symmetry

MDPI

*Article*

# A High-Payload Data Hiding Scheme Based on Absolute Moment Block Truncation Coding for Minimizing Hiding Impact

**Chia-Chen Lin [1,*], Bohan Zhang [2], Wei-Liang Tai [3,*], Pei-Feng Shiu [4] and Jinn-Ke Jan [4]**

[1] Department of Computer Science and Information Engineering, National Chin-Yi University of Technology, Taichung 411, Taiwan
[2] Institute of Information Management, National Yang-Ming Chiao Tung University, Hsinchu 300, Taiwan; zhangbohan.mg10@nycu.edu.tw
[3] Bachelor Degree Program of Artificial Intelligence, National Taichung University of Science and Technology, Taichung 404, Taiwan
[4] Department of Computer Science and Engineering, National Chung Hsing University, Taichung 402, Taiwan; d100056005@cs.nchu.edu.tw (P.-F.S.); jkjan@cs.nchu.edu.tw (J.-K.J.)
* Correspondence: ally.cclin@ncut.edu.tw (C.-C.L.); twl@nutc.edu.tw (W.-L.T.)

**Abstract:** Data hiding encompasses a wide range of applications related to hiding messages in digital images. The visual redundancy of images makes it possible to embed data in the images without attracting attention. Increasing the hiding capacity and decreasing the hiding distortion are prime objectives that data hiding intends to achieve. In this paper, we propose a high-payload data hiding scheme based on absolute moment block truncation coding (AMBTC) to minimize the impact of hiding. A two-level minimum mean square error (MMSE) quantizer generated by AMBTC is used to decrease the distortion associated with hiding. Also, we present a lookup table based on the symmetric property for adaptively hiding secrets in pixels to achieve high hiding capacity. We can embed almost 1.9 bits per pixel (bpp) with a high image quality of an average of 31 dB. Only 5.3% of pixels are changed during the data-hiding process. Compared with other schemes, we can use 1 bpp more relative payload for embedding with the same stego image quality. The experimental results show that the proposed scheme has better hiding performance because it allows a huge amount of secret data to be hidden while maintaining the high visual quality of the stego image.

**Keywords:** data hiding; AMBTC; high-payload; digital images; minimizing hiding impact



**Citation:** Lin, C.-C.; Zhang, B.; Tai, W.-L.; Shiu, P.-F.; Jan, J.-K. A High-Payload Data Hiding Scheme Based on Absolute Moment Block Truncation Coding for Minimizing Hiding Impact. *Symmetry* **2024**, *16*, 64. https://doi.org/10.3390/sym16010064

Academic Editor: Tomohiro Inagaki

Received: 12 November 2023
Revised: 26 December 2023
Accepted: 29 December 2023
Published: 4 January 2024



## 1. Introduction

Cryptography uses mathematical theory to prevent people from gaining access to secret data illegally. However, the secret data can still be threatened by malicious attackers since the meaningless and unintelligible form generated from encryption may attract their attention. To conceal the existence of secret data from the public, data hiding provides a satisfactory solution by making the very existence of the original messages imperceptible. Data hiding [1] is the science of concealed communication, which involves hiding secret data in meaningful cover objects with a slight and imperceptible distortion. It can be used in various applications, such as copyright protection (robust watermarking), secret communication (steganography), and image authentication (fragile watermarking). Various applications have different requirements. For the safety of secret communications, it is important to conceal the existence of the secret data in order to avoid attracting an attacker's attention. People cannot easily distinguish the difference between the meaningful cover object and the corresponding hiding result, i.e., the stego object. To meet the safety requirements, the hiding distortion should be minimized, and the hiding capacity should suffice for embedding the secret data. Intuitively, the goal of data hiding is to design schemes that have high hiding capacity but low distortion introduced by the hiding. Hence, a trade-off must be made between hiding capacity and hiding distortion.

In data hiding, we will assume that the cover image is an 8-bit grayscale digital image, which is the set of all possible pixel values in the range [0, 255]. The most common way used in data hiding is the least significant bit (LSB) of pixel values. The LSB of pixel value $i$ can be computed as $i$ mod 2. The LSB embedding operation is flipping the LSB of the pixel value. Low hiding distortion means that there is a low probability of detecting the existence of a secret message. Mielikainen [2] proposed a modification to the LSB matching that involves designing a binary function of two cover pixels to the desired value, which allows hiding the same payload as LSB matching but with fewer changes to the cover image. Also, van Dijk et al. [3] developed another type of $\pm 1$ steganography for improving embedding efficiency in which two-dimensional codes were proposed to embed a 5-ary message symbol in a group of two pixels by modifying, at most, one pixel in the group by one. In addition, this can be generalized to n-dimensional codes [4,5] that allow $\log_2(2n + 1)$ secret bits to be embedded in n cover pixels by modifying, at most, one pixel in the group by one. Fridrich et al. [6] introduced a wet paper coding mechanism in which the sender embeds messages into content without sharing selection rules with the recipient. In addition, wet paper codes have been combined with most steganographic schemes to improve their embedding efficiencies [7].

However, the LSB-based data-hiding schemes have a low hiding capacity. To improve the hiding capacity, Lin et al. [8] proposed a high-payload, reversible data hiding scheme that is based on the absolute moment block truncation coding (AMBTC) compression domain. They presented four disjointed sets for embeddable blocks to embed data using different combinations of the mean value and the standard deviation. Malik et al. [9] modified the AMBTC compression technique for hiding secret data by first applying the original AMBTC technique, and they identified the smooth and complex blocks using a threshold value. They converted the one-bit plane into a two-bit plane and replaced all bits of the bit plane with the secret bits to obtain better image quality and high capacity. However, this approach permanently destroys the original AMBTC code and requires overhead information. Chen et al. [10] proposed an image authentication scheme for AMBTC of a compressed image using turtle shell-based data hiding.

In 2019, Yu et al. [11] proposed a hybrid data-hiding method for AMBTC compressed images, which combines a turtle-shell reference matrix and (7, 4) Hamming code to enhance the hiding capacity for compressed codes. Lin et al. [12] provided a reversible data-hiding method that uses adaptive block truncation coding based on an edge-based quantization approach. They utilized a Canny edge detector to obtain edge-blocks and non-edge-blocks, and they applied zero-point fixed histogram shifting to embed the secret information into the compressed code. In 2020, Yu et al. [13] proposed an adaptive image steganography method combining matrix coding. They constructed a reference data set by classifying all possible 7-bit binary number combinations and adaptively embedded 3-bit data by choosing a suitable alternative from the reference data. Their approach provided better results than the other existing matrix coding-based data-hiding schemes.

To minimize the risk of hiding data, we aim to decrease the hiding distortion and increase the hiding capacity. In this paper, we propose a high-payload data-hiding procedure based on AMBTC to improve the embedding efficiency further. Our proposed scheme embeds secret data by modifying the quantization level according to the pre-defined lookup table. Moreover, previous AMBTC-based schemes may have suffered from the problem of having a high quantization level lower or equal to a low quantization level caused by hiding the data. We propose an adaptive embedding strategy to solve the above issues and achieve high hiding capacity. We demonstrated that the proposed scheme has better hiding performance in image quality and hiding capacity.

To make this paper self-contained, in Section 2, we review the absolute moment block truncation coding (AMBTC) algorithm for data hiding. High-payload data hiding based on AMBTC is described in Section 3. The experimental results and their analyses are presented in Section 4. Also, in Section 4, the performance is compared to the performances of existing

data-hiding schemes. The brief contribution is concluded in Section 5, where we also outline the future research directions.

## 2. AMBTC

The absolute moment block truncation coding (AMBTC) proposed by Lema and Mitchell [14] is a type of lossy image compression technique. It is a variation of block truncation coding (BTC), but it is simpler in practical implementation. AMBTC preserved the first absolute moment and proposed a two-level, non-parametric minimum mean square error quantizer where the threshold is fixed to the sample mean. A lower mean square error yield by AMBTC is used in our proposed data hiding to minimize the impact of secret data hiding.

In AMBTC, an image is divided into non-overlapping blocks of $n \times n$ pixels, and $x_i$ is the gray level of a pixel in the block where $1 \leq i \leq n^2$. Each block is quantized, and the corresponding resulting block has the same sample mean and the same sample first absolute central moment as each original block. For each block, the sample mean, $\overline{\eta}$, is the decision threshold of the quantizer, which is calculated as

$$\overline{\eta} = \frac{1}{n^2} \sum_{i=1}^{n^2} x_i. \tag{1}$$

In AMBTC encoding, each block is encoded as ($L$, $H$, $BM$), where ($L$, $H$) is a two-level MMSE (minimum mean square error) quantizer, and $BM$ is a bitmap to denote the thresholding result. The bitmap $BM$ is presented as:

$$BM_i = \begin{cases} 1, & \text{if } x_i \geq \overline{\eta}, \\ 0, & \text{otherwise} \end{cases}, \tag{2}$$

where $x_i$ is encoded as 1 when $x_i$ is greater than or equal to the threshold and 0 otherwise. The two-level MMSE quantizers are computed as shown below:

$$\begin{aligned} L &= \tfrac{1}{n^2-q} \sum_{x_i < \overline{\eta}} x_i, \\ H &= \tfrac{1}{q} \sum_{x_i \geq \overline{\eta}} x_i, \end{aligned} \tag{3}$$

where $q$ is the number of pixels above the threshold. Note that $L$ and $H$ are estimated conditional means given that $x_i$ is less than or greater than $\overline{\eta}$, respectively.

In AMBTC decoding, each pixel $x_i'$ of each block is decoded according to the two-level MMSE quantizer and $BM$:

$$x_i' = \begin{cases} L, & \text{if } x_i = 0, \\ H, & \text{otherwise.} \end{cases} \tag{4}$$

## 3. Our Proposed DH Method Based on HP-AMBTC

To improve the hiding capacity and decrease the hiding distortion, the two-level MMSE quantizer used by AMBTC is used in our proposed data hiding to minimize the hiding impact. In addition, we designed a lookup table for adaptively hiding secret information in images. The details of the proposed scheme are described as follows.

### *3.1. Data Embedding*

Assume that the cover image is an 8-bit grayscale digital image, which is the set of all possible pixel values in the range [0, 255]. The cover image is divided into non-overlapping blocks of $4 \times 4$ pixels, and let $\{x_1, x_2, \ldots, x_{16}\}$ be the pixels in a block read in a raster scan where $x_i \in [0, 255]$.

Step 1. For each block, we use AMBTC to encode it to generate the AMBTC compression code ($L$, $H$, $BM$).

Step 2. Reconstruct the block using Equation (4) to obtain the reconstructed pixels $\{x'_1, x'_2, \ldots, x'_{16}\}$ read in a raster scan.

Step 3. For each reconstructed block, we mark $x'_1$ and $x'_{16}$ as the non-embeddable pixels and modify $x'_{16}$ as

$$x'_{16} = \begin{cases} L \text{ if } x'_1 = H \\ H \text{ if } x'_1 = L \end{cases}. \quad (5)$$

Step 4. Embed secret data into embeddable pixels $\{x'_2, x'_3, \ldots, x'_{15}\}$ as

$$y_i = x'_i + mv,$$

where $y_i$ is the stego pixel, and $mv$ is defined as follows:

Case 1: $|H-L| = 0$

| **Secret Data** | ***H*/*L* *mv*** |
|---|---|
| 00 | −1 |
| 01 | 0 |
| 10 | +1 |
| 11 | +2 |

Case 2: $|H-L| = 1$

| **Secret Data** | ***L* *mv*** | ***H* *mv*** |
|---|---|---|
| 0 | 0 | 0 |
| 1 | −1 | +1 |

Case 3: $2 \leq |H-L| < 5$

| **Secret Data** | ***L* *mv*** | ***H* *mv*** |
|---|---|---|
| 00 | −1 | −1 |
| 01 | 0 | 0 |
| 10 | −2 | +1 |
| 11 | −3 | +2 |

Case 4: $|H-L| = 5$

| **Secret Data** | ***L* *mv*** | ***H* *mv*** |
|---|---|---|
| 1111 | −1 | +3 |
| 0000 | −2 | −2 |
| 00 | −1 | −1 |
| 01 | 0 | 0 |
| 10 | +1 | +1 |
| 11 | +2 | +2 |

Case 5: $|H-L| > 5$

| **Secret Data** | ***H/L mv*** |
|---|---|
| 1111 | +3 |
| 0000 | −2 |
| 00 | −1 |
| 01 | 0 |
| 10 | +1 |
| 11 | +2 |

Note that the lookup table, which is our embedding and extraction rule, should be previously shared between the two parties using a secure channel. Taking Figure 1 as the data hiding to conceal 34 secret bits, for example, after decoding the AMBTC encoded code (90, 150, *BM*) to obtain a reconstructed image block, we define $x'_1$ and $x'_{16}$ as the non-embeddable pixels. According to $|H-L| = 60 > 5$ and secret data, we choose the Case 5 lookup table to adaptively increase the embeddable pixels by the corresponding modified value, *mv*. Note that the stego image is also an 8-bit grayscale digital image rather than a compressed image or a decompressed image. From Figure 1, we can see that the stego image block is very similar to the cover image block. After hiding 34 bits of secret data, the hiding distortion is very low since we used a two-level MMSE quantizer to minimize the impact of hiding.

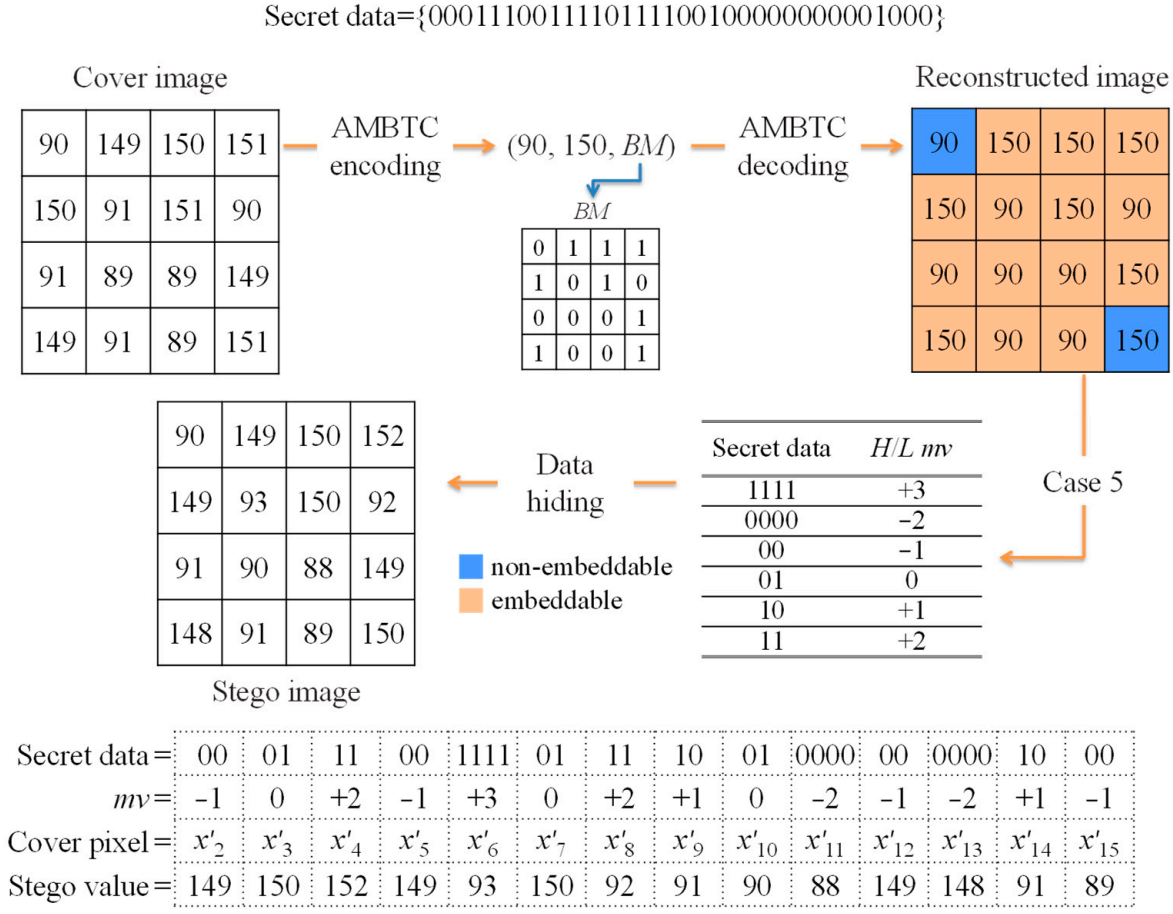



**Figure 1.** Example of data hiding.

### *3.2. Data Extraction*

This process extracts the secret data from the 8-bit grayscale stego image. Assume that the grayscale stego image is divided into non-overlapping blocks of 4 × 4 pixels,

and let $\{y_1, y_2, \ldots, y_{16}\}$ be the pixels in a stego image block and read in raster scan where $y_i \in [0, 255]$.

Step 1. For each block, we mark $y_1$ and $y_{16}$ as the non-embeddable pixels. If $y_1 \geq y_{16}$, it indicates $y_1$ as $H$ and $y_{16}$ as $L$, respectively. Otherwise, $y_1$ is $L$ and $y_{16}$ is $H$, respectively.

Step 2. For each pixel of embeddable pixels $\{y_2, y_3, \ldots, y_{15}\}$, we extract the secret data according to the absolute difference of $H$ and $L$ and the lookup table defined below:

Case 1: $|H-L| = 0$

| **Stego Value** | **Secret Data** |
|---|---|
| $(H-1)/(L-1)$ | 00 |
| $H/L$ | 01 |
| $(H+1)/(L+1)$ | 10 |
| $(H+2)/(L+2)$ | 11 |

Case 2: $|H-L| = 1$

| **Stego Value** | **Secret Data** |
|---|---|
| $H/L$ | 0 |
| $(H+1)/(L-1)$ | 1 |

Case 3: $2 \leq |H-L| < 5$

| **Stego Value** | **Secret Data** |
|---|---|
| $(H-1)/(L-1)$ | 00 |
| $H/L$ | 01 |
| $(H+1)/(L-2)$ | 10 |
| $(H+2)/(L-3)$ | 11 |

Case 4: $|H-L| = 5$

| **Stego Value** | **Secret Data** |
|---|---|
| $(H+3)/(L-3)$ | 1111 |
| $(H-2)/(L-2)$ | 0000 |
| $(H-1)/(L-1)$ | 00 |
| $H/L$ | 01 |
| $(H+1)/(L+1)$ | 10 |
| $(H+2)/(L+2)$ | 11 |

Case 5: $|H-L| > 5$

| **Stego Value** | **Secret Data** |
|---|---|
| $(H+3)/(L+3)$ | 1111 |
| $(H-2)/(L-2)$ | 0000 |
| $(H-1)/(L-1)$ | 00 |
| $H/L$ | 01 |
| $(H+1)/(L+1)$ | 10 |
| $(H+2)/(L+2)$ | 11 |

Figure 2 illustrates the data extraction example. We define $y_1$ and $y_{16}$ as the non-embeddable pixels, and we regard $y_1$ as $L$ and $y_{16}$ as $H$ due to $y_1 < y_{16}$. According to $|H-L| = 60 > 5$, we choose the Case 5 lookup table to extract the secret data. Finally, 34-bits

of secret data can be extracted after scanning 14 pixels. Moreover, the AMBTC decoded image block can be almost restored according to the corresponding *L*s and *H*s except for the first and the last pixels.

We said the AMBTC encoded image block could be almost restored because the first and the sixteenth pixels needed to be modified and then served as the indicators for data extraction later. If these two pixels in a block are the same as the original ones after the data hiding, the original AMBTC-encoded image blocks can be completely restored. Otherwise, only fourteen pixels in a block can be restored to the original AMBTC compression codes after data extraction.

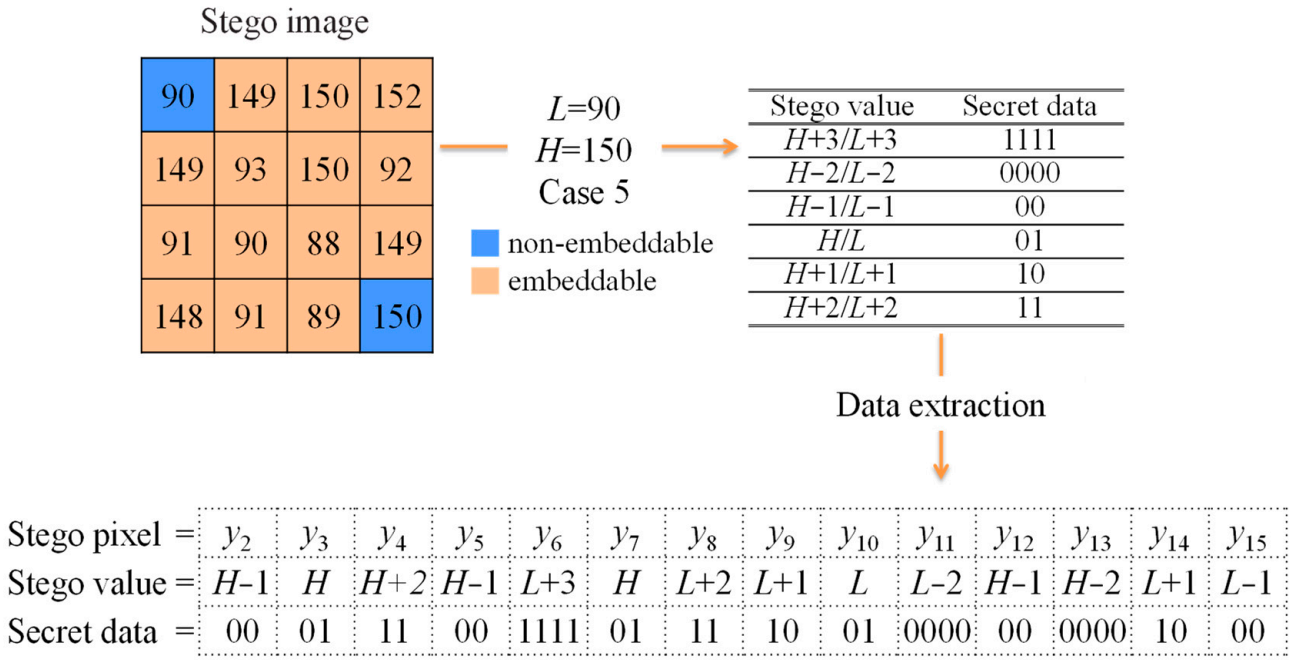



**Figure 2.** Example of data extraction.

## 4. Experimental Results

In this section, a series of analyses and simulation results are demonstrated to prove the performance of the proposed scheme. In addition, comparisons among [15–20] also are presented in this section.

### *4.1. Capacity Versus Distortion Performance*

To test our performance on data hiding and image quality, all experiments were performed with ten commonly used grayscale images sized 512 × 512, i.e., "Lena," "Airplane," "Baboon," "Barbara," "Peppers," "Boat," "F-16," "Boat," "House," "Houses," "Zelda," and "Gold Hill." Six 512 × 512 selected grayscale images are shown in Figure 3 to demonstrate partial test images.

We use the peak signal-to-noise ratio (*PSNR*), which is the most common image quality criterion, to evaluate the visual quality of the reconstructed images. The *PSNR* (unit: dB) is defined as:

$$PSNR = 10 \times \log_{10} \frac{255^2}{MSE}, \tag{6}$$

where *MSE* is the mean square error defined by:

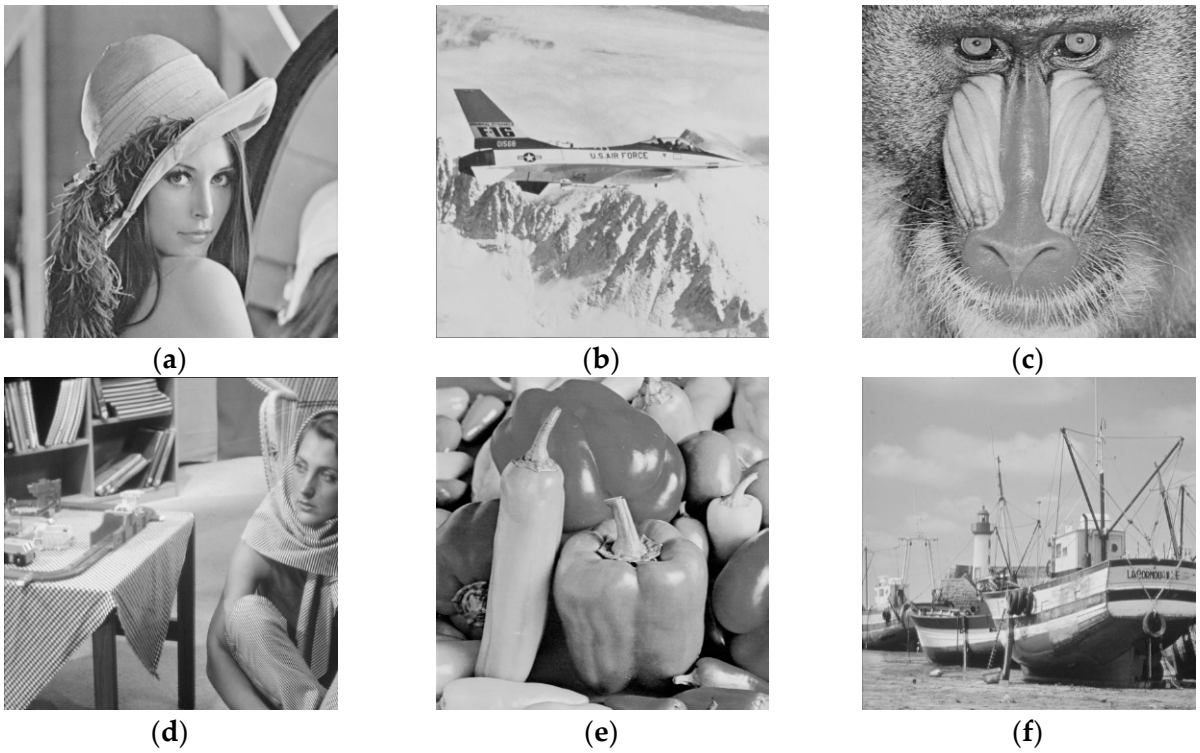
$$MSE = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \left( x_{i,j} - x'_{i,j} \right)^2, \tag{7}$$

where $x_{i,j}$ and $x'_{i,j}$ present the pixel values of the original image and the modified image, respectively, and $M \times N$ indicates the size of the image. Moreover, structural similarity

(*SSIM*) is also used to evaluate the similarity between the stego image and the original cover image. The *SSIM* is defined as:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{\left(\mu_x^2 + \mu_y^2 + c_1\right)\left(\sigma_x^2 + \sigma_y^2 + c_2\right)}, \tag{8}$$

where $\mu_x$ and $\mu_y$ are the mean values of the pixel values of image $x$ and image $y$, respectively. $\sigma_x$ and $\sigma_y$ are the standard deviations of the pixel values of image $x$ and image $y$, respectively. $c_1 = (k_1 L)^2$ and $c_2 = (k_2 L)^2$, and $k_1 = 0.01$ and $k_2 = 0.03$. In general, $SSIM(x, y)$ ranges between 1 and $-1$. When two images are identical, the corresponding *SSIM* value will be 1.




(**a**) (**b**) (**c**)

(**d**) (**e**) (**f**)

**Figure 3.** Greyscale test image. (**a**) Lena; (**b**) F-16; (**c**) Baboon; (**d**) Barbara; (**e**) Peppers; and (**f**) Boat.

Figure 4 presents the relationship between the capacity and the corresponding image distortion for the six test images. It is obvious that the achievable capacity of our proposed HP-AMBTC-based DH scheme depends on the features of the image. It indicates that the smooth images provide a higher capacity at the same embedding distortion; this made it evident that images with high correlation offer a larger hiding capacity than images with low correlation. To further demonstrate the performance of our scheme, Table 1 presents the performance of *PSNR*, maximum hiding capacity, and its corresponding bit per pixel (bpp) with our proposed HP-AMBTC-based DH scheme. Here, we can see that the average bpp can be rounded to 2 bpp while maintaining 31.96 dB for the average image quality of the stego image with our proposed HP-AMBTC-based DH scheme when "Baboon" is excluded. Even if "Baboon" is included, our average *PSNR* still remains at 31.340 dB, and previous works [21,22] have confirmed that the visual quality of AMBTC is acceptable, although the value of its *PSNR* value is less than 30 dB.

To further demonstrate the competitive performance of our proposed HP-AMBTC-based DH scheme in Table 2, we compare our scheme with six other existing BTC or AMBTC-based data hiding schemes [15–20]. It is noted that all data of the other six schemes are cited from the original experimental data presented by the original research teams. Based on the data presented in Table 2, it is obvious that the average maximum hiding

capacity of the test images delivered by the proposed scheme is 1.90 bpp, which is relatively higher than that offered by the other six data-hiding schemes. Our scheme embeds data into the high mean values or the low mean values derived from AMBTC. The maximum change to the mean value is within the range [−2, 3], which means that we can embed 1.9 bpp with an embedding efficiency of almost 2 bits per change. Moreover, there are differences between the original pixel value and the mean value. Embedding data to the mean value makes it possible to get close to the original pixel value, which greatly decreases the embedding distortions; this means that the stego pixel value is nearly the same as the original pixel value. As shown in Figure 5, our scheme achieves the highest average hiding capacity while maintaining better image quality at 32.26 dB. However, a lookup table is required for data hiding and extraction in our scheme; either the corresponding data hiding rules or data extraction rules can be easily programmed to replace the lookup table.

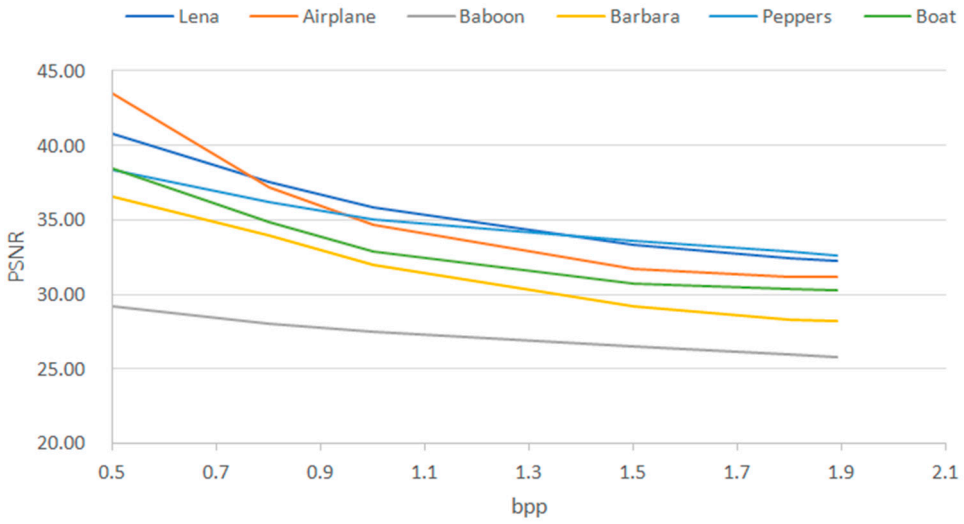



**Figure 4.** PSNR versus payload size for test images.

**Table 1.** Performance on image quality (*PSNR*), max. hiding capacity and bpp with ten 512 × 512 test images.

| Criteria / Test Images | Hiding Capacity (bits) | bpp | *PSNR* |
|---|---|---|---|
| | | | HP-AMBTC |
| Lena | 494,464 | 1.9 | 32.253 |
| Baboon | 509,764 | 1.9 | 25.755 |
| Peppers | 502,476 | 1.9 | 32.624 |
| F-16 | 476,036 | 1.8 | 31.144 |
| Boat | 505,308 | 1.9 | 30.240 |
| House | 457,444 | 1.7 | 35.914 |
| Houses | 480,850 | 1.8 | 29.964 |
| Zelda | 499,818 | 1.9 | 35.667 |
| Gold Hill | 509,688 | 1.9 | 31.660 |
| Barbara | 500,334 | 1.9 | 28.172 |
| Average | 493,618 | 1.9 | 31.334 |

**Table 2.** Performance comparison with our scheme and six existing BTC/AMBTC-based data-hiding schemes [15–20] (Unit: dB for *PSNR*, and bit for bpp).

| Methods | Test Images | Boats | Goldhill | F-16 | Lena | Peppers | Zelda | Average |
|---|---|---|---|---|---|---|---|---|
| Ou and Sun [15] | bpp | 0.79 | 0.81 | 0.9 | 0.92 | 0.93 | 0.85 | 0.87 |
| | *PSNR* | 29.57 | 29.12 | 30.75 | 30.69 | 31.43 | 29.93 | 30.25 |
| Huang et al. [16] | bpp | 0.99 | 1.02 | 1.09 | 1.12 | 1.12 | 1.05 | 1.07 |
| | *PSNR* | 29.30 | 29.41 | 30.34 | 30.44 | 31.06 | 29.79 | 30.06 |
| Hong [17] | bpp | 0.9 | 0.92 | 1.01 | 1.04 | 1.05 | 0.96 | 0.98 |
| | *PSNR* | 29.56 | 29.11 | 30.74 | 30.68 | 31.41 | 29.92 | 30.24 |
| Malik et al. [18] | bpp | 1.52 | - | 1.52 | 1.52 | 1.52 | - | 1.52 |
| | *PSNR* | 31.09 | - | 31.90 | 33.10 | 33.30 | - | 32.35 |
| Yeh et al. [19] | bpp | 1.9 | - | 1.3 | 1.67 | 1.83 | - | 1.675 |
| | *PSNR* | 31.04 | - | 32.09 | 33.06 | 33.01 | - | 32.30 |
| Kim et al. [20] | bpp | 1.02 | 1.03 | 1.11 | 1.14 | 1.13 | 1.08 | 1.09 |
| | *PSNR* | 30.39 | 29.65 | 31.86 | 29.89 | 28.57 | 31.02 | 30.24 |
| Our proposed scheme | bpp | 1.93 | 1.95 | 1.82 | 1.89 | 1.92 | 1.91 | 1.90 |
| | *PSNR* | 30.24 | 31.66 | 31.14 | 32.25 | 32.62 | 35.67 | 32.26 |

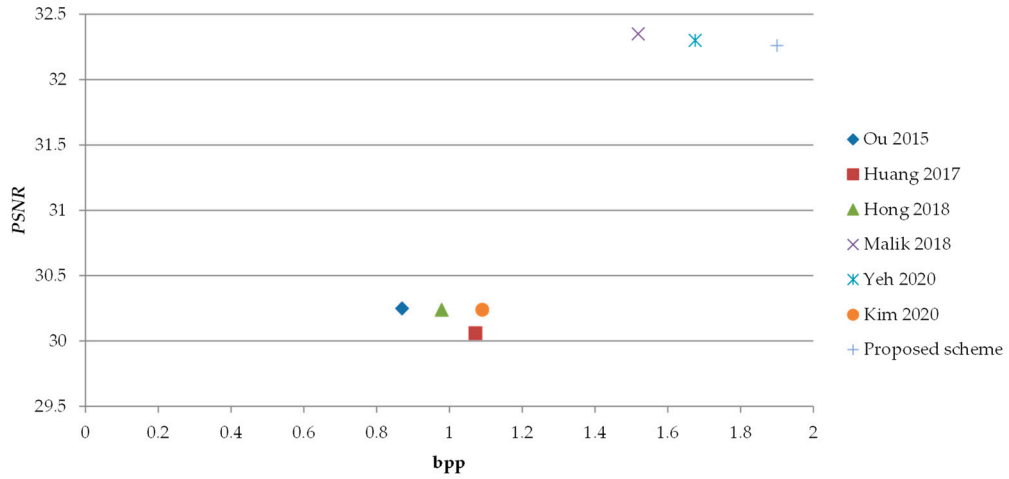



**Figure 5.** Average performance comparison of schemes [15–20].

### *4.2. Security Analysis*

Since the cover image must be modified to conceal secret data with data-hiding strategies, whether the pixel distribution of the stego image remains an unidentified pattern similar to that of the cover image is crucial when evaluating the security of a data-hiding scheme. To demonstrate the security of our proposed HP-AMBTC-based DH scheme, several metrics: *SSIM* (structural similarity), number of changing pixel rate (*NPCR*), unified averaged changed intensity (*UACI*), and peak signal-to-noise ratio (*PSNR*) are used to analyze the stego images derived from our proposed scheme.

*PSNR* has been defined in Equation (6), and *SSIM* has been defined in Equation (8). As for *NPCR* and *UACI*, the corresponding detailed definitions are given in the following equations:

$$D(i',j') = \begin{cases} 0, & if\ O(i',j') = O^*(i',j'), \\ 1, & if\ O(i',j') \neq O^*(i',j'), \end{cases} \tag{9}$$

$$NPCR = \frac{1}{M \times N} \sum_{i',j'} D(i',j') \times 100\%, \tag{10}$$

$$UACI = \frac{1}{M \times N} \sum_{i',j'} \frac{O(i',j') - O^*(i',j')}{255} \times 100\%. \quad (11)$$

The ranges of *NPCR* and *UACI* are both [0, 1], and a higher value indicates a higher difference between the original image and the stego image. At the same time, the value of *UACI* is not as obvious as that of *NPCR*.

The security analyses for our proposed HP-AMBTC-based DH scheme are demonstrated in Table 3. Table 3 shows the similarity of structure between the original image and the stego image, which remains at 0.905, which is very close to 1. In addition, *NPCR* and *UACI* indicate that the pixel difference between the original image and the stego image is quite slight. Take *NPCR*, for example; only 5.3% of pixels between the original image and stego images are different. With the results demonstrated in Table 3, it is concluded that our proposed HP-AMBTC-based DH scheme can guarantee the similarity between the original image and the stego image. In other words, the security of the hidden secret can be guaranteed.

**Table 3.** Security analyses with six different test images.

| Metrics \ Test Images | Baboon | F-16 | Barbara | Boat | House | Peppers | Average |
|---|---|---|---|---|---|---|---|
| *PSNR* | 25.755 dB | 31.144 dB | 28.172 dB | 30.240 dB | 35.914 dB | 32.624 dB | 30.642 dB |
| *SSIM* | 0.8515 | 0.9260 | 0.8978 | 0.8958 | 0.9509 | 0.9104 | 0.905 |
| *NPCR* | 0.0592 | 0.0492 | 0.0556 | 0.0556 | 0.0437 | 0.0539 | 0.053 |
| *UACI* | 0.0002 | 0.0001 | 0.0001 | 0.0002 | 0.0001 | 0.0002 | 0.0001 |

## 5. Conclusions

To mitigate the impact of concealed data while enhancing the hiding capacity for AMBTC compressed images, this paper introduces a high-payload data-hiding scheme based on AMBTC. The proposed scheme leverages a two-level MMSE quantizer generated by AMBTC and our specially designed data-hiding strategies tailored for different quantizer distortions. The collaboration between the quantizer and our strategies aims to strike a balance between induced distortion and hiding capacity. Experimental results validate that our scheme surpasses six existing BTC or AMBTC-based data-hiding schemes in terms of both hiding capacity and visual quality. We can embed almost 1.9 bits per pixel with a high image quality of an average of 31 dB. The data hiding process only modifies 5.3% pixels of the original image on average. The similarity of structure between the original image and the stego image is 0.905, which means that the stego pixel image is nearly the same as the original image. Additionally, security analyses affirm that the increased capacity does not compromise the similarity in either structure or pixel distribution. Based on these findings, future efforts will explore incorporating integrity authentication codes during data-hiding operations. This enhancement will allow the receiver(s) to verify the integrity of received stego images before extracting hidden data, thereby bolstering the usability of the extracted confidential data.

**Author Contributions:** Conceptualization, C.-C.L. and J.-K.J.; formal analysis, C.-C.L.; methodology, C.-C.L. and J.-K.J.; validation, B.Z. and P.-F.S.; writing—original draft, W.-L.T. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the National Science and Technology Council: MOST 111-2410-H-167-005-MY2.

**Data Availability Statement:** Publicly available datasets were analyzed in this study. This data can be found here: https://sipi.usc.edu/database/.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References


1. Wu, M.; Liu, B. Data hiding in image and video. I. Fundamental issues and solutions. *IEEE Trans. Image Process.* **2003**, *12*, 685–695. [PubMed]
2. Mielikainen, J. LSB matching revisited. *IEEE Signal Process. Lett.* **2006**, *13*, 285–287. [CrossRef]
3. van Dijk, M.; Willems, F. Embedding information in grayscale images. In Proceedings of the 22nd Symposium on Information and Communication Theory in the Benelux, Enschede, The Netherlands, 15–16 May 2001; pp. 147–154.
4. Zhang, X.P.; Wang, S.Z. Efficient steganographic embedding by exploiting modification direction. *IEEE Commun. Lett.* **2006**, *10*, 781–783. [CrossRef]
5. Fridrich, J.; Lisoněk, P. Grid colorings in steganography. *IEEE Trans. Inf. Theory* **2007**, *53*, 1547–1549. [CrossRef]
6. Fridrich, J.; Goljan, M.; Lisoněk, P.; Soukal, D. Writing on wet paper. *IEEE Trans. Signal Process.* **2005**, *53*, 3923–3935. [CrossRef]
7. Fridrich, J.; Goljan, M.; Soukal, D. Wet paper codes with improved embedding efficiency. *IEEE Trans. Inf. Forensics Secur.* **2006**, *1*, 102–110. [CrossRef]
8. Lin, C.C.; Liu, X.L.; Tai, W.L.; Yuan, S.M. A novel reversible data hiding scheme based on AMBTC compression technique. *Multimed. Tools Appl.* **2015**, *74*, 3823–3842. [CrossRef]
9. Malik, A.; Sikka, G.; Verma, H.K. A high payload data hiding scheme based on modified AMBTC technique. *Multimed. Tools Appl.* **2017**, *76*, 14151–14167. [CrossRef]
10. Chen, C.C.; Chang, C.C.; Lin, C.C.; Su, G.D. TSIA: A novel image authentication scheme for AMBTC-based compressed images using turtle shell based reference matrix. *IEEE Access* **2019**, *7*, 149515–149526. [CrossRef]
11. Yu, Z.; Lin, C.C.; Chang, C.C.; Su, G.D. HBF-DH: An enhanced payload hybrid data hiding method based on a hybrid strategy and block features. *IEEE Access* **2019**, *7*, 148439–148452. [CrossRef]
12. Lin, C.C.; Chang, C.C.; Wang, Z.M. Reversible data hiding scheme using adaptive block truncation coding based on an edge-based quantization approach. *Symmetry* **2019**, *11*, 765. [CrossRef]
13. Yu, Z.; Lin, C.C.; Chang, C.C. ABMC-DH: An Adaptive Bit-Plane Data Hiding Method Based on Matrix Coding. *IEEE Access* **2020**, *8*, 27634–27648. [CrossRef]
14. Lema, M.; Mitchell, O. Absolute moment block truncation coding and its application to color images. *IEEE Trans. Commun.* **1984**, *32*, 1148–1157. [CrossRef]
15. Ou, D.; Sun, W. High payload image steganography with minimum distortion based on absolute moment block truncation coding. *Multimed. Tools Appl.* **2015**, *74*, 9117–9139. [CrossRef]
16. Huang, Y.H.; Chang, C.C.; Chen, Y.H. Hybrid secret hiding schemes based on absolute moment block truncation coding. *Multimed. Tools Appl.* **2017**, *76*, 6159–6174. [CrossRef]
17. Hong, W. Efficient data hiding based on block truncation coding using pixel pair matching technique. *Symmetry* **2018**, *10*, 36. [CrossRef]
18. Malik, A.; Sikka, G.; Verma, H.K. An AMBTC compression based data hiding scheme using pixel value adjusting strategy. *Multidimens. Syst. Signal Process.* **2018**, *29*, 1801–1818. [CrossRef]
19. Yeh, J.Y.; Chen, C.C.; Liu, P.L.; Huang, Y.H. High-payload data-hiding method for AMBTC decompressed images. *Entropy* **2020**, *22*, 145. [CrossRef]
20. Kim, C.; Yang, C.N.; Leng, L. High-capacity data hiding for ABTC-EQ based compressed image. *Electronics* **2020**, *9*, 644. [CrossRef]
21. Bai, J.; Chang, C.C. A high payload steganographic scheme for compressed images with hamming code. *Int. J. Netw. Secur.* **2016**, *18*, 1122–1129.
22. Hong, W.; Chen, T.S.; Yin, Z.; Luo, B.; Ma, Y. Data hiding in AMBTC images using quantization level modification and perturbation technique. *Multimed. Tools Appl.* **2017**, *76*, 3761–3782. [CrossRef]