for high mean values and low mean values. However, Li et al.'s scheme obtained low embedding capacity when, on average, only one secret bit is embedded into each image block. To further improve embedding capacity, in 2013, Sun et al. [19] proposed a new reversible data hiding scheme based on the joint neighbor coding algorithm (JNC) to embed a secret message into BTC-compressed images. This scheme can obtain a high embedding capacity, which is four times larger than that of Li et al.'s scheme [18]. However, the efficiency rate of this scheme is still low. Later, in 2017, Hong et al. adopted a median edge detection (MED) predictor to predict the quantization values of blocks first. Next, they further designed a centralization error division (CED) technique to modify the classification rule and ensure all zero prediction errors can be categorized into a new division and no more bits are required. With such an arrangement, Hong et al. successfully shrank the size of the original quantization values of blocks and saved more space to embed secret bits [20].

To improve the embedding capacity and efficiency rate further while maintaining the reconstruction of a cover image after extracting a secret message, in this paper, we propose a new, reversible data hiding scheme for an absolute moment block truncation coding (AMBTC) image, which is a variant of the BTC-compressed image. In the proposed scheme, the high correlation of values in two mean tables of an AMBTC-compressed image is explored to embed a secret message. Experimental results demonstrated that the proposed scheme obtained a high embedding capacity and the same PSNRs as the traditional AMBTC algorithm. In addition, the proposed scheme achieved a higher embedding capacity and higher efficiency rate than those of some previous schemes while maintaining an acceptable bit rate.

The remainder of the paper is organized as follows. Section 2 briefly reviews the original AMBTC compression algorithm and previous AMBTC-based data hiding scheme [4]. Section 3 describes details concerning the proposed scheme. Then, experimental results are analyzed and compared to previous schemes in Section 4. Our conclusions and discussion are provided in Section 5.

2. Related Work

AMBTC is a technique that is the variant of the traditional BTC algorithm. However, the AMBTC technique is computationally simpler than the BTC algorithm, while the performance of the AMBTC technique is the same as that of BTC. Therefore, in this paper, the AMBTC technique is used instead of BTC. To provide the background knowledge of our proposed scheme, in the following subsections, the AMBTC technique will be presented in Section 2.1. In addition, Sun et al.'s scheme [19] is introduced in detail in Section 2.2. The main reason is that this scheme obtained high performance when embedding data into the BTC-compressed image.

2.1. Absolute Moment Block Truncation Coding

To compress both grayscale and color images, in 1984, absolute moment block truncation coding (AMBTC) was proposed by Lema and Mitchell [21]. An image is first divided

into $n \times n$ blocks. For example, to compress the image block X , the AMBTC algorithm computes mean value M to classify all pixels of block X into two different groups. If the value of pixels is smaller than M , these pixels belong to the first group. Otherwise, these pixels are considered members of the second group. Concurrently, the corresponding bit value, 0 or 1, is recorded in the bitmap, B , if the pixel belongs to the first group or second group, respectively.

After obtaining the two groups, two mean values, H_k and L_k , of the block X are calculated by Equations (1) and (2), respectively:

$$L_k = \frac{1}{n \times n - Z} \times \sum_{X_i < M} X_i \tag{1}$$

$$H_k = \frac{1}{Z} \times \sum_{X_i \geq M} X_i \tag{2}$$

where Z denotes the number of pixels with the value larger than or equal to the value of M . If $Z = n \times n$, set $H_k = L_k = M$.

After encoding by AMBTC, each block is represented by a compressed trio (H_k, L_k, B) , where H_k and L_k are the two mean values and B is the bitmap. Figure 1 shows an example of the AMBTC algorithm.

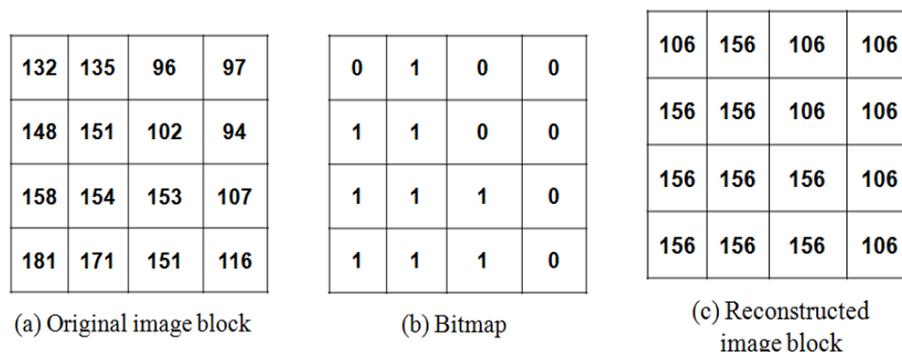


Figure 1. Example of the AMBTC algorithm.

Assume that the original image block size of 4×4 pixels is shown in Figure 1a. The mean value M of the pixel values at each block X is calculated as $M = 134$. Then, each pixel is compared to the mean value M to generate the bitmap table B of this block, as presented in Figure 1b. According to Equations (1) and (2), two mean values, H_k and L_k , are calculated as $H_k = 156$ and $L_k = 106$, respectively. The compressed trio of the block is determined as $(156, 106, 0100110011101110)$, which is sent to a receiver. After obtaining the compressed trio, the receiver can decode it to reconstruct the image block shown in Figure 1c. In general, if an image is sized as 512×512 pixels, then its code stream after encoding is sized as $(4 + 4 + 16) \times \left(\frac{512 \times 512}{4 \times 4}\right) = 524,288$ bits, which is one-quarter the size of the original image.

2.2. Sun et al.'s Scheme

In 2013, Sun et al. [19] proposed a reversible data hiding scheme for BTC-compressed images based on the joint neighbor coding technique, which was inspired by Wang et al. [22]. In this scheme, the $P \times Q$ image is first encoded by the BTC algorithm with an $n \times n$ image block to generate a high mean table, $H = H_1, H_2, \dots, H_{(P \times Q)/(n \times n)}$, low mean table, $L = L_1, L_2, \dots, L_{(P \times Q)/(n \times n)}$, and bitmap table, $B = B_1, B_2, \dots, B_{(P \times Q)/(n \times n)}$. Then, the same embedding process is used to embed a secret message into the two mean table tables, H and L . Therefore, the high mean table H is used to demonstrate this embedding process as follows. Let M be a binary pseudo-random sequence, and the length of M is similar to that of the secret message. Figure 2 shows there are four neighboring high mean values, and

they are selected to encode the current high mean value H_{cur} . The set of four neighboring high mean values is denoted as H_{ref} , which is marked in the light blue area, and their corresponding indicators are (00), (01), (10), and (11), respectively.

(01)	(10)	(11)
(00)	H_{cur}	

Figure 2. The current value H_{cur} and its four reference values.

The embedding process of Sun et al.’s scheme is described in detail as follows.

Step 1: Read two bits b_M from M and two bits b_S from the secret message S .

Step 2: Determine the reference indicator $b_{ref} = b_M \text{ XOR } b_S$.

Step 3: According to the value of b_{ref} , select the corresponding H_{ref} and calculate the difference value $D = H_{cur} - H_{ref}$.

Step 4: Classify D into four cases and encode D to D' using Equation (3)

$$D' = \begin{cases} 11 || (D^m)_2 & 0 \leq D \leq 2^m - 1 \\ 10 || (-D^m)_2 & -(2^m - 1) \leq D \leq 2^m - 1 \\ 01 || (D^8)_2 & D > 2^m - 1 \\ 00 || (D^8)_2 & D \leq -(2^m - 1) \end{cases} \quad (3)$$

where m is a predefined parameter and ranges in $[1,8]$, $||$ is the concatenation function, and $(D^m)_2$ and $(D^8)_2$ represent the m bits and eight bits of binary information of the difference value D .

Step 5: Concatenate the reference indicator b_{ref} and the value D' to generate the final codestream of the current high mean value as $b_{ref} || D'$.

Step 6: Repeat Steps 1 to 5 until the entire high mean values are encoded completely.

After the above six steps are finished, the codestream CS_H of the high mean table is obtained. Obviously, Sun et al. used the joint neighbor coding technique to embed a secret message into the two mean tables, H and L . This scheme could guarantee that the successful restoration of the original BTC-compressed image is achieved after the secret message was extracted. However, their scheme offered a low embedding capacity.

3. Proposed Scheme

In this section, we propose a novel reversible data hiding scheme for AMBTC-compressed images to further improve the performance of previous schemes. The main idea of the proposed scheme is that the AMBTC-compressed data that is generated by the AMBTC algorithm is further encoded losslessly based on the high correlation of neighboring mean values in the two mean tables, H and L . Once the hidden secret data have been extracted, the original AMBTC-based image, as shown in Figure 1c, will be completely restored. To achieve our idea, first, the AMBTC algorithm is performed on the input image to obtain the high mean table H , low mean table L , and bitmap table B . Then, the embedding process is used to hide a secret message S into the two mean tables while the bitmap table stays unchanged. The difference values of these two tables are calculated and used to embed the secret message. The proposed scheme has two phases, i.e., the embedding phase and the extracting phase, which are described in Sections 3.1 and 3.2, respectively.

3.1. Embedding Phase

Assume that a $P \times Q$ image I is encoded as an $M \times N$ high mean table H , an $M \times N$ low mean table L , and a $P \times Q$ bitmap table B , where $M = P/4$ and $N = Q/4$. After obtaining

the three tables, the secret message S can be embedded by encoding the difference values of the two mean tables, H and L , respectively. The final code stream CS generated by the proposed scheme will include four parts, which are guided information (IN), code stream CS_H of the high mean table H , code stream CS_L of the low mean table L , and code stream CS_B of the bit map table B , as shown in Figure 3. As for guide information (IN), it indicates the basic data of the code stream, including the lengths of the secret message S , codestream CS_H , codestream CS_L , size of cover image size, and value of m . IN is used to tell the data receiver how they can determine the information of data during the extracting process. Figure 4 shows the flow chart of the proposed embedding phase.

Guide information IN	Code stream CS_H of the high mean table H	Code stream CS_L of the low mean table L	Code stream CS_B of the bit map table B
------------------------	---	--	---

Figure 3. The structure of the code stream CS .

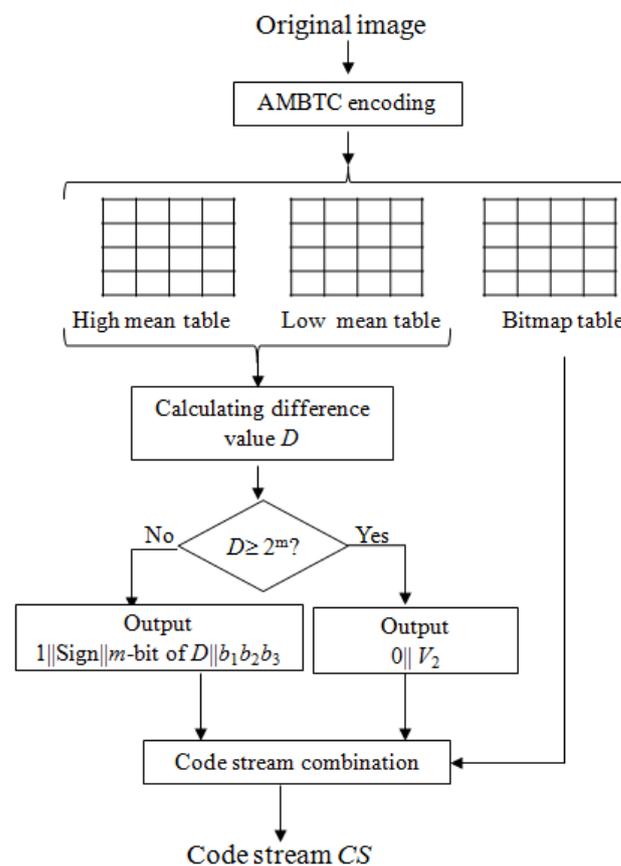


Figure 4. Flow chart of the proposed embedding phase.

The detailed embedding algorithm is shown as follows.

Step 1: Calculating difference value D .

Firstly, an inverse “S” scan path, shown in Figure 5, is used to scan the high mean table H and low mean table L into sequences of mean values $SH (SH_1, SH_2, \dots, SH_{M \times N})$ and $SL (SL_1, SL_2, \dots, SL_{M \times N})$, respectively. Then, calculate two different value sequences, DH and DL , i.e., $DH_{i+1} = SH_{i+1} - SH_i$ and $DL_{i+1} = SL_{i+1} - SL_i$, where $1 \leq i \leq M \times N - 1$.

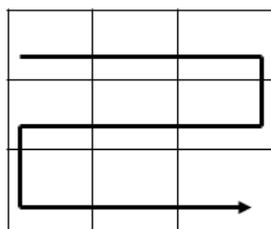


Figure 5. Example of inverse “S” scan path.

It is clear that the values of two difference sequences, DH and DL , are in range $[-255, 255]$. The embedding process of DH is similar to that of DL . Therefore, to better explain, let D_i be DH_i (or DL_i) and V_i be the corresponding mean value of SH_i (or SL_i), respectively.

Step 2: Encoding difference value D .

Due to the high correlation of adjacent values in the two mean tables, most of the difference values of D are ranged around 0. This property is used for embedding data. The value of D_i will be encoded as follows.

$$D'_i = \begin{cases} 1||Sign||m \text{ bits of } |D_i| ||b_1b_2b_3 \text{ if } |D_i| < 2^m \\ 0||(V_i)_2 \text{ if } |D_i| \geq 2^m \end{cases} \tag{4}$$

where, for $|D_i| < 2^m$, the D_i is encoded by indicator “1,” followed by 1-bit of the sign of the value of D_i , m bits of binary representation of the absolute value of D_i , and concatenated by three secret bits $b_1b_2b_3$. If $D_i < 0$, the sign of DH_i will be equal to 1; otherwise, the sign of DH_i will be equal to 0. Otherwise, if $|D_i| \geq 2^m$, it is encoded by indicator “0,” and followed by eight bits of binary representation of the original mean value V_i . Then, the compression code of D_i is concatenated into the corresponding output code stream CS_H (or CS_L).

Step 3: Code stream combination

After all difference values of DH and DL are encoded completely, the final code stream CS is obtained by concatenating IN , CS_H , CS_L , and the code stream CS_B of the bitmap table.

To clarify our embedding phase, Figure 6 shows an example of the embedding phase. Assume the AMBTC algorithm is performed, and a 3×3 high mean table is generated as shown in Figure 6a.

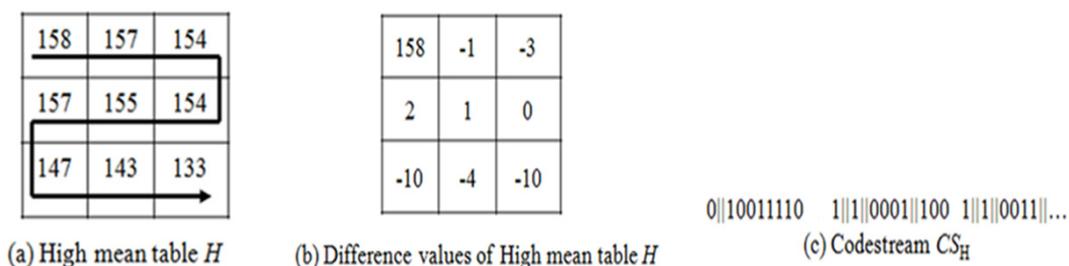


Figure 6. Example of the embedding algorithm for high mean table: (a) high mean table H ; (b) difference value of high mean table H ; (c) codestream CS_H .

Along the inverse “S” scan path, the difference values are calculated and shown in Figure 6b. Assume the secret message S is “100011011011,” and $m = 4$ is used. The first difference value D_i of DH is equal to “158,” so $|D_i| = 158 \geq 2^4$. According to Step 2, no secret message is embedded, and the 1-bit indicator “0” should be used. Then, the difference value D_i is encoded as $0||10011110$. The second difference value D_i is equal to “-1,” so $|D_i| < 2^4$. Thus, read three secret bits $b_1b_2b_3 = 100$ from the secret message S . Finally, D_i is encoded by $D'_i = 1||1||1||0001||100$, and D'_i is sent to code stream CS_H . Using the above-described embedding algorithm, all difference values of the high mean table H will be encoded to generate the code stream CS_H completely.

3.2. Extracting Phase

After obtaining the code stream CS , the receiver can extract the embedded secret message S exactly and recover the AMBTC-compressed image precisely. The detail of the extracting phase is shown as follows.

Step 1: Extract the guide information IN from the code stream CS . According to IN , the information of the embedded message S , CS_H , CS_L , CS_B , the size of the cover image, and the value of m are determined.

Step 2: Read one bit indicator ind from the codestream CS_H (or CS_L).

Step 3: If $ind = 0$, this means that no secret message is extracted. Then, eight consecutive bits are read and converted to decimal value as the reconstructed mean value SH_i (or SL_i).

Step 4: If $ind = 1$, read one next bit d as the sign of the reconstructed difference value D . Then, read m consecutive bits and convert them to decimal values as the reconstructed difference value D . Three next bits $b_1b_2b_3$ are read as three extracted secret bits and concatenated into the secret message S . Then, the value of d is considered to reconstruct the mean value as follows.

If $d = 1$, the reconstructed mean value SH_i (or SL_i) is calculated using Equation (5).

$$\begin{aligned} SH_i &= SH_{i-1} - D \\ SL_i &= SL_{i-1} - D \end{aligned} \tag{5}$$

If $d = 0$, the reconstructed mean value SH_i (or SL_i) is calculated using Equation (6).

$$\begin{aligned} SH_i &= SH_{i-1} + D \\ SL_i &= SL_{i-1} + D \end{aligned} \tag{6}$$

where $2 \leq i \leq M \times N$. The reconstructed mean value SH_i (or SL_i) is sent to the corresponding high or low mean tables.

Step 5: Steps 2 to 4 are implemented repeatedly until all values of the two mean tables are reconstructed completely.

Once the extracting algorithm is finished, the secret message S is extracted exactly. Moreover, the two mean tables are recovered accurately. Obviously, the original AMBTC-compressed image is recovered without any distortion. Following the example of the embedding phase shown in Figure 6, Figure 7 shows the example of the proposed extracting phase. First, the first bit, $ind = 0$, is read from code stream CS_H . According to Step 3, the consecutive eight bits, 10011110, are read and converted to decimal value as $SH_i = 158$. The next indicator bit, $ind = 1$, is read from CS_H . According to Step 4, read one bit $d = 1$, then four subsequent bits, 0001, are read and converted to a decimal value as $D = 1$. Three next bits, 100, are read as three extracted secret bits, $b_1b_2b_3$. Since $d = 1$, according to Step 4, the reconstructed mean value SH_i is calculated as $SH_i = SH_{i-1} - D = 158 - 1 = 157$. The extracting phase is terminated when the entire bits in the code stream CS are processed completely.

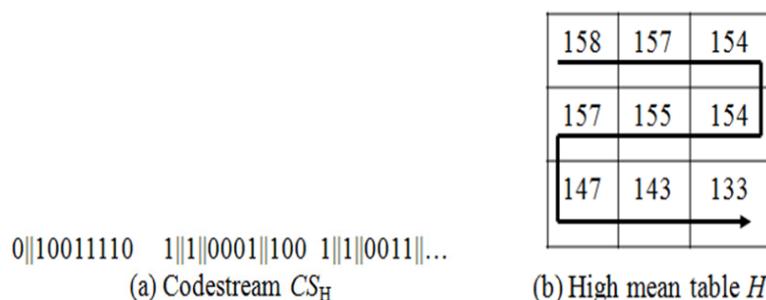


Figure 7. Example of the extracting algorithm for high mean table: (a) codestream CS_H ; (b) high mean table H .

4. Experimental Results

In this section, tests of the performance of the proposed scheme on six public 512×512 images shown in Figure 8 are described. MATLAB R2010a was used to program the proposed scheme and five previous schemes in the experiments, including AMBTC, Chuang and Chang [14], Chang et al. [15], and Hong et al. [16,20]. As for the remaining existing schemes, we cite the experimental results from their published articles. The objective measures, peak signal-to-noise ratio (PSNR), embedding capacity (EC), efficiency rate, and bit rate, were performed to estimate performance in this paper. For the $P \times Q$ grayscale image, the PSNR is calculated by Equation (7), where MSE is the mean square error and is defined as shown in Equation (8). It is noted that the proposed scheme has the reversibility feature, and the reconstructed image will be the same as the AMBTC-reconstructed image after hidden secret data have been successfully extracted. To prove our proposed scheme is reversible data hiding for AMBTC-compressed images, the definitions of the original image and reconstructed image are the same as those defined by AMBTC. The bit rate and efficiency rate are calculated by Equations (9) and (10), respectively.

$$PSNR = 10 \log_{10} \left(\frac{255^2}{MSE} \right) \quad (7)$$

$$MSE = \frac{1}{P \times Q} \sum_{i=1}^P \sum_{j=1}^Q (X_{ij} - X'_{ij})^2 \quad (8)$$

$$bit\ rate = \frac{|CS|}{P \times Q} \quad (9)$$

$$efficiency = \frac{EC}{|CS|} \quad (10)$$

where X_{ij} and X'_{ij} are the pixel values of the original image and the AMBTC-reconstructed image, respectively. EC is the embedding capacity, and $|CS|$ denotes the size of the code stream CS . EC is calculated as $EC = 3 \times N - 62$ bits, where N is the number of $|D_i| < 2^m$.

In the proposed scheme, m is the parameter that can affect the performance of the proposed scheme; therefore, in Table 1, difference values of m were used to test the results of the proposed scheme. As can be observed in Table 1, the PSNRs of the image were not changed with the difference values of m . Basically, the value of m is only ranged from 3 to 5. This is because when $m = 3$, it only takes eight bits at most to encode each difference value for either the high mean table or low mean table. In other words, the size of the final code stream of our proposed scheme will be the same as that of conventional AMBTC. Based on the experimental results, we can observe that when the value of m is increased from 3 to 5, the embedding capacity is gained while the value of the bit rate is also increased. Therefore, it is suggested that the value of m should be set to 4 in the proposed scheme to achieve both a high embedding capacity and an acceptable bit rate at the same time.

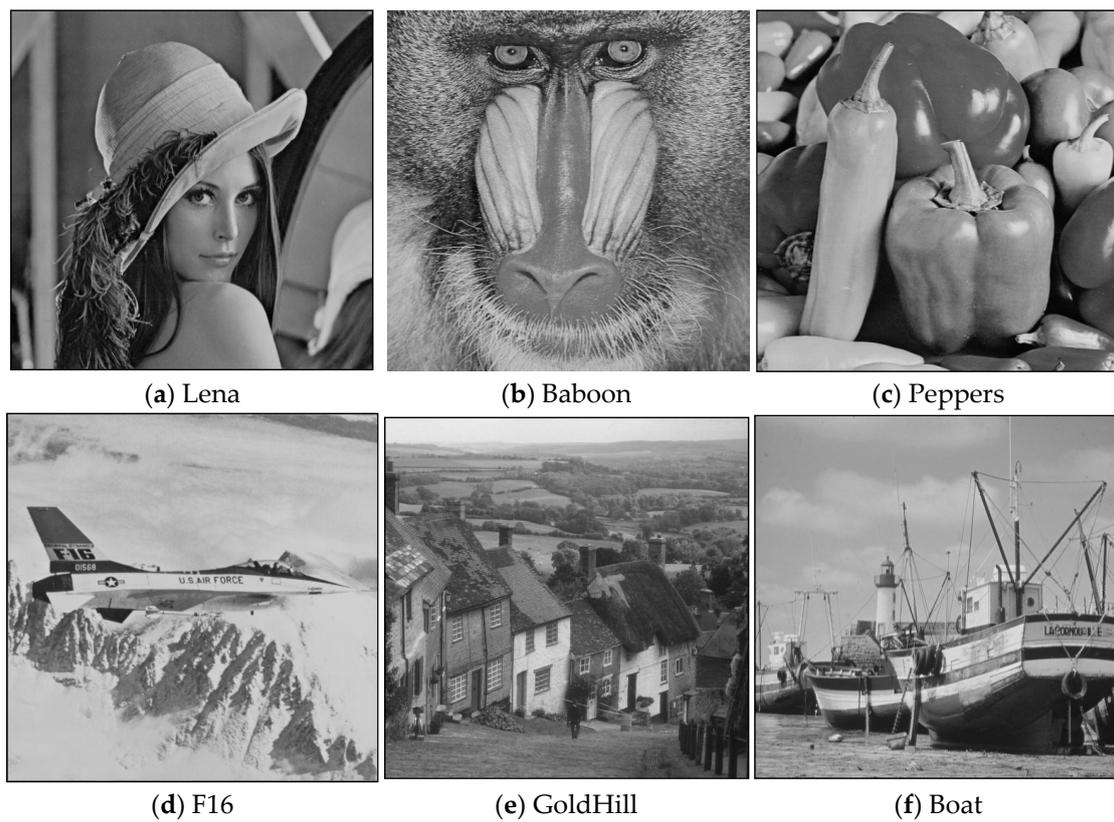


Figure 8. Six test grayscale images sized 512×512 .

Table 1. Performance of the proposed scheme under difference values of m .

Images	Performance	$m = 3$	$m = 4$	$m = 5$
Lena	PSNR (dB)	33.97	33.97	33.97
	EC (bits)	60,010	75,128	87,199
	bit rate (bpp)	2.05	2.13	2.24
Baboon	PSNR (dB)	26	26	26
	EC (bits)	40,048	64,882	86,902
	bit rate (bpp)	2.07	2.13	2.24
Peppers	PSNR (dB)	33.4	33.4	33.4
	EC (bits)	63,010	78,799	88,771
	bit rate (bpp)	2.04	2.13	2.24
F16	PSNR (dB)	31.95	31.95	31.95
	EC (bits)	71,233	81,097	88,648
	bit rate (bpp)	2.03	2.13	2.24
GoldHill	PSNR (dB)	33.16	33.16	33.16
	EC (bits)	56,710	76,603	89,569
	bit rate (bpp)	2.05	2.13	2.24
Boat	PSNR (dB)	31.15	31.15	31.15
	EC (bits)	65,620	78,178	88,321
	bit rate (bpp)	2.04	2.13	2.24

Table 2. Cont.

Schemes	Performance	Lena	Baboon	Peppers	F16	Goldhill	Boat	Average
Chang et al. [15]	PSNR (dB)	33.97	26.00	33.40	31.95	33.16	31.15	31.60
	EC (bits)	31,011	28,146	31,208	30,520	29,874	28,784	29,924
	Bit rate (bpp)	2.00	2.00	2.00	2.00	2.00	2.00	2.00
Hong et al. [20]	PSNR (dB)	33.97	26.00	33.40	31.95	33.16	31.15	31.60
	EC (bits)	64,516	64,516	64,516	64,516	64,516	64,516	64,516
	Bit rate (bpp)	2.00	2.13	2.00	1.99	1.99	2.00	2.00

As can be observed in Table 2, the embedding capacity offered by the proposed scheme, on average, was higher than that of Hong et al.'s [20] and Sun et al.'s scheme [19] by more than 10,000 bits, was nearly 2.53 times higher than Chang et al.'s scheme [15], and was four times higher than four other schemes [14,16–18], with a slightly higher bit rate of the proposed scheme. Even though a high bit rate is required, however, Figure 9 presents that the proposed scheme has a much higher efficiency rate than the seven previous schemes. Specifically, the efficiency rate of the proposed scheme, on average, outperformed Sun et al.'s scheme [19] by 2%, while it was two times larger than Chang et al.'s scheme and four times that of four other schemes [14,16–18].

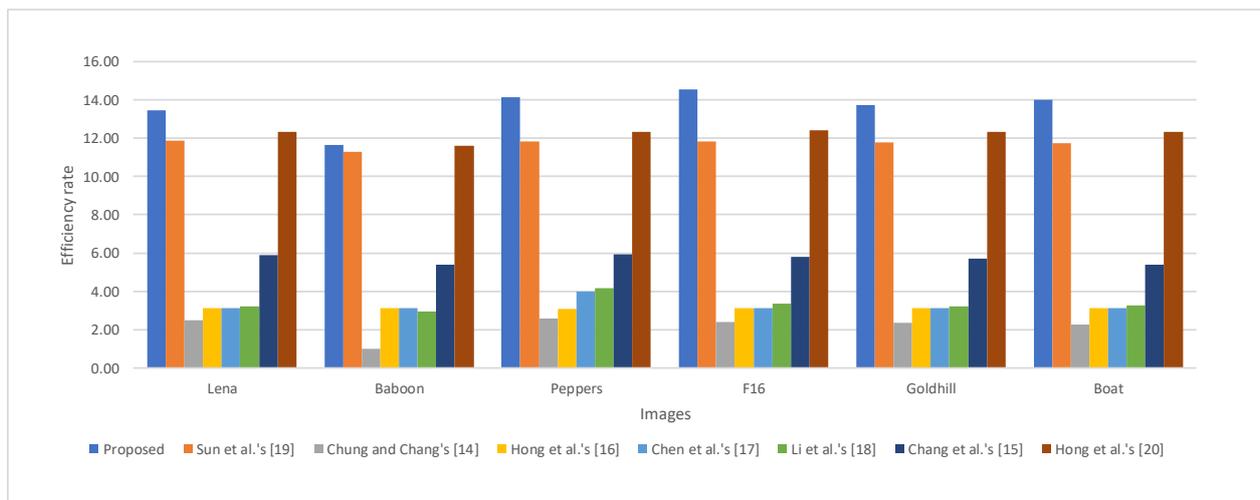


Figure 9. Comparison of efficiency rate between the proposed scheme and seven previous schemes [14–20].

To demonstrate our performance on image quality (PSNR), embedding capacity (EC), and embedding efficiency can compete with the latest existing scheme, we compare our proposed scheme with Hong et al.'s scheme [20] in Table 3. To provide fair comparisons, all data related to Hong et al.'s scheme are based on the best combinations of (α, β) , and the corresponding (α, β) for each test image, as listed in Table 3. From Table 3, we can observe the average image quality is the same as that of Hong et al.'s scheme since both two schemes are reversible data hiding schemes. However, when m is up to 5, the average hiding capacity will be up to 74,977 bits, which is 10,461 bits higher than that of Hong et al.'s scheme. In other words, the average embedding efficiency offered by our scheme is 0.085 bpp higher than that of Hong et al.'s scheme based on the definition given in Equation (10).

Table 3. Performance comparison of the proposed scheme and Hong et al.'s scheme [20] on PSNR, EC, and embedding efficiency.

Images	Performance	Proposed Scheme			Hong et al. [20]	
		$m = 3$	$m = 4$	$m = 5$	Best Combination of α & β	(α, β)
Lena	PSNR (dB)	33.97	33.97	33.97	33.97	(1,3)
	EC (bits)	60,010	75,128	87,199	64,516	
	Embedding Efficiency	0.23	0.29	0.33	0.25	
Baboon	PSNR (dB)	26	26	26	26	(3,4)
	EC (bits)	40,048	64,882	86,902	64,516	
	Embedding Efficiency	0.15	0.25	0.33	0.25	
Peppers	PSNR (dB)	33.40	33.40	33.40	33.40	(1,3)
	EC (bits)	63,010	78,799	88,771	64,516	
	Embedding Efficiency	0.24	0.3	0.34	0.25	
F16	PSNR (dB)	31.95	31.95	31.95	31.95	(1,3)
	EC (bits)	71,233	81,097	88,648	64,516	
	Embedding Efficiency	0.27	0.31	0.34	0.25	

5. Conclusions

In this paper, we introduced a new reversible data hiding scheme for AMBTC-compressed images by depending on the adjacent relation of mean values. To achieve a high embedding capacity and efficiency rate, the high correlation of values in two mean tables was explored and used to embed a secret message. The experimental results demonstrated that the proposed scheme obtained a high embedding capacity while guaranteeing the same PSNRs as the traditional AMBTC algorithm. In addition, the proposed scheme achieved a higher embedding capacity and efficiency rate than those of previous schemes while maintaining an acceptable bit rate. Obviously, the experimental results confirm that the proposed scheme can be used not only to protect secret messages but also as a practical algorithm for real-time applications.

Author Contributions: Conceptualization, C.-C.L. and T.-S.N.; methodology, C.-C.L. and T.-S.N.; software, T.-S.N. and W.-C.C.; validation, C.-C.L., and C.-C.C.; writing—original draft preparation, T.-S.N.; writing—review and editing, C.-C.L.; project administration, C.-C.C.; funding acquisition, C.-C.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Ministry of Science and Technology, Taiwan (grant number MOST 109-2410-H-167-014).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All test data can be accessed in public repository.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wang, W.-J.; Huang, C.-T.; Wang, S.-J. VQ Applications in Steganographic Data Hiding Upon Multimedia Images. *IEEE Syst. J.* **2011**, *5*, 528–537. [[CrossRef](#)]
2. Chen, S.; Leung, H. Chaotic Watermarking for Video Authentication in Surveillance Applications. *IEEE Trans. Circuits Syst. Video Technol.* **2008**, *18*, 704–709. [[CrossRef](#)]
3. Tai, W.-L.; Yeh, C.-M.; Chang, C.-C. Reversible Data Hiding Based on Histogram Modification of Pixel Differences. *IEEE Trans. Circuits Syst. Video Technol.* **2009**, *19*, 906–910. [[CrossRef](#)]
4. Nguyen, T.S.; Chang, C.C.; Lin, M.C. Adaptive lossless data-hiding and compression scheme for SMVQ indices using SOC. *Smart Comput. Rev.* **2014**, *4*, 230–245. [[CrossRef](#)]
5. Luo, L.; Chen, Z.; Chen, M.; Zeng, X.; Xiong, Z. Reversible Image Watermarking Using Interpolation Technique. *IEEE Trans. Inf. Forensics Secur.* **2010**, *5*, 187–193. [[CrossRef](#)]
6. Chang, C.-C.; Nguyen, T.S.; Lin, C.-C. Reversible data embedding for indices based on histogram analysis. *J. Vis. Commun. Image Represent.* **2014**, *25*, 1704–1716. [[CrossRef](#)]
7. Hong, W.; Chen, T.-S. A Novel Data Embedding Method Using Adaptive Pixel Pair Matching. *IEEE Trans. Inf. Forensics Secur.* **2011**, *7*, 176–184. [[CrossRef](#)]
8. Huang, F.; Huang, J.; Shi, Y.-Q. New Channel Selection Rule for JPEG Steganography. *IEEE Trans. Inf. Forensics Secur.* **2012**, *7*, 1181–1191. [[CrossRef](#)]
9. Tseng, H.; Chang, C. High Capacity Data Hiding in JPEG-Compressed Images. *Informatica* **2004**, *15*, 127–142. [[CrossRef](#)]
10. Chang, C.C.; Nguyen, T.S.; Lin, C.C. A reversible data hiding scheme for VQ indices based on absolute difference trees. *KSII Trans. Internet Inf. Syst.* **2014**, *8*, 2572–2589.
11. Lee, J.-D.; Chiou, Y.-H.; Guo, J.-M. Lossless data hiding for VQ indices based on neighboring correlation. *Inf. Sci.* **2013**, *221*, 419–438. [[CrossRef](#)]
12. Qin, C.; Chang, C.-C.; Chiu, Y.-P. A Novel Joint Data-Hiding and Compression Scheme Based on SMVQ and Image Inpainting. *IEEE Trans. Image Process.* **2014**, *23*, 969–978. [[CrossRef](#)] [[PubMed](#)]
13. Lin, M.-H.; Chang, C.-C. A novel information hiding scheme based on BTC. In Proceedings of the Fourth International Conference on Computer and Information Technology, Wuhan, China, 16 September 2004; pp. 66–71.
14. Chuang, J.C.; Chang, C.C. Using a simple and fast image compression algorithm to hide secret information. *Int. J. Comput. Appl.* **2006**, *28*, 329–333.
15. Chang, C.-C.; Lin, C.-Y.; Fan, Y.-H. Lossless data hiding for color images based on block truncation coding. *Pattern Recognit.* **2008**, *41*, 2347–2357. [[CrossRef](#)]
16. Hong, W.; Chen, T.-S.; Shiu, C.-W. Lossless Steganography for AMBTC-Compressed Images. In Proceedings of the 2008 Congress on Image and Signal Processing, Sanya, China, 27–30 May 2008; pp. 13–17. [[CrossRef](#)]
17. Chen, J.; Hong, W.; Chen, T.-S.; Shiu, C.-W. Steganography for BTC compressed images using no distortion technique. *Imaging Sci. J.* **2010**, *58*, 177–185. [[CrossRef](#)]
18. Li, C.-H.; Lu, Z.-M.; Su, Y.-X. Reversible Data Hiding for Btc-compressed Images Based on Bitplane Flipping and Histogram Shifting of Mean Tables. *Inf. Technol. J.* **2011**, *10*, 1421–1426. [[CrossRef](#)]
19. Sun, W.; Lu, Z.-M.; Wen, Y.-C.; Yu, F.-X.; Shen, R.-J. High performance reversible data hiding for block truncation coding compressed images. *Signal Image Video Process.* **2013**, *7*, 297–306. [[CrossRef](#)]
20. Hong, W.; Ma, Y.-B.; Wu, H.-C.; Chen, T.-S. An efficient reversible data hiding method for AMBTC compressed images. *Multimedia Tools Appl.* **2016**, *76*, 5441–5460. [[CrossRef](#)]
21. Lema, M.; Mitchell, O. Absolute Moment Block Truncation Coding and Its Application to Color Images. *IEEE Trans. Commun.* **1984**, *32*, 1148–1157. [[CrossRef](#)]
22. Wang, J.-X.; Lu, Z.-M. A path optional lossless data hiding scheme based on VQ joint neighboring coding. *Inf. Sci.* **2009**, *179*, 3332–3348. [[CrossRef](#)]