Mathematical Biosciences and Engineering

MBE, 16(6): 7934–7949.
DOI: 10.3934/mbe.2019399
Received: 29 April 2019
Accepted: 14 July 2019
Published: 29 August 2019

http://www.aimspress.com/journal/MBE

Research article

AMBTC based high payload data hiding with modulo-2 operation and Hamming code

Li Li[1], Min He[1], Shanqing Zhang[1,*], Ting Luo[2], and Chin-Chen Chang[3]

[1] School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China
[2] College of Science and Technology, Ningbo University, Ningbo 315000, China
[3] Department of Information Engineering and Computer Science, Feng Chia University, Taichung 40724, Taiwan

* **Correspondence:** Email: sqzhang@hdu.edu.cn; Tel: +8618668196769.

Abstract: An efficient data hiding method with modulo-2 operation and Hamming code (3, 2) based on absolute moment block truncation coding (AMBTC) is proposed. In order to obtain good data hiding performance, different textures are assigned to different embedding strategies. The AMBTC compressed codes are divided into smooth and complex blocks according to texture. In the smooth block, the secret data and the four most significant bits plane of the two quantization levels are calculated using modulo-2 operation to replace the bitmap in order to improve the security of data transmission. Moreover, Hamming code (3, 2) is used to embed the two additional secret bits in the three significant bits planes of the two quantization levels. In the complex block, one secret bit is embedded by swapping the order of two quantization levels and flipping the bitmap. Experimental results show that the proposed method achieves higher capacity than the existing data hiding methods and maintains good visual quality.

Keywords: data hiding; AMBTC; modulo-2 operation; hamming code; high payload

1. Introduction

With the rapid development of network and multimedia technologies, illegal attackers can easily grab, tamper or copy secret data during transmission. Data hiding technologies embed secret data into the digital media without attracting attacker's attention for disguise, which is an efficient way to

protect data [1–6]. There are spatial domain based [7–12], frequency domain based [13,14] and compressed domain based [15–17] data hiding methods. Least significant bit (LSB) replacement is a classical spatial domain based data hiding method with little visual distortion [18]. However, in reality, compressed image format is often used for image transmission due to the limitation of storage space and bandwidth. If data are embedded in spatial domain, and the stego image is compressed, it will cause the loss of the secret data. As for the frequency domain method, the original images are transformed into frequency format for data embedding, such as Discrete Fourier Transformation (DFT) [19], Discrete Cosine Transformation (DCT) [20]. This kind of method is more robust against compression compared with the spatial domain based data hiding methods. The compressed domain baseddata hiding method embeds secret data by modifying the compressed code. Most of data hiding methods use redundancy in the image to embed data, and however, the compressed image greatly decreases redundancy. Thus, the compressed domain baseddata hiding method has a smaller payload compared with the spatial domain based data hiding method. However, since the image was usually stored or transmitted with the compressed format so that the compressed domain based data hiding method attracted researchers. Some compressed domain baseddata hiding methods embedded data into vector quantization (VQ) format [21], joint photographic experts group (JPEG) format [22], and absolute moment block truncation coding (AMBTC) format [23–28] images. Compared to VQ and JPEG, AMBTC requires less computing cost but achieves acceptable image quality. Therefore, it is more suitable for real-time and low-power applications such as portable digital devices. Due to the increasing demand for low-cost image compression, several data hiding methods for AMBTC codes were proposed.

In 2006, Chuang et al. [23] proposed a high payload data hiding method based on BTC images. Specifically, each block is classified as a smooth type or a complex type, and secret data are embedded into the bitmap of the selected smooth blocks. In 2015, Ou and Sun [24] presented a data hiding method based on AMBTC, and the bitmaps of smooth blocks were used to embed the data. Moreover, two quantization levels in the smooth block were recalculated to reduce the image distortion. However, recalculation needs the original image block, which decreases the scope of data hiding application. In 2016, Bai and Chang [25] used a two-phase strategy to embed the data into AMBTC compressed images by using matrix encoding. The Hamming code (7, 4) was performed on the quantization levels and the bitmap to embed the data in the first and second phases, respectively. However, modification of the bitmaps in complex blocks will cause visual distortion, especially in edge blocks. In 2018, Kim and Shin [26] proposed an efficient reversible data hiding algorithm based on the histogram modification of AMBTC-compressed images. However, they could not decode properly after embedding data. In 2019, Kumar et al.'s [27] used two thresholds to classify three categories of AMBTC compressed images. This method increased embedding capacity without much improvement on image quality. At the same time, Kumar et al.'s [28] used adaptive quantization and dynamic bit plane to embed data into AMBTC compressed images by using bit-replacement strategy. Although this method improved the embedding capacity, the lengths of segments in the compression code were variable after embedding different secret data, which could not maintain 2 bits per pixel the same as the original AMBTC had.

Considering embedding capacity and visual quality, we propose a data hiding method by modifying the quantization level and the bitmap of AMBTC compressed codes to minimize the distortion between the original AMBTC image and the stego image. Firstly, an image is divided into non-overlapping blocks, and these blocks are classified into smooth blocks and complex blocks. The

smooth blocks are combined with two quantization levels, and secret data are embedded by using modulo-2 operation. Hamming code (3, 2) is used to embed extra two bits of the secret data into the three least significant bits of two quantization levels. For the complex blocks, lossless embedding is used to embed one bit of secret data by swapping the orders of two quantization levels. Experimental results demonstrate that the proposed method achieves higher payload and better image quality compared with the existing data hiding methods.

The rest of this paper is organized as follows. Section 2 gives the introduction of related works. The proposed method is described in detail in Section 3. The experimental results are analyzed in Section 4. Section 5 gives the conclusion.

2. Related works

The AMBTC compression technique, Hamming code (3, 2) and modulo-2 operation are briefly introduced in this section.

2.1. Absolute Moment Block Truncation Coding (AMBTC)

AMBTC is a lossy compression algorithm for grayscale or color images. AMBTC, which was proposed by Lema and Michell [29] in 1984, is a modified version of Block Truncation Coding (BTC) [30]. It requires less computing time than BTC and can further improve the performance of traditional BTC technology. Take the gray image as an example, firstly, the image is divided into non-overlapping blocks with a size of $n \times n$. The average value of pixels in each block is calculated as:

$$\bar{x} = \frac{1}{W \times H} \sum_{i=1}^{N} x_i \tag{1}$$

where x_i represents the i^{th} pixel value, N is the total number of pixels, and W and H represent the width and height of each block, respectively. The bitmap B of AMBTC is constructed as:

$$B_i = \begin{cases} 1 & if\, x_i \geq \bar{x} \\ 0 & if\, x_i < \bar{x} \end{cases} \tag{2}$$

Equations (3) and (4) are used to calculate the higher quantization level a and lower quantization level b of each block.

$$a = \left\lfloor \frac{1}{t} \sum_{x_{i \geq \bar{x}}} x_i \right\rfloor \tag{3}$$

$$b = \left\lfloor \frac{1}{(W \times H) - t} \sum_{x_{i < \bar{x}}} x_i \right\rfloor \tag{4}$$

where t is the number of pixels whose values are greater than $\bar{x}$. When decoding, each "1" in the bitmap B is replaced by a, and each "0" is replaced by b, as shown in Eq (5).

$$g_i = \begin{cases} a & if\ (B_i = 1) \\ b & if\ (B_i = 0) \end{cases} \tag{5}$$

where g_i represents the pixel of the reconstructed grayscale image.

Figure 1 shows an example of the encoding and decoding processes of AMBTC. Let $x_i = \{150, 151, 148, 150;\ 146, 147, 147, 148;\ 146, 147, 144, 143;\ 144, 143, 141, 143\}$. The average value $\bar{x}$ of x_i is 146.13. Therefore, we have $a = 148$, $b = 143$, and $B_i = \{1111;\ 1111;\ 1100;\ 0000\}$. To construct $x_i^{'}$, 0 and 1 in B_i are replaced by 148 and 143, respectively. Therefore, the reconstructed image block $x_i^{'} = \{148, 148, 148, 148;\ 148, 148, 148, 148;\ 148, 148, 143, 143;\ 143, 143, 143, 143\}$.

$$\begin{bmatrix} 150 & 151 & 148 & 150 \\ 146 & 147 & 147 & 148 \\ 146 & 147 & 144 & 143 \\ 144 & 143 & 141 & 143 \end{bmatrix} \xrightarrow{\bar{x} = 146.13} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \xrightarrow[b = 143]{a = 148} \begin{bmatrix} 148 & 148 & 148 & 148 \\ 148 & 148 & 148 & 148 \\ 148 & 148 & 143 & 143 \\ 143 & 143 & 143 & 143 \end{bmatrix}$$

Figure 1. An illustration of AMBTC compression method. (a) Original image block (b) Bitmap (c) Reconstructed image block.

2.2. Hamming code (3, 2)

Hamming code (3, 2) [31] is used in the proposed method. Its parity-check matrix H is shown in Eq (6).

$$H = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} \tag{6}$$

The check value $E(P)$ of 3-bit code $P = (p_1, p_2, p_3)$ can be obtained from Eq (7).

$$E(P) = (H \times P^T)^T mod\ 2 \tag{7}$$

For example, if the received codeword is $P = (1,0,1)$ and the check value $E(P) = (1,0)$ is obtained according to Eq (7), which denotes that the second bit is wrong during transmission. The receiver only needs to flip the second bit to get the correct codeword $P^{'} = (1,1,1)$.

2.3. Modulo-2 operation

Modulo-2 operations [32] include modulo-2 addition, modulo-2 subtraction, modulo-2 multiplication and modulo-2 division. Modulo-2 multiplication and modulo-2 division are used in the proposed method. The modulo-2 multiplication is used to deal with the intermediate results by using modulo-2 addition. The modulo-2 division is actually an XOR operation. The relationship between modular division and multiplication is shown in Eqs (8) and (9).

$$S \div K = q \cdots\cdots r \tag{8}$$

$$K \times q + r = S \tag{9}$$

where S represents the dividend, K represents the divisor, q represents the quotient, and r represents the remainder. In addition, the divisor K has the highest bit of 1, and the length of the remainder r is one bit less than the divisor K. Figure 2 shows an example of modulo-2 multiplication and modulo-2 division.

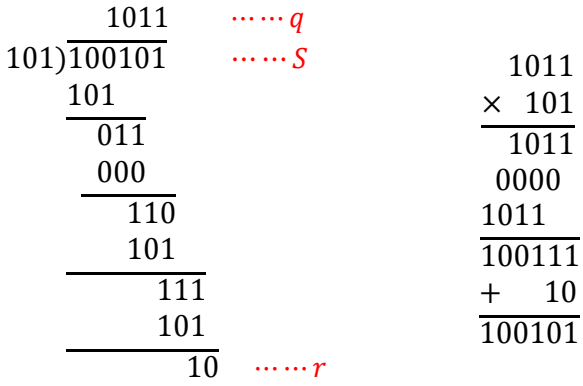

Figure 2. An illustration of modulo-2 operations. (a) Modulo-2 division. (b) Modulo-2 multiplication.

3. Proposed method

In this paper, an AMBTC based high payload data hiding with modulo-2 operation is proposed. The blocks of the AMBTC compressed codes are classified as smooth blocks and complex blocks. In the smooth block, firstly, the secret data and the four most significant bits of two quantization levels are computed using modulo-2 division, and the obtained quotient and remainder replace the bitmap. Then, the three least significant bits of the two quantization levels are used to embed the two additional secret bits with Hamming code (3, 2), respectively. If the modified quantization level is still within the range of the smooth block, the original quantization level will be replaced by the modified quantization level. In the complex block, a secret bit is embedded by swapping the order of two quantization levels and flipping the bitmap. The embedding strategy of the complex blocks can further improve the capacity of the proposed method without causing any distortion.

3.1. Embedding method

In this section, secret data are embedded using modulo-2 division and Hamming code (3, 2) to modify the bitmap and quantization levels of AMBTC compressed codes. The flowchart of the embedding process is described in Figure 3.

Input: Original grayscale image GI with a size of $W \times H$, threshold Thr and secret data $S = (s_1, s_2, \ldots, s_n)$.

Output: Stego AMBTC compressed codes SI, and matrix Y with a size of $W \times H$.

Step 1. GI is divided into $n \times n$ non-overlapping image blocks, and n is usually set to 4.

Step 2. The compressed trio (a, b, B) of the AMBTC compressed codes is obtained according to Eqs (1) and (2), where a and b are the higher quantization level and the lower quantization level,

respectively, and B is the bitmap.

Step 3. Compute the absolute difference value of two quantization levels a and b.

Step 4. If $|a-b| \leq Thr$, it is a smooth block. For the smooth block, modulo 2 division and Hamming code (3, 2) are used to embed secret data. There are two main embedding phases, as shown below.

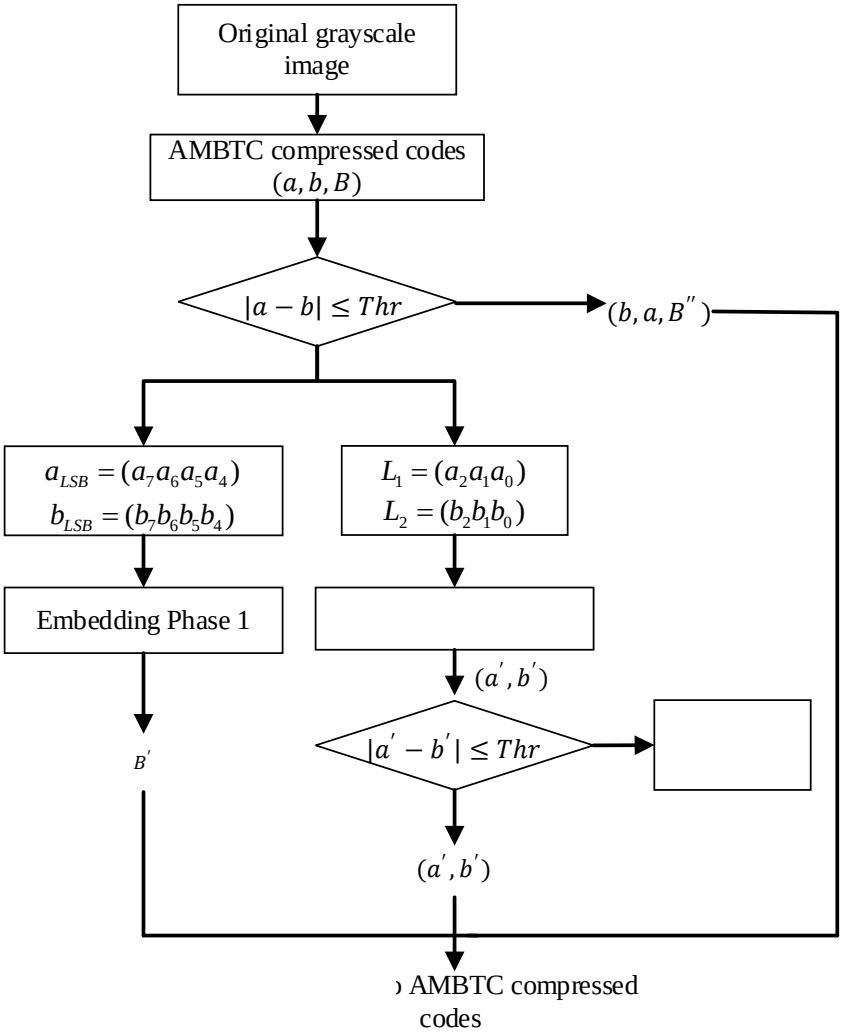

Figure 3. The flowchart of data embedding process.

Embedding phase 1:

Step 4.1. For the smooth block SB, the binary representations of the higher quantization level a and the lower quantization level b are set as $a = (a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0)_2$ and $b = (b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0)_2$, respecively. The four most significant bits of a and b are obtained according to Eqs (10) and (11), respectively. Then, the vector K is obtained by combing a_{LSB} and b_{LSB} as $K = (a_{\text{LSB}} \parallel b_{\text{LSB}}) = (a_7 a_6 a_5 a_4 b_7 b_6 b_5 b_4)$. If $a_7 = 0$, then $a_7 = 1$.

$$a_{LSB} = (a_7 a_6 a_5 a_4) \tag{10}$$

$$b_{LSB} = (b_7 b_6 b_5 b_4) \tag{11}$$

Step 4.2. Sixteen secret bits $S_0 = (s_1, s_2, \dots, s_{16})$ and row vector K are encoded by Eq (8) to obtain the 9 bits of quotient $q = (q_1, q_2, \dots, q_9)$ and 7 bits of remainder $r = (r_1, r_2, \dots, r_7)$. The quotient q and the remainder r are combined into a new 16 bits bitmap from $B' = (qr)$. The bitmap B of the original block is directly replaced by the new bitmap B'.

Embedding phase 2:

Step 4.3. The three least significant bits of a and b in the smooth block are obtained according to Eqs (12) and (13), respectively. Let secret data bits embed into a and b be $S_1 = (s_1, s_2)$ and $S_2 = (s_3, s_4)$, respectively.

$$L_1 = (a_2 a_1 a_0) \tag{12}$$

$$L_2 = (b_2 b_1 b_0) \tag{13}$$

Step 4.4. Calculate the check value $n_i = (H \times {L_i}^T)^T mod\ 2$ according to Eq (7), where $i = 1, 2$. Then $y_i = n_i \oplus S_i$. Next y_i is converted to decimal number to get d_i. The three least significant bits L_i' are obtained by inverting the d_i in L_i.

Step 4.5. The new two quantization levels a' and b' are obtained by combining L_i' with left bits of a and b. If $|a' - b'| \le Thr$, the modified block still belongs to the smooth block, and the new bitmap (a', b', B') is obtained, and $Y(SB) = 1$. Otherwise, the bitmap (a, b, B') is obtained, and $Y(SB) = 0$.

Step 5. If $|a - b| > Thr$, it is a complex block. For complex blocks, a lossless embedding algorithm is proposed. One secret bit $S_3 = (s_1)$. If $s_1 = 1$, the order of the two quantization levels is swapped to get (b, a), and $B'' = 1 - B$. If $s_1 = 0$, the order of two quantization levels (a, b) is unchanged.

Step 6. Repeat Steps 3–5 until all image blocks are processed. Then, the stego AMBTC compressed codes SI is constructed. The compressed codes SI consist of $(W \times H)/(4 \times 4)$ trios, which contains two quantization levels and one bitmap.

Figure 4 shows an example of constructing a stego AMBTC compressed codes.

3.2. Extraction and reconstruction method

The processes of secret data extraction are organized as in Figure 5.

Input: Stego AMBTC compressed codes SI, matrix Y of size $W \times H$, and threshold Thr.

Output: Secret data $S = (s_1, s_2, \dots, s_n)$.

Step 1. For each trio (a, b, B) in SI, compute the absolute difference between value of a and b.

Step 2. If $|a - b| \le Thr$, it is a smooth block SB, and two extraction phases are used.

Extraction phase 1:

Step 2.1. The first 9 bits of bitmap B are retrieved to get quotient $q' = (B_1, B_2, \dots, B_9)$ and the last 7 bits are extracted to get the remainder $r' = (B_{10}, B_{11}, \dots, B_{16})$.

Step 2.2. The four most significant bits of a and b are obtained using Eqs (10) and (11),

respectively. Then $K = (a_{\text{LSB}} \parallel b_{\text{LSB}}) = (a_7 a_6 a_5 a_4 b_7 b_6 b_5 b_4)$. If $a_7 = 0$, then set $a_7 = 1$.

Step 2.3. The quotient q', the remainder r' and the row vector K are calculated using Eq (8) to obtain the embedded secret data $S_0' = K \times q' + r' = (s_1, s_2, \ldots, s_{16})$.

Extraction phase 2:

Step 2.4. If $Y(SB) = 1$, L_1' and L_2' are obtained by the three least significant bits a and b by using Eqs (12) and (13), respectively.

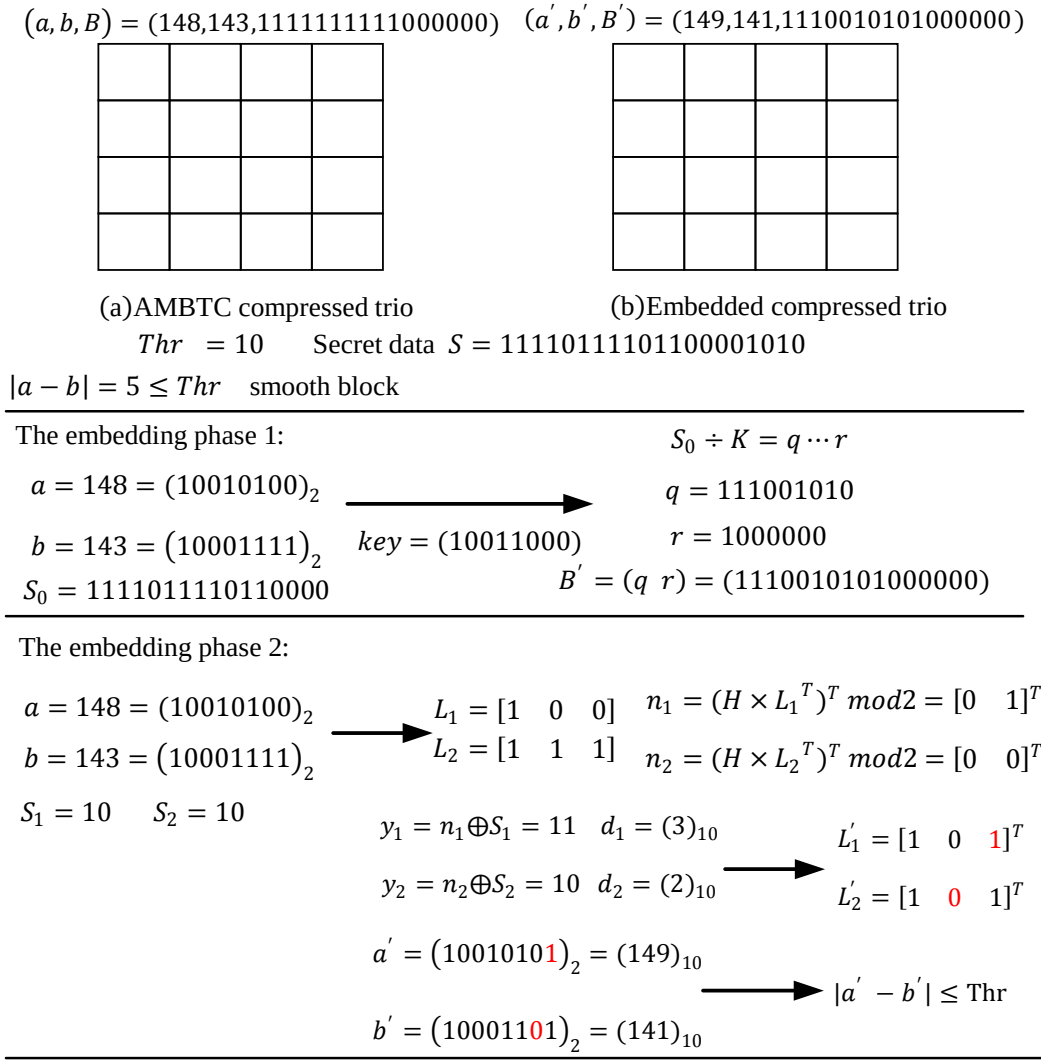

Figure 4. Illustration of data embedding.

Step 2.5. The embedded secret data is calculated by using $S_i' = (H \times L_i'^T)^T mod\ 2$, where $i = 1,2$.

Step 3. If $|a - b| > Thr$, it is a complex block. If $a - b < 0$, the order of two quantization levels is swapped, and secret data $S_3' = 1$; otherwise $S_3' = 0$.

Step 4. Repeat Steps 2–3 until all the trios are completely processed, and the extracted bit sequence constitutes the secret data S.

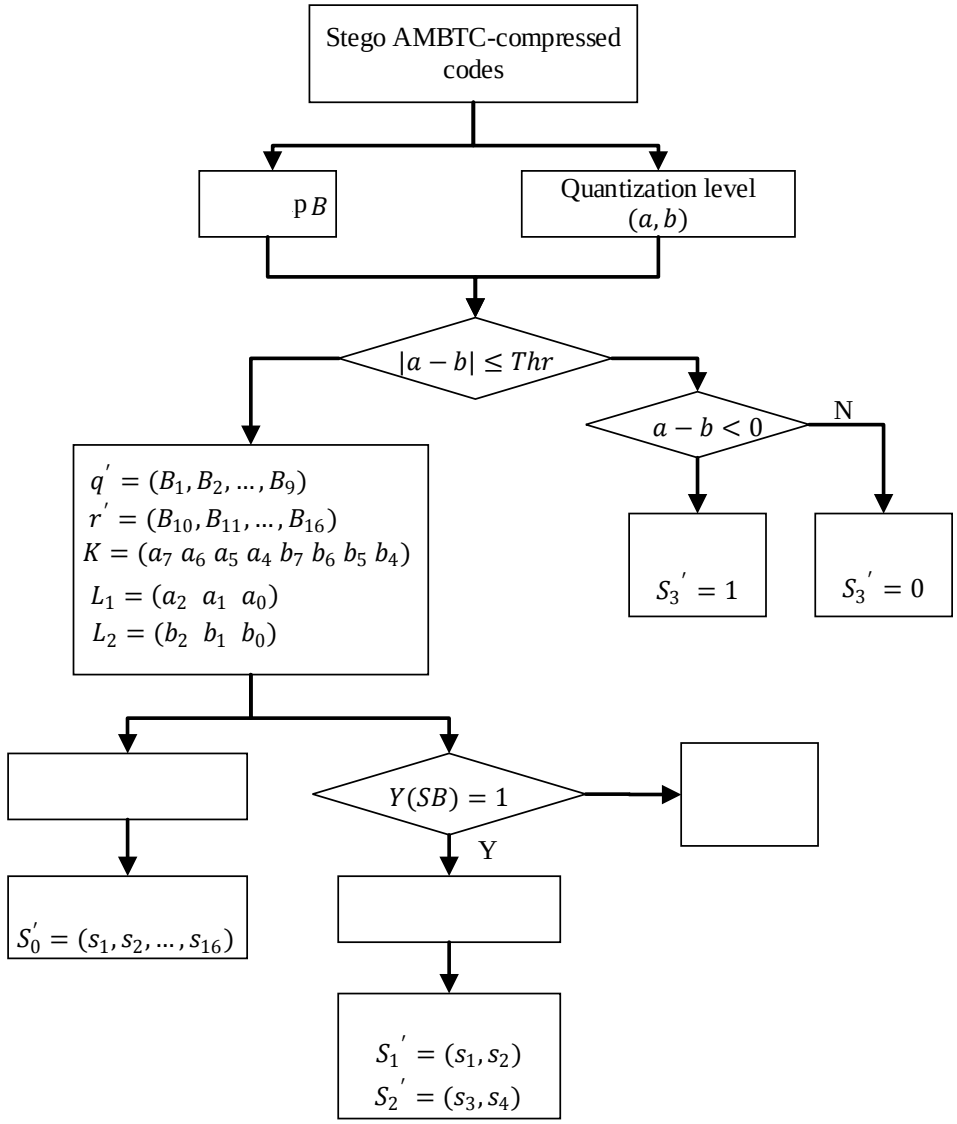

Figure 5. The flowchart of extracting process.

4. Experimental results

In order to prove the effectiveness of the proposed method, Ou and Sun's [24], Bai and Chang's [25], Kumar et al.'s [28] and Chuang et al.'s [23] methods are used for comparison. As shown in Figure 6, six 512×512 images are used in the experiment. In addition, the block size of AMBTC compression is set to 4×4, and secret data are generated by the same pseudo-random generator.

4.1. The performance of the proposed method

Peak Signal to Noise Ratio (PSNR) [33] is the most widely used objective image evaluation index. The higher the PSNR value, the better the quality performance of the hidden image. The PSNR value is defined as

$$\text{PSNR} = 10\log_{10}\frac{255^2}{MSE} \tag{14}$$

where MSE is defined as

$$\text{MSE} = \frac{1}{W \times H}\sum_{i=1}^{W}\sum_{j=1}^{H}(x_{ij} - x_{ij}^{'})^2 \tag{15}$$

where $W \times H$ is the size of the image, and x_{ij} and $x_{ij}^{'}$ indicates the pixel values of the original image and stego image at the same position (i, j), respectively.

In addition to the PSNR, Mean Structural Similarly Index Measure (MSSIM) [34] is also used to measure the similarity between the original image and the stego image. The MSSIM is between 0 and 1. In general, the MSSIM is close to 1, indicating that the stego image is structurally more similar to the original image.

(a) (b) (c)

(d) (e) (f)

Figure 6. Test images (a) Lena (b)Baboon (c) Peppers (d) Plane (e) Clown (f) Boats.

The bit rate BR is the number of bits required to measure a pixel and is calculated by

$$\text{BR} = \frac{L}{W \times H}(bpp) \tag{18}$$

where L represents the length of AMBTC compressed codes.

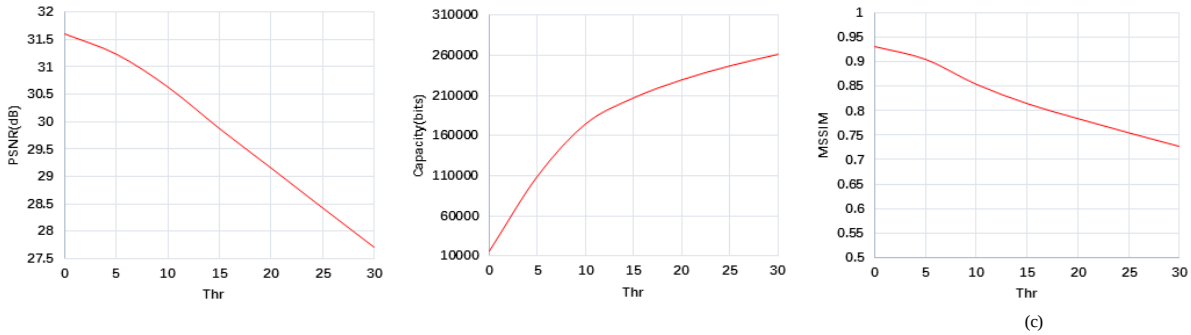

Figure 7. PSNRs, MSSIMs and Embedding Capacity with different values of Thr (a) the average PSNR (b) the average MSSIM (c) the average Capacity.

Table 1. PSNR, MSSIM and embedding capacity with different thresholds for Thr.

Method	Thr	Performances	Lena	Baboon	Peppers	Plane	Clown	Boats
AMBTC	N/A	PSNR (dB)	33.24	26.98	33.42	31.97	32.51	31.54
		MSSIM	0.9413	0.8869	0.9349	0.9505	0.9413	0.9360
Proposed	5	PSNR (dB)	32.79	26.97	33.05	31.61	32.27	30.73
		MSSIM	0.9037	0.8851	0.9055	0.9093	0.9201	0.9102
		Capacity(bits)	132,422	21,280	94,902	169,726	102,044	138,369
	10	PSNR (dB)	31.78	26.82	31.60	31.09	31.73	30.71
		MSSIM	0.8373	0.8527	0.8052	0.8664	0.8867	0.8798
		Capacity(bits)	215,463	62,142	219,291	220,139	147,199	184,780
	15	PSNR (dB)	30.88	26.47	30.61	30.54	30.94	29.81
		MSSIM	0.7948	0.8044	0.7546	0.8393	0.8526	0.8464
		Capacity(bits)	251,063	102,758	260,514	244,390	177,777	207,205
	20	PSNR (dB)	30.13	26.07	29.98	29.95	29.82	29.02
		MSSIM	0.7689	0.7637	0.7338	0.8198	0.8128	0.8082
		Capacity(bits)	270,236	131,196	278018	260,268	206,115	232,767
	25	PSNR (dB)	29.44	25.57	29.44	29.44	28.40	28.29
		MSSIM	0.7492	0.7279	0.7186	0.8062	0.7556	0.7759
		Capacity(bits)	282,952	154,624	288044	270,109	236,692	250,673

Table 1 shows PSNR and MSSIM of different images with different Thr. As Thr increases, so does the embedding capacity, and the PSNRs gradually decrease but still maintains good visual quality as the SSIMs are stable. With different thresholds, such as $Thr = 5$ and $Thr = 10$, the embedding capacity of Lena is 132,422 bits and 215,463 bits, respectively.

The embedding capacity has noticeably improved, and PSNRs of the stego images decreases slightly when Thr is increased, which denotes that the proposed method can obtain high payload without image distortion. Moreover, the MSSIM is very close to 1, which denotes good visual quality as well. The average PSNRs, MSSIMs and embedding capacities of all six images are computed for different Thr as illustrated in Figure 7. It is clearly observed that as Thr increases,

PSNR and MSSIM decrease slightly, but the embedding capacity increases more. For example, when Thr is increased from 0 to 20, the average values of PSNRs and MSSIMs are only reduced by around 1.45 dB and 0.1473, respectively, but the average value of embedding capacity is increased by around 213382 bits.

Table 2. Comparison of embedding capacity and PSNR ($Thr = 10$).

Method	Performances	Lena	Baboon	Peppers	Plane	Clown	Boats
AMBTC	PSNR (dB)	33.24	26.98	33.42	31.97	32.51	31.54
	BR (bpp)	2.000	2.000	2.000	2.000	2.000	2.000
Proposed	PSNR (dB)	31.78	26.82	31.60	31.09	31.73	30.81
	Capacity (bits)	215,463	62,142	219,291	220,139	147,199	184,780
	BR (bpp)	2.000	2.000	2.000	2.000	2.000	2.000
Ou and Sun's [24]	PSNR (dB)	32.67	26.92	33.36	31.91	31.52	31.32
	Capacity (bits)	172,579	50,374	183,529	178,099	136,774	148,609
	BR (bpp)	2.000	2.000	2.000	2.000	2.000	2.000
Bai and Chang's [25]	PSNR (dB)	28.92	24.17	28.95	28.11	32.25	27.56
	Capacity (bits)	169,250	176,178	169,644	168,388	168,480	170,103
	BR (bpp)	2.000	2.000	2.000	2.000	2.000	2.000
Kumar et al.'s [28]	PSNR (dB)	36.11	29.1	36.74	36.96	35.75	35.96
	Capacity (bits)	115,060	125,900	112,859	139,125	118,452	121,569
	BR (bpp)	1.988	2.06	1.930	1.785	2.014	2.054
Chuang et al.'s [23]	PSNR (dB)	32.03	26.85	32.37	31.54	31.03	31.04
	Capacity (bits)	166,608	36,256	178,288	172,496	141,040	128,416
	BR (bpp)	2.000	2.000	2.000	2.000	2.000	2.000

4.2. Comparison with other related works

A comparison of four existing data hiding methods are listed in Table 2. From Table 2, we can see that the embedding capacity ofthe proposed method is higher than that of Chuang et al.'s [23]. It is mainly because the proposed method can replace the bit plane with some secret data bits while Chuang and Chang's method cannot do so in the smooth blocks. Compared with Ou and Sun's [24] method, extra two bits are embedded in the two quantization levels in the proposed method, which leads to a higher embedding capacity. However, the PSNR of the proposed method are 0.645 dB lower than that of Ou and Sun's method on average. The gain of embedding capacity considered, the proposed method is superior to Ou and Sun's. From Table 2, the PSNRs and embedding capacity of the proposed method are all higher than those of Bai and Chang's [25] method. For instance, when $Thr = 10$ for Lena, our PSNR and embedding capacity are 31.78 dB and 215463 bits, respectively. Thus, PSNR gain and embedding capacity gain are 2.86 dB and 46213 bits compared with those of Bai and Chang's method. Kumar et al.'s method [28] embeds secret data into the bit plane of each block using a bit replacement strategy. This method improves the quality of the image, but our method has a slightly higher embedding capacity. Kumar and Kim's [27] method is divided into three states by setting two different thresholds. Although the embedding capacity is improved, the image quality is not significantly

improved. Based on the above analysis, the proposed method outperforms these four methods.

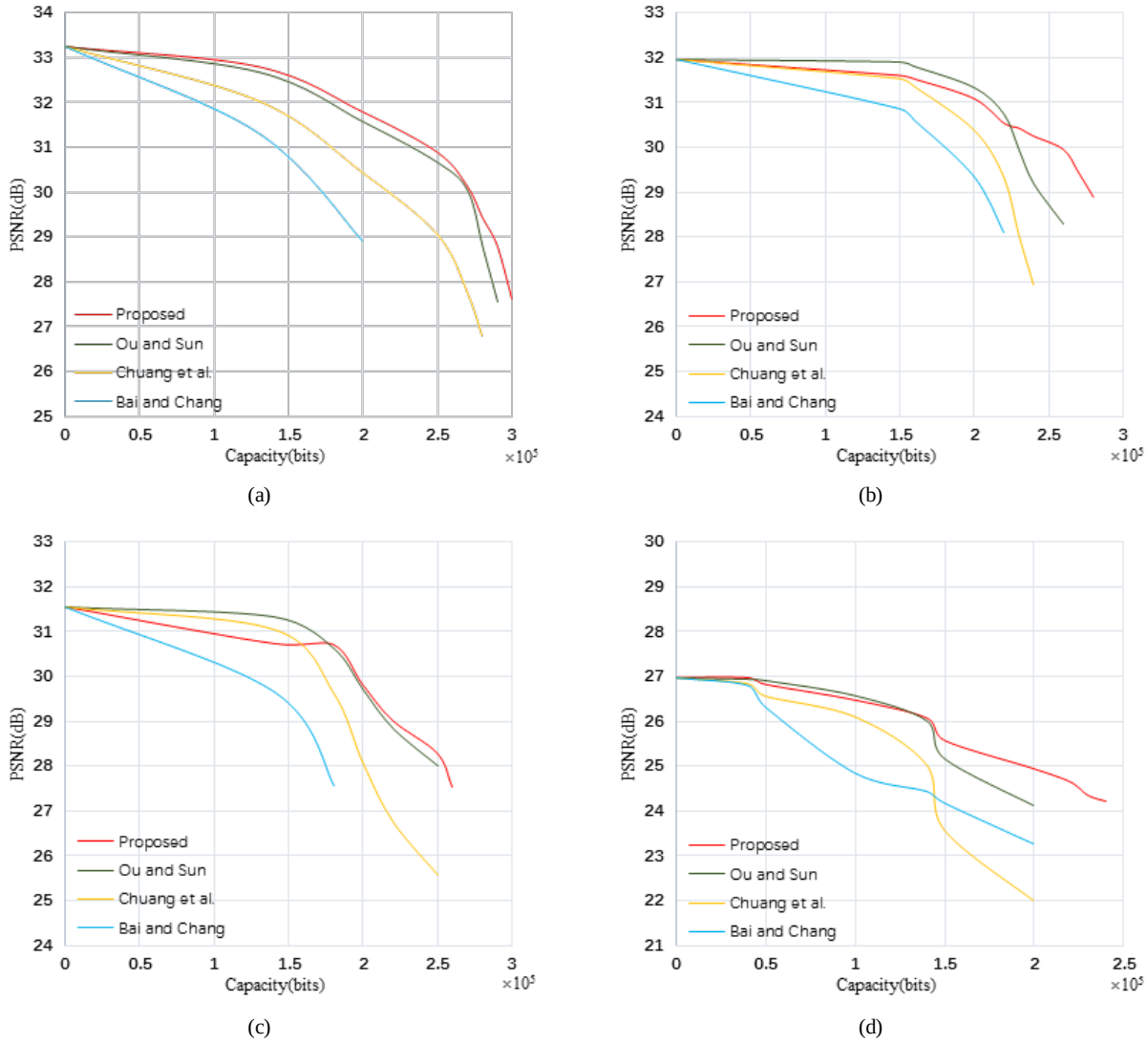

Figure 8. Performance comparison of four methods. (a) Lena (b) Plane (c) Boats (d) Baboon.

In the following experiments, the best thresholds are set for the compared and proposed methods to ensure that the best image quality can be achieved for a given embedding capacity. The Capacity-PSNR curves are shown in Figure 8. Although Figure 8 only shows testing results for Lena, Plane, Boats, and Baboon images, experiments on other images also suggest the similar trends. Bai and Chang's [25] method has the lowest image quality because this method embeds secret data into the bitmap and quantization levels by using Hamming code (7, 4). Ou and Sun's [24] method gives an optimal modification to the quantization levels to minimize distortion. However, this method does not utilize quantization levels to embed the secret data, and therefore the embedding capacity is limited. Chuang et al. [23] performs better than Bai and Chang's method because it embeds secret data only into smooth blocks. Nevertheless, the proposed method provides the best results for various embedding capacities, especially at high embedding capacities. It is mainly because the proposed

method selectively embeds the data into quantization level using a Hamming code (3, 2).

Compared to other compression formats, AMBTC has a lower compression ratio than JPEG, but it is more suitable for real-time applications. It has become the subject of real-time image transmission. Since JPEG has lower redundancy than AMBTC, the embedding capacity of the proposed method is thirty times compared with JPEG based data hiding method [22]. Therefore, the embedding capacity of the proposed method is much higher than those of JPEG compression based data hiding methods.

5. Conclusion

An efficient data hiding method based on modulo-2 operation and Hamming (3, 2) of AMBTC compressed images is proposed. A block in a grayscale image is compressed to a bitmap and a pair of quantization levels. The blocks of the AMBTC compressed codes are classified as smooth blocks and complex blocks. In the smooth block, the secret data and the four most significant bits of two quantization levels are computed using modulo-2 division, and the result replaces the bitmap. The three least significant bits of the two quantization levels are used to embed the two additional secret bits with Hamming code (3, 2), respectively. In the complex block, a secret bit is embedded by swapping the order of two quantization levels and flipping the bitmap. The experimental results show that the proposed method inherits the low complexity and easy implementation of AMBTC. Compared with other existing AMBTC-based algorithms, the proposed method achieves higher capacity and maintains good visual quality. Since BTC has many variations, we will study data hiding methods for some more advanced BTC-variant in the future.

Acknowledgments

This work was mainly supported by Public Welfare Technology, Industry Project of Zhejiang Provincial Science Technology Department and Natural Science Foundation of China under Grant. (No. LGG18F020013, No. LGG19F020016, No. 61971247).

Conflict of Interest

All authors declare no conflicts of interest in this paper.

References

1. F. A. P. Petitcolas, R. J. Anderson and M. G. Kuhn, Information hiding-a survey, *Proc. IEEE*, **87** (1999), 1062–1078.
2. D. Xiao, J. Liang, Q. Ma, et al., High capacity data hiding in encrypted image based on compressive sensing for nonequivalent resources, *CMC Comput. Mater. Continua*, **58** (2019), 1–13.
3. Y. Du, Z. Yin and X. Zhang, Improved lossless data hiding for JPEG images based on histogram modification, *CMC Comput. Mater. Continua*, **55** (2018), 495–507.

4. Y. Chen, B. Yin, H. He, et al., Reversible data hiding in classification-scrambling encrypted-image based on iterative recovery, *CMC Comput. Mater. Continua*, **56** (2018), 299–312.
5. J. W. Wang, T. Li, X. Y. Luo, et al., Identifying computer generated images based on quaternion central moments in color quaternion wavelet domain, *IEEE Trans. Circuits Syst. Video Technol.*, (2018).
6. T. Qiao, R. Shi, X. Luo, et al., Statistical model-based detector via texture weight map: Application in re-sampling authentication, *IEEE Trans. Multimedia*, **21** (2018), 1077–1092.
7. J. Fridrich, M. Goljan and R. Du, Detecting LSB steganography in color and gray-scale images, *IEEE Multimedia*, **8** (2001), 22–28.
8. M. Omoomi, S. Samavi and S. Dumitrescu, An efficient high payload ±1 data embedding scheme, *Multimedia Tools Appl.*, **54** (2011), 201–218.
9. Y. Zhang, C. Qin, W. Zhang, et al., On the fault-tolerant performance for a class of robust image steganography, *Signal Process.*, **146** (2018), 99–111.
10. Y. Ma, X. Luo, X. Li, et al., Selection of rich model steganalysis features based on decision rough set α-positive region reduction, *IEEE Trans. Circuits Syst. Video Technol.*, **29** (2018), 336–350.
11. W. Luo, F. Huang and J. Huang, Edge adaptive image steganography based on LSB matching revisited, *IEEE Trans. Inf. Forensics Secur.*, **5** (2010), 201–214.
12. W. Hong, Adaptive image data hiding in edges using patched reference table and pair-wise embedding technique, *Inf. Sci.*, **221** (2013), 473–489.
13. V. Kumar and D. Kumar, A modified DWT-based image steganography technique, *Multimedia Tools Appl.*, **77** (2018), 13279–13308.
14. C. C. Lin, C. C. Chang and Y. H. Chen, A novel SVD-based watermarking scheme for protecting rightful ownership of digital images, *IEEE Trans. Multimedia*, **5** (2014), 124–143.
15. C. C. Chang, T. D. Kieu and W. C. Wu, A lossless data embedding technique by joint neighboring coding, *Pattern Recognit.*, **42** (2009), 1597–1603.
16. C. C. Chang, Y. H. Chen and C. C. Lin, A data embedding scheme for color images based on genetic algorithm and absolute moment block truncation coding, *Soft Comput.*, **13** (2009), 321–331.
17. A. J. Zargar and A. K. Singh, Robust and imperceptible image watermarking in DWT-BTC domain, *Int. J. Electron. Secur. Digital Forensics*, **8** (2016), 53–62.
18. C. K. Chan and L. M. Cheng, Hiding data in images by simple LSB substitution, *Pattern Recognit.*, **37** (2004), 469–474.
19. J. A. Fessler and B. P. Sutton, Nonuniform fast Fourier transforms using min-max interpolation, *IEEE Trans. Signal Process.*, **51** (2003), 560–574.
20. B. Chen, S. Latifi and J. Kanai, Edge enhancement of remote image data in the DCT domain, *Image Vision Comput.*, **17** (1999), 913–921.
21. C. Qin and Y. C. Hu, Reversible data hiding in VQ index table with lossless coding and adaptive switching mechanism, *Signal Process.*, **129** (2016), 48–55.
22. Y. Qiu, H. He, Z. Qian, et al., Lossless data hiding in JPEG bitstream using alternative embedding, *J. Visual Commun. Image Representation*, **52** (2018), 86–91.
23. J. C. Chuang and C. C. Chang, Using a simple and fast image compression algorithm to hide secret information, *Int. J. Comput. Appl.*, **28** (2006), 329–333.

24. D. Ou and W. Sun, High payload image steganography with minimum distortion based on absolute moment block truncation coding, *Multimedia Tools Appl.*, **74** (2015), 9117–9139.
25. J. Bai and C. C. Chang, High payload steganographic scheme for compressed images with Hamming code, *Int. J. Network Secur.*, **18** (2016), 1122–1129.
26. C. Kim, D. Shin, L. Leng, et al., Lossless data hiding for absolute moment block truncation coding using histogram modification, *J. Real-Time Image Process.*, **14** (2018), 101–114.
27. R. Kumar, D. S. Kim and K. H. Jung, Enhanced AMBTC based data hiding method using hamming distance and pixel value differencing, *J. Inf. Secur. Appl.*, **47** (2019), 94–103.
28. R. Kumar, N. Kumar and K. H. Jung, *A new data hiding method using adaptive quantization & dynamic bit plane based AMBTC*, 2019 6th International Conference on Signal Processing and Integrated Networks (SPIN), 2019, 854–858. Available from: https://ieeexplore.ieee.org/abstract/document/8711774.
29. M. Lema and O. Mitchell, Absolute moment block truncation coding and its application to color images, *IEEE Trans. Commun.*, **32** (1984), 1148–1157.
30. P. Fränti, O. Nevalainen and T. Kaukoranta, Compression of digital images by block truncation coding: A survey, *Comput. J.*, **37** (1994), 308–332.
31. C. Kim, D. Shin, B. G. Kim, et, al.,Secure medical images based on data hiding using a hybrid scheme with the Hamming code, LSB, and OPAP, *J. Real-Time Image Process.*, 14 (2018), 115–126.
32. E. Tsimbalo, X. Fafoutis and R. Piechocki, CRC error correction in IoT applications, *IEEE Trans. Ind. Inf.*, **13** (2017), 361–369.
33. C. Kim, Data hiding by an improved exploiting modification direction, *Multimedia Tools Appl.*, **69** (2014), 569–584.
34. Z. Wang, A. C. Bovik, H. R. Sheikh, et al., Image quality assessment: From error visibility to structural similarity, *IEEE Trans. Image Process.*, **13** (2004), 600–612.

AIMS Press