A Tamper-Detection Scheme for BTC-Compressed Images with High-Quality Images

Thai-Son Nguyen[1, 2], Chin-Chen Chang[1, 3] and Ting-Feng Chung[1]
[1] Department of Information Engineering and Computer Science, Feng Chia University,
Taichung 40724, Taiwan, R.O.C,
[e-mail: carter629629@gmail.com]
[2] Department of Information Technology, Tra Vinh University
Tra Vinh Province, Vietnam
[e-mail: thaison@tvu.edu.vn]
[3] Department of Biomedical Imaging and Radiological Science, China Medical University
Taichung 40402, Taiwan, R.O.C
[e-mail: alan3c@gmail.com]
*Corresponding author: Chin-Chen Chang

Received January 6, 2014; revised March 27, 2014; accepted April 22, 2014; published June 27, 2014

Abstract

This paper proposes a novel image authentication scheme, aiming at tampering detection for block truncation coding (BTC) compressed image. The authentication code is generated by using the random number generator with a seed, and the size of the authentication code is based on the user's requirement, with each BTC-compressed image block being used to carry the authentication code using the data hiding method. In the proposed scheme, to obtain a high-quality embedded image, a reference table is used when the authentication code is embedded. The experimental results demonstrate that the proposed scheme achieves high-quality embedded images and guarantees the capability of tamper detection.

Keywords: Block truncation coding (BTC), high-quality image, image authentication, tamper detection

http://dx.doi.org/10.3837/tiis.2014.06.011

1. Introduction

With the rapid development of multimedia and the Internet, a massive amount of digital information (i.e., images, video, and audio) is transmitted daily over the Internet which is a public channel. In fact, a huge amount of digital information is stored and transmitted each second. This means that transmitted data have many treats for being easily tampered or copied by malicious users. Therefore, the protection of transmitted data has become an issue of increasing concern. Many researchers have proposed data hiding and watermarking techniques to solve this security problem. Data hiding is a technique by which invisible secret data can be delivered securely by hiding the data in cover media, such as the 'Hello Kitty' image. Watermarking is another technique that is used to protect the copyright of digital data.

All current digital image compression techniques are based on exploiting the redundancy of information that is inherent in most digital images. This redundancy stems from the statistics of the image data, which are directly related to the probability distribution of the image data and can be treated by information theory techniques using image entropy concepts. Basically, image compression techniques can be divided into two types. The first type is called lossless compression, i.e., Huffman coding [1, 2] or arithmetic coding [3, 4]. The property of lossless compression is that the original image can be recovered after decompression. The second type of image compression technique, i.e., block truncation coding (BTC) [5-10,], wavelet coding and discrete cosine coding [11-13, 24], or vector quantization (VQ) [14,15, 26], is lossy compression [5-16, 26], which is often used to compress digital images. This is because the latter has a better compression ratio than the former.

In the past five years, many watermarking schemes have been proposed in compression domain. In 2008, Lee and Lin [17] introduced a new watermark scheme that not only detects the tampered areas in the embedded image, but also enables to recover the areas where are tampered. In Lee and Lin's scheme, each block in the image is used to carry watermark of other two blocks. Therefore, there are two copies of watermark for each block in the watermarked image. In 2010, Ahmed and Siyal [18] proposed a hash-based authentication scheme to verify the image by using a hash function. Ahmed and Siyal's scheme obtained the good resilience against some attacks, i.e., JPEG compression, low-pass and high-pass filtering. In 2011, Chuang and Hu [19] introduced a new authentication scheme that is used to detect illegal modifications in the vector quantization compressed image. In their scheme, two sets of authentication data are needed to perform the tamper detection process and authenticate the given VQ compressed image. Their scheme achieved acceptable image quality of the embedded image and it provided accurate detection capability. From 2012 to 2013, Hu et al. proposed a three different tamper detection schemes [20, 23, 25] for the compressed images of BTC. In [23], the authentication codes of the image blocks are created from the quantization levels. Then, many copies of the authentication codes are embedded into the bit maps based on the permutation operation. In [25], a joint image compressed and image authentication technique had been proposed for the compressed images of BTC. In this scheme, pseudo random generator is used to generate the authentication code that is embedded into the bit maps of AMBTC-compressed image blocks. The embedded bit maps and these quantization levels are further compressed to reduce the storage cost. In [20], multiple sizes of authentication code are also embedded into the BTC-compressed image. Their scheme can obtain the clear tamper detection capability. However, the visual quality of the embedded image obtained by this scheme is still low, with the average quality of the embedded image being less than 39 dB. To obtain the higher visual quality while maintaining clear tamper detection capability for the BTC-compressed images, in this paper, we propose a new tamper detection scheme. In the proposed scheme, each quantization level of the BTC-compressed image block is used to carry the authentication code by referring to a reference table. Then, to detect any areas that have been tampered, the embedded authentication code is extracted and verified. The mathematical morphology operation can be used to improve the detection results [21]. Experimental results demonstrate that the proposed scheme achieves both high-quality of the embedded images and clear verification of any tampering that occurred.

The rest of this paper is organized as follows. Section 2 provides a review of related work, such as absolute moment block truncation coding [6] and Hu et al.'s scheme [20]. In Section 3, the details of our proposed scheme are discussed. The experimental results are provided in Section 4, and our conclusions are presented in Section 5.

2. Related Work

2.1 Absolute Moment Block Truncation Coding (AMBTC)

To compress images, Lema and Mitchell [6] introduced an AMBTC technique in 1984. In this scheme, the image is first partitioned into an image block with the size of $n \times n$. Then, to encode the image block P, the mean value, *mean*, of the image block is computed and used to separate all of the pixels of P into two clusters. The first cluster L_1 is used to contain all pixels that have value smaller than *mean*. The second cluster L_2 is used to contain the rest pixels.This means that the corresponding bit value to be stored into the bit map, *BM*, is 0 or 1, when the pixel is in the cluster L_1 or in the cluster L_2, respectively.

After two clusters are determined, Equations (1) and (2) are used to compute two quantization levels, p and q, respectively:

$$p = \frac{1}{n-m} \times \sum_{P_i < mean} P_i \tag{1}$$

$$q = \frac{1}{m} \times \sum_{P_i \geq mean} P_i \tag{2}$$

where m denotes the number of pixels that have values that are equal to or larger than the mean.

In AMBTC technique, a compressed trio, also called as compressed block, (p, q, *BM*) is generated for each image block. **Fig. 1** shows an example of the AMBTC technique. Assume that **Fig. 1(a)** is the original image block with the size of 4 × 4 pixels. The mean value, *mean*, of this block is 127.75. The bit map *BM* of this block, which is generated by using AMBTC, is presented in **Fig. 1(b)**. According to Equations (1) and (2), the values of (p, q) is obtained as (103, 159). Then, the compressed trio, (103, 159, 0000110011001110), is sent to the receiver. Then receiver decodes the compressed trio to obtain the reconstructed image block, as shown in **Fig. 1(c)**.

126	120	86	70
161	154	106	85
168	162	121	103
172	167	131	112

(a) Original image block of 4×4 pixels

0	0	0	0
1	1	0	0
1	1	0	0
1	1	1	0

(b) Bit map

103	103	103	103
159	159	103	103
159	159	103	103
159	159	159	103

(c) Reconstructed image block

Fig. 1. Example of the AMBTC technique

2.2 Hu et al.'s Scheme

In this section, we introduce Hu et al.'s scheme [20]. The main purpose of their scheme is to protect the integrity of the BTC-compressed images from being tampered, where the BTC-compressed images are generated by using the AMBTC technique [6]. To achieve high accuracy in detecting tamper areas and less distortion of the BTC-compressed image, for each compressed trio (*p*, *q*, *BM*), a different value *d* between *p* and *q* are computed. Then, the different value *d* is used to embed the authentication code that is generated in advance using a pseudo random-number generator (PRNG). Then, to detect the tamper areas of the BTC-compressed image, the embedded authentication codes are extracted and verified. This scheme consisted of three procedures, i.e., authentication code generator, embedding, and detection procedures, each of which is described below.

Assume that the BTC-compressed image *I* with a size of $W \times H$ is processed by using Hu et al.'s scheme, and assume that the block size is set to $n \times n$. Therefore, the entire number of $k \times l$ compressed image blocks is processed, meaning that $k \times l$ of compressed trios (*p*, *q*, *BM*) are generated, where $k = W/n$ and $l = H/n$. Let *b* be the number of bits is embedded into each different value *d*.

For each compressed trio (*p*, *q*, *BM*), Hu et al. generates a random number *r*, meaning that $k \times l$ random values are generated for $k \times l$ compressed blocks of BTC-compressed image. Then, authentication code, *w*, is calculated by using Equation (3). According to Equation (3), the value of *w* is in the range of $[0, 2^b-1]$. Then, to embed *w* into the compressed block, the value *w* is converted to the *b*-bit of binary form.

$$w = r \bmod 2^b \tag{3}$$

For example, random value *r* is generated as 26534, and $b = 3$. By using Equation (3), the authentication code, *w*, is 6, and its binary form is $(110)_2$.

To embed *w* into the compressed trio (*p*, *q*, *BM*), the differenct value *d* is calculated as $d = q - p$ and converted to binary form *bd*. Then, the last *b*-bit of *db* is compared with the binary form of *w*. If they are equal, the quantization levels are kept unchanged. Otherwise, the quantization levels are modulated to ensure that the last *b*-bit of *db* has the same value as the *b*-bit binary form of authentication code, *w*. When the binary form of *w* is not equal to *db*, the quantization levels are modulated. The first quantization level, *p*, is fixed, and two candidates, i.e., q_1, q_2, are computed by using Equations (4) and (5), respectively. These two candidates are used for replacing the second quantization level, *q*.

$$q_1 = p + \lfloor d / 2^b \rfloor \times 2^b + w \tag{4}$$

$$q_2 = \begin{cases} q_1 - 2^b & if\ q_1 > q \\ q_1 + 2^b & if\ q_1 < q \end{cases} \tag{5}$$

where $\lfloor\ \rfloor$ is the floor function.

After the two candidates, q_1 and q_2, have been generated, which one that is closer to *q* is used for replacing *q*. The selected candidate is denoted as q_s, and the quantization level, *q*, is replaced by q_s.

After the replacement, to ensure that difference value between the new quantization level, q_s, and the original quantization level, *q*, is small, the new quantization levels (p, q_s) are adjusted. Then, the displacement of the adjustment of quantization levels is computed by Equation (6):

$$dis = \lfloor ds - d \rfloor / 2 \tag{6}$$

where *dis* is the displacement value, and *ds* is the absolute difference value between *p* and q_s. Then, the quantization levels (p, q_s) are adjusted to (p', q') by using Equations (7) and (8).

$$p' = p + dis \tag{7}$$

$$q' = q_s + dis \tag{8}$$

The above-mentioned steps are repeated until all of the quantization levels of the BTC-compressed image are processed completely. This means that each pair, i.e., (p, q), is used to carry b-bit of authentication code. To better explain the procedure used to embed the authentication code in Hu et al.'s scheme, the following example is provided. Assuming that *b* is 3, the BTC-compressed trio (82, 140, 1000110011101111) is used for embedding the 3-bit authentication code, *w*, and $w = (101)_2$. To embed w into the BTC-compressed trio, the different value $d = 58$ is first obtained by subtracting 82 from 140. Since the last 3-bit of *db* being determined as $(010)_2$, which is different from the authentication code *w*, $(101)_2$. Therefore, according to Equations (4) and (5), two candidates, q_1 and q_2, are computed, where q_1 is equal to $(82 + \lfloor 58/2^3 \rfloor \times 2^3 + 5) = 143$, and q_2 is equal to $(143 - 2^3) = 135$. In this case, the first candidate $q_1 = 143$ is chosen for replacing the quantization code $q = 140$ because it is closer to q than the second candidate, $q_2 = 135$. Then, the new quantization levels are (82, 143). To adjust these new quantization levels, the absolute difference value, *ds*, and the displacement of the adjustment of the quantization levels, *dis*, are computed as 61 and 1, respectively. According to Equations (7) and (8), the adjusted quantization levels (p', q') are obtained as (81, 142).

Once the owner of the BTC-compressed image suspects that her/his image has been illegally tampered by others, he/she will use the tamper detection procedure to detect this image. Specifically, in this procedure, the embedded authentication code is extracted and compared with the original authentication code generated by PRNG with seed. By doing that, tamper locations are determined.

3. Proposed Scheme

After carefully considering Hu et al.'s scheme [20], we discovered that it embeds the authentication code into each BTC-compressed block (p, q, *BM*) by modifying the different value *d* between p and q. In their scheme, to minimize the distortion of the embedded BTC-compressed image, the two candidates, q_1 and q_2, are used to determine whether $q_1 < q < q_2 = q_1 + 2^b$ (if $q_1 < q$) or whether $q_1 - 2^b < q < q_1$ (if $q_1 > q$), where *b* is the number of bits of authentication code. Then, the closer one is chosen to replace the second quantization level, q, and the new quantization levels are adjusted to generate the embedded quantization levels (p', q'). Therefore, the original quantization levels (p, q) must be changed to the embedded quantization levels (p', q'), which must be from 0 to $2^b/2$. In other words, for each block, the mean square error (MSE) must be increased from 0 to $(2^b/2)^2$. This means that, when *b* increases, the visual quality of the embedded image decreases. In this paper, to further improve the quality of the embedded image while maintaining the clear tamper detection

capability, we propose a new tamper detection scheme for BTC-compressed images. In the proposed scheme, first, a reference table and the authentication code are generated. Then, the authentication code is hidden into the BTC-compressed block (*p*, *q*, *BM*) by referring to the reference table. The proposed scheme consists of two phases, i.e., embedding the authentication code and tamper detection.

3.1 Authentication Code Embedding Phase

In this section, the main idea of the authentication code embedding phase is presented. In the proposed scheme, the authentication code for each component of the compressed trio is generated by using PRNG, which is the same as was done in Hu et al.'s scheme, and the secret key, *K*, is used as the seed. Let *AC* be the authentication code, which is defined as $AC = \{ac_1, ac_2,..., ac_{k\times l}\}$. Assume that image *I* with a size of $W \times H$ is compressed by using the AMBTC technique, with block size of $n \times n$. Therefore, the entire number, $k \times l$, of the BTC-compressed image block (*p*, *q*, *BM*) is obtained, where $k = W/n$ and $l = H/n$. **Fig. 2** shows the flowchart of the main processes in the authentication code embedding phase of our proposed scheme.

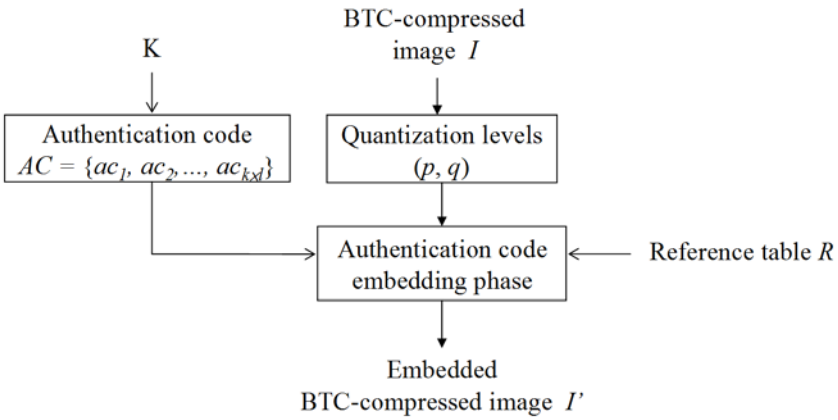

Fig. 2. Flowchart of the main processes in the authentication code embedding phase

To generate reference table *R*, which is based on the number of bits *b* of the authentication code, *ac*, and the extraction function *h* [22], the extraction function *h* is defined by Equation (9)

$$h(g_1, g_2,..., g_n) = (\sum_{i=1}^{n} g_i \times i) \bmod (2^b) \tag{9}$$

where g_i is gray pixel values, which are integers from [0, 255], and *b* is the number of bits of the authentication code. **Figs. 3(a)** and **(b)** show the simplest cases of $b = 2$ and $b = 3$, respectively. Here, each square in reference table *R* is represented by an *h* value. The *h* values of any square and its 2^b neighbors that are integers from 0 to $2^b - 1$, are mutually different.

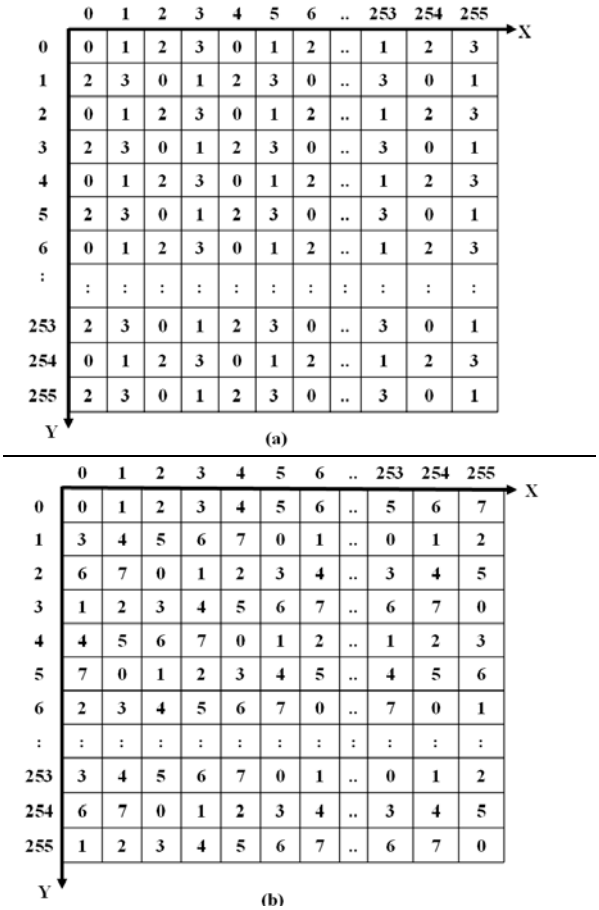

Fig. 3. Reference table *R* with different values of b: (a) $b = 2$; (b) $b = 3$

Our proposed authentication code embedding phase can be broken into five steps. The corresponding algorithm is described in detail below:

Embedding authentication code algorithm

Input: BTC-compressed image *I*, the number of bits *b* of the authentication code, and the secret key *K*

Output: Embedded BTC-compressed image *I'*

Step 1: Generate the reference table *R* and the authentication code *AC*.

Step 2: For each of the quantization levels (*p*, *q*) that is located on the reference matrix *R* at row *p* and column *q*, read the authentication code, ac_i, from *AC*.

Step 3: From the value of *b*, a set, *CS*, of candidate elements within the reference table *R* is constructed, as shown in **Fig. 4**. Note that the 2^b elements in the candidate set *CS* are exactly composed of non-repeating integers between 0 and 2^b-1. **Fig.s 4(a)**, **(b)**, and **(c)** show the candidate sets when the number of bits *b* are 2, 3, and 4, respectively.

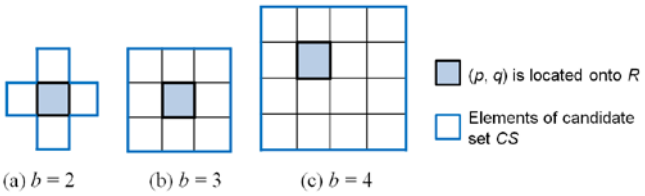

Fig. 4. Candidate set with different values of b

Step 4: To embed the authentication code, ac_i, into quantization levels (p, q), find a position (p', q') in the candidate set CS that satisfies $R(p', q') = ac_i$.
Step 5: Repeat Steps 2 through 4 until all quantization levels of the BTC-compressed image I have been processed completely.

After all of the steps have been completed, the authentication code AC is embedded into the entire image I. To better explain the authentication code embedding phase, an example is provided in **Fig. 5**. Assume that the quantization levels (p, q) of the BTC-compressed image block are (3, 2), the authentication code, ac_i, is 4, b is 3, and the reference table R is constructed as shown in **Fig. 3(b)**. According to Step 2, the quantization levels (3, 2) are located on the reference table R, and the candidate set CS is determined as shown in **Fig. 5**. Then, to embed the authentication code, ac_i, into the quantization level (3, 2), the element $R(3, 3)$ is selected from the set CS due to R(3, 3) = 4 = ac_i. Then, the current quantization levels (3, 2) are replaced by (3, 3).

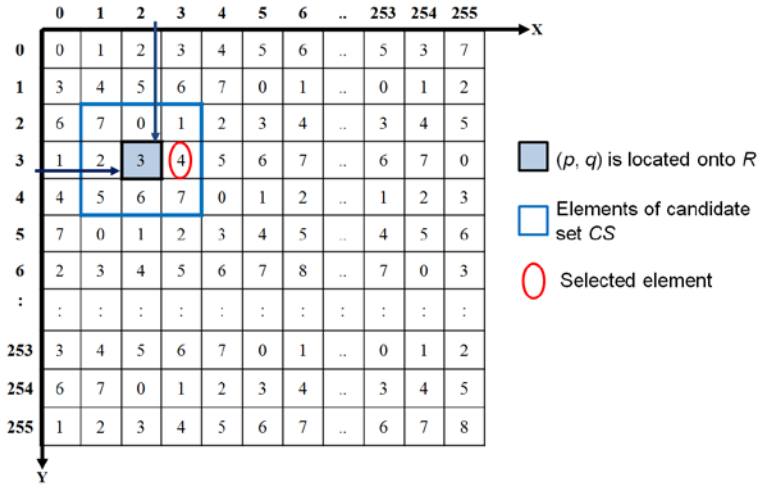

Fig. 5. Example of the embedding authentication code phase

3.2 Tamper Detection Phase

Assume that the owner of the embedded BTC-compressed image I' suspects that a published image has been tampered from her/his embedded BTC-compressed image. In this scenario, this phase is used to authenticate that there is any modifications in the embedded BTC-compressed image I'.

To extract and verify the embedded authentication code, two parameters, b and K, are required in this phase. **Fig. 6** shows the flowchart of main processes in the tamper detection phase.

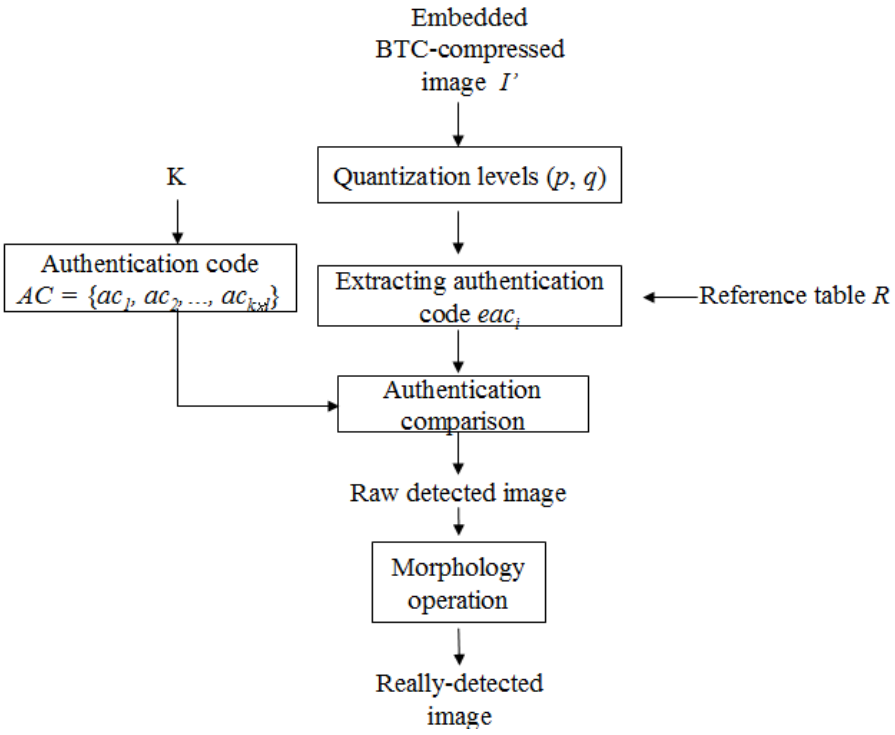

Fig. 6. Flowchart of main processes in the tamper detection phase

The following algorithm shows the tamper detection phase in detail.
Input: The embedded BTC-compressed image I', the number of bits b of the authentication code, and the secret key K
Output: Tampered image
Step 1: Generate reference table R, and authentication code, AC, based on b and the secret key K, respectively.
Step 2: For each of the quantization levels (p', q') that is located on the reference matrix R at the row p' and the column q', the embedded authentication code, eac_i, is extracted as the value of $R(p', q')$ in the reference table R.
Step 3: Read the corresponding authentication code, ac_i, from the AC.
Step 4: If $ac_i = eac_i$, the BTC-compressed image block is marked as a clear block; otherwise, the BTC-compressed image block is marked as a tampered block.
Step 5: Repeat Steps 2 through 4 until all quantization levels of the embedded BTC-compressed image I' have been processed completely and the roughly tampered image is generated by combining of all the clear blocks and tampered blocks.
Step 6: Estimate the roughly tampered image by using a morphological operation [21], and, then, the really-tampered areas are located correctly.

4. Experimental Classification Results and Analysis

To demonstrate the extensive experiments conducted with the proposed scheme, some of the experimental results are discussed in this section. Eight general gray images with sizes of 512×512 were test in these experiments. The eight gray images that were used were entitled

F16, Boat, Goldhill, Lena, Couple, Clock, Peppers, and Sailboat, and they are shown in **Fig. 7**. In addition, the AMBTC scheme was used to compress the original images. This was done because the AMBTC scheme, which is a proven scheme, obtained the optimal value of MSE [6]. All computations were conducted on a computer with an Intel(R) Core™ i7-3770 CPU @ 3.4 GHz and an 8-GB RAM with Windows 7 Professional as the operating system, and Microsoft Visual Studio 2005 C# was used to implement the experiments.

To estimate the visual quality of the embedded image, the peak signal-to-noise ratio (*PSNR)* was defined in Equation (10):

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

The mean square error (MSE) for a $W \times H$ grayscale image was defined as shown in Equation (11):

$$MSE = \frac{1}{H \times W}\sum_{i=1}^{H}\sum_{j=1}^{W}\left(X_{ij} - Y_{ij}^{'}\right)^2 \tag{11}$$

where W and H are the dimensions of the images, and X_{ij} and Y'_{ij} are the pixel values of the original image and the embedded image, respectively. In principle, a lower value for *MSE* means less error, and, as considered in Equation (10), the inverse relationship between *MSE* and *PSNR* translates into a high value of *PSNR*, which is desirable. Here, the "signal" is the original image, and the "noise" is the error in the embedded image. Therefore, the lower the *MSE* (or the higher the *PSNR*) is, the better the visual quality of the image is.

(a) F16 (b) Boat (c) Goldhill (d) Lena

(e) Couple (f) Clock (g) Peppers (h) Sailboat

Fig. 7. Eight grayscale test images with sizes of 512×512

Table 1. Qualities of images obtained by the AMBTC scheme [6].

Value of n / Images	n = 2	n = 4	n = 8
Airplane	40.579	33.253	30.153
Boat	39.344	31.958	28.995
Goldhill	40.287	33.236	30.136
Lenna	40.491	33.643	30.216
Couple	38.716	31.262	28.106
Clock	42.587	33.519	30.300
Peppers	41.248	34.010	30.237
Sailboat	37.947	31.129	28.094
Average	40.150	32.751	29.530

Table 1 shows the visual qualities of images obtained by the AMBTC scheme [6] with three different values of *n*, i.e., 2, 4, and 8. Obviously, the visual quality of these images increase as the value of *n* decreases. When the values of *n* were 2, 4, and 8, the average visual qualities are achieved as 40.150 dB, 32.751 dB, and 29.530 dB, respectively, and the bit rates were 5 bits per pixel (bpp), 2 bpp, and 1.25 bpp, respectively.

Table 2. Comparison of the image quality of our proposed scheme and Hu et al.'s scheme when *n* was 2.

b	b = 2		b = 3		b = 4	
Scheme / Images	Hu et al.	Proposed	Hu et al.	Proposed	Hu et al.	Proposed
Airplane	38.874	40.576	36.241	39.092	33.162	35.700
Boat	38.093	39.342	35.915	38.561	33.368	36.733
Goldhill	38.814	40.282	36.581	39.254	32.672	37.058
Lena	38.907	40.488	36.379	39.421	33.634	36.926
Couple	37.681	38.715	35.880	38.283	32.785	37.481
Clock	40.314	42.577	37.461	40.875	32.006	36.899
Peppers	39.431	41.245	36.689	40.111	33.967	37.526
Sailboat	37.053	37.946	35.425	37.526	32.514	36.752
Average	38.646	40.146	36.321	39.140	33.014	36.884

Tables 2 to 4 present the qualities of the embedded images of the proposed scheme and that of Hu et al.'s scheme [20] when using different values of *n*, i.e., 2, 4, and 8. As can be seen from **Table 2**, when the value of *n* was 2, and *b* had the values of 2, 3, and 4. The average visual qualities of the embedded images obtained by our proposed scheme were 40.146, 39.140, and 36.884 dB, respectively. It means that the losses of image quality by our proposed scheme were 0.004, 1.010, and 3.266 dB, for b values of 2, 3, and 4, respectively. In the same configuration, the losses for Hu et al.'s scheme were 1.504, 3.829, and 7.136 dB, respectively. Our scheme performed better than Hu et al.'s scheme because their scheme embeds the *b* bits of authentication code into the quantization values (*p*, *q*) by modifying the difference value between *p* and *q*. In Hu et al.'s scheme, two candidates, q_1 and q_2, are generated by using Equations (4) and (5), and the different value between q_1 and q_2 is 2^b. The quantization level *q*

will be in the middle of q_1 and q_2. Then, to embed ac into the quantization levels, q is replaced by the closer candidate, meaning that the value of q will be changed from 0 to $2^b/2$. Therefore, the different value between the original quantization levels (p, q) and the embedded quantization values (p', q') also is changed from 0 to $2^b/2$. For example, when b is equal to 3, the amount of the change will be from 0 to 4. Conversely, the proposed scheme is based on the reference table to ensure that the smallest change of quantization levels. For example, to embed the b bits of authentication code, (p, q) is located in the reference table R, and the candidate set is determined as shown in **Fig. 4**. **Fig. 4** shows that when b is equal to 3, the embedded quantization value (p', q') will be located in the range [p-1, q-1] and [p+1, q+1], meaning that the difference value between the original quantization values (p, q) and the embedded quantization values (p', q') is from 0 to $\sqrt{2}$. Obviously, the embedded quantization levels in the proposed scheme will be closer to the original quantization levels than they are in Hu et al.'s scheme. Therefore, we can conclude that the proposed scheme achieved better visual quality of reconstructed images than Hu et al.'s scheme.

Table 3. Comparison of the image quality of our proposed scheme and Hu et al.'s scheme when n was 4.

b	b = 2		b = 3		b = 4	
Scheme / Images	Hu et al.	Proposed	Hu et al.	Proposed	Hu et al.	Proposed
Airplane	32.899	33.252	32.077	33.075	30.836	32.602
Boat	31.701	31.958	31.107	31.884	30.138	31.746
Goldhill	32.922	33.235	32.250	33.109	30.314	32.960
Lena	33.278	33.642	32.499	33.526	31.086	33.308
Couple	31.059	31.262	30.607	31.204	29.208	31.186
Clock	33.146	33.518	32.309	33.353	29.416	32.834
Peppers	33.618	34.010	32.809	33.895	31.250	33.756
Sailboat	30.928	31.129	30.485	31.067	29.242	31.017
Average	32.444	32.751	31.768	32.639	30.186	32.426

Table 4. Comparison of the image quality of our proposed scheme and Hu et al.'s scheme when n was 8.

b	b = 2		b = 3		b = 4	
Scheme / Images	Hu et al.	Proposed	Hu et al.	Proposed	Hu et al.	Proposed
Airplane	29.976	30.153	29.585	30.098	28.768	30.014
Boat	28.865	28.995	28.560	28.961	27.926	28.922
Goldhill	29.978	30.135	29.642	30.087	28.553	30.067
Lena	30.054	30.216	29.683	30.170	28.709	30.136
Couple	28.006	28.106	27.776	28.077	26.948	28.072
Clock	30.122	30.299	29.698	30.241	27.996	30.150
Peppers	30.075	30.237	29.722	30.190	28.686	30.169
Sailboat	27.996	28.094	27.779	28.065	27.003	28.056
Average	29.384	29.529	29.056	29.486	28.074	29.448

In **Tables 3** and **4**, it is clear that, when the values of *n* were 4 and 8, our proposed scheme provided better image quality than Hu et al.'s scheme in all cases. The results in **Tables 3** and **4** shows that, as the value of *b* increased, the better the quality of the embedded image became in the proposed scheme. For example, in **Table 3**, when the value of *b* was 2, an average image quality of our proposed scheme is greater than that of Hu et al.'s scheme. The average gain rate is 0.307 dB. However, when the value of *b* is increased to 4, the average image quality of the proposed scheme was 2.24 dB greater than that of Hu et al.'s scheme. This can be explained by the fact that a reference table was used in the proposed scheme. In addition, in the proposed scheme, when the smaller block size *n* is used, the better quality of the embedded image is obtained. The quantization levels are utilized to compress image block. Therefore, if the block size n is small, the quantization levels will be closer to the original pixel value. Consequently, the higher quality of BTC-compressed images is achieved. However, more bits are required to compress the image.

(a) Tampered object (b) Binary tampered object

Fig. 8. Tampered image used in tamper detection test

In the tamper test, **Fig. 8(a)** was inserted on the wall of the embedded image. **Fig. 8(b)** shows the binary version of the tamper area of the tampered object.

(a) Embedded image (b) Tampered image

(c) Raw detected image NC = 0.7233 (d) Really-detected image NC = 0.9672

Fig. 9. Tamper test of embedded image "F16" when the value of n was 4, and b was equal to 3: (a) embedded compressed image with a PSNR of 33.075 dB; (b) tampered image; (c) raw detected image; (d) really-detected image after morphology operation

Fig. 9 shows the tamper test result of the proposed scheme for image "F16" when the value of *n* was 4, and b was equal to 3. **Fig. 9(d)** shows the refined detected image after the morphology operation. Normalized correlation coefficient (NC) is used to measure the similarity between the tamper binary image and the detected image. Basically, the nearer the value *NC* to value 1 is obtained, the more accuracy the detected image is achieved. It is easy to see that the proposed scheme provides the high accuracy of tamper detection, when the *NC* value is larger than 0.96. Here, *NC* is calculated using Equation (12)

$$NC = \frac{1}{TI_h \times TI_w} \sum_{i=0}^{TI_{h-1}} \sum_{j=0}^{TI_{w-1}} TI(i,j) \times DI(i,j), \quad (12)$$

where *TI* is the tamper binary image, *DI* is the detected image, and TI_h and TI_w are the height and width of the tamper binary image, respectively.

Table 5. Comparison of the proposed scheme and Hu et al.'s scheme when the value of n was 2 and b was equal to 2.

Schemes	Image quality of embedded image	Clear tamper area	Technique for improving detection accuracy
Hu et al.	38.874	Yes	Tamper refinement mechanism
Proposed	40.576	Yes	Morphology operation

Table 5 shows the comparison of the proposed scheme and Hu et al.'s scheme [20] when *n* was equal to 2 and the value of *b* was 2. It is apparent that the proposed scheme and Hu et al.'s schemes can be used for tamper detection. However, the quality value of the embedded image obtained in the proposed scheme is better than that obtained in Hu et al.'s scheme.

5. Conclusion

In this paper, we propose a new tamper detection scheme for BTC-compressed images that can detect and locate the tamper areas in such images. Our proposed scheme achieved

high-quality images due to its use of the reference table, which was generated by using an extraction function. The experimental results show that our scheme obtained better visual quality of the embedded images than Hu et al.'s scheme irrespective of the size of blocks used in the AMBTC compression technique. For example, the image quality loss of the proposed scheme (0.004 dB) is much less than that of Hu et al.'s scheme (1.504 dB) when the *b* value is set to 2 and the value of *n* is 2. In addition, the proposed scheme achieves a clear tampered area. In the future, approaches capable of recovering the original information will be studied.

References

[1] P. G. Howarda and J. S. Vitter, "Parallel lossless image compression using Huffman and arithmetic coding," *Information Processing Letters*, vol. 59, no. 2, pp. 65-73, 1996. Article (CrossRef Link)

[2] Y. C. Hu and C. C. Chang, "A new lossless compression scheme based on Huffman coding scheme for image compression," *Signal Processing: Image Communication*, vol. 16, no. 4, pp. 367-372, 2000. Article (CrossRef Link)

[3] P. G. Howard and J. S. Vitter, "New methods for lossless image compression using arithmetic coding," *Information Processing & Management*, vol. 28, no. 6, pp. 765-779, 1992. Article (CrossRef Link)

[4] J. Wu, Z. Xu, G. Jeon, X. Zhang and L. Jiao, "Arithmetic coding for image compression with adaptive weight-context classification," *Signal Processing: Image Communication*, vol. 28, no.7, pp. 727-735, 2013. Article (CrossRef Link)

[5] E. J. Delp and O. R. Mitchell, "Image compression using block truncation coding," *IEEE Transactions on Communication*, vol. 27, no. 9, pp. 1335-1342, 1979. Article (CrossRef Link)

[6] M. Lema and O. R. Mitchell, "Absolute moment block truncation coding and its application to color images," *IEEE Transactions on Communication*, vol. 32, no. 10, pp. 1148-1157, 1984. Article (CrossRef Link)

[7] P. Fränti, O. Nevalainen and T. Kaukoranta, "Compression of digital images by block truncation coding: a survey," *The Computer Journal*, vol. 37, no. 4, pp. 308-332, 1994. Article (CrossRef Link)

[8] B. C. Dhara and B. Chanda, "Block truncation coding using pattern fitting," *Pattern Recognition*, vol. 37, no. 11, pp. 2131-2139, 2004. Article (CrossRef Link)

[9] B. C. Dhara and B. Chanda, "Color image compression based on block truncation coding using pattern fitting principle," *Pattern Recognition*, vol. 40, no. 9, pp. 2408-2417, 2007. Article (CrossRef Link)

[10] Y. C. Hu, B. H. Su and P. Y. Tsai, "Color image coding scheme using absolute moment block truncation coding and block prediction technique," *Imaging Science Journal*, vol. 56, no. 5, pp. 254-270, 2008. Article (CrossRef Link)

[11] P. Schelkens, A. Munteanu and J. Cornelis, "Wavelet-based compression of medical images: protocols to improve resolution and quality scalability and region-of-interest coding," *Future Generation Computer Systems*, vol. 15, no. 2, pp. 171-184, 1999. Article (CrossRef Link)

[12] V. Bruni, D. Vitulano, "Combined image compression and denoising using wavelets," *Signal Processing: Image Communication*, vol. 22, no. 1, pp. 86-101, 2007. Article (CrossRef Link)

[13] A. M. Rufai, G. Anbarjafari and H. Demirel, "Lossy image compression using singular value decomposition and wavelet difference reduction," *Digital Signal Processing*, vol. 24, pp. 117-123, January, 2014. Article (CrossRef Link)

[14] M. H. Horng, "Vector quantization using the firefly algorithm for image compression," *Expert Systems with Applications*, vol. 39, no. 1, pp. 1078-1091, 2012. Article (CrossRef Link)

[15] D. Tsolakis, George E. Tsekouras and J. Tsimikas, "Fuzzy vector quantization for image compression based on competitive agglomeration and a novel codeword migration strategy," *Engineering Applications of Artificial Intelligence*, vol. 25, no. 6, pp. 1212-1225, 2012. Article (CrossRef Link)

[16] W. N. Lie, T. C. I. Lin, S. L. Cheng, "Dual protection of JPEG images based on informed

embedding and two-stage watermark extraction techniques," *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 3, pp. 330-341, 2006. Article (CrossRef Link)
[17] T. Y. Lee and S. D. Lin, "Dual watermark for image tamper detection and recovery," *Pattern Recognition*, vol. 41, pp. 3497-3506, 2008. Article (CrossRef Link)
[18] F. Ahmed and M. Y. Siyal, "A secure and robust hash-based scheme for image authentication," *Signal Processing*, vol. 90, pp. 1456–1470, 2010. Article (CrossRef Link)
[19] J. C. Chuang and Y. C. Hu, "An adaptive image authentication scheme for vector quantization compressed image," *Journal of Visual Communication and Image Representation*, vol. 22, pp. 440-449, 2011. Article (CrossRef Link)
[20] Y. C. Hu, C. C. Lo, C. M. Wu, W. L. Chen, and C. H. Wen, "Probability-based tamper detection scheme for BTC-compressed images based on quantization levels modification," *International Journal of Security and Its Applications*, vol. 7, no. 3, pp. 11-32, 2013. Article (CrossRef Link)
[21] J. Serra, "Image analysis and mathematical morphology," *Academic Press, Inc. Orlando, FL, USA*, 1983. Article (CrossRef Link)
[22] X. Zhang and S. Wang, "Efficient steganographic embedding by exploiting modification direction," *IEEE Communication Letters*, vol. 10, no. 11, pp. 1-3, 2006. Article (CrossRef Link)
[23] Y. C. Hu, W. L. Chen, C. C. Lo, C. M. Wu, "A novel tamper detection scheme for BTC compressed images," *Opto-Electronics Review*, vol. 21, no. 1, pp. 137-146, 2013. Article (CrossRef Link)
[24] C. Qin, C. C. Chang, P. Y. Chen, "Self-embedding fragile watermarking with restoration capability based on adaptive bit allocation mechanism," *Signal Processing*, vol. 92, no. 4, pp. 1137-1150, 2012. Article (CrossRef Link)
[25] Y. C. Hu, C. C. Lo, W. L. Chen, C. H. Wen, "Joint image coding and image authentication based on absolute moment block truncation coding," *Journal of Electronic Imaging*, vol. 22, no. 1, pp. 1-12, January- March, 2013. Article (CrossRef Link)
[26] C. Qin, C. C. Chang, K. N. Chen, "Adaptive self-recovery for tampered images based on VQ indexing and inpainting," *Signal Processing*, vol. 93, no. 4, pp. 933-946, 2013. Article (CrossRef Link)

Thai-Son Nguyen received the bachelor degree in information technology from Open University, HCM city, Vietnam, in 2005. From December 2006, he has been lecturer of TraVinh University, TraVinh, Vietnam. In 2011, he received M.S. degree in computer sciences from FengChia University, TaiChung, Taiwan. He is currently pursuing the Ph.D. degree with the Department of Information Engineering and Computer Science, Feng Chia University, Taichung, Taiwan. His current research interests include data hiding, image processing, and information security.

Chin-Chen Chang received the B.S. degree in applied mathematics and the M.S. degree in computer and decision sciences from National Tsing Hua University, Hsinchu, Taiwan, R.O.C., in 1977 and 1979, respectively. He received the Ph.D degree in computer engineering from National Chiao Tung University, Hsinchu, in 1982. From July 1998 to June 2000, he was Director of the Advisory Office, Ministry of Education, R.O.C. From 2002 to 2005, he was a Chair Professor at National Chung Cheng University. From February 2005, he has been a Chair Professor at Feng Chia Univeristy. In addition, he was severed as a consultant to several research institutes and government departments. His current research interests include database design, computer cryptography, image compression, and data structures.

Ting-Feng Chung, He is currently pursuing the M.S. degree with the Department of Information Engineering and Computer Science, FengChia University, Taichung, Taiwan. His current research interests include data hiding, and image processing.