IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 29, 2019, accepted March 1, 2019, date of publication March 20, 2019, date of current version April 5, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2906500

Reversible AMBTC-Based Data Hiding With Security Improvement by Chaotic Encryption

HSIANG-YING WANG[1], HSIN-JU LIN[1], XIANG-YUN GAO[1], WEN-HUANG CHENG[2],(Member, IEEE), AND YUNG-YAO CHEN[1], (Member, IEEE)

[1]Graduate Institute of Automation Technology, National Taipei University of Technology, Taipei 106, Taiwan
[2]Institute of Electronics, National Chiao Tung University, Hsinchu 300, Taiwan

Corresponding author: Yung-Yao Chen (yungyaochen@mail.ntut.edu.tw)

This work was supported by the Ministry of Science and Technology under Grant 107-2221-E-027-123-.

ABSTRACT Social networking and cloud computing are being extensively used, and in this era, the frequency of sending information or images to each other is increasing. The prevention of private information leakage during communication over the Internet has become a concern in the past decades. Several data protection methods, such as cryptographic, watermarking, and steganography techniques, have been proposed to protect private data. In this paper, an embedding method is proposed based on an absolute moment block truncation coding (AMBTC)-compressed image. High and low mean tables are extracted from a compressed image and are divided into non-overlapping blocks. An adaptive variable N-bit bit plane truncation image embedding method is proposed to embed the secret data in each block. In this method, at the receiver end, the secret data are extracted, and the original AMBTC image could be recovered by recalling the stored peak and zero points. In addition, a chaotic encryption scheme is integrated into the proposed system to improve robustness against security vulnerability. The results show that the proposed method has superior performance and higher payload compared with the reference methods.

INDEX TERMS Reversible data hiding, absolute moment block truncation coding (AMBTC), encryption, chaos.

I. INTRODUCTION

In smart cities, a large amount of digital data are communicated over the Internet everyday because of the rapid development of networking technologies. The protection of this information is a major concern and must be discussed. Security problems include different activities such as interception, illegal duplication, and counterfeiting through an open network. A data hiding technique is one of the effective solutions that maintains the secrecy of the transmitted information.

This study proposes a data hiding scheme for the AMBTC-compressed image because most images on the Internet are compressed. A block truncation coding (BTC) scheme [1] was proposed by Delp and Mitchell, which is also called moment-preserving block truncation coding (MPBTC). The BTC scheme is considered a lossy compression method, i.e., although it reduces the memory of the image or video, it loses some information of the cover image at the same time. However, BTC has simple image encoding or decoding procedures and involves low computational cost. Because of this advantage, BTC has been used in several applications related to grayscale/color image compression and video compression. Lema and Mitchell presented an improved version of BTC called AMBTC [2] to preserve higher image performance.

The goal of data hiding is to embed secret data in an input image (i.e., cover image) and generate a data-embedded image (i.e., stego image), which is visually similar to the cover image. Data hiding schemes can be classified into two categories: reversible [3]–[7] and irreversible data hiding schemes [8]–[11]. In irreversible data hiding schemes, although hidden data can be extracted from stego images, original cover images cannot be restored. By contrast, in addition to data extraction, reversible data hiding schemes can restore cover images. Reversible data hiding schemes have lower hiding capacities than irreversible data hiding schemes; however, it can preserve better image performance. In general, because reversible data hiding must carry extra information to restore original images, the compression ratio is often lower. Overall, the difficulty associated with reversible data

The associate editor coordinating the review of this manuscript and approving it for publication was Carlisle Adams.

hiding schemes is higher than that associated with irreversible data hiding schemes.

Many reversible data hiding schemes have been proposed. Histogram shifting (or modification) schemes [12]–[14] are one of the most common methods. Ni *et al.* [15] presented a histogram shifting-based data hiding scheme, which explores the histograms of pixels in the cover image. First, the pairs of peak point (PP) and zero point (ZP) are found, and the pixels between the peak and zero points are modified in order to embed data. Each pixel in a PP is used for 1-bit embedding. Therefore, the number of pixels in the PPs determines the maximal hiding capacity for the secret data to be embedded.

To increase the hiding capacity of the histogram shifting technique, Tsai *et al.* [16] presented a novel data hiding method using a residual histogram. A cover image is divided into non-overlapping blocks, and the center pixel in each block is selected as a basic pixel for linear prediction. Other pixels in block are processed using the linear prediction technique to generate the residual values and residual histogram. Similar to [15], multiple pairs of the PP and ZP can be used in residual histogram shifting to increase the payload. Chang *et al.* [17] further improved the scheme proposed by Tsai *et al.* by using residual histogram shifting in AMBTC-compressed images, thus simultaneously reducing the storage memory. Some improved histogram-based hiding methods have been recently proposed.

With mass and heterogeneous data transmitted everywhere, vicious attacks become invincible. The requirement of a cryptosystem pushes the technology to protect our information [18]–[20]. An appropriate cryptosystem must have sufficient key space against attacks. The histogram of the cipher image must be random, and adjacent pixels must be irrelevant. In recent years, some studies have analyzed the security of cryptosystems. Traditional encryption technology is advancing; however its application in digital images can be further improved. Some challenges are still encountered in digital image encryption.

Chaos is an interdisciplinary theory concerning factors, such as a few parameters, related dynamical systems, and randomness of chaotic complexity. Recently, many image encryption methods involving Chaos have been proposed. The image to be encrypted is first transformed into a scrambled sequence, which is controlled using the initial values (keys) of a chaotic system. It is further sorted into a random set so that outsiders cannot attack. Only legal receivers can extract the hidden data or retrieve an image that is similar to the original image. Chaos can be used for real-time applications because of its simple algorithm and low computational complexity.

This study presents a reversible data hiding scheme based on AMBTC. The histogram modified embedding method, which contains the concept of reversible macro- and micro-block embedding, is utilized for obtaining a high payload and low storage memory. Attacks on stego images can be prevented by integrating chaotic-based encryption into our system. Only users with the data hiding key can decrypt, decode, and restore the original image.

The remaining of this paper is organized as follows. In Section 2, the related works are reviewed. Section 3 presents the proposed reversible data hiding scheme and a chaotic encryption method which is used to improve the security level. Section 4 provides the experimental results of the proposed method, which are compared with those of the state-of-the-art methods. The conclusions are presented in Section 5.

II. RELATED WORKS

A. BLOCK TRUNCATION CODING (BTC)

This subsection briefly describes the BTC and AMBTC methods. The BTC method is a lossy compression method, which divides the input grayscale image into non-overlapped $n \times n$ blocks. In each block, the mean (μ) and standard deviation (σ) are calculated. To approximate the original image, each block is stored only as one bitmap and two corresponding quantization levels: low mean value and high mean value. Without lack of generality, n is set as 4 in this study, and the two mean values A and B are calculated by

$$A_{BTC} = \mu - \sigma \times \sqrt{\frac{q}{16-q}}, \tag{1}$$

and

$$B_{BTC} = \mu + \sigma \times \sqrt{\frac{q}{16-q}}, \tag{2}$$

where q represents the number of pixels whose pixel value is larger than μ. Therefore, for an input image with size of 512×512, two BTC (low and high) mean tables with size of 128×128 are generated.

Later, Lema and Mitchell proposed an improved method of BTC called AMBTC, in which the two mean values are redefined as

$$A_{AMBTC} = \frac{1}{q'} \sum_{x_i < \mu} x_i, \tag{3}$$

and

$$B_{AMBTC} = \frac{1}{16-q'} \sum_{x_i \geq \mu} x_i, \tag{4}$$

where x_i denotes the i-th pixel value, and q' denotes the number of pixels whose values are less than μ. In AMBTC, although it reduces the computational complexity, it preserves the absolute central moment of each original block and the high image performance. In order to achieve compression, the bitmap is defined as follows. When a pixel value is less than μ of the block, it is set to "0" at that position; and otherwise, it is set to "1". When approximating the uncompressed grayscale image, each bit value "0" of bitmap is restored as the low mean value, and each bit value "1" of bitmap is restored as the high mean value.

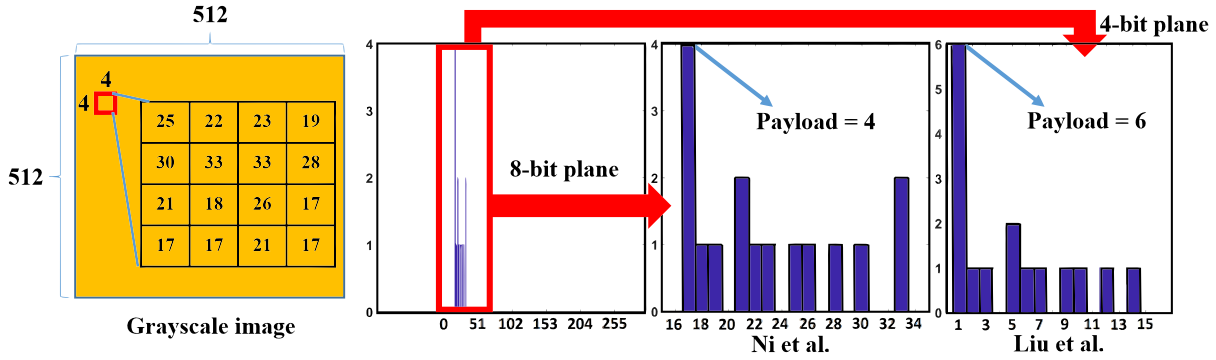


FIGURE 1. Comparison between the histograms generated by the methods of Ni *et al.* [15] and Liu *et al.* [21], respectively.

B. BIT PLANE TRUNCATION IMAGE (BPTI)

Ni *et al.* proposed a histogram-modified method for reversible data hiding [15] in grayscale images. In [15], an input image is first transformed into a histogram, where the occurrence frequencies of all pixel values in the image are calculated. The following points are searched from the histogram: 1) the PP defined as the pixel value with the highest occurrence frequency, and 2) the ZP defined as the pixel value with the lowest occurrence frequency. The data are embedded by modifying the pixels with values ranging between the PP and ZP. For obtaining the reversible property, values of the PP and ZP must be recorded in addition to a stego image. In [21], Liu *et al.* proposed a bit plane truncation image (BPTI) scheme that truncates an 8-bit plane into the least significant N-bit plane (N-BP). That is, the N least significant bit of each pixel is extracted and the binary-to-decimal conversion is applied as follows:

$$\breve{p} = p^1 \times 2^0 + p^2 \times 2^1 + \ldots + p^N \times 2^{N-1}, \tag{5}$$

where $p^i \in \{0, 1\}$ is i from the last bit of the pixel value [i.e., taking the N least significant bit using (5)] and $\breve{p}$ is the new pixel value of the N-BP in the BPTI. The BPTI is used to increase the payload of the histogram-modified method. As shown in Fig. 1, by truncating the bit plane, the histogram generated by Liu *et al.*'s scheme is more concentrated, which is likely to have larger payload.

III. PROPOSED JOINT DATA-HIDING AND ENCRYPTION SYSTEM

A. DUAL-BLOCK SETTING TO EMBED SECRET DATA

This work aims to provide a double protection scheme (i.e. image-based data hiding and image encryption) for image security. Figure 2 shows the flowchart of the proposed system. First, a dual-block setting, with a macro-block and a micro-block, is proposed to embed the secret data. After the AMBTC compression, the AMBTC mean table T is subsampled into 128 × 128 macro-blocks as follows:

$$B^k_{Macro}(i,j) = T\left(8i + \left\lfloor \frac{k-1}{8} \right\rfloor, 8j + \mathrm{mod}(k-1, 8)\right), \tag{6}$$

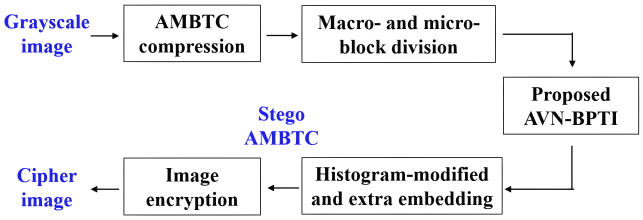


FIGURE 2. Flowchart of the proposed joint data-hiding and encryption system.

where k and (i, j) are the order of macro-blocks and row or column positions of the mean table, respectively. In this study, the secret data are separately embedded in each macro-block. Each macro-block comprises four micro-blocks as follows:

$$B_{Macro} = \left\{B^{\mathrm{TL}}_{Micro}, B^{\mathrm{TR}}_{Micro}, B^{\mathrm{BL}}_{Micro}, B^{\mathrm{BR}}_{Micro}\right\}, \tag{7}$$

where TL, TR, BL, and BR indicate top-left, top-right, bottom-left, and bottom-right, respectively. Each micro-block has an individual scanning order (indicated by arrows in Fig. 3), from which 15 residual values are orderly calculated as follows:

$$\{r_i = p_i - p_{i+1} | \, i = 1, 2, \ldots, 15\}, \tag{8}$$

where p_i is the i^{th} grid value in a micro-block. Because the mean values in adjacent AMBTC blocks generally have close

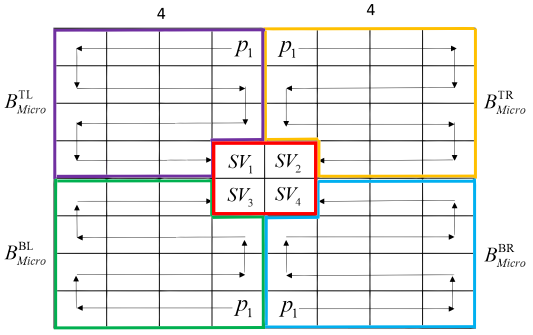


FIGURE 3. Blocks used in the proposed method: 8 × 8 macro-block and 4 × 4 micro-block, where each grid indicates an AMBTC high mean (or low mean) value.

values, this study utilizes the advantage of local similarity among neighboring mean values to increase the payload by using serpentine scanning orders. Moreover, the center 2×2 block of a macro-block is considered a steady block (the red square in Fig. 3), which comprises four 16^{th} grids of micro-blocks. Values in the steady block are maintained because they are the key to retrieve the original AMBTC image and secret data at the receiver end. In addition, another embedding approach is proposed to embed extra payload in the steady block.

B. ADAPTIVE VARIABLE N-BIT BPTI

In the histogram-modified embedding method, it is necessary to find at least one pair of the PP and ZP to embed the data, and from these data, the payload size is determined using the number of pixels (n_P) in the PP, and the size of extra storage is determined using the number of pixels (n_Z) in the ZP. In the easiest case, n_Z is equal to zero that only the values of the PP and ZP need to be recorded. If n_Z is larger than zero, the locations of pixels in the ZP must be recorded and are recalled when recovering the original image, thus inevitably increasing the storage size. To avoid this problem, data are not embedded in [21] for blocks with $n_Z \neq 0$. However, it does not completely solve the problem of requiring extra storage because in addition to recording of the pixel locations, the locations of these blocks must be recorded to maintain reversibility.

To solve the abovementioned problem, this study proposes an adaptive variable N-bit (AVN)-BPTI embedding scheme. The term "AVN" indicates that the value of N in the N-BP is varied and dynamically selected. From (8), a residual histogram (R_H) is generated by 15 r_i values of a micro-block, and the range of the residual histogram is determined using N-BP truncation as follows:

$$Range_H^N = \left\{0, 1, \ldots, 2^N - 1\right\}, \tag{9}$$

In general, the smaller the N is, the larger the n_P is, which implies higher payload is possibly achieved. Nevertheless, n_Z in a narrow histogram range is generally not equal to zero. Therefore, the AVN-BPTI embedding scheme starts at $N = 1$ with $Range_H^1 = \{0, 1\}$. If n_Z is not equal to zero, the N value is increased by one to generate a new residual histogram, and the n_Z value of the new R_H is revaluated. By contrast, if n_Z is equal to zero, the subsequent step for histogram-modified embedding can be performed. Because the histogram range for $N = 4$ is $Range_H^4 = \{0, 1, \ldots, 15\}$ with 16 levels, n_Z must be zero because the residual histogram contains only 15 residual values. Fig. 4 presents the pseudocode of the proposed AVN-BPTI scheme.

C. HISTOGRAM-MODIFIED EMBEDDING

This section describes the process of embedding in the N-BP residual histogram (R_H^N), followed by the AVN-BPTI scheme. Similar to (5), the N-BP residual value (NBRV) is defined as

```
N = 1;
Generate the N-BP residual histogram;
Calculate n_Z;
while (n_Z ≠ 0 in the current histogram)
{
   N = N + 1;
   Renew the N-BP residual histogram;
   Calculate n_Z;
}
Save the N value;
Embed data by Histogram modification;
Output stego - block;
```

FIGURE 4. Pseudocode of the AVN-BPTI scheme.

follows:

$$\breve{r} = r^1 \times 2^0 + r^2 \times 2^1 + \ldots + r^N \times 2^{N-1}. \tag{10}$$

From (8), there are 15 NBRVs ($\breve{r}_i, i = 1, 2, \ldots, 15$) in a micro-block that construct an N-BP residual histogram. A pair of (PP, ZP) is assumed to be found in N-BP, and thus, two cases are considered for histogram-modified embedding. In histogram modification, only the values within the PP and ZP range are modified by maximum one, and the values outside this range are unchanged. Consequently, when embedding secret data, the highest quality of the image is maintained. For case 1, where the PP is smaller than the ZP, the NBRV within the range of (PP + 1, ZP − 1] is first increased by one to create a space for secret bits, and the remaining NBRVs are modified as follows:

$$\breve{r}_i' = \begin{cases} \breve{r}_i + 1, & \text{if } \breve{r}_i \in (\text{PP} + 1, \text{ZP} - 1] \\ \breve{r}_i, & \text{if } \breve{r}_i \notin [\text{PP}, \text{ZP}] \end{cases}. \tag{11}$$

The secret bits are embedded as follows:

$$\breve{r}_{ij}' = \breve{r}_i + s_j, \quad \text{if } \breve{r}_i = \text{PP}. \tag{12}$$

where $s_j \in \{0, 1\}$ is the j^{th} secret bit to be embedded with $j = 1, \ldots, n_P$, and $\breve{r}_{ij}'$ represents the i^{th} modified NBRV; moreover, it is ranked at the j^{th} order among the pixels with its NBRV value equal to the PP. Fig. 5 presents an example of histogram-modified embedding. Similar to (11) and (12), for case 2, where the PP is larger than the ZP, the embedding

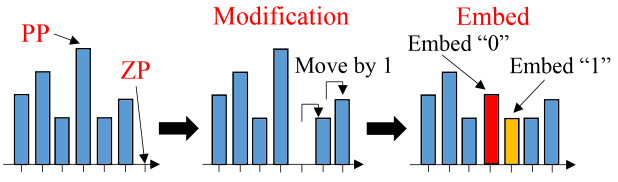


FIGURE 5. Histogram-modified embedding (for 3-BP and ZP > PP).

procedure is performed as follows:

$$\check{r}'_i = \begin{cases} \check{r}_i - 1, & \text{if } \check{r}_i \in (\text{ZP}+1, \text{PP}-1] \\ \check{r}_i, & \text{if } \check{r}_i \notin [\text{PP}, \text{ZP}], \end{cases} \tag{13}$$

and

$$\check{r}'_{ij} = \check{r}_i + s_j, \quad \text{if } \check{r}_i = \text{PP}. \tag{14}$$

The construction of the stego micro-block is described as follows. To maintain reversibility, the modified NBRVs are further processed by:

$$\widetilde{r}_i = \begin{cases} r_i - \check{r}_i + \check{r}'_i, & \text{if } r_i \geq 0 \\ r_i + \check{r}_i - \check{r}'_i, & \text{if } r_i < 0, \end{cases} \quad i = 1, 2, \ldots, 15 \tag{15}$$

where $\widetilde{r}_i$ represent the stego residual values. The stego values ($\widetilde{p}_i$) are calculated by performing (8) inversely as follows:

$$\{\widetilde{p}_i = \widetilde{p}_{i+1} + \widetilde{r}_i | \, i = 14, 13, \ldots, 1\}, \tag{16}$$

where the initial value ($\widetilde{p}_{15}$) is obtained by

$$\widetilde{p}_{15} = p_{16} + \widetilde{r}_{15}. \tag{17}$$

D. EXTRA-EMBEDDING BASED ON STEADY BLOCK AND LOSSLESS FLIPPING

In the steady block of a macro-block, four steady values ($SV_i, i = 1, \ldots, 4$ as shown in Fig. 3) are present, which are exploited for additional embedding. First, a prediction value (PV_i) is defined as a rounded-down nearest integer of the average of two nearest values. For example, PV_1 of SV_1 is defined as $\lfloor (SV_2 + SV_3)/2 \rfloor$. Subsequently, an absolute difference value is defined as $D_1 = |SV_1 - PV_1|$. The relationship between D_i and a predefined threshold T_{std} (T_{std} is set as 1) is used to determine the ability of pair (SV_i, PV_i) to carry 1-bit data. The relationship is as follows:

$$\begin{cases} \text{Embedding 1-bit in } (SV_i, PV_i), & \text{if } D_i \leq 1 \\ \text{No embedding in } (SV_i, PV_i), & \text{otherwise.} \end{cases} \tag{18}$$

The following hiding rules are designed for reversible data hiding through the modification of the steady value:

• Rule for embedding 1 bit: If the steady value is larger than or equal to the prediction value, the steady value is modified as follows: $SV'_i = PV_i + 2 \times D_i + s_i$, where SV'_i and $s_i \in \{0, 1\}$ are the modified value and the secret bit to be embedded, respectively. Otherwise, $SV'_i = PV_i - 2 \times D_i - s_i$.

• Rule for no embedding: If the steady value is larger than or equal to the prediction value, the steady value is modified as $SV'_i = SV_i + 2$. Otherwise, $SV'_i = SV_i - 2$.

When embedding data, the four steady values are modified from $i = 1$ to $i = 4$ in sequence. When extracting data, an inverse process is performed from $i = 4$ to $i = 1$. Moreover, the property of absolute moment block truncation coding (AMBTC) is utilized to carry the additional data of 1 bit in each AMBTC block through a lossless flipping strategy. If code 0 is embedded, the bitmap is flipped by applying a Boolean NOT operator and the orders of high and low means are reversed. Otherwise, the bitmap is unchanged. Extra embedding is performed to create additional embedding capacity for storing the information of an AVN value and peak/zero points.

E. SECURITY IMPROVEMENT BY CHAOTIC ENCRYPTION

The security of the data-embedded AMBTC (i.e. the stego AMBTC) images is further enhanced using the proposed chaotic encryption scheme. Chaotic mapping is nonperiodic and unpredictable. Thus, disrupting the characteristics of the image by chaotic mapping is appropriate, and the chaotic sequences are suitable for image encryption. To scramble the data-embedded image, the image of size $M \times N$ is reshaped into a one-dimensional sequence $E_i, i = 1, 2, \ldots, M \times N$. To create chaotic behavior, the following chaotic equations are selected: 1) chaos logistic equation

$$x^L_{n+1} = \mu_1 x^L_n (1 - x^L_n), \tag{19}$$

and 2) chaos sine equation

$$x^S_{n+1} = \mu_2 \sin(\pi x^S_n), \tag{20}$$

where $\mu_1 = 3.9$ and $\mu_2 = 0.89$. Two sequences, namely x^L_i and x^S_i, are generated by setting x_0 as 0.1 in (19) and (20), respectively. To obtain mapping positions, two mapping sequences (E^L_i and E^S_i) are generated by rearranging the chaotic sequences as follows:

$$E^L_i = sort(unique(x^L_i)), \tag{21}$$

and

$$E^S_i = sort(unique(x^S_i)), \tag{22}$$

where *unique* indicates a delete function which deletes repetitive elements of the sequences, and *sort* indicates a sorting function which sorts the sequence by size and records the rearrangement. The image sequence E_i is mapped into image chaotic sequences P^L_i and P^S_i by applying E^L_i and E^S_i, respectively. Finally, a Boolean XOR operator is used to diffuse the pixels as follows:

$$C_i = \begin{cases} P^L_i \oplus rescale(x^S_i), & \text{if } i \in \text{even} \\ P^S_i \oplus rescale(x^L_i), & \text{if } i \in \text{odd}, \end{cases} \tag{23}$$

where *rescale* indicates a normalized function which scales x^L_i (and x^S_i) into an integer sequence within the interval of [0, 255], and C_i indicates the pixels of a cipher image in a one-dimensional sequence. The decryption system is based on the inverse process of the above encryption algorithm.

F. DATA EXTRACTION AND IMAGE RECOVERY

This subsection describes the process of extracting the hidden data and recovering the original AMBTC image. First, the stego AMBTC is divided into 8×8 macro-blocks. For the stego values in the steady block, the prediction value (PV'_i) and absolute difference (D'_i) are recalculated. Note that in the extra-embedding of steady blocks, the predefined threshold T_{std} is set as 1. As this threshold becomes larger, the embedding capacity becomes larger but the image quality degrades.

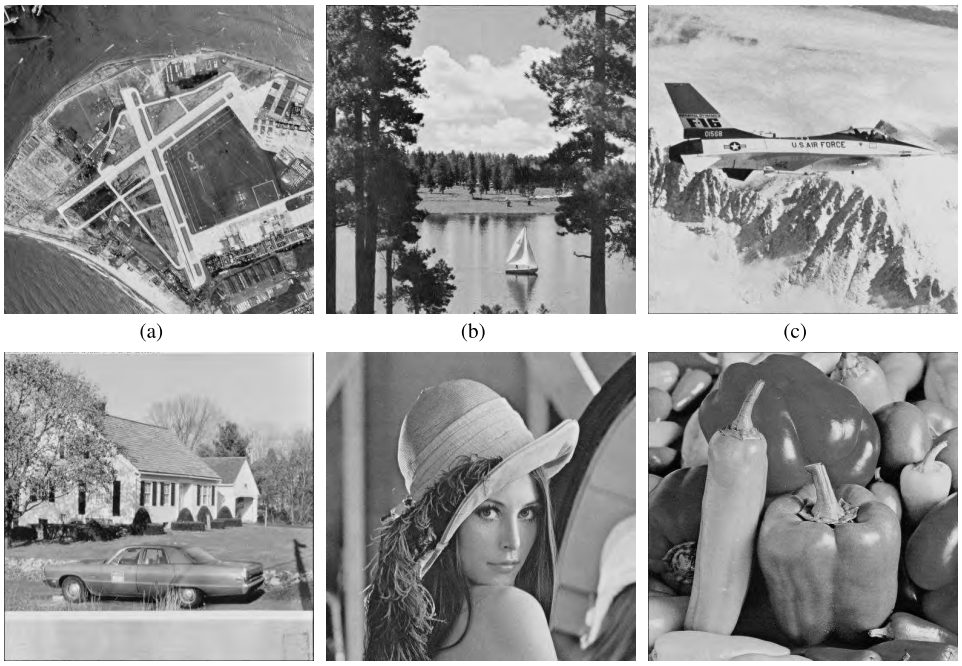


FIGURE 6. The six test images. (a) North Island, (b) Sailboat, (c) F-16, (d) House, (e) Lena, and (f) Peppers.

The absolute difference is comparing with $2 \times T_{\text{std}} + 1$. That is, if $D'_i \leq 3$, one secret bit is decoded as $\text{mod}(D'_i, 2)$.

In addition, the following recovery rules are used to restore the original steady values (SV_i):

- Recovery rule 1 for $D'_i \leq 3$: If SV'_i is larger than or equal to PV'_i, SV_i is recovered as $SV_i = PV'_i + \lfloor D'_i/2 \rfloor$. Otherwise, $SV_i = PV'_i - \lfloor D'_i/2 \rfloor$.
- Recovery rule 2 for $D'_i > 3$: If SV'_i is larger than or equal to PV'_i, SV_i is recovered as $SV_i = PV'_i - 2$. Otherwise, $SV_i = PV'_i + 2$.

After restoring the steady values, the hidden data embedded through lossless flipping can be extracted by evaluating the mean value pair of each AMBTC. Note that a macro-block contains 64 AMBTC blocks and 128 mean tables. Therefore, the AVN values and peak/zero points are recalled. In each micro-block, 15 residual values are recovered as follows:

$$r_i = \begin{cases} \widetilde{r}_i - \breve{r}'_i + \breve{r}_i, & \text{if } \widetilde{r}_i \geq 0 \\ \widetilde{r}_i + \breve{r}'_i - \breve{r}_i, & \text{if } \widetilde{r}_i < 0, \end{cases} \quad i = 1, 2, \ldots, 15 \qquad (24)$$

The remainder of the recovery procedure is similar to inversing the embedding process.

IV. EXPERIMENTAL RESULTS

This section evaluates the performance of the proposed method, and compare it with the three state-of-the-art methods from [17], [21], and [22].The six 512×512 standard test images are selected: North Island, Sailboat, F-16, House, Lena and Peppers, as shown in Fig. 6. The embedded secret data are generated from the same pseudorandom generator.

A. DEFINITION OF EVALUATION MEASURES

To quantitatively and fairly compare the performance of the different methods, this work utilizes the following evaluation measures.

1) *Peak signal-to-noise ratio*: To compare the image quality of the stego image, the peak signal-to-noise ratio (PSNR) is used for evaluating the quality performance. The PSNR is defined as follows:

$$10 \times \log_{10} \left(\frac{W \times H \times 255^2}{\sum_{W,H} \left[\sum_{m,n} g_{i+m,j+n} - h_{i+m,j+n} \right]^2} \right), \qquad (25)$$

where W and H represent the image width and the image height, respectively. The variable g represents the original image, and g represents the output stego AMBTC image. The PSNR value is the ratio between the maximum possible power of a signal and that of the corrupting noise, which is commonly used to measure the image quality. As PSNR value goes higher, it indicates the better quality of the stego images.

2) *Bit per pixel*: To compare the relationship between the storage memory and image size, the bit per pixel (Bpp) is used for evaluating the embedding performance. The Bpp is defined as follows:

$$Bpp = \frac{CS_f}{\text{Total number of pixels in the test image}}, \qquad (26)$$

where CS_f is the length of the final code of a stego AMBTC image. The Bpp measures the number of bits

TABLE 1. Comparison of payload, efficiency and Bpp of various methods*.

Metrics	Methods	Test Images						
		North Island	Sailboat	F-16	House	Lena	Peppers	Avg.
Payload (bits)	Chang et al. [17]	7638	8931	10654	8414	9897	9627	9194
	Liu et al. [21]	6622	7238	8541	8737	7261	7113	7585
	Yin et al. [22]	983	1019	2315	2174	2343	1973	1801
	Ours	**20946**	**22050**	**23610**	**23671**	**22287**	**22142**	22451
Efficiency (%)	Chang et al. [17]	1.29	1.51	1.81	1.42	1.68	1.63	1.56
	Liu et al. [21]	1.22	1.33	1.57	1.61	1.34	1.31	1.40
	Yin et al. [22]	0.19	0.19	0.44	0.44	0.45	0.38	0.35
	Ours	**3.89**	**4.11**	**4.41**	**4.41**	**4.15**	**4.12**	4.18
Bpp	Chang et al. [17]	2.25	2.25	2.25	2.25	2.25	2.25	2.25
	Liu et al. [21]	2.07	2.07	2.07	2.07	2.07	2.07	2.07
	Yin et al. [22]	**2.00**	**2.00**	**2.00**	**2.00**	**2.00**	**2.00**	2.00
	Ours	2.05	2.05	2.04	2.04	2.04	2.04	2.04
Mean Square Error (MSE)	Chang et al. [17]	79.81	79.85	47.87	60.55	34.60	32.89	55.93
	Liu et al. [21]	77.82	77.28	46.57	57.29	33.76	32.51	54.21
	Yin et al. [22]	80.37	80.73	49.21	64.28	36.15	36.06	57.80
	Ours	**75.00**	**69.03**	**43.66**	**55.47**	**32.89**	**31.92**	51.33

∗ Bold number indicates the best performing method in each image.

required to record a pixel of an image. Therefore, this value must be as low as possible.

3) *Efficiency*: To compare the storage memory and data capacities, the efficiency is measured. The efficiency is defined as follows:

$$Efficiency = \frac{S}{CS_f}, \tag{27}$$

where S is the length of the hidden message, i.e. the payload. The efficiency represents the payload size under a particular (fixed) length of the final code. Therefore, this value must be as high as possible.

B. PERFORMANCE COMPARISON

A general property of different histogram shifting methods is discussed in this section. For a reasonable and fair comparison, the method proposed by Liu *et al.* [21] is modified to be compatible with the AMBTC format. Similar to the proposed method, in [21], the high (and low) mean tables are divided into 4 × 4 blocks. In each block, the least significant 4 bits of each value are extracted. Then, binary-to-decimal transformation is applied to generate a histogram. According to the existence of the zero point, the blocks can be categorized as embeddable blocks or non-embeddable blocks, which are recorded by the location map. In the methods proposed by Chang *et al.* [17], the pixel vale at the center position of a block is used as a reference value. Thus the remaining pixels can be further computed as the residual values, and the residual histogram is generated. In the methods proposed by Yin *et al.* [22], a two-stage data embedding scheme is used to change the high (and low) mean values. The difference between the high and low means is used to generate a histogram. The first and second peak points are used to complete histogram-modified embedding.

Table 1 summarizes the overall comparison of the proposed method with three other methods. All the four methods are reversible data hiding schemes. Table 1 indicates that the proposed method has the highest payload. In the method proposed by Liu *et al.*, AMBTC-BPTI must record the peak (requiring 4 bits) and zero (requiring 4 bits) points for every

TABLE 2. The number of different AVN blocks in the six test images.

Number of blocks	Test Images					
	North Island	Sailboat	F-16	House	Lena	Peppers
Blocks with AVN = 1	0	0	7	7	0	0
Blocks with AVN = 2	131	190	343	394	183	161
Blocks with AVN = 3	1438	1518	1469	1354	1533	1594
Blocks with AVN = 4	479	340	229	293	332	293
Total	2048	2048	2048	2048	2048	2048

block and the location map. In the method proposed by Chang *et al.*, each block must record two pairs of peak (requiring 8 bits) and zero (requiring 8 bits) points. In the method proposed by Yin *et al.*, 1 bit must record the overflow problem. In the proposed method, each block records a pair of peak and zero points. Because of the proposed AVN-BPTI approach, the memory required for the peak and zero points is decreased and AVN information does not require additional memory for storage. Each block can carry data. Therefore, no location map is required. Although the method proposed by Yin *et al.* has the lowest Bpp, the proposed method has ten times higher payload than Yin *et al.*'s method (averagely 22451 bits versus 1801 bits). Moreover, the proposed method exhibits the highest efficiency and the least mean squared error (MSE) rate among all the methods.

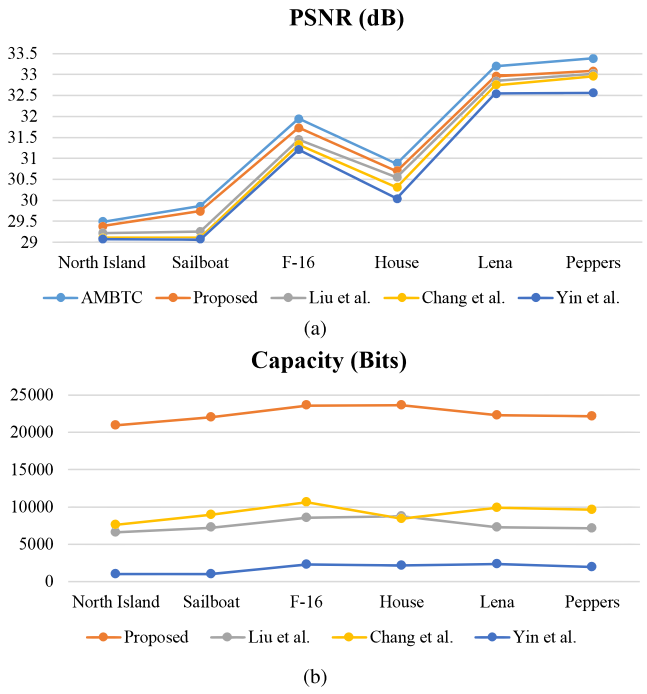


FIGURE 7. Comparisons among different methods. (a) PSNR, and (b) Embedding capacity.

Fig. 7(a) illustrates a comparison of the four methods in terms of the image quality (PSNR). The uncontaminated AMBTC images have optimal image quality because they do not embed the secret data. The proposed method has the second-best PSNR, and its PSNR value is considerably close to that of the uncontaminated AMBTC. Because a two-stage encryption procedure is used in the method proposed by Yin *et al.*, the high and low mean values change significantly, which results in the lowest PSNR. Compared with the other three data hiding methods, the proposed method has the maximum data capacity in all the six test images (Fig. 7b). Fig. 8 displays the stego AMBTC compressed image using the proposed method.

In Section III(B), an AVN-BPTI scheme is proposed, which adaptively selects the appropriate N to generate the N-BP residual histogram. For each test image, two 128×128 mean tables are present (high and low means). Therefore, each test image comprises 2048 micro-blocks. Table 2 presents the number of different AVN blocks in the six test images. As shown in Table 2, blocks with AVN $=3$ have the highest occurrence in the six test images. In general, a high data capacity is observed for a low AVN. The F-16 and House images have a relatively high number of blocks from AVN $=1$ to AVN $=3$. Thus, they exhibit the largest data capacity among the six test images. Moreover, the proposed AVNBPTI scheme reduces the memory required for storing the peak and zero points. Therefore, the F-16 and House images have the highest efficiency among the six test images (as shown in Table 1).

To reinforce stego image protection, chaotic functions are used to encrypt the stego AMBTC image. As shown in Fig. 9, the test image Lena is used to demonstrate the overall process of the proposed method. The memory size of the input grayscale image is $512 \times 512 \times 8$ bits (Fig. 9a). By performing AMBTC compression and data hiding, the size of the stego AMBTC (Fig. 9b) becomes $128 \times 128 \times 32$ bits, which has the image compression ratio of four. The stego AMBTC is transformed into a scrambled stego image with $256 \times 256 \times 8$ bits (Fig. 9c). After chaotic encryption, the cipher image has dimensions of $256 \times 256 \times 8$ bits (Fig. 9d). As displayed in Figs. 9e and 9f, the histograms of the original grayscale and stego AMBTC images are significantly similar. Therefore, the third party does not anticipate that the secret data are embedded. As illustrated in Fig. 9g, the histogram of the scrambled stego image is irregular and different from that of the stego AMBTC image. In addition, the information entropy is calculated to evaluate the randomness of pixel

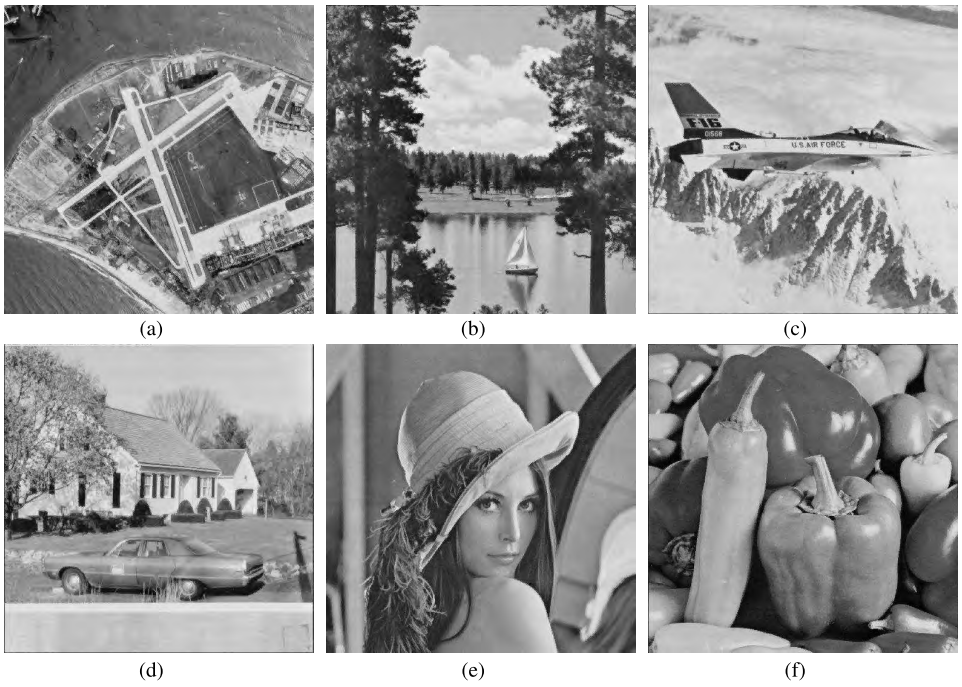


FIGURE 8. The resulting stego AMBTC images using the proposed method. (a) Stego North Island (embedding 20946 bits, PSNR = 29.38 dB). (b) Stego Sailboat (embedding 22050 bits, PSNR = 29.74 dB). (c) Stego F-16 (embedding 23610 bits, PSNR = 31.73 dB). (d) Stego House (embedding 23671 bits, PSNR = 30.69 dB). (e) Stego Lena (embedding 22287 bits, PSNR = 32.96 dB). (f) Stego Peppers (embedding 22142 bits, PSNR = 33.09 dB).

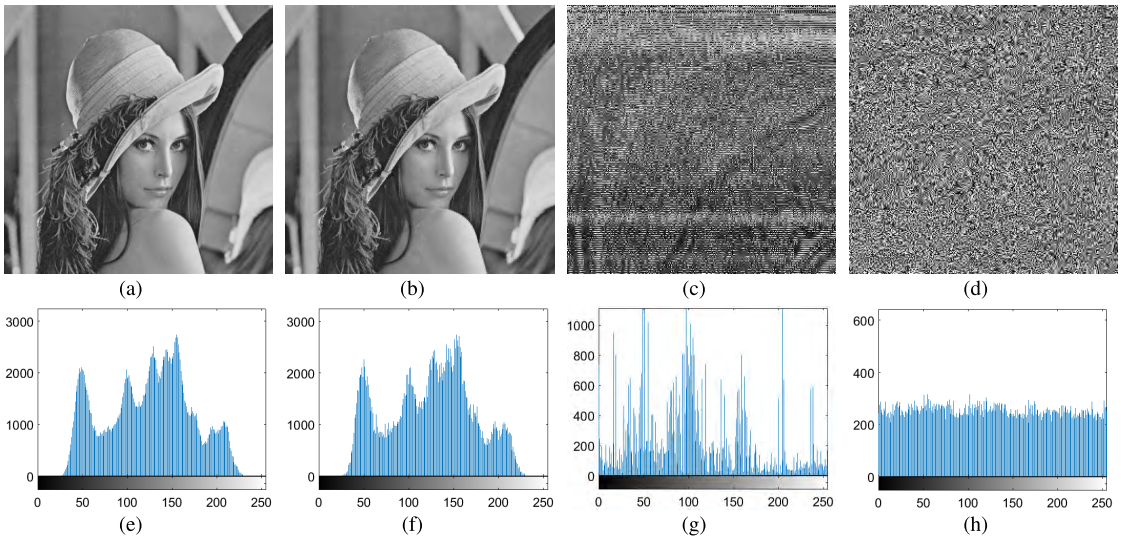


FIGURE 9. (a) Original grayscale image, (b) stego AMBTC image, (c) scrambled image, (d) cipher image, (e) histogram of (a), (f) histogram of (b), (g) histogram of (c), and (h) histogram of (d).

distribution as follows:

$$Entropy(s) = \sum_{i=0}^{255} p(s_i) \log_{10} \frac{1}{p(s_i)}, \tag{28}$$

where $p(s_i)$ indicates the probability of a pixel with pixel value i. According to the analysis of entropy, from Figs. 9(e) to 9(h), the entropy values are 7.4451, 7.4505, 7.1558, and 7.9873, respectively. Because the maximum information entropy of a 8-bit image is eight, the entropy result of Fig. 9(h) demonstrates that the chaotic mapping effectively increases the randomness of pixel distribution. As depicted in Fig. 9h, the histogram of the cipher image is uniform and contains no characteristics of the original image, which makes it increasingly difficult for a malicious attacker to extract information.

V. CONCLUSION

With the development of smart city and cloud computing, a large amount of digital images are distributed through the Internet worldwide daily. The topic of information security which prevents unauthorized use, copy, and disclosure becomes important and has widespread applications. Because most images on the Internet are compressed images, this work proposes a reversible data hiding method for the AMBTC compressed images. A chaos-based image encryption scheme is further combined with the proposed data hiding method. Therefore, the security of transiting them over public networks is enhanced and protected. From the experimental results, we demonstrate the superiority of the proposed method by using different objective evaluation measures and comparing with existing methods.

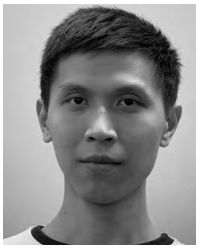
REFERENCES

[1] E. Delp and O. Mitchell, "Image compression using block truncation coding," *IEEE Trans. Commun.*, vol. 27, no. 9, pp. 1335–1342, Sep. 1979.
[2] M. Lema and O. Mitchell, "Absolute moment block truncation coding and its application to color images," *IEEE Trans. Commun.*, vol. 32, no. 10, pp. 1148–1157, Oct. 1984.
[3] Y. Qiu, Z. Qian, and L. Yu, "Adaptive reversible data hiding by extending the generalized integer transformation," *IEEE Signal Process. Lett.*, vol. 23, no. 1, pp. 130–134, Jan. 2016.
[4] H. Chen, J. Ni, W. Hong, and T.-S. Chen, "High-fidelity reversible data hiding using directionally enclosed prediction," *IEEE Signal Process. Lett.*, vol. 24, no. 5, pp. 574–578, May 2017.
[5] Z. Qian, H. Zhou, X. Zhang, and W. Zhang, "Separable reversible data hiding in encrypted JPEG bitstreams," *IEEE Trans. Dependable Secure. Comput.*, vol. 15, no. 6, pp. 1055–1067, Dec. 2018. [Online]. Available: https://ieeexplore.ieee.org/document/7763833
[6] D. Hou, W. Zhang, Y. Yang, and N. Yu, "Reversible data hiding under inconsistent distortion metrics," *IEEE Trans. Image Process.*, vol. 27, no. 10, pp. 5087–5099, Oct. 2018.
[7] P. Rahmani and G. Dastghaibyfard, "Two reversible data hiding schemes for VQ-compressed images based on index coding," *IET Image Process.*, vol. 12, no. 7, pp. 1195–1203, Jul. 2018.
[8] D. Xu, R. Wang, and Y. Shi, "An improved scheme for data hiding in encrypted H.264/AVC videos," *J. Vis. Commun. Image Represent.*, vol. 36, pp. 229–242, Apr. 2016.
[9] C.-N. Yanga, S.-C. Hsua, and C. Kim, "Improving stego image quality in image interpolation based data hiding," *Comput. Standards Interfaces*, vol. 50, pp. 209–215, Feb. 2017.
[10] T. Rabie and I. Kamel, "High-capacity steganography: A global-adaptive-region discrete cosine transform approach," *Multimedia Tools Appl.*, vol. 76, no. 5, pp. 6473–6493, Mar. 2017.
[11] K. Jung, "Data hiding scheme improving embedding capacity using mixed PVD and LSB on bit plane," *J. Real-Time Image Process.*, vol. 14, no. 1, pp. 127–163, Jan. 2018.
[12] D. Wang, W. Sun, S. Yu, L. Li, and W. Liu, "A novel background-weighted histogram scheme based on foreground saliency for mean-shift tracking," *Multimedia Tools Appl.*, vol. 75, no. 17, pp. 10271–10289, Sep. 2016.
[13] M. Li and Y. Li, "Histogram shifting in encrypted images with public key cryptosystem for reversible data hiding," *Signal Process.*, vol. 130, pp. 190–196, Jan. 2017.
[14] J. Wang, J. Ni, X. Zhang, and Y.-Q. Shi, "Rate and distortion optimization for reversible data hiding using multiple histogram shifting," *IEEE Trans. Cybern.*, vol. 47, no. 2, pp. 315–326, Feb. 2017.
[15] Z. Ni, Y.-Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 3, pp. 354–362, Mar. 2006.
[16] P. Tsai, Y.-C. Hu, and H.-L. Yeh, "Reversible image hiding scheme using predictive coding and histogram shifting," *Signal Process.*, vol. 89, no. 6, pp. 1129–1143, Feb. 2009.
[17] I.-C. Chang, Y.-C. Hu, W.-L. Chen, and C.-C. Lo, "High capacity reversible data hiding scheme based on residual histogram shifting for block truncation coding," *Signal Process.*, vol. 108, pp. 376–388, Mar. 2015.
[18] A. Kumar and A. Kumar, "A cell-array-based multibiometric cryptosystem," *IEEE Access*, vol. 4, pp. 15–25, Feb. 2016.
[19] Y. Lee, Y.-S. Kim, and J.-S. No, "Ciphertext-only attack on linear feedback shift register-based Esmaeili-Gulliver cryptosystem," *IEEE Commun. Lett.*, vol. 21, no. 5, pp. 971–974, May 2017.
[20] A. Belazi, M. Khan, A. Abd El-Latif, and S. Belghith, "Efficient cryptosystem approaches: S-boxes and permutation–substitution-based encryption," *Nonlinear Dyn.*, vol. 87, no. 1, pp. 337–361, Jan. 2017.
[21] L. Liu, C.-C. Chang, and A. Wang, "Reversible data hiding scheme based on histogram shifting of *n*-bit planes," *Multimedia Tools Appl.*, vol. 75, no. 18, pp. 11311–11326, Sep. 2016.
[22] Z. Yin, X. Niu, X. Zhang, J. Tang, and B. Luo, "Reversible data hiding in encrypted AMBTC images," *Multimedia Tools Appl.*, vol. 77, no. 14, pp. 18067–18083, Jul. 2018.

HSIANG-YING WANG received the M.S. degree in business administration from National Sun Yat-sen University, Kaohsiung, Taiwan, in 2010. She is currently pursuing the Ph.D. degree in mechanical and electrical engineering with the National Taipei University of Technology, Taipei, Taiwan. She is having more than 20 years solid working experience.

Since 2015, she has been the Assistant Research Fellow with the Research Division of Business Model Innovation, Commerce Development Research Institute, Taipei, Taiwan. Her current research interests include computer vision, vision-based automation, pattern recognition, color image processing, imaging systems, machine learning, and human–computer interaction.

HSIN-JU LIN received his B.S. degree in electronic engineering from National Taiwan Ocean University, Keelung, Taiwan, in 2016, and the M.S. degree from the Graduate Institute of Automation Technology, National Taipei University of Technology, Taipei, Taiwan, in 2019. His current research interests include computer vision and information security.



XIANG-YUN GAO received the B.S. degree in electronic engineering from Chang Gung University, Taoyuan, Taiwan, in 2017. He is currently pursuing the M.S. degree with the Graduate Institute of Automation Technology, National Taipei University of Technology, Taipei, Taiwan. His current research interests include digital image processing and information security.

WEN-HUANG CHENG received the B.S. and M.S. degrees in computer science and information engineering from National Taiwan University, Taipei, Taiwan, in 2002 and 2004, respectively, and the Ph.D. degree (Hons.) from the Graduate Institute of Networking and Multimedia, in 2008. He is currently a Professor with the Institute of Electronics, National Chiao Tung University (NCTU), Hsinchu, Taiwan, where he is also the Founding Director with the Artificial Intelligence and Multimedia Laboratory. Before joining NCTU, he led the Multimedia Computing Research Group, Research Center for Information Technology Innovation, Academia Sinica, Taipei, Taiwan, from 2010 to 2018. His current research interests include multimedia, artificial intelligence, computer vision, machine learning, social media, and financial technology. He has received numerous research and service awards, including the K. T. Li Young Researcher Award from the ACM Taipei/Taiwan Chapter in 2014, the Top 10% Paper Award from the 2015 IEEE MMSP, the Outstanding Youth Electrical Engineer Award from the Chinese Institute of Electrical Engineering in 2015, the 2016 Y. Z. Hsu Scientific Paper Award, the 2017 Ta-Yu Wu Memorial Award from Taiwan's Ministry of Science and Technology, the 2017 Significant Research Achievements of Academia Sinica, the 2018 MSRA Collaborative Research Award, and the Outstanding Reviewer Award of 2018 IEEE ICME. He is an APSIPA Distinguished Lecturer.

YUNG-YAO CHEN received the B.S. and M.S. degrees in electrical and control engineering from National Chiao Tung University, in 2004 and 2006, respectively, and the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN, USA, in 2013. He has worked in HP Labs–Printing and Content Delivery Lab (HPL–PCDL) as a Full-Time Research Intern for 1 year, from 2012 to 2013. He has been an Associate Professor with the Graduate Institute of Automation Technology and the Chief of the Chinese Language Training Center, National Taipei University of Technology (NTUT), Taipei, Taiwan, since 2018. His current research interests include computer vision, vision-based automation, pattern recognition, color image processing, imaging systems, machine learning, and human–computer interaction.

Dr. Chen is a member of Golden Key International Honor Society and Phi Tau Phi. He was a recipient of the First Prize Paper Award of the 2015 International conference on Advanced Robotics and Intelligent Systems, the Best Paper Award of the 2018 International Conference on Communications, Computation, Network and Technologies, and the Outstanding Reviewer Award of 2018 IEEE International Conference on Visual Communications and Image Processing.