electronics

MDPI

Article

Dual Reversible Data Hiding Based on AMBTC Using Hamming Code and LSB Replacement

Cheonshik Kim

Department of Computer Engineering, Sejong University, Seoul 05006, Korea; mipsan@sejong.ac.kr

Abstract: The existing data hiding schemes conceal the data in the cover image and then communicate secretly on the channel. The weakness of these methods is that the security aspect is somewhat lacking, and there is a limit to hiding enough data. In this paper, we propose a reversible data hiding method based on dual AMBTC images. It improves security, which is a weakness of data hiding. AMBTC has strengths in low-bandwidth channel environments with simple calculations and efficient data performance. HC(7,4) and LSB replacement methods are applied to each block of AMBTC to hide secret data. After the embedding process, the two AMBTC-marked images are obtained, and these images are sent to different recipients. The recipients can extract hidden messages and restore the cover AMBTC image by using the proposed method and two marked images. Our proposed data hiding method guarantees sufficient data hiding, proper cover image quality, and restoration of the original cover image. Experimental results show that our method is efficient in terms of image quality and embedding ratio.

Keywords: dual reversible data hiding; block truncation coding (BTC); hamming code (HC); absolute moment BTC (AMBTC); reversible data hiding (RDH)

Citation: Kim, C. Dual Reversible Data Hiding Based on AMBTC Using Hamming Code and LSB Replacement. *Electronics* **2022**, *11*, 3210. https://doi.org/10.3390/electronics11193210

Academic Editors: Andrei Kelarev and Krzysztof Szczypiorski

Received: 22 August 2022
Accepted: 3 October 2022
Published: 6 October 2022

1. Introduction

Recently, many people have used community services such as Social Networking Services (SNS) to exchange information and share digital media with many people. Digital media may include text, images, video, and audio, and these resources are at risk of being forged or altered by the latest digital software at any time [1]. Additionally, the Internet, Internet of Things, and email providers that we use are not as reliable as we may imagine, making it possible for any or all of the content to be hijacked by a knowledgeable network attacker, which would cause problems for innocent victims.

A traditional method of solving this problem in terms of security is to apply encryption to digital media and transmit it, and a method to transmit the encryption key using a separate channel can be used. The disadvantage of encrypted messages is that they are easily targeted by attackers. One of the suggested solutions to this problem is data hiding. Data hiding (DH) [2–5] is a method for hiding a message in digital media and secretly delivering it to the receiver. The advantage of this method is that it may not attract the attention of the attacker. If a secret message is surreptitiously hidden on the cover medium, part of the cover signal will inevitably be permanently damaged. The reason is that the relationship between the secret message to be hidden and the image quality is inversely proportional. That is, as the amount of data to be hidden increases, the quality of the image decreases.

Reversible DH (RDH) [3] is a solution that can avoid permanent damage to the cover image. That is, RDH can restore the image losslessly after extracting information from the cover image. RDH provides solutions for application scenarios that require non-destructive capabilities, such as privacy protection in medical, judicial, and military scenarios. RDH technology can be broadly classified into three methods: pixel extension (DE) [6–8] histogram shift (HS) [9], and prediction error extension (PE) [10]. DE is a method

for hiding data by first doubling the difference h between two adjacent pixels, hiding one bit from the LSB (least significant bit) of h, and then adjusting the two pixels using h' and two pixels. In HS [9], the zeros and peak points of the cover image histogram are used to hide secret metadata. That is, while scanning the image, if the pixel (when the grayscale value is α) is the peak point (α) and the bit to hide is '1', the pixel value is changed to $\alpha + 1$. If the bit is '0', the pixel value remains α. In the existing position of a+1, the value is moved to an adjacent pixel. the information of the moved pixel is stored separately and is used for image restoration. Therefore, the smaller the pixel value of $\alpha + 1$, the better. PE [10] is a method of hiding data by using the difference in prediction error between the original pixel and the prediction pixel.

Recently, RDH has expanded the research area to RDH of the encrypted domain [11–13] and dual-image-based RDH [14–25]. Dual-image-based RDH is a research field in which RDH and secret sharing [26,27] are partially combined. Secret sharing was introduced for the purpose of improving the vulnerability of secret security compared to data hiding. Encoding uses n shadow images, and decoding is made possible by reconstruction of $k(\leq n)$ shadow images. Thus, it is known as (k, n) threshold secret sharing.

Dual-image-based RDH can be considered as another form of secret sharing using two identical cover images. In dual-image-based RDH, the owner hides data in two identical original cover images; creates two marked images that maintain the original image quality; and delivers them to the receiver. The receiver restores the original cover image after extracting the data. Therefore, the dual-image RDH method can provide higher embedding capacity and security than the traditional method.

Figure 1 shows a diagram of the dual-image RDH strategy. The dual-image method is a model suitable for data delivery that requires the reliability of security. Additionally, it can be used to distribute military deployment images and various fields based on medical images within medical groups.

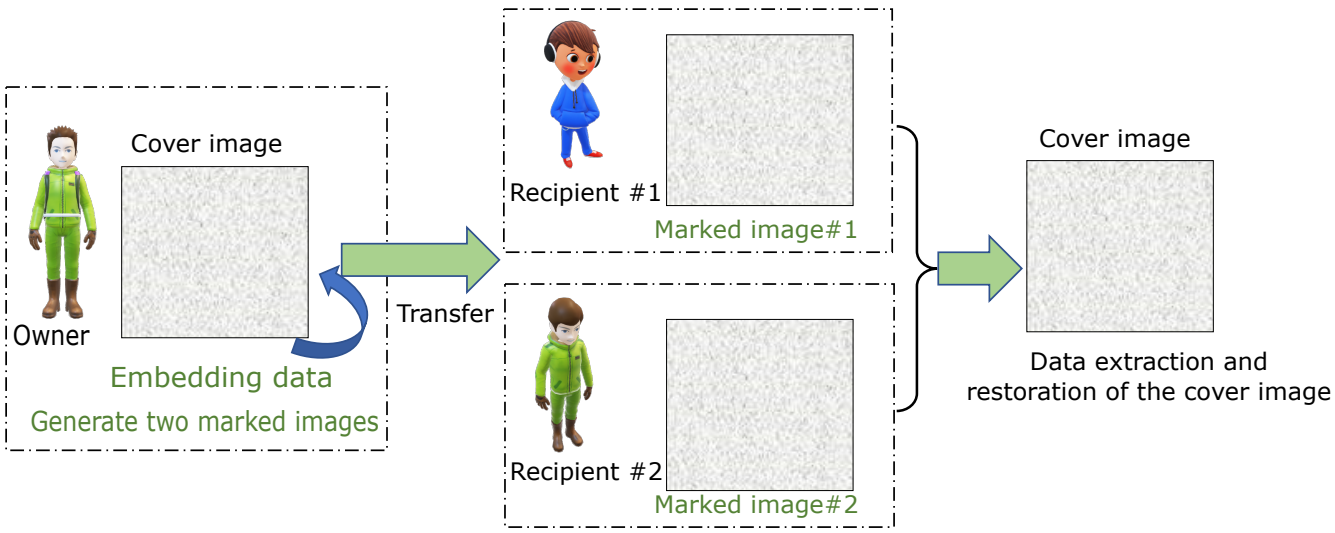

Figure 1. Schematic diagram of the dual-image RDH application scenario.

Chang et al. [14] generate a 5×5 modulus matrix using EMD and modulus functions, and then the message (5-ary value) intersects the two diagonals of the submatrix. A modulus function is applied to this to insert data. Chang et al. [15] improved the embedding capacity by increasing the submatrix size from 5×5 to 9×9. Lee et al. [16] constructed a crosshair to indicate the relationship between the original pixel and the displayed pixel, precorrected the pixel pairs according to the positioning strategy, and inserted the pixel values after confirming their recoverability. When hiding a secret bit in one pixel, it is possible to hide

about 2 bits of data with only a maximum of plus or minus one operation, so that the high quality of marked images can be maintained.

Lin et al. [18] simultaneously embed two five-base secret numbers in each pixel pair of the cover image according to the EMD matrix to generate two stego pixel pairs. In some cases, moving these pairs of stego pixels into the proper position will produce two meaningful shadows. Lu et al. [19] proposed a novel dual-image RDH method based on a center-folding strategy (CFS). The k-bit binary secret was first converted to decimal and then mapped from the range $[0, 2^k - 1]$ to the range $[-2^k - 1, 2^{k-1} - 1]$ via center folding. Lu et al. [20] propose an alternative approach. It is based on the least significant bit matching (LSB matching) method for embedding using dual imaging technique. Jana et al. [21] proposed an embedding scheme based on the (7, 4) Hamming code, where the secret message bits are embedded through error generation and the original image is recovered using the Hamming error correction code.

Meanwhile, Block Truncation Coding (BTC) [28] is one of the compression methods, and the configuration of BTC is very simple compared to conventional JPEG. BTC compression operation is simple and the quality of BTC-based images is not significantly lower than that of the original image, so it is suitable for applications that do not require high image quality. Lema and Mitchell proposed a variant of BTC, Absolute Moment BTC (AMBTC) [29]. AMBTC adopts a bi-clustering approach, so it uses two quantization levels in blocks, similar to BTC. Compared to BTC, the calculation time is shorter and the image quality is excellent.

Chuang and Chang [30] proposed a DH method in which each block of the bitmap is divided into a smooth block and a complex block, and then the bitmap of the smooth block is replaced with a secret bit. The image quality is controlled by dividing the block into a smooth block and a complex block, creating a difference value (threshold value: T) between two quantization levels representing the block. That is, when the threshold value T is lowered, the image quality is improved, but the DH capacity is deteriorated. If the size of the threshold is increased, the image quality may decrease and the hidden data capacity may increase. Ou and Sun [31] proposed a method to adjust image distortion by adjusting the two quantization levels, but the original image is required for recomputation. Chen et al. [32] proposed a lossless DH method using two quantization level orders. This method is named the Order of Two Quantization Level (OTQL) method, which can conceal one bit per block. This method does not change the coefficients of both quantization levels, so it does not affect the quality of the image.

The BTC-based RDH proposed by Sun et al. [33] obtains BTC-compressed data with an average table, a low average table, and a bitplane sequence. The secret data are then included without loss into the high-average and low-average tables. This is a lossless method based on the relationship between the current value and the neighboring values in the mean table. Chang et al. [34] introduced RDH based on the AMBTC compression code. Secret information is hidden mainly by using joint neighborhood coding (JNC). Hong et al. [35] proposed a RDH for AMBTC compressed codes. Here, this method transforms the quantization levels into means and differences, which are used to carry data bits. Moreover, the classifications of prediction errors are adaptively assigned, and varied length indicators are utilized to effectively reduce the bitrate.

In this paper, we propose a double reversible data hiding (RDH) method based on AMBTC images. This study is the first case of using AMBTC as a cover image. AMBTC is very efficient for image and video transmission in IoT-based systems due to its simple compression calculation and relatively good compression efficiency. The receiver can extract data using data extraction and restoration methods. The proposed method is focused on how to avoid the attacker's eyes by hiding data while hiding data in the image with less damage to the image. Moreover, the proposed RDH can be used for copyright protection of images and videos. The proposed dual RDH is performed using Hamming codes [36,37] and LSB replacement for bitmaps constituting each block of AMBTC.

The main contributions of this study are summarized in two points: First, HC(7,3) is a code capable of correcting an error of 1 bit in a 7-bit codeword, and is a very useful method for hiding 3 bits when applied to the bitmap constituting the block. Using this method, errors in the data hiding process can be reduced to a minimum. In addition, sufficient data can be hidden by applying the LSB replacement method to the bitmap. Second, it is possible to adjust the amount of hidden data and the quality of the cover image by adjusting the difference value T of the coefficients of the two quantization levels constituting each block of AMBTC.

The rest of this paper is organized as follows. Section 2 introduces the preliminary knowledge for our work. Section 3 describes the proposed dual-AMBTC RDH method. Section 4 shows the experimental results, and Section 5 concludes this paper.

2. Preliminary Knowledge

2.1. AMBTC

BTC is a relatively straightforward and simple lossy image compression method. The BTC method has a low computational cost, making it suitable for non-critical applications where image quality is critical. Later, Lema and Mitchell [29] presented the AMBTC method, which is an improved method of the existing BTC method. The main difference between AMBTC and BTC is that they compute two quantization levels. To compress an image with AMBTC, the image is divided into non-overlapping blocks of $\ell \times \ell$ pixels. For each block, x is the pixels that make up the block, and the average pixel value $\bar{x}$ is calculated by:

$$\bar{x} = \frac{1}{\ell \times \ell} \sum_{j=1}^{\ell \times \ell} x_j \tag{1}$$

where x_j represents the j^{th} pixel of this block. Each pixel value x_j is compared with the average value $\bar{x}$ using Equation (2). If x_j is greater than or equal to $\bar{x}$, b_j is 1, otherwise b_j is 0. That is, a bitmap $\mathcal{M} = [b_j]$ consisting of two groups (a set of '1 s' and a set of '0 s') is obtained.

$$b_i = \begin{cases} 1, & if\ (x_j \geq \bar{x}), \\ 0, & if\ (x_j < \bar{x}). \end{cases} \tag{2}$$

AMBTC has two quantization values per block: an upper mean and a lower mean. Equation (3) calculates two quantized values in each block, where t represents the number of ones in each bitmap $\mathcal{M}$ (condition: $x_j \geq \bar{x}$). $\lfloor \cdot \rfloor$ is a function that takes a real number x as input and gives the largest integer less than or equal to x, denoted as $floor(x)$. The two quantization levels Γ_1 and Γ_0 are the upper and lower means, respectively, based on $\bar{x}$.

$$\Gamma_1 = \left\lfloor \frac{1}{\ell} \sum_{x_j \geq \bar{x}} x_j \right\rfloor \text{ and } \Gamma_0 = \left\lfloor \frac{1}{(\ell \times \ell) - t} \sum_{x_j < \bar{x}} x_j \right\rfloor. \tag{3}$$

Finally, the image block is compressed into two quantization levels (Γ_0, Γ_1) and a bitmap $\mathcal{M}$, i.e., a $trio(\Gamma_0, \Gamma_1, \mathcal{M})$. If $\ell = 4$, i.e., process the image as a (4×4) block-wise operation. The 16 pixels that make up a block are $8 + 8 + 16 = 32$ bits, so the CR is $(16 \times 8)/32 = 4$. A file size of $2M$ bits can be reduced to $0.5M$ bits. In the decoding step, when two quantization levels and a bitmap are obtained, the corresponding image block can be easily reconstructed by replacing all '1 s' in bitmap $\mathcal{M}$ with Γ_1 and all '0 s' with Γ_0.

2.2. Hamming Code

The Hamming code [36,37] is a single error correction linear block code with $(n, k) = (2^r - 1, 2^r - 1 - r)$, where $r = n - k$ is the number of check bits and k is the number of message bits in the codeword. A binary linear $[n, k]$ code y of length n and dimension k is a k-dimensional linear subspace of $\mathbb{F}_2^n$, where the sum of two vectors and the scalar product of the vectors are defined using ordinary binary operations. Let m be a

message of length k and let $m \in \mathbb{F}_2^k$. For the mapping from any message m to codeword y, we use a generator matrix G. That is, $y = mG$ Hamming code has a parity check matrix $\mathcal{H}$ expressed as:

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

The syndrome decoding algorithm for correcting single errors in Hamming code is given as follows. Given the received codeword y, to estimate e, it first forms the product $\mathcal{H}\tilde{y}^T = \mathcal{H}y^T + \mathcal{H}e^T = \mathcal{H}e^T$. Once we know which bit was corrupted, we can recover the codeword by using the syndrome. The syndrome φ can be obtained as

$$\varphi = (\mathcal{H} \cdot y^T). \tag{5}$$

If the syndrome φ is equal to zero, there is no error in the codeword y, otherwise the syndrome $j = \varphi$ is the location of the error. The error can be corrected by inverting the j-th pixel in the codeword $\tilde{y}$. That is, $y = (1 - y_j)$. Given y, compute $\mathcal{H}y^T$. This involves multiplying a $\Theta(log\ n) \times n$ matrix with an $n \times 1$ vector, which can be achieved in time, i.e., $\Theta(n\ log\ n)$.

Example 1. *If John sends the codeword* $y = [1\ 0\ 0\ 1\ 0\ 1\ 0]$ *to Mary, it is assumed that the codeword y causes an error in the third transmission process. That is, Mary receives the codeword* $\tilde{y} = [1\ 0\ 1\ 1\ 0\ 1\ 0]$ *with an error. Mary computes the syndrome (*$\varphi = \mathcal{H} \cdot y^T = [0\ 1\ 1]$*) and finds the e error. As a result, she obtains the error-corrected codeword* $y = [1\ 0\ 0\ 1\ 0\ 1\ 0]$.

3. Proposed Scheme

This section introduces a dual-image-based RDH method that applies HC(7,4), LSB direct replacement, and OTQL to each block's bitmap and two quantization levels. Figure 2 shows the general embedding framework of the proposed method, which consists of two main steps. The first step is to obtain two covered AMBTC images using the AMBTC compression algorithm. The second step is to apply HC(7,4) to the bitmap of each block (4×4 pixels) of the two cover AMBTC images and covertly hide the message using OTQL for the two quantization levels of the same block. In this process, two marked AMBTCs with hidden data are created, and then these images are transmitted to the receiver. The receiving side can extract data and restore the cover image with the proposed decoding method.

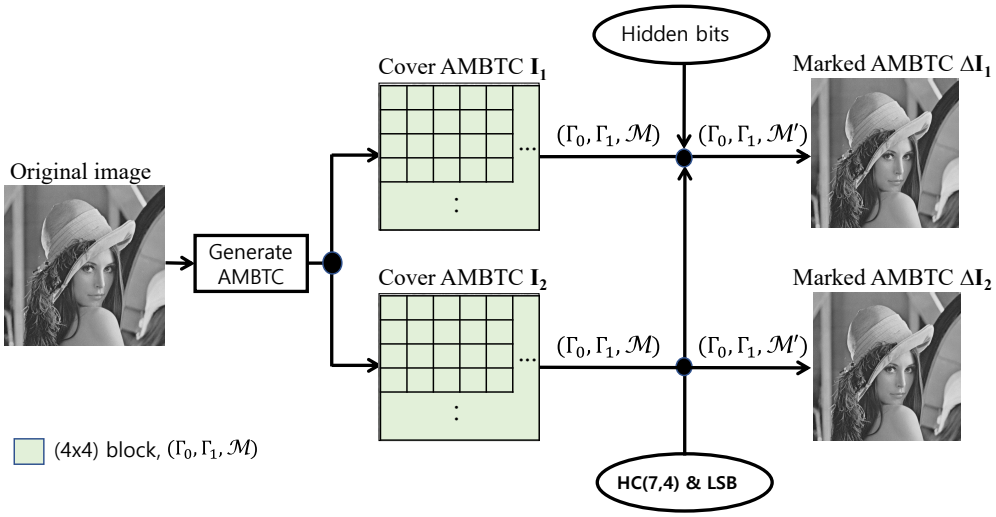

Figure 2. Schematic diagram for the proposed model.

3.1. Data Embedding Procedure

Using AMBTC method (Section 2.1), two cover AMBTC images, I_1 and I_2, are obtained. In this section, we describe the data embedding procedure (Figure 3) in detail, with two cover images, step by step.

Input: Two cover AMBTC images I_1 and I_2, secret bits $m = (m_1, m_2, \ldots, m_n)$.
Output: Two marked images $\triangle I_1$ and $\triangle I_2$.

Step 1: The size of two images is $N \times N$ and the variable i denotes $1 \leq i \leq (N \times N)/(4 \times 4)$.

Step 2: Read a sized 4×4 block $\mathcal{B}_i^1$ and $\mathcal{B}_i^2$ of two cover AMBTC I_1 and I_2, respectively.

Step 3: A codeword is constructed with 7 bits each of blocks $\mathcal{B}_i^1$ and $\mathcal{B}_i^2$. That is, $y_1 = (b_1b_2b_3b_4b_5b_6b_7) \in \mathcal{B}_i^1$ and $y_2 = (b_1b_2b_3b_4b_5b_6b_7) \in \mathcal{B}_i^2$.

Step 4: Equation (6) is applied to codeword y_1 to calculate the syndrome for data hiding, i.e., $\varphi_1 = \mathcal{H}y_1^T$. The result of XOR operation on the syndrome and 3-bit m is assigned to φ_1', i.e., $\varphi_1' = \varphi_1 \oplus m_j^{j+3}$. If the result of converting φ_1' to a decimal number is not zero, it is calculated as in Equation (7). Assign the syndrome φ_1' to (l_1, l_2, l_3) of block $\mathcal{B}_i^2$ (Figure 3), where $b2d$ is a function that converts a binary number to a decimal number.

$$\varphi_1 = \mathcal{H} \cdot y_1^T \text{ and } \varphi_2 = \mathcal{H} \cdot y_2^T \tag{6}$$

$$y_{i,j}^1 = \begin{cases} 1 - y_{i,j}^1, \; if \; j = b2d(\varphi_1), \\ y_{i,j}^1, \text{ otherwise.} \end{cases} \text{ and } y_{i,j}^2 = \begin{cases} 1 - y_{i,j}^2, \; if \; j = b2d(\varphi_2), \\ y_{i,j}^2, \text{ otherwise.} \end{cases} \tag{7}$$

Step 5: In order to hide data in $\mathcal{B}_i^2$, Equations (6) and (7) are applied to codeword y_2, and the method proceeds in the same way as in Step 4, i.e., $\varphi_2' = (\mathcal{H} \cdot y_2^T) \oplus m_j^{j+3}$. Assign the syndrome φ^2 to (l_1, l_2, l_3) of block $\mathcal{B}_i^1$ (Figure 3).

Step 6: For $F = (f_1, f_2, \ldots, f_5)$ of blocks $\mathcal{B}_i^1$ and $\mathcal{B}_i^2$, respectively, data are hidden by directly replacing m_j^{j+10} bits with the bitmap. That is, $f_1^5 \in \mathcal{B}^1 = m_j^{j+5}$ and $f_1^5 \in \mathcal{B}^2 = m_j^{j+5}$.

Step 7: A bit of m is hidden using the order of the two quantization levels. If 1 bit of m is '0', the order of the two quantization levels (Γ_0, Γ_1) is not changed. If $m =$ '1', the order of the two quantization levels (Equation (8)) is changed to (Γ_1, Γ_0).

$$\begin{cases} swap(\Gamma_0, \Gamma_1), \text{ if } (m =' 1') \\ \text{no change, otherwise.} \end{cases} \tag{8}$$

Step 8: If variable i is less than $(N \times N)/(4 \times 4)$, go to Step 2.

If this process is repeatedly applied as much as the size of the image, marked images $\triangle I_1$ and $\triangle I_2$ are obtained, respectively.

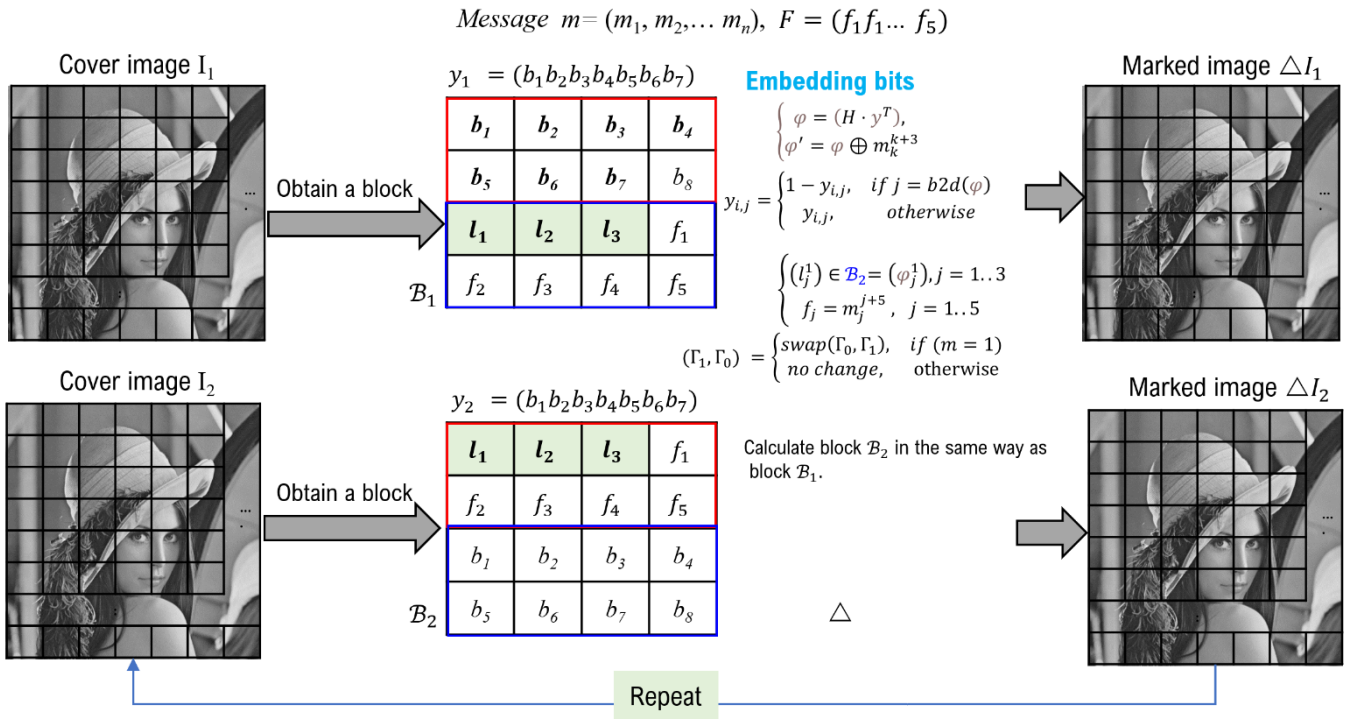

Figure 3. Diagram of embedding procedure.

3.2. Data Extraction and Recovering Procedure

In this section, we extract hidden data from two marked images $\triangle I_1$ and $\triangle I_2$ and also reconstruct the cover AMBTC image. The detailed process is described in detail (Figure 4) step by step as follows.

Input: Two marked AMBTC $\triangle I_1$ and $\triangle I_2$.
Output: The reconstructed AMBTC I, secret bit $m = (m_1, m_2, \dots, m_n)$.

Step 1 : The size of two images is $N \times N$ and the variable i is the block index, i.e., $1 \leq i \leq (N \times N)/(4 \times 4)$. The variable j is initialized to one.

Step 2: Read a block $\mathcal{B}_i^1$ and $\mathcal{B}_i^2$ of two marked AMBTC $\triangle I_1$ and $\triangle I_2$, respectively.

Step 3: A codeword is constructed with 7 bits each of blocks $\mathcal{B}_i^1$ and $\mathcal{B}_i^2$. That is, $y_1 = (b_1 b_2 b_3 b_4 b_5 b_6 b_7) \in \mathcal{B}_i^1$ and $y_2 = (b_1 b_2 b_3 b_4 b_5 b_6 b_7) \in \mathcal{B}_i^2$.

Step 4: In order to extract three hidden bits m in the block $\mathcal{B}_i^1$, the syndrome φ_1 is calculated for the codeword y_1 by Equation (6). Put the extracted bits into m, i.e., $m_j^{j+3} = \varphi_1, j = j+3$.

Step 5: To extract three hidden bits m in the block $\mathcal{B}_i^2$, the syndrome φ_2 is calculated for the codeword y_2 using Equation (6). Put the extracted bits into m, i.e., $m_j^{j+3} = \varphi_2, j = j+3$.

Step 6: Extract hidden bits from directly $F = (f_1, f_2, \dots f_5) \in \mathcal{B}_i^1$ and $\mathcal{B}_i^2$, i.e., $m_j^{j+5} = F \in \mathcal{B}_i^1$, $m_j^{j+5} = F \in \mathcal{B}_i^2$ and $j = j+5$, respectively.

Step 7: According to the order of the two quantization levels, 1 bit is extracted by using Equation (9), and then put into m_j.

$$m_j = \begin{cases}\text{'1'}, \text{ if } (\Gamma_0 > \Gamma_1),\\ \text{'0'}, \text{ otherwise.}\end{cases} \tag{9}$$

Step 8: (l_1, l_2, l_3) of $\mathcal{B}_i^2$ is a syndrome for y_1, which is converted to a decimal number, i.e., $pos = b2d([l_1 l_2 l_3])$ and then the pixel corresponding to y_1 is restored to the original pixel using Equation (10).

$$y(pos) = 1 - y(pos). \tag{10}$$

Step 9: (l_1, l_2, l_3) of $\mathcal{B}_i^1$ is a syndrome for y_2, which is converted to a decimal number, and then the pixel corresponding to y_2 is restored to the original pixel by using Equation (10).

Step 10: The restored bitmap block is completed by copying y_1 of $\mathcal{B}_i^1$ and y_2 of $\mathcal{B}_i^2$ to the bitmap block $\mathcal{B}_i$, and then copied to the corresponding positions of I.

Step 11: If variable index i is less than $(N \times N)/(4 \times 4)$, go to *Step 2*.

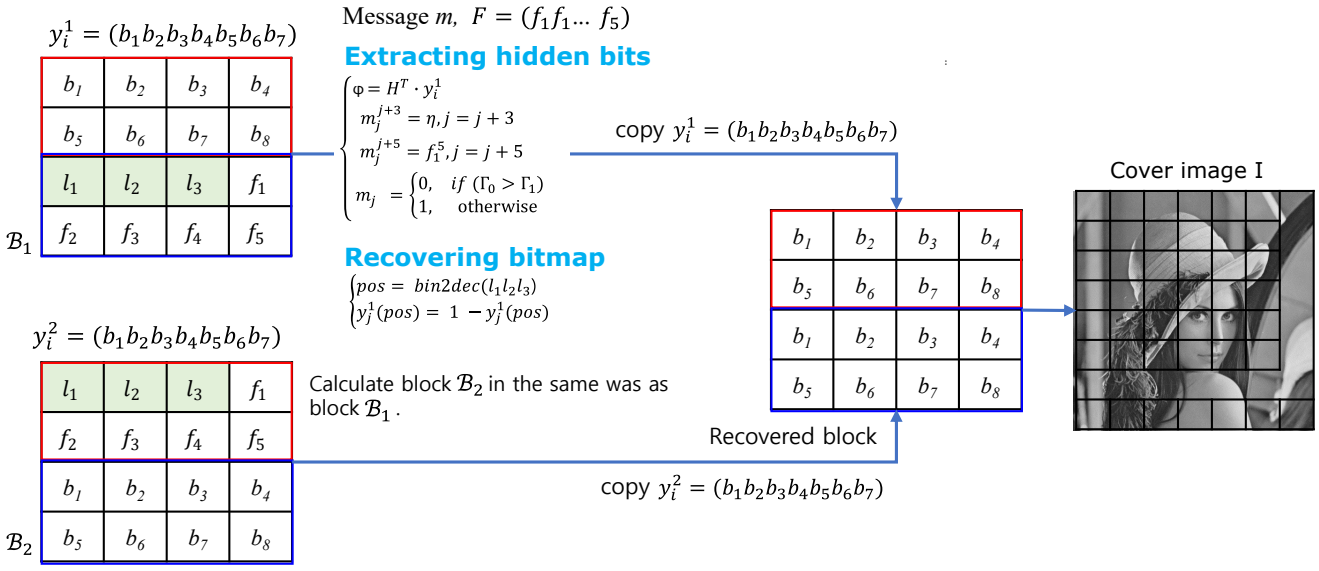

Figure 4. Diagram of extraction and recovering procedure.

4. Experimental Results and Discussions

In this section, we simulate the performance of the proposed AMBTC-based dual data hiding and describe the possibilities and strengths of the proposed method. The platform used in the experiment has a Core i5-8250U processor, 1.60 GHz speed, 8 GB RAM, and the software for the simulation is MATLAB R2019b. Our proposed experimental model selects several images from the standard USC-SIPI image database [38] and uses them for the experiment. The image used is a 512×512 grayscale image. Figure 5 is the original grayscale image for simulation (e.g., Lena, Pepper, Airplane, Boat, Goldhill, Couple, Baboon, Zelda, Barbara, Tiffany).

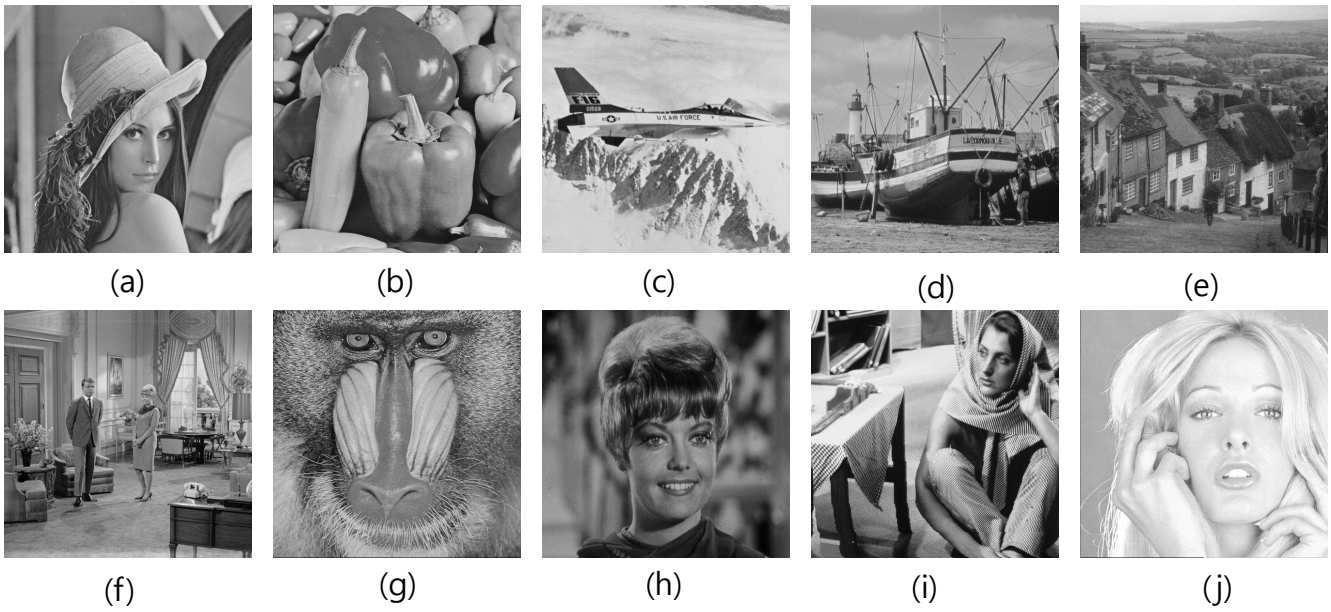

Figure 5. Test images: (**a**) Lena, (**b**) Peppers, (**c**) Airplane, (**d**) Boat, (**e**) Goldhill, (**f**) Couple, (**g**) Baboon, (**h**) Zelda, (**i**) Barbara, (**j**) Tiffany (512×512).

In order to evaluate the performance of the proposed method, ER, PSNR, and SSIM are introduced. ER is commonly used to estimate the performance of the embedding capacity and is calculated by

$$ER = \frac{tot}{(N \times N \times 2)} \tag{11}$$

where *tot* refers to the total number of hidden bits, N is the height and the width of the cover image, and 2 denotes the number of marked images.

PSNR [39] provides an objective criterion for evaluating image quality. As for PSNR, the larger the calculated value, the higher the image quality recognized, and if it is 30 dB or more, the human visual system recognizes it as sufficiently similar to the original image. PSNR is calculated using Equations (12) and (13).

$$PSNR = 10 \times log_{10}\left(\frac{255^2}{MSE}\right). \tag{12}$$

Mean-Squared Error (MSE) is a formula for the difference in mean intensity between a marked image and the original image. That is, the smaller the difference between the two images, the lower the MSE and the higher the PSNR. The notations y and y' denote the pixels of the cover image and the marked image, respectively. The formula for MSE is:

$$MSE = \frac{1}{N \times N} \sum_{i=1}^{N \times N} (y_i - y'_i)^2. \tag{13}$$

In other word, the MSE is the mean $(\frac{1}{N \times N} \sum_{i=1}^{N \times N})$ of the squares of the errors $(y_i - y'_i)$. 255^2 means maximum pixel intensity.

In addition, another measurement introduced for performance evaluation is SSIM [39], which is a formula (Equation (14)) that measures the similarity between the original image and the marked image.

$$SSIM(p, p') = \frac{(2\mu_p\mu'_p + c_1)(2\sigma_{pp'} + c_2)}{(\mu_p^2 + \mu_p'^2 + c_1)(\sigma_p^2 + \sigma_p'^2 + c_2)} \tag{14}$$

where μ_p and μ'_p are the mean values of p and p', respectively, and c_1 is the stabilization constant and σ_p^2, $\sigma_{p'}^2$, and $\sigma_{pp'}$ are the variances and covariances of the cover image and the stego image. c_1 and c_2 are constant values used to avoid division by zero problems.

Table 1 compares the PSNR of the cover AMBTC image and the two marked AMBTCs with the original image. A message is hidden in a bitmap that satisfies the condition $T \leq 5$, the difference between the two quantization levels representing each block. Table 1 shows the size of the actual hidden bit under this condition. $\text{PSNR}_{(O-I)}$ is a measure of the PSNR of the cover image with respect to the original image. $\text{SSIM}_{(O-I)}$ measures the SSIM of the cover image with respect to the original image. Except for the high-frequency Baboon image, it can be seen that the measurement is very similar to the original image. $\text{PSNR}_{(I-I')}$#1 is measured PSNR of marked image#1 including cover image and data, and it was measured with high PSNR of 45dB or more. $\text{PSNR}_{(I-I')}$#2 is the PSNR evaluation for the second marked image. The PSNR measurement comparing the marked image with the original image is the $\text{PSNR}_{(O-I')}$. It can be seen that most of the measurement results are very close to $\text{PSNR}_{(O-I)}$. Therefore, it can be confirmed that the damage to the cover image is not significant.

Table 1. Comparison of PSNR for all dual RDH AMBTC images (when threshold $T = 5$).

Images	$\textbf{PSNR}_{(O-I)}$	$\textbf{SSIM}_{(O-I)}$	**EC**	$\textbf{PSNR}_{(I-I')}$ #1	$\textbf{PSNR}_{(I-I')}$ #2	$\textbf{PSNR}_{(O-I')}$ #1	$\textbf{PSNR}_{(O-I')}$ #2
Lena	33.6556	0.9468	148705	45.2557	45.2875	33.3654	33.3702
Pepper	34.0968	0.9395	114593	45.7492	45.7017	33.8055	33.8068
Airplane	32.0372	0.9504	168193	45.7197	45.756	31.8562	31.8587
Boat	31.5700	0.9388	142865	46.603	46.6046	31.4365	31.4365
Goldhill	32.836	0.9208	67537	50.1685	50.1580	32.7545	32.7551
Couple	31.2567	0.9241	70193	48.8354	48.8542	31.1806	31.1809
Baboon	26.9765	0.8872	34449	57.7565	57.494	26.9726	26.9726
Zelda	36.6537	0.9476	123201	45.1824	45.1478	36.0805	36.075
Barbara	27.0756	0.9248	89553	48.9218	48.9832	27.0478	27.0477
Tiffany	35.6576	0.9473	147521	45.1695	45.1695	35.1957	35.206
Average	31.8235	0.9311	106588	48.2436	48.2208	31.6111	31.6115

Table 2 shows EC and PSNR measurements while increasing the difference T between two quantization levels representing a block to 5, 10, 15, and 20. For the Lena image, the EC is measured to be 148705 for $T = 5$ and the EC is measured to be 247313 for $T = 20$. In this case, the mean PSNR is 33.3678 and 32.2417, respectively. In Table 2, the reason for the measurement based on the difference between the two quantization levels is that EC and PSNR can be adjusted according to the difference.That is, as the two differences decrease, the EC also decreases. Meanwhile, PSNR increases. EC and PSNR are inversely proportional.

Table 3 compares the maximum EC for the existing method and our proposed method. However, since there has been no previous study on the AMBTC-based dual RDH method, it is compared with the studies of the AMBTC-based RDH (Sun et al. (2013) [33], Chang et al. (2018) [34], and W. Hong (2018) [35]) among related studies. In this experiment, the EC of our proposed method shows the highest performance compared to the existing methods. The PSNR is relatively low, because it can be seen that the PSNR decreased inversely because the EC was increased to the maximum.

Table 2. PSNR of dual stego images with embedding capacity.

Original Image I	Threshold T	EC	PSNR^1	PSNR^2	Avg. PSNR
Lena	5	148,705	33.3654	33.3702	33.3678
	10	206865	32.8208	32.8335	32.8271
	15	232545	32.2292	32.2542	32.2417
	20	247313	31.6594	31.699	31.6792
Boat	5	142865	31.4365	31.4365	31.4365
	10	174529	31.2241	31.2172	31.2206
	15	199249	30.8229	30.8091	30.816
	20	219425	30.2416	30.2505	30.246
Pepper	5	114593	33.8055	33.8068	33.8061
	10	213985	32.8545	32.8637	32.8591
	15	241681	32.2336	32.2521	32.2428
	20	254593	31.7156	31.7641	31.7398
Goldhill	5	67537	32.7545	32.7551	32.7548
	10	141073	32.0922	32.0991	32.0956
	15	195777	31.0982	31.1116	31.1049
	20	225857	30.2253	30.2423	30.2338
Zelda	5	123201	36.0805	36.075	36.0777
	10	226369	34.5375	34.5615	34.5495
	15	253425	33.6805	33.6865	33.6835
	20	267457	32.9338	32.9668	32.9503
Barbara	5	89553	27.0478	27.0477	27.0477
	10	135969	26.9345	26.9329	26.9337
	15	158497	26.7962	26.7973	26.7967
	20	267457	26.6557	26.6545	26.6551
Airplane	5	168193	31.8562	31.8587	31.8574
	10	208545	31.5844	31.584	31.5842
	15	227873	31.232	31.2363	31.2341
	20	240625	30.8257	30.8355	30.8306
Couple	5	70193	31.1806	31.1809	31.1807
	10	143009	30.7385	30.7422	30.7403
	15	180721	30.1983	30.1906	30.1944
	20	205217	29.6088	29.6046	29.6067
Tiffany	5	147521	35.1957	35.206	35.2008
	10	211825	34.3111	34.3197	34.3154
	15	240433	33.4334	33.4311	33.4322
	20	255985	32.6425	32.6755	32.659
Baboon	5	34449	26.9726	26.9726	26.9726
	10	70481	26.8768	26.8781	26.8774
	15	104593	26.664	26.6665	26.6652
	20	128225	26.4056	26.4056	26.4056

Table 3. Comparing the maximum EC and PSNR of the existing method and our proposed method.

Images	Sun et al. [33]		Chang et al. [34]		W. Hong et al. [35]		The Proposed			
	EC	PSNR	EC	PSNR	EC	PSNR	EC	PSNR^1	PSNR^2	Avg. PSNR
Baboon	64008	26.98	151439	26.34	64516	26.98	128225	26.4056	26.4056	26.4056
Boats	-	-	166786	31.16	64516	31.12	219425	30.2416	30.2505	30.2460
Goldhill	-	-	168185	33.72	64516	32.83	225857	30.2253	30.2423	30.2338
Airplane	64008	31.97	177814	33.29	64516	31.97	240625	30.8257	30.8355	30.8306
Lena	64008	33.24	175145	33.72	64516	33.24	247313	31.6594	31.6990	31.6792
Peppers	64008	33.42	174222	34.10	64516	33.42	254593	31.7156	31.7641	31.7398
Tiffany	64008	35.77	-	-	64516	35.77	255985	32.6425	32.6755	32.6590
Average	64008	32.28	168932	32.06	64516	32.19	224575	30.5308	30.5500	30.5400

Table 4 compares the ER of the proposed method and the existing method (Lee et al. (2013) [15], Liu et al.(2018) [23], Lin et al. (2019) [18]). The ER of the existing method is measured from the uncompressed image, and the ER of the proposed method is obtained from the compressed image. The file size of our proposed AMBTC-compressed image is

four times smaller, with an ER of 0.56. Therefore, it can be seen that the performance of our proposed method is not bad.

Table 4. Comparison of maximum embedding ratio with different schemes.

Images	File Size	Lee et al. [15]	Liu et al. [23]	Lin et al. [18]	The Propose	
		ER	ER	ER	File Size	ER
Baboon	258 KB	1.09	1	1.07	64 KB	0.56
Barbara	258 KB	1.09	1	1.07	64 KB	0.56
Lena	258 KB	1.09	1	1.07	64 KB	0.56
Pepper	258 KB	1.09	1	1.07	64 KB	0.56
Goldhill	258 KB	1.09	1	1.07	64 KB	0.56
Airplane	258 KB	1.09	1	1.07	64 KB	0.56
Boat	258 KB	1.09	1	1.07	64 KB	0.56
Couple	258 KB	1.09	1	1.07	64 KB	0.56
Zela	258 KB	1.09	0.99	1.07	64 KB	0.56

Compared to data hiding for a single image, data throughput may be slightly longer when using two images. However, encoding does not require real-time processing. Additionally, in this paper, two identical cover-image-based data hiding methods use some of the advantages of the existing secret sharing method, and secret sharing improves security by itself. This is an advantage of dual image-based data hiding. The fact that the amount of data hiding can be increased can also be an advantage.

5. Conclusions

In this paper, we proposed AMBTC-based dual-image RDH using HC (7,3) and LSB replacement. This method introduces AMBTC for the first time and provides high ER = 0.56 and PSNR. Because dual-RDH is used, it has security advantages over traditional data hiding methods. Since our proposed method is RDH, the receiver can restore the original cover image by using two marked images and a data restoration method. By using the AMBTC compressed image as a cover image, fast transmission in a low-traffic network may be possible. In addition, the proposed method has the advantage that the quality of the two marked images is almost the same because the payload is properly distributed. In the future, we would like to develop a new technology that can improve the performance of ER and PSNR by further developing research on dual data hiding based on AMBTC.

Funding: This research received no external funding.

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

Abbreviations

The following abbreviations are used in this manuscript:

AMBTC	Absolute Moment Block Truncation Coding.
BTC	Block Truncation Coding.
DH	Data Hiding.
RDH	Reversible Data Hiding.
Dual RDH	Dual Reversible Data Hiding.
HC	Hamming Code.
I_1	Cover Image 1.
I_2	Cover Image 2.
$\triangle I_1$	Marked Image 1.
$\triangle I_2$	Marked Image 2.
m	Secret bits.
$\mathcal{B}$	A 4×4 block of the cover image.
y	A codeword.

$b2d$	A function that converts a binary number to decimal number.
φ	Syndrome.
x	A pixel.
$\bar{x}$	Average pixel value.
Γ_0 and Γ_1	Two quantization levels.
$\mathcal{H}$	A parity check matrix.

References

1. Suryawanshi, P.; Padiya, P.; Mane, V. Detection of Contrast Enhancement Forgery in Previously and Post Compressed JPEG Images. In Proceedings of the 2019 IEEE 5th International Conference for Convergence in Technology (I2CT), Bombay, India, 29—31 March 2019; pp. 1–4. https://doi.org/10.1109/I2CT45611.2019.9033764.
2. Bender, W.; Gruhl, D.; Morimote, N.; Lu, A. Techniques for data hiding. *IBM Syst. J.* **1996**, *35*, 313–336.
3. Shi, Y.Q.; Li, X.; Zhang, X.; Wu, H.-T.; Ma, B. Reversible data hiding: Advances in the past two decades. *IEEE Access* **2016**, *4*, 3210–3237.
4. Kim, C.; Shin, D.-K.; Yang, C.-N.; Leng, L. Hybrid data hiding based on AMBTC using enhanced Hamming code. *Appl. Sci.* **2020**, *10*, 5336.
5. Yang, C.N.; Wu, S.Y.; Chou, Y.S.; Kim, C. Enhanced stego-image quality and embedding capacity for the partial reversible data hiding scheme. *Multimed. Tools Appl.* **2019**, *78*, 18595–18616.
6. Tian, J. Reversible data embedding using a difference expansion. *IEEE Trans. Circuits Syst. Video Technol.* **2003**, *13*, 890–896.
7. Wu, D.C.; Tsai, W.H. A steganographic method for images by pixel-value differencing. *Pattern Recognit. Lett.* **2013**, *24*, 1613–1626.
8. Alattar, A.M. Reversible watermark using the difference expansion of a generalized integer transform. *IEEE Trans. Image Process.* **2004**, *13*, 1147–1156.
9. Ni, Z.; Shi, Y.Q.; Ansari, N.; Su, W. Reversible data hiding. *IEEE Trans. Circuits Syst. Video Technol.* **2006**, *16*, 354–362.
10. Tsai, P.Y.; Hu Y.C.; Yeh, H.L. Reversible image hiding scheme using predictive coding and histogram shifting. *Signal Process.* **2009**, *89*, 1129–1143.
11. Zhang, W.; Wang, H.; Hou, D.; Yu, N. Reversible Data Hiding in Encrypted Images by Reversible Image Transformation. *IEEE Trans. Multimed.* **2016**, *18*, 1469–1479.
12. Cao, X.; Du, L.; Wei, X.; Meng, D.; Guo, X. High Capacity Reversible Data Hiding in Encrypted Images by Patch-Level Sparse Representation. *IEEE Trans. Cybern.* **2016**, *46*, 1132–1143.
13. Ke, Y.; Zhang, M.; Zhang, X.; Liu, J.; Su, T.; Yang, X. A Reversible Data Hiding Scheme in Encrypted Domain for Secret Image Sharing Based on Chinese Remainder Theorem. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, **32**, 2469–2481.
14. Chang, C.C.; Kieu T.D.; Chou Y.C. Reversible data hiding scheme using two steganographic images. In Proceedings of IEEE Region 10 International Conference (TENCON), Taipei, Taiwan, 30 October–2 November 2007; pp. 1–4.
15. Chang, C.C.; Chou, Y.C.; Kieu, T.D. Information hiding in dual images with reversibility. In Proceedings of the 3rd International Conference on Multimedia and Ubiquitous Engineering, Qingdao, China, 4–6 June 2009; pp. 145–152.
16. Lee, C.F.; Huang, Y.L. Reversible data hiding scheme based on dual stegano-images using orientation combinations. *Telecommun. Syst.* **2013**, *52*, 2237–2247.
17. Chang, C.C.; Lu, T.C.; Horng, G.; Huang, Y.H.; Hsu, Y.M. A high payload data embedding scheme using dual stego-images with reversibility. In Proceedings of the 3rd International Conference on Information, Communications and Signal Processing, Islamabad, Pakistan, 16–18 December 2013; pp. 1–5.
18. Lin, J.Y.; Chen, Y.; Chang, C.C.; Hu, Y.C. Dual-image with integrity verification using exploiting modification direction. *Multimed. Tools Appl.* **2019**, *78*, 25855–25872.
19. Lu, T.C.;Wu, J.H.; Huang, C.C. Dual-image-based reversible data hiding method using center folding strategy. *Signal Process.* **2015**, *115*, 195–213.
20. Lu, T.C.; Tseng, C.Y.; Wu, J.H. Dual imaging-based reversible hiding technique using LSB matching. *Signal Process.* **2015**, *108*, 77–89.
21. Jana, B.; Giri, D.; Mondal, S.K. Dual image based reversible data hiding scheme using (7,4) hamming code. *Multimed. Tools Appl.* **2018**, *77*, 763–785.
22. Yao, H.; Qin, C.; Tang, Z.; Tian, Y. Improved dual-image reversible data hiding method using the selection strategy of shiftable pixels' coordinates with minimum distortion. *Signal Process.* **2017**, *135*, 26–35.
23. Liu, Y.; Chang, C.C. A turtle shell-based visual secret sharing scheme with reversibility and authentication. *Multimed. Tools Appl. 2018*, *77*, 25295–25310.
24. Chen, X.; Hong, C. An efficient dual-image reversible data hiding scheme based on exploiting modification direction. *J. Inf. Secur. Appl.* **2021**, *58*, 102702
25. Sun, Y.X.; Li, Q.; Yan, B.; Pan, J.S.; Yang, H.M. Reversible data hiding in dual encrypted halftone image using matrix embedding, *Multimed. Tools Appl.* **2020**, *79*, 27659–27682.
26. Shamir A. How to share a secret. *Commun. Assoc. Comput. Mach.* **1979**, *22*, 612–613.
27. Naor, M., Shamir, A. Visual cryptography. In *Advances in Cryptology—EUROCRYPT'94. EUROCRYPT 1994, Perugia, Italy, 9–12 May 1994*; De Santis, A., Ed.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 1995; Volume 950.

28. Delp, E.; Mitchell, O. Image compression using block truncation coding. *IEEE Trans. Commun.* **1979**, *27*, 1335–1342.
29. Lema, M.D.; Mitchell, O.R. Absolute moment block truncation coding and its application to color images. *IEEE Trans. Commun.* **1984**, *32*, 1148–1157.
30. 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.
31. 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.
32. Chen, J.; Hong, W.; Chen, T.S.; Shiu, C.W. Steganography for BTC compressed images using no distortion technique. *Imaging Sci. J.* **2013**, *58*, 177–185.
33. 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.
34. Chang, C.C.; Chen, T.S.; Wang, Y.K.; Liu, Y.J. A reversible data hiding scheme based on absolute moment block truncation coding compression using exclusive OR operator. *Multimed. Tools Appl.* **2018**, *77*, 9039–9053.
35. Hong,W.; Zhou, X.Y.; Weng, S.W. Joint adaptive coding and reversible data hiding for AMBTC compressed images. *Symmetry* **2018**, *10*, 254.
36. Rurik, W.; Mazumdar, A. Hamming codes as error-reducing codes. In Proceedings of the 2016 IEEE Information Theory Workshop (ITW), Cambridge, UK, 11–14 September 2016; pp. 404–408.
37. Moon, T.K. *Error Correction Coding—Mathematical Methods and Algorithms*; John Wiley & Sons: Hoboken, NJ, USA, 2005; pp. 2001–2006.
38. Image Databases. Available online: https://www.imageprocessingplace.com/root_files_V3/image_databases.htm (accessed on 5 May 2022).
39. Horé, A.; Ziou, D. Image Quality Metrics: PSNR vs. SSIM. In Proceedings of the 20th International Conference on Pattern Recognition, Istanbul, Turkey, 23–26 August 2010; pp. 2366–2369. https://doi.org/10.1109/ICPR.2010.579.