Joint image coding and image authentication based on absolute moment block truncation coding

Yu-Chen Hu
Chun-Chi Lo
Wu-Lin Chen
Chia-Hsien Wen
Joint image coding and image authentication based on absolute moment block truncation coding

Yu-Chen Hu
Providence University
Department of Computer Science and Information Management
200 Chung Chi Road, Taichung 43301, China
E-mail: ychu@pu.edu.tw

Chun-Chi Lo
Providence University
Department of Computer Science and Information Engineering
200 Chung Chi Road, Taichung 43301, China

Wu-Lin Chen
Chia-Hsien Wen
Providence University
Department of Computer Science and Information Management
200 Chung Chi Road, Taichung 43301, China

Abstract. A joint image coding and image authentication scheme based on absolute moment block truncation coding (AMBTC) is proposed. In the proposed scheme, the authentication data is generated by using the pseudo random sequence. Then, the authentication codes are embedded into the bit maps of AMBTC-compressed image blocks. The embedded bit maps and these quantization levels are further losslessly compressed to cut down the required storage cost. Experimental results demonstrate that the proposed scheme achieves good image quality of the embedded image while keeping good detecting accuracy. © 2013 SPIE and IS&T [DOI: 10.1117/1.JEI.22.1.013012]

1 Introduction

Along with the rapid growth of computer hardware and software, the use of digital images becomes more and more popular. Digital images are generally stored in compressed formats because the raw image files require a lot of storage cost. To cut down the storage cost of the digital images, some lossy image coding techniques, such as block truncation coding (BTC), vector quantization (VQ), sub-band coding, JPEG, and JPEG 2000, had been proposed. In addition, some lossless image coding techniques, such as Huffman coding, arithmetic coding, and JPEG/LS, had been proposed. The lossy compression techniques are often used to compress the general-purpose images. Special-purpose images, such as military images and medical images, are often compressed by using lossless image compression techniques.

Basically, BTC has the advantage of requiring very low computational cost for image compression. In addition, the reconstructed image quality of BTC is quite good. The major problem of BTC is that the compression ratio is low compared to state-of-art coding standards such as JPEG and JPEG 2000. Some improved methods had been proposed to cut down the bit rate of BTC. From the literature, some extended applications based on BTC such as edge detection, digital watermarking, and data hiding have been proposed.

In addition to image compression, research toward image integrity protection becomes more and more important in recent decades because digital images can be easily copied and modified by using image processing software. The traditional cryptography methods, such as advanced encryption standard (AES), message-digest algorithm 5 (MD5), and Ron Rivest, Adi Shamir and Leonard Adleman (RSA), can be used to protect the integrity of digital data. However, they are not suitable for the protection of digital images. When the traditional cryptograph method is employed to encrypt the digital image, the encrypted data is meaningless, and any changes from the encrypted data can be detected. Nevertheless, the tamper areas of the image cannot be located.

To protect the integrity of digital images, the concept of image authentication had been proposed. Generally, image authentication schemes can be classified into two categories: signature-based schemes and fragile watermark-based schemes. In a signature-based scheme, the image to be protected is processed by using the hash function and the hashed result is then encoded via the public key cryptosystem to generate the signature of the image. Then the signature of the image is stored with a trusted third party. When the image is to be authenticated, the signature that was stored with a third party is extracted and it is then...
compared to the other signature that is generated from the image to be authenticated to detect the tampered areas.

In a fragile watermark-based scheme, the watermark data is embedded into the image to be protected. The watermark data can be generated by using either the random number sequence or the image features extracted from the given image. No extra data are to be stored in a trust third party. When the image is to be authenticated, the watermark data are extracted from the image to verify the image integrity. The image quality of the embedded image and the detection accuracy are the major considerations of the fragile watermark-based approach.

In 2001, Lin and Chang proposed a semi-fragile watermarking scheme\textsuperscript{17} which can be resistant to lossy compression for JPEG images. Wong and Memon proposed the secret and public key image watermarking schemes\textsuperscript{18} for image authentication and ownership verification in 2001. Lie et al.\textsuperscript{19} proposed a compression-domain scheme that provides dual protection for JPEG images in 2006. Lee and Lin proposed the dual watermark for both image tamper detection and image recovery\textsuperscript{20} in 2008. Qi and Xin proposed a quantization-based semi-fragile watermarking scheme\textsuperscript{21} for image content authentication in 2011. In addition, Chung and Hu proposed an adaptive image authentication scheme for VQ compressed images in 2011.\textsuperscript{22} Basically, the fragile watermark-based approach cannot provide strong robustness against common image processing attacks. Some published image authentication techniques of this approach provide the robustness analysis against attacks. Some tampering attacks for image authentication are discussed.\textsuperscript{23} In addition, the analysis and design of secure watermark-based authentication system had been proposed in 2006.\textsuperscript{24}

From the literature, the compressed-domain image authentication schemes often embed the watermark data into the compressed codes of the digital images to protect the image integrity. The image quality may significantly degrade when a great deal of watermark data is embedded into the compressed images. In this paper, we design a joint image compression and image authentication scheme for grayscale images. The authentication data are embedded into the compressed trios of absolute moment block truncation coding (AMBTC) and then the compressed trios are losslessly compressed to cut down the required storage cost. Doing so incurs less image degradation than directly embedding the watermark data into the compressed images.

The rest of this paper is organized as follows. We will review some block truncation coding schemes in Sec. 2. Section 3 will present the proposed scheme including the authentication data generation procedure, the secret embedding procedure, the image compression and authentication data embedding procedure, and the tamper detection procedure. The experimental results will be discussed in Sec. 4. Finally, conclusions and discussions will be given in Sec. 5.

2 Review on Block Truncation Coding

Delp and Ritcell introduced the block truncation coding scheme in 1979,\textsuperscript{4} which is also called the moment-preserving block truncation coding (MPBTC) scheme. The BTC scheme has the advantage of requiring very little computational time. In addition, the absolute moment block truncation coding\textsuperscript{5} that preserves the sample mean and the sample first absolute central moment had been proposed. In this section, the MPBTC scheme will be described. Next, the AMBTC scheme will be introduced.

2.1 Moment Preservation Block Truncation Coding

In MPBTC, each grayscale image to be compressed is first divided into a set of nonoverlapped image blocks of \( n \times n \) pixels. Each \( n \times n \) image block can be viewed as an image vector of \( k \) dimensions, where \( k = n \times n \). Each image vector is sequentially processed in the order of left-to-right and top-to-down. In the image encoding procedure, the mean value (\( \bar{x} \)) and the standard deviation (\( \sigma \)) for each block \( x \) are calculated. All the pixels in the image block are classified into two groups according to its block mean value \( \bar{x} \). If the intensity of one pixel is less than or equal to \( \bar{x} \), it is classified as the first group \( (G_0) \). Otherwise, it is classified as the second group \( (G_1) \). A corresponding bit with value 0 or 1 is stored in the bit map (BM) when this pixel is classified as \( G_0 \) or \( G_1 \), respectively. Then, two quantization levels \( a \) and \( b \) for these two groups are computed according to the following equations, respectively.

\[
a = \bar{x} - \sigma \times \frac{q}{k - q},
\]

\[
b = \bar{x} + \sigma \times \frac{k - q}{q}.
\]

Here, \( q \) stands for the number of pixels whose values are greater than or equal to \( \bar{x} \).

In MPBTC, the quantization levels \( a \) and \( b \) are designed so that the first and the second sample moments of each image block are preserved. Each compressed image block produces a trio \( (a, b, BM) \). Each quantization level is stored in 8 bits. A total of \((8 + 8 + k)\) bits are needed to store the compressed codes \( (a, b, BM) \) of MPBTC.

The image decoding procedure of BTC is very simple. After receiving the compressed trio \( (a, b, BM) \) of each image block, the corresponding pixel is reconstructed by quantization level \( a \) if a corresponding bit valued 0 is found in the bit map. Otherwise, it is recovered by quantization level \( b \).

2.2 Absolute Moment Block Truncation Coding

In 1984, Lema and Mitchell proposed the AMBTC scheme for grayscale and color image compression. AMBTC is proved to be the minimal mean squared error BTC scheme when the block mean value is taken as the threshold value to classify the pixels in each image block into two groups. The basic image encoding/decoding procedures of AMBTC are the same as those of MPBTC. The major difference between AMBTC and MPBTC is the calculation of the two quantization levels. The details of the image encoding/decoding procedures of AMBTC are skipped.

In AMBTC, two quantization levels \( a \) and \( b \) for these two groups can be computed according to the following two equations, respectively.

\[
a = \frac{1}{k - q} \times \sum_{x_i \leq \bar{x}} x_i,
\]

\[
b = \frac{1}{k - q} \times \sum_{x_i > \bar{x}} x_i.
\]
\[ b = \frac{1}{q} \sum_{i \leq x} x_i, \quad (4) \]

Here, \( q \) denotes the number of pixels whose values are greater than or equal to \( \bar{x} \).

### 3 Proposed Scheme

The goal of the proposed scheme is to perform both image coding and image authentication based on AMBTC. To protect the image integrity, the authentication data are embedded into the bit map of each AMBTC compressed block. Then, the quantization levels and the bit map of each image block are compressed losslessly. The size of the authentication data for each image block can be selected according to the user requirement to comprise the embedded image quality and the accuracy of tamper detection. To improve the readability, Table 1 lists the abbreviations and the relations of the variables used in the proposed scheme.

#### 3.1 Authentication Data Generation Procedure

Suppose the grayscale image of \( W \times H \) pixels is to be processed. It is divided into \( n \times n \) nonoverlapping image blocks. A total of \( w \times h \) image blocks of \( n \times n \) pixels are divided where \( w = W/n \) and \( h = H/n \). Let \( eb \) denote the embedded bits of authentication data for each compressed image block. A total number of \( w \times h \) authentication data of \( eb \) bits will be generated. To generate the authentication data, the pseudo random number generator (PRNG) with a predefined seed is used to generate \( w \times h \) random values. Each random value \( rv \) is then converted to the authentication code (ac) by using the following equation:

\[ ac = rv \mod 2^{eb}. \quad (5) \]

Each authentication code will be embedded into the bit map of the corresponding compressed block in the image compression and the authentication data embedding procedure.

#### 3.2 Image Compression and Authentication Data Embedding Procedure

The grayscale image of \( W \times H \) pixels is divided into \( n \times n \) nonoverlapping image blocks. A total of \( w \times h \) image blocks of \( n \times n \) pixels are to be processed. Each image block \( x \) is first compressed by AMBTC as mentioned in Sec. 2.2 to generate the compressed trio \((a, b, BM)\). Here, \( a \) and \( b \) are two quantization levels for these two groups. In addition, \( BM \) denotes the bit map of \( n \times n \) bits. The reason why we choose AMBTC instead of MPBTC is that AMBTC provides better reconstructed image quality than MPBTC. To embed the authentication code \( ac \) into \( BM \), the \( BM \) is divided into \( eb \) equal-sized blocks. To simplify the bit map division process, the \( eb \) value is restricted to the factor of \( n \times n \).

After the bit map \( BM = [bm_1, bm_2, \ldots, bm_{n\times n}] \) is divided into \( eb \) equal-sized blocks, the authentication code of \( eb \) bits is to be embedded into the bit map. Each subdivided bit map will be used to embed 1-bit authentication data. Let \( gsize \) denote the size of the subdivided bit map where \( gsize = n \times n/eb \). Let \( SDBM_i \) denote the \( i \)’th subdivided bit map where \( 1 \leq i \leq eb \). The subdivided bit map \( SDBM_i \) of size \( gsize \) bits can be generated by the following equation:

\[
SDBM_i = [bm_{1+(i-1)\times gsize}, bm_{2+(i-1)\times gsize}, \ldots, bm_{gsize+(i-1)\times gsize}].
\]

(6)

An example to divide the bit map of \( 4 \times 4 \) bits into two equal-sized blocks is described in the following. The size of each subdivided bit map is 8 bits. In other words, \( gsize \) equals 8. The first subdivided bit map \( SDBM_1 = [bm_1, bm_2, \ldots, bm_8] \) can be generated by using Eq. (6). Similarly, the second subdivided bit map \( SDBM_2 = [bm_9, bm_{10}, \ldots, bm_{16}] \) can be generated.

Let \( ac_i \) denote the \( i \)’th bit of the authentication code to be embedded into \( SDBM_i \), where \( 1 \leq i \leq eb \). The parity of \( SDBM_i \) can be computed according to the following equation:

\[ p_i = \sum_{j=1}^{gsize} bm_{j+(i-1)\times gsize} \mod 2. \]

(7)

If \( p_i \) equals \( ac_i \), \( SDBM_i \) is unchanged. Otherwise, one candidate in \( SDBM_i \) will be selected and changed so that the parity value of the modified \( SDBM_i \) will equal \( ac_i \). To select the best candidate in \( SDBM_i \), the incurred squared Euclidean distance for each candidate is computed. Let \( x = \{x_1, x_2, x_3, \ldots, x_{n\times n}\} \) denote the original image block of \( n \times n \) pixels. It can be divided into \( eb \) equal-sized blocks of size \( gsize \) pixels. Let \( SDB_i \) denote the \( i \)’th subdivided image block where \( 1 \leq i \leq eb \). The subdivided block
SDIB<sub>j</sub> of gsize pixels can be generated by the following equation:

\[
SDIB_j = [x_{1+(i-1)\times gsize}, x_{2+(i-1)\times gsize}, \ldots, x_{\text{gsize}+(i-1)\times \text{gsize}}].
\] (8)

For the \( j \)’th pixel \( x_{j+(i-1)\times \text{gsize}} \) in SDIB\(_j\) satisfying that \( \text{bm}_{j+(i-1)\times \text{gsize}} \) equals 0 where \( 1 \leq j \leq \text{gsize} \), the incurred distortion to replace \( \text{bm}_{j+(i-1)\times \text{gsize}} \) by 1 can be computed as follows:

\[
\text{dist}_j = [x_{j+(i-1)\times \text{gsize}} - b]^2 - [x_{j+(i-1)\times \text{gsize}} - a]^2.
\] (9)

For the \( j \)’th pixel \( x_{j+(i-1)\times \text{gsize}} \) in SDIB\(_j\) satisfying that \( \text{bm}_{j+(i-1)\times \text{gsize}} \) equals 1, the incurred distortion to replace \( \text{bm}_{j+(i-1)\times \text{gsize}} \) by 0 can be computed as follows:

\[
\text{dist}_j = [x_{j+(i-1)\times \text{gsize}} - a]^2 - [x_{j+(i-1)\times \text{gsize}} - b]^2.
\] (10)

After computing the incurred distortions of these gsize candidates, the candidate with the least distortion is selected and the corresponding grouping information of this candidate is changed to embed \( a\). After the authentication code is embedded into the bit map of each compressed image block, the compressed code of each image block is a trio of \((a, b, \text{BM}')\). Here, \( \text{BM}' \) denotes the resultant bit map after authentication data embedding. The grouping information recording in the bit map for some pixels in each image block may be changed after the authentication data embedding process is performed. In other words, the original quantization levels may not still be the mean values of these two groups after bit map modification. To improve the image quality of the compressed image block, these two quantization levels are recomputed as follows:

\[
a' = \frac{1}{k-q} \times \sum_{x_i \in G_1} x_i
\] (11)

and

\[
b' = \frac{1}{q} \times \sum_{x_i \in G_0} x_i.
\] (12)

Here, \( q \) stands for the number of pixels belonging to \( G_1 \). Now, the compressed code of each image block is a trio of \((a', b', \text{BM}')\).

To cut down the storage cost of the compressed codes, the compressed trios are further losslessly compressed. Since the quantization levels and the bit maps have different properties, two different lossless approaches are used. To process the quantization levels, all these \( a' \) values of \( w \times h \) blocks are collected to form \( A \). In addition, the difference value between \( b' \) and \( a' \) of each compressed image block is computed and collected to form \( D \). The size of \( D \) is also of \( w \times h \) elements. Then, \( A \) and \( D \) are encoded by the two-stage lossless coding approach consisting of the linear prediction technique and the Huffman coding technique. First, the given data are processed by the linear prediction technique to generate the predicted errors. The following prediction function is used to generate the predicted value for the currently processing value \( v \):

\[
f(v) = (I + u)/1,
\] (13)

where \( l \) and \( u \) denote the adjacent left and up values of \( v \), respectively. Then, these prediction errors are compressed by using the Huffman coding technique. The reason why the two-stage lossless approach is employed to encode the quantization levels is that the neighboring elements in both \( A \) and \( D \) are quite similar.

To cut down the storage cost of the bit maps for the compressed image blocks, all the bit maps are collected and then compressed by using the Huffman coding technique. Since the similarity among neighboring bit maps is quite low, the two-stage lossless coding approach for the compression of \( A \) and \( D \) is not suitable for the compression of the bit maps.

\[\text{Fig. 2} \quad \text{Four test cases for tamper refinement.}\]
3.3 Tamper Detection Procedure

The goal of the tamper detection procedure is to detect whether the AMBTC compressed image is modified or not. Some system parameters, such as $W$, $H$, $n$, $eb$, and random seed, should be known. Before the block-based tamper detection process is executed, we need to reconstruct the compressed trio $(a, b, BM)$ of each image block. First, quantization level $a$ of each compressed image block can be reconstructed by performing the Huffman decoding procedure followed by the reverse linear prediction technique on the received compressed codes of $A$. Similarly, the difference value between the two quantization levels of each compressed image block can be reconstructed by performing the Huffman decoding procedure followed by the reverse linear prediction technique on the received compressed codes of $D$. Then, quantization level $b$ of each compressed image block can be computed. Finally, the bit map BM of each compressed block can be recovered by performing the Huffman decoding procedure on the received compressed codes of bit maps.

To determine whether these $w \times h$ image blocks are tampered or not, two sets of authentication codes are to be generated. The first set of authentication codes of $eb$ bits is generated by using the random values induced with the random seed. To generate the first set of the authenticated codes, a total number of $w \times h$ random values are generated by using the random number seed. Each random value $r_v$ is converted into the authentication code $ac$ of $eb$ bits by Eq. (5). The set of the authentication code is the same as that generated by using the authentication data generation procedure as mentioned in Sec. 3.1.

The second set of the authentication codes will be generated by using the bit maps. Each bit map of $n \times n$ bits is divided into equal-sized blocks of $gsze$ bits where $gsze$ equals $(n \times n)/eb$. The parity value of each subdivided image block is computed. These parity values of $eb$ subblocks are collected to form the extracted authentication code (eac) of each image block.

When two sets of the authentication codes are available, we can determine whether each image block $x$ is tampered or not. If $ac$ equals $eac$, each image block $x$ is classified as a clear block. Otherwise, $x$ is classified as a modified block. The flag values 0 and 1 are used to represent the clear and modified blocks, respectively. The flowchart of the block-based tamper detection procedure is shown in Fig. 1.

<table>
<thead>
<tr>
<th>Table 2</th>
<th>Reconstructed image qualities of AMBTC with different block sizes.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block size</td>
<td>2 × 2</td>
</tr>
<tr>
<td>Images</td>
<td>(5 bpp)</td>
</tr>
<tr>
<td>Airplane</td>
<td>40.695</td>
</tr>
<tr>
<td>Girl</td>
<td>41.862</td>
</tr>
<tr>
<td>Lenna</td>
<td>40.615</td>
</tr>
<tr>
<td>Pepper</td>
<td>41.404</td>
</tr>
<tr>
<td>Tiffany</td>
<td>43.110</td>
</tr>
<tr>
<td>Zelda</td>
<td>44.207</td>
</tr>
<tr>
<td>Average</td>
<td>41.982</td>
</tr>
</tbody>
</table>
The detected result needs to be further refined because it is possible that some modified bit maps may have the same eb-bit parity values as the original bit maps. The possibility of this condition decreases as the eb value increases. To improve the detecting accuracy, an iterative refinement mechanism is employed to refine the roughly detected image of size $w \times h$. In each round, we need to check whether each white pixel will be changed to a black one or not. Here, the white pixels and the black pixels stand for the clear and modified blocks, respectively. Four test cases as shown in Fig. 2 are sequentially checked to decide whether each white pixel will be changed to a black one or not. In Fig. 2, $p$ denotes the selected white pixel to be processed. In the first case in Fig. 2(a), if the adjacent up and down pixels of $p$ are black, $p$ is changed to a black pixel. In the second case in Fig. 2(b), if the adjacent left and right pixels of $p$ are black, $p$ is changed to a black one. Similarly, two additional cases for the 45 deg and 135 deg splay black pixels of $p$ are listed in Fig. 2(c) and 2(d).

After checking each white pixel to determine whether it should be changed to a black one or not, the total number of modified pixels in each round is calculated. If the number of modified pixels is greater than or equal to one, the same refinement process is iterated. Otherwise, the tamper refining process is stopped. The above process does not guarantee to remove the white pixels within the tampered objected. To solve the problem, an additional process to remove the small-sized connected component of white pixels within the tampered area is executed.

4 Experimental Results

Our experiments are performed on Windows XP PC with an Intel Core Duo 2.2 GHz CPU and 512 MB RAM. The testing programs are implemented using Bloodshed Dev C++. In our experiments, six grayscale images of $512 \times 512$ pixels, as shown in Fig. 3, are used. Results of the reconstructed image qualities of AMBTC with different block sizes are listed in Table 2. It is shown that the reconstructed image quality decreases with the increase of the block size. Average image qualities of 41.982, 34.915, and 31.341 dB are obtained when the block sizes are set to $2 \times 2$, $4 \times 4$, and $8 \times 8$, respectively. The bit rates of AMBTC equal 5, 2, and 1.25 bpp when the block sizes are set to $2 \times 2$, $4 \times 4$, and $8 \times 8$, respectively. We find that the reconstructed

Table 3

<table>
<thead>
<tr>
<th>Images</th>
<th>$eb = 1$</th>
<th>Bit rate</th>
<th>PSNR</th>
<th>Bit rate</th>
<th>PSNR</th>
<th>Bit rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airplane</td>
<td>35.633</td>
<td>3.178</td>
<td>32.926</td>
<td>3.201</td>
<td>31.692</td>
<td>3.186</td>
</tr>
<tr>
<td>Girl</td>
<td>37.540</td>
<td>3.330</td>
<td>35.058</td>
<td>3.365</td>
<td>33.774</td>
<td>3.353</td>
</tr>
<tr>
<td>Lenna</td>
<td>38.030</td>
<td>3.284</td>
<td>35.654</td>
<td>3.308</td>
<td>34.077</td>
<td>3.287</td>
</tr>
<tr>
<td>Pepper</td>
<td>38.157</td>
<td>3.218</td>
<td>35.644</td>
<td>3.233</td>
<td>34.140</td>
<td>3.211</td>
</tr>
<tr>
<td>Tiffany</td>
<td>37.023</td>
<td>3.056</td>
<td>34.106</td>
<td>3.078</td>
<td>33.161</td>
<td>3.068</td>
</tr>
<tr>
<td>Zelda</td>
<td>40.507</td>
<td>3.080</td>
<td>37.791</td>
<td>3.102</td>
<td>36.464</td>
<td>3.087</td>
</tr>
<tr>
<td>Average</td>
<td>37.815</td>
<td>3.191</td>
<td>35.197</td>
<td>3.215</td>
<td>33.885</td>
<td>3.199</td>
</tr>
</tbody>
</table>

Table 4

<table>
<thead>
<tr>
<th>Images</th>
<th>$eb = 1$</th>
<th>Bit rate</th>
<th>PSNR</th>
<th>Bit rate</th>
<th>PSNR</th>
<th>Bit rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airplane</td>
<td>32.724</td>
<td>1.385</td>
<td>31.794</td>
<td>1.398</td>
<td>29.984</td>
<td>1.427</td>
</tr>
<tr>
<td>Girl</td>
<td>34.380</td>
<td>1.364</td>
<td>33.528</td>
<td>1.386</td>
<td>31.931</td>
<td>1.431</td>
</tr>
<tr>
<td>Lenna</td>
<td>33.653</td>
<td>1.386</td>
<td>33.156</td>
<td>1.400</td>
<td>31.900</td>
<td>1.429</td>
</tr>
<tr>
<td>Pepper</td>
<td>33.960</td>
<td>1.406</td>
<td>33.217</td>
<td>1.419</td>
<td>31.729</td>
<td>1.448</td>
</tr>
<tr>
<td>Tiffany</td>
<td>35.689</td>
<td>1.367</td>
<td>34.783</td>
<td>1.378</td>
<td>32.708</td>
<td>1.398</td>
</tr>
<tr>
<td>Zelda</td>
<td>36.481</td>
<td>1.337</td>
<td>35.881</td>
<td>1.355</td>
<td>34.496</td>
<td>1.395</td>
</tr>
<tr>
<td>Average</td>
<td>34.481</td>
<td>1.374</td>
<td>33.727</td>
<td>1.389</td>
<td>32.125</td>
<td>1.421</td>
</tr>
</tbody>
</table>
image quality of AMBTC is poor when the block size is set to 8×8.

Table 3 lists the results of the proposed scheme when the block size is set to 2×2. It is shown that the image quality of the embedded image rapidly decreases with the increase of the eb value. Average embedded image qualities of 37.815, 35.197, and 33.885 dB are obtained when the eb values are set to 1, 2, and 4, respectively. Average bit rates of 3.191, 3.215, and 3.199 bpp are achieved when the eb values are set to 1, 2, and 4, respectively. It is suggested that the eb value should be set to 1 when the block size of 2×2 is to be used in the proposed scheme.

Table 4 lists the results of the embedded image of the proposed scheme when the block size is set to 4×4. Average embedded image qualities of 34.481, 33.727, and 32.125 dB are obtained when the eb values are set to 1, 2, and 4, respectively. Average bit rates of 1.374, 1.389, and 1.421 bpp are achieved when the eb values are set to 1, 2, 4, 8, and 12, respectively. It is suggested that the eb value should be less than or equal to 4. Some embedded images of the proposed scheme when the block size is set to 4×4 are listed in Fig. 4. The visual qualities of these embedding images are quite good.

In the simulations, the block size is set to 4×4 and two eb values 1 and 2 are tested. In the first tamper test, a fish as shown in Fig. 5(a) is added on the shoulder of each embedded image. Two tampered images with the eb values equal to 1 and 2 are shown in Fig. 5(b) and 5(c), respectively. In this test, 5660 pixels are modified when the fish is added. A total number of 389 blocks of 4×4 pixels is affected. The pixel difference images and the block difference images for the first tamper test are listed in Fig. 6.

The detected results of the proposed tamper detection procedure are listed in Fig. 7. The roughly detected images when the eb values are set to 1 and 2 are shown in Fig. 7(a) and 7(d), respectively. It is shown that the number of the white spots within the modified area decreases when the eb value increases. That is because the probability that

![Fig. 4 Some embedded images of the proposed scheme when the block size equals 4×4.](image)

![Fig. 5 The first tamper example.](image)
Fig. 6 The difference images for the first tamper test.

(a) Pixel difference image \((eb=1)\)  
(b) Pixel difference image \((eb=2)\)

(c) Block difference image \((eb=1)\)  
(d) Block difference image \((eb=2)\)

Fig. 7 The detected images for the first tamper test.

(a) Roughly detected image \((eb=1)\)  
(b) Detected image \((eb=1)\)  
(c) Error image \((eb=1)\)

(d) Roughly detected image \((eb=2)\)  
(e) Detected image \((eb=2)\)  
(f) Error image \((eb=2)\)
the original bit map and the modified bit map have the same eb-bit remainders decreases when the eb value increases.

The detected results of the proposed tamper detection procedure with the iterative refinement mechanism when the eb values equal 1 and 2 are shown in Fig. 7(b) and 7(e), respectively. No white spots are found within the modified area in these two detected images. Compared to the difference image as shown in Fig. 6, the tampered area is clearly detected. To understand the performance of the proposed scheme, the error images between the tampered objects and the detected images when the eb values equal 1 and 2 are listed in Fig. 7(c) and 7(f), respectively. It is shown that only these modified blocks in the boundary of the tampered area cannot be correctly detected by using the proposed tamper detection procedure.

In the second tamper test, a rose as shown in Fig. 8(a) is added on the shoulder of each embedded image. Two tampered images with the eb values equal to 1 and 2 are shown in Fig. 8(b) and 8(c), respectively. In this test, 7120 pixels are modified when the rose is added. A total number of 495 blocks of 4x4 pixels is affected. The pixel difference images and the block difference images for the second tamper test are listed in Fig. 9.

The detected results of the proposed tamper detection procedure are listed in Fig. 10. The roughly detected images when the eb values are set to 1 and 2 are shown in

![Fig. 8](image_url) The second tamper example.

![Fig. 9](image_url) The difference images for the second tamper test.
Fig. 10(a) and 10(d), respectively. Similarly, it is shown that the number of the white spots within the modified area decreases when the eb value increases. The refined results of the proposed tamper detection procedure when the eb values equal 1 and 2 are shown in Fig. 10(b) and 10(d), respectively. The tampered area is clearly detected. The error images between the tampered objects and the detected images when the eb values equal 1 and 2 are listed in Fig. 10(c) and 10(f), respectively. Some modified blocks in the boundary of the tampered area cannot be detected by using the proposed tamper detection procedure.

The analysis of the detecting accuracy for these two tamper tests is listed in Table 5. The results of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) are provided for these tests. There are 16,384 image blocks for each 512×512 image when the block size is set to 4×4. Recall that 389 and 495 blocks are affected when a fish and a rose are added into the embedded images in the first and second tamper tests, respectively. According to the results, 363 and 378 tampered blocks are detected for the first tamper test when the eb values are set to 1 and 2, respectively. In addition, 462 and 480 tampered blocks are found for the second tamper test when the eb values are set to 1 and 2, respectively.

In the first tamper test, 26 and 14 modified blocks are not correctly detected when the eb values are 1 and 2, respectively. According to the error images shown in Fig. 7(c) and 7(f), they exist in the boundary of the tampered object. In addition, two and five clear blocks are mistakenly detected as modified blocks due to the tamper refinement strategy. Similarly, 34 and 19 modified blocks are not correctly detected in the second tamper test when the eb values are 1 and 2, respectively. They are located in the boundary of the tampered object according to the error images in Fig. 10(c) and 10(f).

### Table 5 Analysis of these two tamper tests (unit: pixel).

<table>
<thead>
<tr>
<th>Test</th>
<th>First tamper test</th>
<th>Second tamper test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Factors</td>
<td>eb = 1</td>
<td>eb = 2</td>
</tr>
<tr>
<td>TP</td>
<td>15995</td>
<td>15992</td>
</tr>
<tr>
<td>TN</td>
<td>361</td>
<td>373</td>
</tr>
<tr>
<td>FP</td>
<td>26</td>
<td>14</td>
</tr>
<tr>
<td>FN</td>
<td>2</td>
<td>5</td>
</tr>
</tbody>
</table>

5 Conclusions and Discussions

A joint image coding and tamper detection scheme based on AMBTC is proposed in the paper. In the proposed scheme, the number of embedded bits for each bit map can be determined by users to reach a compromise between the embedded image quality and the detection accuracy in the proposed scheme. From these two tamper tests, the image qualities of the embedded images decrease when the number of the embedded bits increases. The clear shapes of the tampered objects can be detected for the tampered images. To provide good image qualities of the embedded images, we suggest that the number of embedded bits should be set to 1 in the proposed scheme. The proposed scheme can...
be used to detect the tamper areas based on the object insertion. But it is not robust to the common image processing attacks.

The reason why the authentication codes are embedded into the bit maps instead of the quantization levels of AMBTC compressed codes is that a slight modification on the quantization levels will cause significant image degradation of the embedded images. The use of the random number sequence induced by the specific random seed to generate the authentication codes is very simple and secure. Without the same random seed, the authentication codes cannot be generated by the illegal users. To be specific, the proposed framework for image authentication can be easily extended to the proposed image coding schemes based on BTC.

Acknowledgments
This research was supported by Providence University, Taichung, Taiwan, under contract the National Science Council, Taipei, China, under contract NSC 99-2221-E-126-004-MY2.

References