IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 19, 2019, accepted January 1, 2020, date of publication January 13, 2020, date of current version February 13, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2966231

ABMC-DH: An Adaptive Bit-Plane Data Hiding Method Based on Matrix Coding

ZHAN YU[1,2], CHIA-CHEN LIN[3], AND CHIN-CHEN CHANG[4,5], (Fellow, IEEE)

[1]School of Electronics and Information Engineering, Fuqing Branch of Fujian Normal University, Fuzhou 350300, China
[2]Engineering Research Center for ICH Digitalization and Multi-Source Information Fusion (Fuqing Branch of Fujian Normal University), Fujian Province University, Fuzhou 350300, China
[3]Computer Science and Information Management, Providence University, Taichung 43301, Taiwan
[4]Department of Information Engineering and Computer Science, Feng Chia University, Taichung 40724, Taiwan
[5]School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China

Corresponding authors: Chia-Chen Lin (mhlin3@pu.edu.tw) and Chin-Chen Chang (alan3c@gmail.com)

This work was partially supported by the Education-Scientific Research Project for Middle-aged and Young of Fujian Province under Grant No. JAT170683, No. JAT190478, and No. JAS 180625, the project of Ministry of Science and Technology of Taiwan under Grant No. MOST 106-2221-E-035-013-MY3, MOST 108-2410-H-126-021, and MOST 106-2221-E-035-018-MY3.

ABSTRACT This paper proposes a novel adaptive image steganography method combining matrix coding, which can achieve better visual quality according to the given embedding rate compared with other existing schemes. The embedding processes are divided into two phases: preparation and implementation. In the preparation phase, all 128 combinations of a 7-bit binary number are classified into eight groups by a parity check matrix based on matrix coding. Thus, each group corresponds to an octal number, and contains 16 candidates. Correspondingly, all pixels of a cover image are classified into three categories by a predefined embedding rate level: low, middle, and high. In the implementation phase, each pixel in the high level can be embedded with 3 bits by substituting its 7 LSBs (Least Significant Bit) using a close candidate from the group determined by the to-be-embedded 3 bits. For the middle level, pixel pairs are used to hide 3-bit data by substituting its (3, 4) LSBs combination using a close candidate form the group determined by the to-be-embedded 3 bits. For pixels in the low level, a triple pixel combination is used to embed 3-bit data by substituting its (2, 2, 3) LSBs combination using a close candidate form the group determined by the to-be-embedded 3 bits. The experimental results indicate that the proposed scheme is optimal compared to other existing matrix-coding data hiding schemes in terms of visual quality, while providing an identical embedding capacity.

INDEX TERMS Adaptive data hiding, steganography, matrix coding, optimal visual quality.

I. INTRODUCTION

With the development of science and technology, it is increasingly convenient to transmit text messages or digital media on the Internet. If the information or media being transmitted is encrypted in advance, the process of transmission may elicit some security concerns because the encrypted message is unreadable and could spur malicious attention. In addition, as the acquisition, replication and modification of digital media is relatively easy, a number of security problems arise. Digital media may be tampered with, stolen [1], [2], etc., thus the development of multimedia information security technology is increasingly important.

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

Many proposed data hiding schemes [3]–[8] use digital images as a cover media with a data steganographic method to decrease the opportunity of disclosure. Least significant bit (LSB) replacement is a widely used method in data hiding [9], [10]. This method uses secret data to directly substitute the least significant bit of pixels in a cover image. With such a strategy, as long as the first four or five bits are not modified, the modification is not visually noticeable. However, the distortion could be obvious when the embedding strategy is conducted on a fourth bit or fifth bit, which results in obvious distortion. Therefore, additional methods have been proposed to overcome these shortcomings.

For instance, the LSB matching revisited method [11] uses a pair of pixels as a unit to carry two bits of information, and achieves the same payload at less cost than traditional methods. The EMD method [12] uses a group of n grayscale pixels to embed $k - digit$ secret data in a $(2n + 1)$ numeral system thru an extraction function that results in high image quality. A matrix encoding method was firstly proposed by Crandall in 1998 as a state-of-the-art information hiding [13]. In this scheme, it is only necessary to modify one specific $2^k - 1$ consecutive pixels, thus the remaining pixel values are unchanged or increase or decrease by 1 to hide k bit information, and the embedding capacity reaches $\frac{k}{2^k-1}$bpp. in 2007 [14], embeds one more bit of binary secret data than the former, with an unchanged cost, successfully increasing the embedding capacity to $\frac{k+1}{2^k}$bpp. A series of methods based on (7, 4) Hamming code were proposed in subsequent years, and all aimed to increase the embedding capacity [16]–[18]. A high capacity data hiding scheme proposed by Cao et al. in 2016 [15] present an algorithm to hide 3-bit binary secret information into one to three cover pixels, which makes the highest capacity rate reaches 3bpp.

In the past few years, some scholars have proposed different data hiding methods by using compressed code, such as BTC, LZW and VQ [33]–[40] to hide secret data. Lin *et al.* [33] proposed a reversible data hiding method for vector quantization (VQ) compressed images, which utilizes search-order-coding and state-codebook mapping techniques to reduce the size of the VQ index table and maps the indices of blocks in the codebook for more redundancy, thus, it has higher capacity than similar methods. Yu *et al.* [34] proposed a hybrid data hiding method for ABMCT compressed images, which utilizes block features and hybrid embedding strategy to enhance hiding capacity for compressed codes. Malik *et al.* [35], [36] proposed a reversible data hiding method, which utilizes LZW compression algorithm and odd-even strategy to hide secret data into LZW codes. Thus, it has higher hiding capacity and better compression ratio. Kumar *et al.* [37] proposed an improved histogram-shifting-imitated reversible data hiding method, which utilizes the characteristics of Human Visual System (HVS) to group intensity levels and embed segment of secret into corresponding peak point of each group for improving the image quality. Kumar *et al.* [38] proposed a reversible data hiding method, which divides the secret data into 2-bit segments and embeds them using blocks with the size of 2×2 by pixel location. It further provides addition hiding capacity by increasing the layers, without deteriorating the image quality. Kumar *et al.* [40] proposed a reversible data hiding method using MFT coding for LZW codes, which improves hiding capacity by optimally using LZW codes, and utilizes the MFT to encode the cover image and further increase hiding capacity. Jung [22] proposed an improved data hiding scheme where the cover image is firstly divided into consecutive pixel pairs, and then two bit-planes are generated using a modulo operation, where finally the secret bits are embedded into each bit plane using the PVD and LSB techniques separately. A high embedding capacity maintaining satisfying visual quality was achieved thru this aforementioned method. Weng *et al.* [23] proposed an improved k-pass PVO revisable data hiding by utilizing the location relationship of the largest and the second largest or the smallest and the second smallest pixels in a block to increase the number of embeddable pixels. Moreover, the remaining pixels are exploited together with neighbor pixels to increase the estimation accuracy of local complexity. Thereby, the preferable embedding performance was achieved compared with previous. A data hiding method based on block truncation coding using pixel pair matching technique was proposed by Hong [24], which classified the compressed AMBTC blocks into smooth and complex blocks using a predefined threshold, and adopted the bitmap replacement strategy in smooth blocks and Chen et al.'s method [25] in complex blocks, and moreover, two quantization levels were recalculated to minimize distortion and $\log_2^{\partial}$ data was embedded into two quantization levels using a pixel pair matching technique, thus the method achieves better image quality with a high payload and comparable image quality with low payload compared to previous schemes. A high-capacity data hiding method was proposed by Shukla *et al.* [30], which uses arithmetic coding to compress secret data to provide extra capacity and encrypts the compressed data by AES for security, and then divides the cover image into non-overlapping pixel blocks size of 3×3 or 2×2, using a PVD+LSB method to embed equal or more than 3-bits into each pixel. Finally, an optimization strategy achieves the smallest distortion compared to previous methods.

In this paper, we propose an innovative adaptive data hiding scheme that provides fine-tuned adjustment for specific applications with superior performance by classifying all pixels of a cover image are classified into three categories: low, middle and high according to a predefined embedding rate level.

The rest of this paper is organized as follows. A brief overview of the (7, 4) Hamming code is presented in Section II. Our proposed scheme is presented in Section III, and the experimental results and analysis are discussed in Section IV. Finally, conclusions are presented in Section V.

II. RELATED WORK

In this section, two primary techniques, (7,4) Hamming code and matrix coding, are introduced in Subsections II-A and II-B, respectively, to provide readers with background knowledge.

A. THE (7, 4) HAMMING CODE

The (7, 4) Hamming code [21] was first proposed by Richard Hamming in 1950 as a linear error correction code, and it has subsequently been widely used in data hiding as an efficient steganography method to achieve satisfactory image visual quality. The advantage of the (7, 4) Hamming code is that it can detect and correct 1 error bit for one code composed of 4 original bits and 3 parity check bits with the help of a parity check matrix.

Specifically, four original bits d_1, d_2, d_3, d_4 are used to yield three parity check bits p_1, p_2, p_3, by multiplying with the code generator matrix G of the (7, 4) Hamming code.

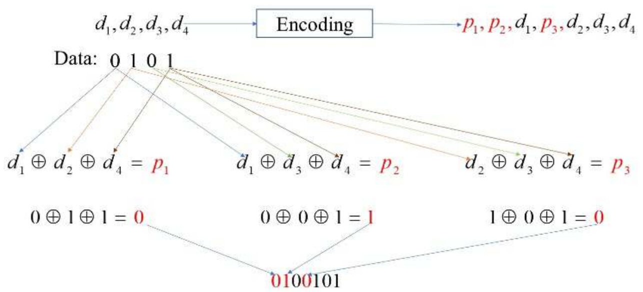


FIGURE 1. The 3 parity check bits of 4 original bits.

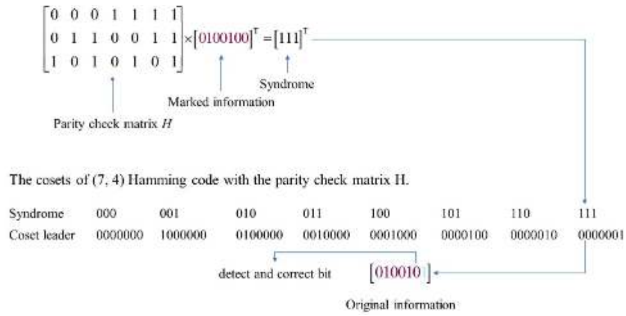


FIGURE 2. The error detection based on (7, 4) Hamming code.

Therefore, one code C with size of 7 is formed by combining 4 original bits with 3 parity bits. The detailed procedure is represented by:

$$\begin{aligned} C &= d \times G \\ &= (d_1, d_2, d_3, d_4) \times \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \\ &= (p_1, p_2, d_1, p_3, d_2, d_3, d_4). \end{aligned} \tag{1}$$

And, three parity check bits p_1, p_2, p_3 can be obtained by the following Eq. (2).

$$\begin{aligned} p_1 &= d_1 \oplus d_2 \oplus d_4, \\ p_2 &= d_1 \oplus d_3 \oplus d_4, \\ p_3 &= d_2 \oplus d_3 \oplus d_4, \end{aligned} \tag{2}$$

where $\oplus$ is the exclusive-or operation.

Fig. 1 is utilized to explain the detailed process of generating C based on the principle of the (7, 4) Hamming code. In Fig. 1, if the original four bits $(d_1, d_2, d_3, d_4) = (0101)_2$ then three parity check bits are $(p_1, p_2, p_3) = (010)_2$. Therefore, $C = (0100101)_2$.

On the decoding side, the receiver can use the same parity check matrix H as in data embedding to detect and correct whether the message has been tampered or not. Assuming the received message is R, Eq. (3) is utilized to judge whether the R is tampered or not by the value of z.

$$z = H \times R^T. \tag{3}$$

where z is called the syndrome vector. Specifically, $z = 0$ indicates the R is not tampered, i.e., $R = C$. Otherwise, R is tampered. Taking $C = (0100101)_2$ for example, if the seventh bit of C is flipped, then the $R = (0100100)_2$. We can calculate $z = (111)_2 = 7$ using Eq. (3), $z \neq 0$ implies that one error bit occurs in the seventh bit of R, and therefore, the original data can be recovered by flipping the seventh bit of R as shown in Fig 2. Finally, $C = (0100101)_2$.

B. MATRIX CODING

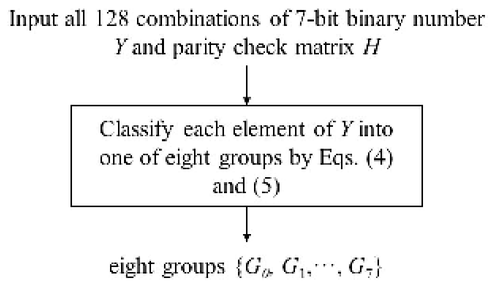
The main idea of matrix code is as follows. To begin, use a pseudo-random number generator to generate a decimal array S which is used to represent the secret data, i.e., $S = \{s_j | j = 1, 2, \cdots, n\}$, where s_j is used to denoted the j-th element of the secret data S, and $s_j \in \{0, 1, \cdots, 7\}$. For the simplicity of a description, we use s to replace s_j by ignoring the subscript of s_j in the rest of this paper. According to the criterion of (7, 4) Hamming code mentioned in Subsection II-A, we collect seven LSBs of n original pixels to form a 7-bit binary number x and embed s into x to generate y by keeping x unaltered or only flipping one bit of x, where x and y are used to denote the original and marked 7-bit binary numbers, respectively, and $s \in \{0, 1, \cdots 7\}$. According to the description above, the advantage of the matrix coding lies in the fact that it can embed 3 bits in x by at most changing one bit of x. After embedding, all the bits of y are appended to the LSBs of n original pixels to form the corresponding stego pixels. On the decoding side, seven LSBs of n stego pixels are extracted to construct y, and then s is extracted via the following equation:

$$s = conv(\bmod(H \times y^T, 2))_{10}. \tag{4}$$

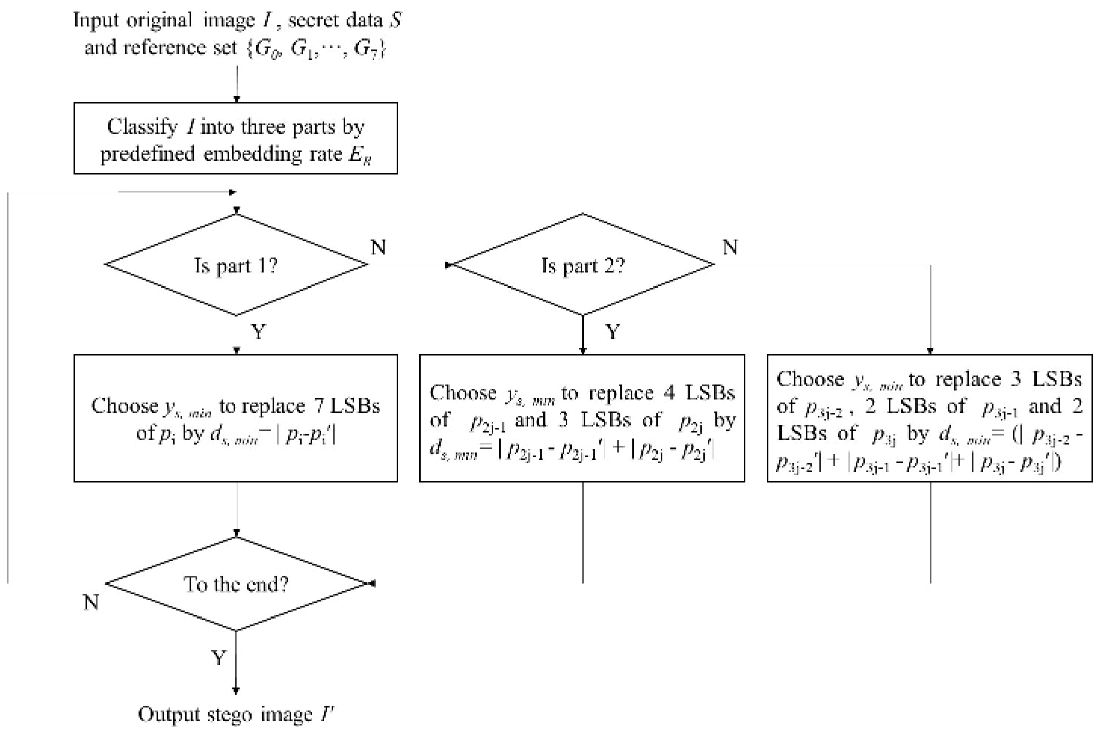
where the superscript T represents the transpose operation, $\bmod(\cdot, 2)$ is the modulo 2 operation which is used to obtain 3-bit binary secret data, the notation $conv(\cdot)$ is a function used to convert numbers from binary representation to decimal representation. If $n = 7$, then the matrix coding can achieve satisfactory embedding performance because only one LSB of seven pixels are modified to embed 3 bits. However, $n \neq 7$ may lead to large modification for some original pixel. Taking $n = 3$ for example, suppose that x is generated by extracting the 2 LSBs (i.e., the 2nd and 1st bits) of the first pixel, two bits (i.e., the 2nd and 1st bits) of the second pixel and three bits (i.e., the 3rd, 2nd and 1st bits) of the third pixel. If the 3rd bit of the third pixel is flipped during data embedding, the embedding distortion for the third pixel is unacceptable. To this end, we proposed an adaptive embedding method.

III. THE PROPOSED SCHEME

The matrix coding technique has been widely used in data hiding in recent years, which can embed three bits by only changing one bit in 7-bit binary data. Therefore, we conclude that matrix coding can achieve higher capacity while introducing less distortion than other existing methods. Inspire by Cao *et al.* [15], we proposed a new data hiding method



(a) The preparation flowchart



(b) The embedding flowchart

FIGURE 3. **(a) The preparation flowchart and (b) embedding flowchart.**

based on matrix coding, called ABMC-DH method, which enhanced hiding capacity by extracting three types of LSBs from the bit-plane of the original pixels as an extension of the matrix coding method. Our method consists of two phases: data embedding and data extraction, which are described in Subsections III-A and III-B, respectively. The embedding procedure is further divided two phases. The first phase is the preparation phase, which is used to construct reference data and classify pixels of the cover image. The second phase is the data hiding phase based on matrix coding, which embeds 3-bit secret data into the LSBs of the original pixels. To give readers a better understanding, five examples are demonstrated at the end of these two subsections.

A. THE EMBEDDING PHASE

This phase can be further classified into five independent sub-processes to satisfy the requirements of various application scenarios in terms of payload and the quality of the stego image. A flowchart is shown in Fig. 3.

1) PREPARATION PHASE

All 128 combinations of 7-bit binary number ranging from 0000000 to 1111111 are classified into 8 groups with the help of H, and thus each group contains 16 combinations, e.g., $G_m = \{y_{1,m}, \cdots, y_{k,m}, \cdots, y_{16,m}\}$, where G_m represents the m-th group, $y_{k,m}$ is the k-th element of G_m, and $m \in \{0, \cdots, 7\}$, $k \in \{1, \cdots, 16\}$. All 128 combinations of a 7-bit binary number denoted by Y are constructed using the following equation

$$\begin{aligned} Y &= \{G_0, \cdots, G_m \cdots, G_7\} \\ &= \{y_{1,0}, \cdots, y_{16,0}, \cdots, \\ &\quad y_{1,m}, \cdots, y_{16,m} \cdots, y_{1,7}, \cdots, y_{16,7}\}, \end{aligned} \tag{5}$$

Eq. (5) indicates that if we want to embed s into x based on (7, 4) Hamming code, we firstly need to know which group s belongs to. Generally, s is classified into group G_s. Afterwards, the difference between x and each element of G_s is calculated, and thus 16 difference values are obtained. Suppose that the difference between x and $y_{k,s}$ is the minimum, therefore, we can utilize that $y_{k,s}$, which is strongly related to x. When y is replaced by $y_{k,s}$ during data embedding, we achieve the optimal quality of stego images because of the high correlation between x and $y_{k,s}$, and also embed s bits into x.

The detailed procedure for generating eight groups is described as follows:

Input: H(The parity check matrix of the (7, 4) Hamming code) and Y (all 128 combinations of 7-bit binary number).

Output: Eight groups $\{G_0, G_1, \cdots, G_7\}$.

Step 1: Exploit the pseudo-random number generator to generate a decimal liner array d ranging from 0 to 127;

Step 2: Import $\{s|0 \leq s \leq 7\}$ which represents the result set of the secret data;

Step 3: Convert the decimal array d into 7-bit binary data array $Y = \{y_{k,m}|1 \leq k \leq 16, 0 \leq m \leq 7\}$;

Step 4: Traverse each element of Y and classify each element into one of eight groups (i.e., $\{G_m|0 \leq m \leq 7\}$) with the help of Eqs. (4) and (5);

Step 5: Go to Step 4 until the last element of Y is classified into one of eight groups;

Step 6: Complete and get eight groups $\{G_0, G_1, \ldots, G_7\}$.

An example is given to demonstrate the classification phase. Assuming s is equal to 6, group G_6 is generated depending on Eqs. (4) and (5), i.e., G_6 = {′0000010′, ′0001101′, ′0010100′, ′0011011′, ′0100111′, ′0101000′, ′0110001′, ′0111110′, ′1000001′, ′1001110′, ′1010111′, ′1011000′, ′1100100′, ′1101011′, ′1110010′, ′1111101′}.

2) DATA HIDING PHASE

The hiding strategies defined in our proposed ABMC-DH method provides a variable payload in an adaptive manner. Specifically, this subsection is divided into five parts: 1. A discussion about the mechanism for adaptive data hiding based on matrix coding; 2. Exploit a single pixel hiding 3-bit secret data; 3. Hide 3-bit secret data by a pixel pair; 4. Embed 3-bit secret data to a combination of three pixels; and finally 5. The proposed general framework of secret data hiding based on above three data hiding scenarios.

a: DEFINITIONS OF THREE DATA HIDING LEVELS

The proposed method focuses on pursuing optimal image visual quality for a given embedding rate. Before introducing the mechanism of the proposed method, we need to give a detailed definition of E_R. Specifically, the embedding rate E_R of the cover image is an important parameter used to evaluate the performance of an embedding algorithm which refers to the ratio of the total pure payload to the image size:

$$E_R = \frac{L}{H \times W} (bpp), \tag{6}$$

where bpp (namely bit-per pixel) presents the unit embedding capacity of the cover image.

In the proposed method, all the pixels of cover image I are divided into three category levels: high, middle and low. The percentages of the pixels belonging to the high, middle, low levels in I are N_1, N_2 and N_3, respectively. The we have $N_1 + N_2 + N_3 = 1$. The proposed method focuses on pursuing optimal image visual quality according to a given E_R. Therefore, we need to get three optimal percentages N_1^*, N_2^* and N_3^* by experiments which can achieve the highest *PSNR* (Peak Signal-to-Noise Ratio) value under a given E_R for an image.

As mentioned previously, one pixel in the high level can carry three bits, two pixels in the middle level can carry two bits and three pixels in the low level can carry a bit, and therefore, the total E_R provided by the pixels belonging to three levels is calculated as follows:

$$3 \times N_1 + 1.5 \times N_2 + N_3 = E_R, \tag{7}$$

For any given E_R, there is an optimal combination of percentage for the pixels. For example, when $E_R = 1$, we easily know that $N_1 = 0, N_2 = 0, N_3 = 1$ satisfy Eq. (7). $N_3 = 1$ means that all the pixels are split into the low level. Since only one of every three pixels is modified to carry 3 bits, the cover image can provide the highest visual quality while maintaining the lowest distortion. When $E_R = 3$, we select $N_1 = 1, N_2 = 0, N_3 = 0$, which implies that all pixels are divided into the high level. Thus, the cover image can provide the highest embedding capacity while introduce the lowest embedding distortion. For $E_R = 1.5$, there exist multiple combinations of N_1, N_2 and N_3 which can achieve the given E_R. Experimental results indicate that when all the pixels are separated into the middle level (i.e., $N_2 = 1$), our method can acquire optimal visual quality compared with other methods based on matrix coding. These three cases can be presented by the following equation:

$$\begin{cases} N_1 = 0, N_2 = 0, N_3 = 1, & \textit{if } E_R = 1, \\ N_1 = 0, N_2 = 1, N_3 = 0, & \textit{if } E_R = 1.5, \\ N_1 = 1, N_2 = 0, N_3 = 0, & \textit{if } E_R = 3. \end{cases} \tag{8}$$

b: THE DATA HIDING PROCEDURE FOR HIGH LEVEL

According to the description mentioned in subsection III-A2.a, when $N_1 = 1$, the data hiding level is defined as high level and the data hiding procedure is performed in a pixel-by-pixel manner. Since each pixel can be embedded with 3 bits, the maximum pure payload can be achieved. Here, we will give a detailed introduction on how to embed 3 bits into each pixel during data embedding.

Input: A cover image I with the size of $H \times W$, the reference set $\{G_0, G_1, \cdots, G_7\}$.

Output: Stego image I'.

Step 1: Scan the cover image I according to the raster scan order to get a pixel list $\{p_i | i = 1, 2, \cdots, H \times W\}$;

Step 2: Exploit the pseudo-random number generator to generate a decimal liner array $S = \{s_i | i = 1, 2, \cdots, H \times W\}$ ranging from 0 to 7, where s_i denotes the i-th 3-bit data to be embedded;

Step 3: Convert p_i to its 8-bits binary representation, extract 7 bits (i.e., the 7th, 6th, 5th, 4th, 3rd, 2nd, 1st LSBs) of p_i and assign these 7 bits to x_i, where i in $\{1, H \times W\}$;

Step 4: Obtain the reference set G_s by Eq. (5);

Step 5: Calculate the difference value $d_{s,k} = |x_i - y_{s,k}|$, where $k \in \{1, 2, \cdots, 16\}$, $|\cdot|$ represents the absolute operator;

Step 6: Select the minimum difference $d_{s,min}$ from $\{d_{s,1}, d_{s,2}, \cdots, d_{s,16}\}$, and p'_i is generated by replacing 7 bits of p_i with $y_{s,min}$.

Step 7: Repeat Step 4 to 6 until the last pixel (i.e., $x_{H\times W}$) is embedded, and the stego image I' is obtained.

An example is now given to demonstrate the data hiding process corresponding to $N_1 = 1$ shown in Fig. 4 (a). Suppose that $p_1 = 15$ and $s = 5$, and reference set $\{G_0, G_1, \ldots, G_7\}$. If we want to embed the secret data $s_1 = 5$ into the pixel $p_1 = 15$, firstly, we convert p_1 to its 8-bit binary representation, and then extract its 7 bits to generate $x_1 = (0001111)_2 = 15$. Where the subscript 2 indicates a binary representation of x_1; next, we get G_5 from the reference set $\{G_0, G_1, \ldots, G_7\}$ due to $s = 5$, then traverse all 16 elements of G_5 one-by-one and calculate the difference value between x_1 and $y_{5,k}$, where $k \in \{1, 2, \cdots, 16\}$; after 16 difference values are obtained. The difference value $d_{5,3} = |x_1 - y_{5,3}| = 3$ is the smallest, where $y_{5,3} = (0010010)_2 = 18$; finally, $p'_i = 18$.

c: DATA HIDING PROCEDURE FOR MIDDLE LEVEL

$N_2 = 1$ means the data hiding level is middle according to subsection 2.1 and every two pixels constructing a pixel pair to carry 3 secret bits. Similar to $N_1 = 1$, we will give a detailed introduction on how to embed 3 bits into a pixel pair during data embedding.

Input: Cover image I with a size of $H \times W$, and reference set $\{G_0, G_1, \cdots, G_7\}$.

Output: Stego image I'.

Step 1: Scan cover image I according to the raster scan order to get a pixel list $\{p_i | i = 1, 2, \cdots, H \times W\}$, and then for every two pixels constructs one pixel pair (p_{2j-1}, p_{2j}), where $j \in \{1, 2, \cdots, \lfloor \frac{H\times W}{2} \rfloor\}$;

Step 2: Exploit the pseudo-random number generator to generate a decimal liner array $S = \{s_i | i=1, 2, \cdots, \lfloor \frac{H\times W}{2} \rfloor\}$ ranging from 0 to 7, where s_i denotes the i-th 3-bit data to be embedded;

Step 3: Convert the pixel pair (p_{2j-1}, p_{2j}) to its 8-bit binary representation (b_{2j-1}, b_{2j});

Step 4: Obtain the reference set G_s depending on Eq. (5);

Step 5: b_{2j-1} is changed to $b'_{2j-1,k}$ by replacing the last 4 bits (i.e., the 4th, 3rd, 2nd, 1st LSBs) of b_{2j-1} with the first 4 bits (i.e., the 7th, 6th, 5th and 4th LSBs) of $y_{s,k}$, and simultaneously, b_{2j} is modified to $b'_{2j,k}$ by replacing the last 3 bits (i.e., the 3rd, 2nd, 1st LSBs) of b_{2j} with the last 3 bits (i.e., the 3rd, 2nd, 1st LSBs) of $y_{s,k}$, where $k \in \{1, \cdots, 16\}$; next, the binary pixel pair $\left(b'_{2j-1,k}, b'_{2j,k}\right)$ is converted to its decimal representation $\left(p'_{2j-1,k}, p'_{2j,k}\right)$. Afterwards, calculate the difference value $d_{s,k}$ between (p_{2j-1}, p_{2j}) and $\left(p'_{2j-1,k}, p'_{2j,k}\right)$ by the following equation, $d_{s,k} = \left|p_{2j-1} - p'_{2j-1,k}\right| + \left|p_{2j} - p'_{2j,k}\right|$;

Step 6: Select the minimum difference $d_{s,min}$ from $\{d_{s,1}, d_{s,2}, \cdots, d_{s,16}\}$, and p'_{2j-1} is generated by replacing the last four bits of p_{2j-1} with $y_{s,min}$, at the same time, p'_{2j} is generated by replacing the last three bits of p_{2j} with the last three bits of $y_{s,min}$.

Step 7: Repeat Step 4 to 6 until the last pixel pair is embedded with 3 bits, and stego image I' is obtained.

An example is now given to demonstrate the process of data hiding corresponding to $N_2 = 1$ shown in Fig. 4 (b). Assume that cover image I = [15,15,17,9; 17,15,10,12; 20,9,15,7; 17,10,10,9], and reference set $\{G_0, G_1, \ldots, G_7\}$, with secret data S = [5,2,5,5,3,4,5,7]. The first pixel pair $(p_1, p_2) = (15, 15)$ is converted to its binary representation $(0001111, 0001111)_2$. Since the first to-be-embedded data is 5, the reference set $G_5 = \{y_{5,1}, \cdots, y_{5,16}\}$ is chosen to find the most similar substitute for (p_1, p_2). Specifically, taking $y_{5,1} = (0000100)_2$ for example, $p'_{1,1} = (0000000)_2$ is obtained by modifying the four bits (i.e., the 4th, 3rd, 2nd, 1st LSBs) of p_1 with the first four bits of $y_{5,1}$, and meanwhile, $p'_{2,1} = (0001100)_2$ is obtained by modifying three bits (i.e., the 3rd, 2nd, 1st LSBs) of p_2 with the last three bits of $y_{5,1}$. The distortion between (p_1, p_2) and $\left(p'_{1,1}, p'_{2,1}\right)$ can be computed as $d_{5,1} = |15 - 0| + |15 - 12| = 18$. Similar to $d_{5,1}$, all other 15 distortion values $\{d_{5,k}\} | 2 \leq k \leq 16$ are obtained. Therefore, we can easily find the minimum distortion value $d_{5,4} = 4$ and the corresponding element $y_{5,4} = (1101101)_2$. Finally, (p_1, p_2) is modified to $(13, 13)$ by means of Step 6. After all original pixels are modified to the corresponding marked pixel pairs, stego image I' = [13,13,17,11; 16,14,10,9; 20,8,15,6; 17,10,11,9] is generated.

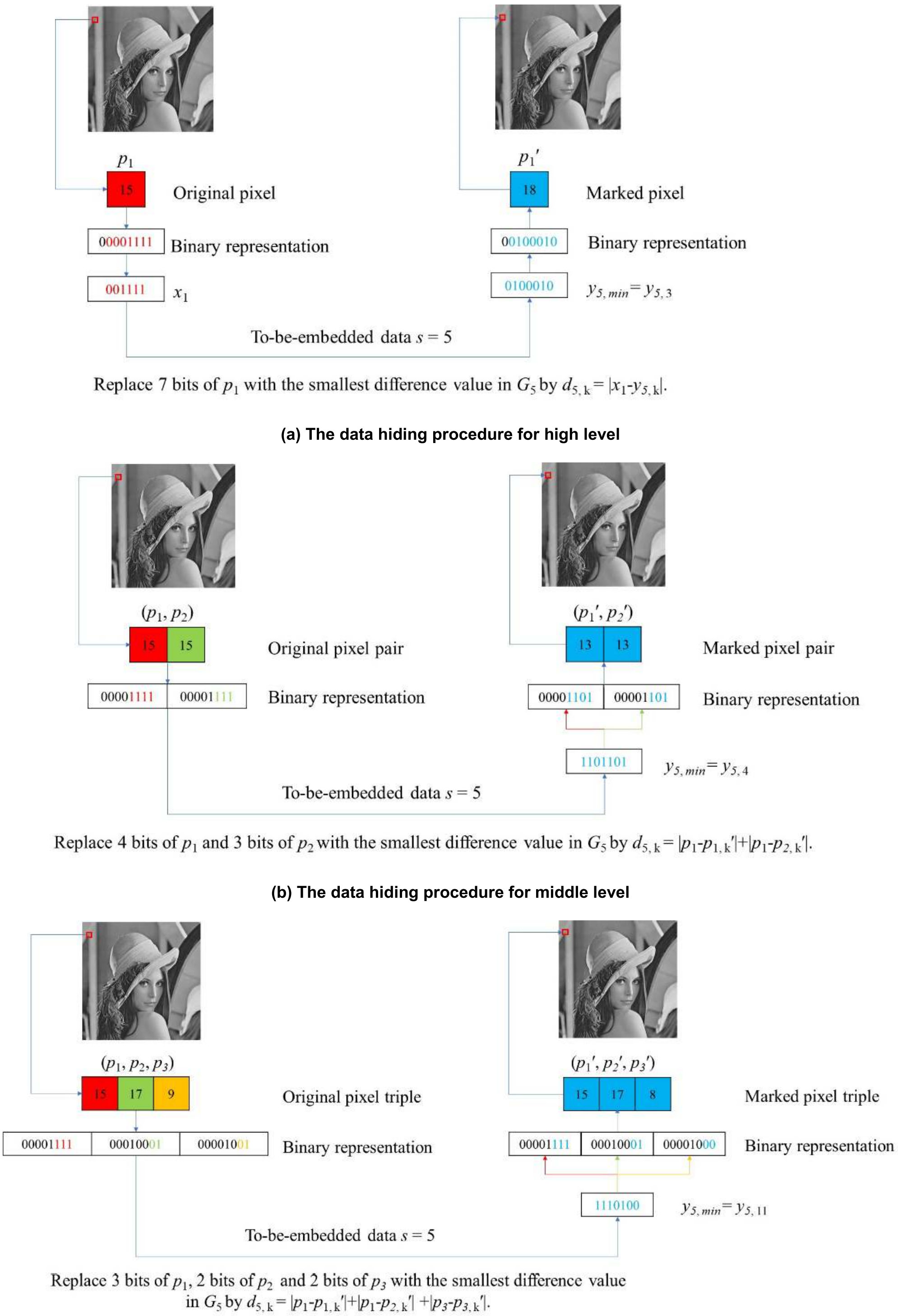
d: THE DATA HIDING PROCEDURE FOR LOW LEVEL

$N_3 = 1$ means that every three pixels constructing a pixel triple can be embedded with 3 bits for low level defined in subsection 2.1. In other words, since a pixel triple is embedded with 3 bits, the embedding distortion is the smallest compared with $N_1 = 1$ and $N_2 = 1$. Similar to $N_1 = 1$ and $N_2 = 1$, we will give a detailed introduction on how to embed 3 bits into a pixel triple during data embedding.

Input: Cover image I with a size of $H \times W$, and reference set $\{G_0, G_1, \cdots, G_7\}$.

Output: Stego image I'.

Step 1: Scan cover image I according to the raster scan order to generate a pixel list $\{p_i | i = 1, 2, \cdots, H \times W\}$,



(c) The data hiding procedure for low level

FIGURE 4. The data hiding procedure for (a) high level, (b) middle level and (c) low level.

and then every three pixels constructs one pixel triple $(p_{3j-2}, p_{3j-1}, p_{3j})$, where $j \in \{1, 2, \cdots, \lfloor \frac{H \times W}{3} \rfloor\}$;

Step 2: Exploit the pseudo-random number generator to generate a decimal liner array $S = \{s_i | i = 1, 2, \cdots, \lfloor \frac{H \times W}{3} \rfloor\}$ ranging from 0 to 7, where s_i denotes the i-th 3-bit data to be embedded;

Step 3: Convert the pixel triple $(p_{3j-2}, p_{3j-1}, p_{3j})$ to its 8-bit binary representation $(b_{3j-2}, b_{3j-1}, b_{3j})$;

Step 4: Obtain reference set G_s depending on Eq. (5);

Step 5: b_{3j-2} is changed to $b'_{3j-2,k}$ by replacing the 3rd, 2nd and 1st LSBs of b_{3j-2} with the 7th, 6th and 5th LSBs of $y_{s,k}$, b_{3j-1} is modified to $b'_{3j-1,k}$ by

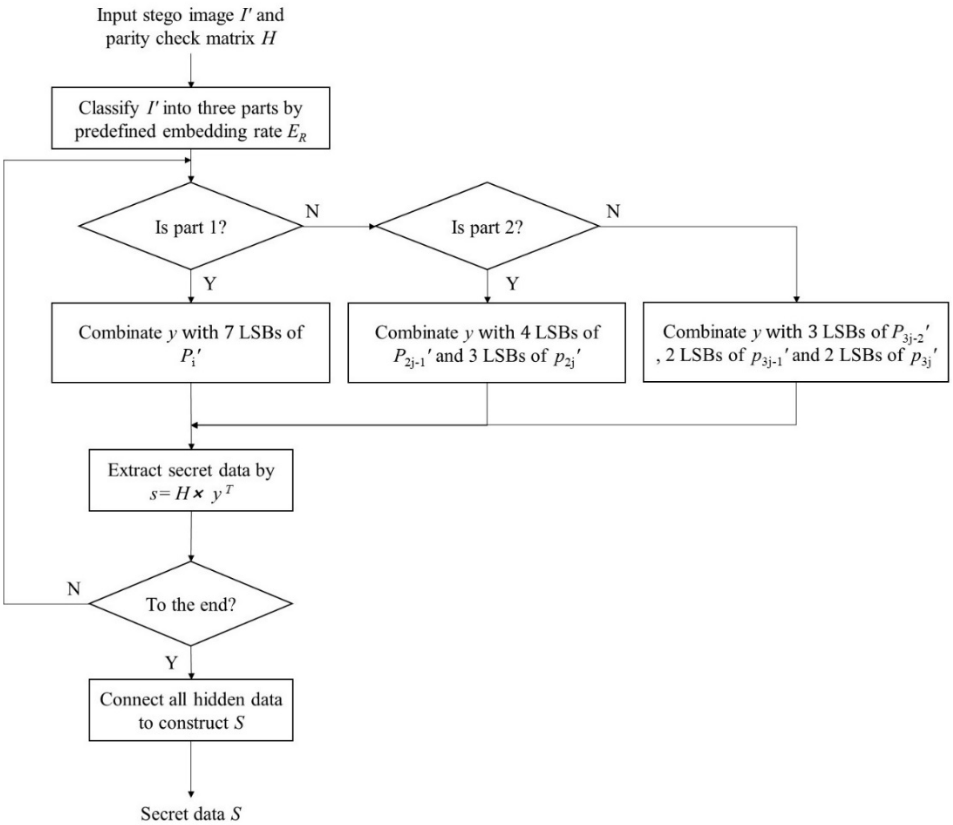


FIGURE 5. The extracting flowchart.

replacing the 2nd and 1st LSBs of b_{3j-1} with the 4th and 3rd LSBs of $y_{s,k}$, and simultaneously, b_{3j} is modified to $b'_{3j,k}$ by replacing the 2nd and 1st LSBs with the 2nd and 1st LSBs of $y_{s,k}$, where $k \in \{1, 2, \cdots, 16\}$; next, the binary pixel triple $\left(b'_{3j-2,k}, b'_{3j-1,k}, b'_{3j,k}\right)$ is converted to its decimal representation $\left(p'_{3j-2,k}, p'_{3j-1,k}, p'_{3j,k}\right)$; afterwards, Calculate the difference value $d_{s,k}$ between $(p_{3j-2}, p_{3j-1}, p_{3j})$ and $\left(p'_{3j-2,k}, p'_{3j-1,k}, p'_{3j,k}\right)$ by the following equation, $d_{s,k} = \left|p_{3j-2} - p'_{3j-2,k}\right| + \left|p_{3j-1} - p'_{3j-1,k}\right| \left|p_{3j} - p'_{3j,k}\right|$;

Step 6: Select the minimum difference $d_{s,min}$ from $\left\{d_{s,1}, d_{s,2}, \cdots, d_{s,16}\right\}$, and p_{3j-2} is changed to p'_{3j-2} by replacing the 3rd, 2nd and 1st LSBs of p_{3j-2} with the 7th, 6th and 5th LSBs of $y_{s,min}$, p_{3j-1} is modified to p'_{3j-1} by replacing the 2nd and 1st LSBs of p_{3j-1} with the 4th and 3rd LSBs of $y_{s,min}$, and at the same time, p_{3j} is modified to p'_{3j} by replacing the 2nd and 1st LSBs of p_{3j} with the 2nd and 1st LSBs of $y_{s,min}$;

Step 7: Repeat Step 4 to 6 until the last pixel triple is embedded with 3 bits, and stego image I' is obtained.

An example is now given to demonstrate the process of data hiding corresponding to $N_3 = 1$ shown in Fig. 4 (c). Assume that cover image $I = [15,17,9; 17,10,12; 20,15,7]$, reference set $\{G_0, G_1, \ldots, G_7\}$, with secret data $S = [5,2,5]$. The first pixel triple $(p_1, p_2, p_3) = (15, 17, 9)$ is converted to its binary representation $(0001111, 0010001, 0001001)_2$. Since the first to-be-embedded data is 5, reference set $G_5 = \left\{y_{5,1}, \cdots, y_{5,16}\right\}$ is chosen to find the most similar substitute for (p_1, p_2, p_3). Specifically, taking $y_{5,1} = (0000100)_2$ for example, $p'_{1,1} = (0001000)_2$ is obtained by modifying the 3rd, 2nd, 1st LSBs of p_1 with the first three bits of $y_{5,1}$, and meanwhile, $p'_{2,1} = (0010001)_2$ is obtained by modifying the 2nd and 1st LSBs of p_2 with the 4th and 5th bits of $y_{5,1}$, and finally, $p'_{3,1} = (0001000)_2$ is obtained by modifying the 2nd and 1st LSBs of p_3 with the last two bits of $y_{5,1}$. The distortion between (p_1, p_2, p_3) and $\left(p'_{1,1}, p'_{2,1}, p'_{3,1}\right)$ can be computed as $d_{5,1} = |15 - 8| + |17 - 17| + |9 - 8| = 8$. Similar to $d_{5,1}$, all other 15 distortion values $\left\{d_{5,k}\right\} |2 \leq k \leq 16$ are obtained. Therefore, we can easily find the minimum distortion value $d_{5,15} = 1$ and the corresponding element $y_{5,15} = (1110100)_2$. Finally, (p_1, p_2, p_3) is modified to (15, 17, 8) by means of Step 6. After all original pixel triples are modified to the corresponding marked pixel pairs, stego image $I' = [15,17,8; 17,11,2; 20,13,7]$ is generated.

e: PROPOSED GENERAL FRAMEWORK FOR DATA HIDING PROCEDURE BASED ON ABOVE THREE DATA HIDING STRATEGIES

Based on three data hiding strategies, a larger N_1 will lead to a higher embedding capacity and embedding distortions.

$N_1 = 1$ means that the obtained embedding capacity is the largest, but the introduced distortion due to data embedding is also the highest. Compared with pixels in high and middle levels, the pixels belonging to the low level can preserve better visual quality because every three pixels are modified to embed 3 bits. In order to maximize the embedding capacity as much as possible on the basis of maintaining high visual quality, N_3 must be reduced, and correspondingly, N_1 and N_2 must be enlarged. Therefore, we need to predefine a suitable N_1, N_2 and N_3 by Eq. (7). Correspondingly, cover image I is split into three parts: I_1, I_2 and I_3 according to N_1, N_2 and N_3. That is to say, all the pixels of I_1, I_2 and I_3 belong to the high, middle and low levels, respectively. The detailed data hiding processes of these three parts have been described in Subsections III-A2.b, III-A2.c and III-A2.d. After the data hiding process of each part is performed, stego image I' is generated. Here, we give a complete description for the data hiding process of cover image I and we also give an example to further explain the data hiding process.

Input: Cover image I with a size of $H \times W$, and reference set $(G_0, G_1, \cdots, G_7)$.

Output: Stego image I'.

Step 1: Scan cover image I according to the raster scan order to generate a pixel list $\{p_i | i = 1, 2, \cdots, H \times W\}$, and then all the pixels are divided into three parts: I_1, I_2 and I_3 according to N_1, N_2 and N_3.

Step 2: Exploit the pseudo-random number generator to generate a decimal liner array $S = \{s_i | i = 1, 2, \cdots, H \times W\}$ ranging from 0 to 7, where s_i denotes the i-th 3-bit data to be embedded;

Step 3: Scan image I, call the corresponding data hiding method in different parts, until all of the payload has been embedded.

Step 4: Reconstruct stego image I' with the marked pixels.

An example is given to explain the process above. Assume I = [162,164,155,153; 162,153,155,156; 163,161,155,154; 163,164,150,152], and S = [0,1,4,2,5,3,5,6,0]. We preset N_1, N_2 and N_3 as 1/4, 1/2 and 1/4. That is to say, a quarter, a half and a quarter of all the pixels are assigned respectively to the high, middle and low levels. Namely, I is divided into three parts: I_1 = [162,164,155,153], I_2 = [162,153,155,156; 163,161,155,154] and I_3 = [163,164,150,152]. The data hiding processes in Subsections III-A2.b, III-A2.c and III-A2.d are performed, respectively, until S is embedded into the original image I based on matrix coding. Finally, stego image I' = [162,165,156,159; 162,153,155,157; 162,160,155,155; 163,157,150,153] is generated.

B. THE DATA EXTRACTION PHASE

The data extracting procedure, which is the reverse operation of the embedding procedure is discussed in this subsection. The extracting process is very simple, and can be further divided into two phases: preparation and extraction. A flowchart is shown in Fig. 4 and a detailed description follows in this subsection.

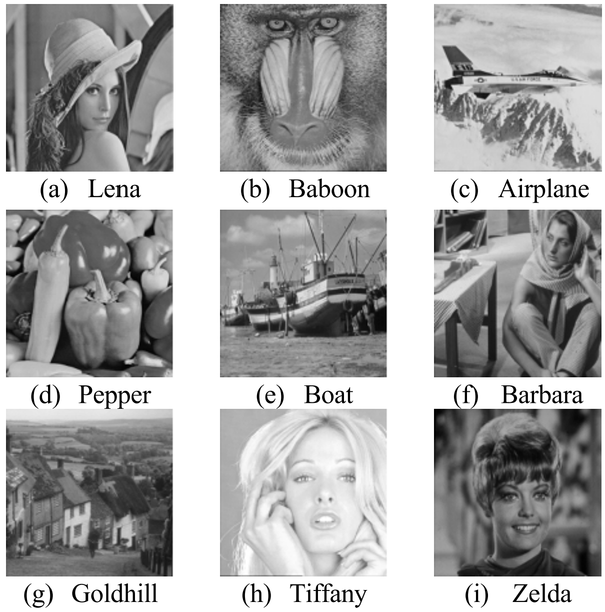


FIGURE 6. Nine grayscale test images.

Assume the stego image is I', and the N_1, N_2 and N_3 are known. Correspondingly, we divide stego image I' into three parts: I'_1, I'_2 and I'_3 according to N_1, N_2 and N_3. Eq. (9) is applied to extract one hidden data s from y, that is:

$$s = H \times y^T \tag{9}$$

where y has three possible values depending on which part it belongs to. Specially, if $y \in I'_1$, we can generate y by extracting the 7 bits (i.e., the 7th, 6th, 5th, 4th, 3rd, 2nd, 1st LSBs) of one marked pixel; if $y \in I'_2$, for a marked pixel pair (p'_1, p'_2), y is formed by extracting 4 bits (i.e., 4th, 3rd, 2nd, 1st LSBs) of p'_1 and the 3 bits (i.e., 3rd, 2nd, 1st LSBs) of p'_2; if $y \in I'_3$, for a marked pixel triple (p'_1, p'_2, p'_3), y is created by extracting 4 bits (i.e., 3rd, 2nd, 1st LSBs) of p'_1, 2 bits (i.e., 2nd and 1st LSBs) of p'_2 and 2 bits (i.e., 2nd and 1st LSBs) of p'_3. The data extraction process is conducted until each hidden bit is extracted one by one. The detailed extraction process is illustrated as follows:

Input: Stego image I' and the proportion of the N_1, N_2 and N_3.

Output: Secret data S.

Step 1: Scan stego image I', and divide I' into three parts: I'_1, I'_2 and I'_3;

Step 2: For one marked pixel p' in I'_1, we can generate y by extracting the 7 bits (i.e., the 7th, 6th, 5th, 4th, 3rd, 2nd, 1st LSBs) of p'; for marked (p'_1, p'_2) in I'_2, y is formed by extracting 4 bits (i.e., 4th, 3rd, 2nd, 1st LSBs) of p'_1 and the 3 bits (i.e., 3rd, 2nd, 1st LSBs) of p'_2; for a marked pixel triple (p'_1, p'_2, p'_3) in I'_3, y is created by extracting 3 bits (i.e., 3rd, 2nd, 1st LSBs) of p'_1, 2 bits (i.e., 2nd and 1st LSBs) of p'_2 and 2 bits (i.e., 2nd and 1st LSBs) of p'_3;

Step 3: Compute secret data s by Eq. (9);

Step 4: Repeat Step 2-3 until all hidden data are extracted;

Step5: Connect all hidden data to construct S.

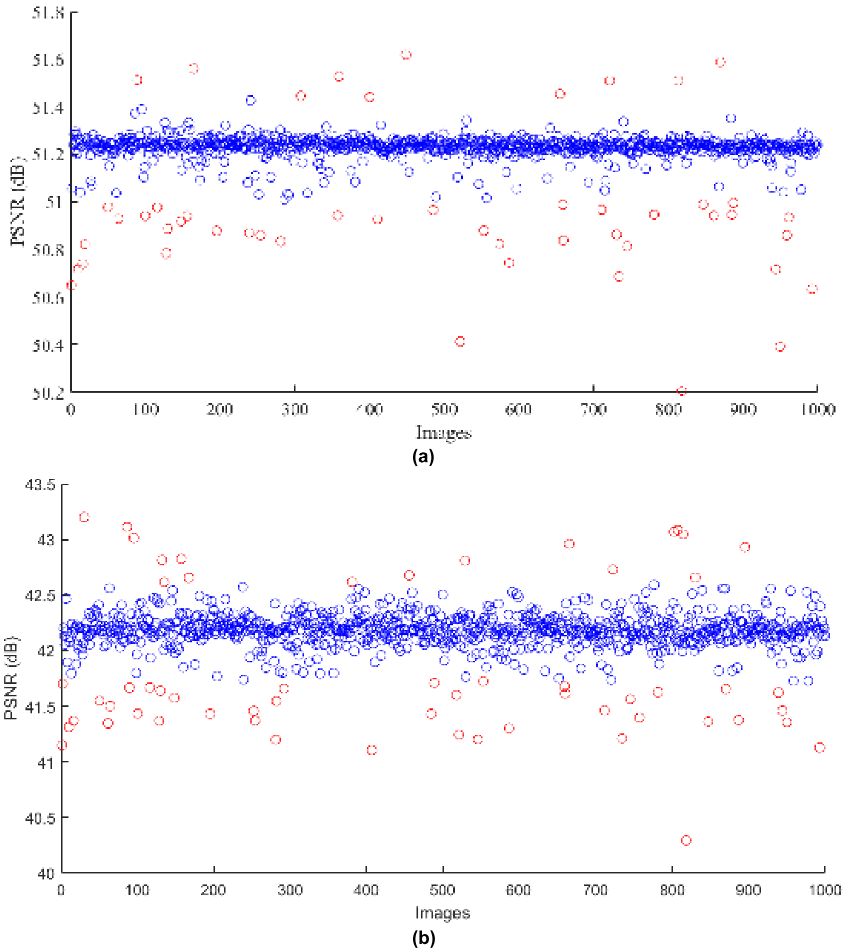


FIGURE 7. The PSNR distribution for 1000 test images when (a) $E_R = 1.0$ and (b) $E_R = 2.0$.

The data extraction process can be further illustrated by the following process. Assume that stego image I' =[162,165,156,159; 162,153,155,157; 162,160,155,155; 163,157,150,153] and N_1, N_2 and N_3 are set at 1/4, 1/2 and 1/4, respectively. Firstly, stego image I' is divided into three parts: i.e., the I'_1 = [162,165,156,159], the I'_2 = [162,153,155,157; 162,160,155,155] and I'_3 = [163,157,150,153]. Secret data *s* is extracted by Eq. (9). Then, secret data $S = [0,1,4,2,5,3,5,6,0]$ is extracted from the stego image.

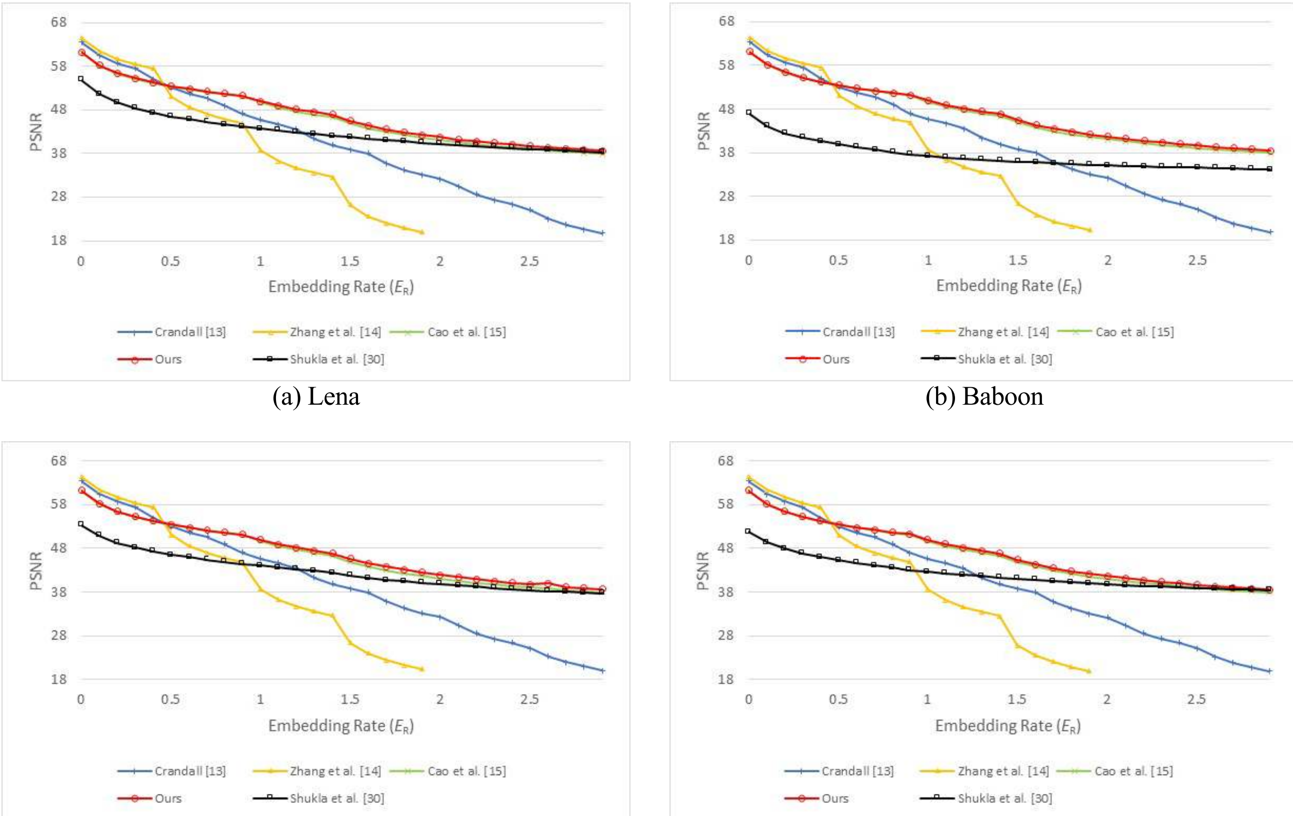
IV. EXPERIMENTAL RESULTS

To demonstrate the effectiveness and superiority of the proposed ABMC-DH method, we conducted comparison experiments against methods by Crandall [13], Zhang *et al.* [14], Cao *et al.* [15], and Shukla *et al.* [30] with our method from two aspects: embedding capacity (or E_R) and visual quality (*PSNR*). Nine grayscale test images shown in the Fig. 6 were utilized in the experiments to compare performance among our proposed method and existing eight methods. All the experiments are conducted by MATELAB R2017a software on a personal computer with a 2.5 GHz Intel core i7 processor, 8 GB memory and preloaded Windows 10 operation system. Besides, in order to verify the proposed ABMC-DH method can maintain consistent performance, we test our method on 1000 grayscale images with a size of 512 × 512 randomly chosen from a popular image dataset called BOSSBase [32]. The related experiment result will be discussed in the next subsections.

Embedding capacity is commonly measured by the standard of how many bits can be carried by per cover pixel denoted as bits-per-pixel (*bpp*), which also reflects the embedding capacity of the cover image. The stego image usually is very similar to the original image, and *PSNR* (Peak-Sign-to-Noise-Ratio) [18], is a commonly-used objective assessment index for image quality, which measures the visual quality of the hidden image according to Eq. (10):

$$PSNR = 10 \log \frac{255^2}{MSE} \, (dB) \, , \tag{10}$$

where $MSE = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} \left(I_{ij} - I'_{ij} \right)^2$ is the degree of the difference between a cover image and its corresponding stego image. The larger the value of *MSE*, the smaller the value of *PSNR*, and vice versa.



(a) Lena (b) Baboon (c) Airplane (d) Peppers

FIGURE 8. The comparison the PSNR under the same E_R between ours and other works [13]–[15], [30] for vary images (a) Lena, (b) Baboon, (c) Airplane and (d) Peppers.

As we know, when *PSNR* is higher than 30 dB, it is hard for human eyes to distinguish any modifications to the original image introduced by data embedding [19]. The proposed ABMC-DH method can achieve good visual quality in the data embedding process, where the modification of a stego image is too insignificant to be detected by the human eyes. The first experiment is to verify the stability of our proposed method. The results of experiment indicate that the average *PSNR* reaches 51.22 dB and 42.17 dB for 1000 test images when $E_R = 1.0$ and $E_R = 2.0$, respectively. In addition, the difference in *PSNR* among images is slight (e.g., although there are 5% of them more than 0.22 dB or 0.44 dB but do not exceed 1 dB or 1.88 dB when $E_R = 1.0$ and $E_R = 2.0$, respectively). That is, every cover image can maintain a high visual quality (e.g., more than 50.2 dB for each test image) when the embedding capacity is 262,144 bits. Detailed results are summarized in Fig. 7.

Next, table 1 demonstrates comparisons *PSNR* under maximum capacity or embedding rate between our method and other existing related work. From the table, the comprehensive performance of our method is optimal. We have the highest *PSNR* among all methods with a comparable embedding capacity or E_R, e.g., Shukla *et al.*'s method [30] $PSNR = 37.23$ when $E_R = 3.11$ and ours $PSNR = 38.59$ when $E_R = 3$. In order to make the comparison more intuitive, we fixed the hidden capacity and only compared the performance of different methods under the same hidden capacity or E_R.

And then, table 2 shows performance comparisons between our proposed ABMC-DH method and four other data hiding methods based on matrix coding. We choose nine test images as shown in Fig. 6, and adjusted the hiding capacity by regulating the embedding rate (e.g., $E_R = 1$, $E_R = 1.5$, $E_R = 2$ and $E_R = 3$). The experimental results show that *PSNR* and hiding capacity is a trade-off, with an increasing E_R, *PSNR* decreases. Taking "Lena" for example, our method is superior to other methods in terms of *PSNR* (e.g., exceeds 4 dB, 7 dB, 0.12 dB and 8.01 dB when $E_R = 1$). And with the E_R is increasing, the advantage is increasing at the same time (e.g., exceed 9 dB, 22 dB, 0.6 dB and 2.88 dB when $E_R = 2$). For more embedding capacity, Crandall's [13] and Zhang *et al.*'s [14] schemes need to provide more binary bits to participate in the hiding procedure, and there is a great impact on *PSNR* when the number of LSBs exceeds four. In Shukla *et al.*'s scheme [30], for each pixel the number of LSBs replaced is greater or equal to three. Therefore, though the method can provide a large embedding capacity, it is inferior to ours in term of *PSNR* under the same embedding rate. Cao *et al.*'s scheme [15] can promise a distortion that is smaller than 7 when embedding 3-bit data, but is inferior to

TABLE 1. Comparisons of PSNR under highest embedding capacity (or E_R) between ours and existing related works [13]–[15], [26]–[30].

Methods	Metrics	Lena	Baboon	Peppers	Airplane	Boats	Average
Crandall [13]	CAP	786,432	786,432	786,432	786,432	786,432	786,432
	E_R	3.00	3.00	3.00	3.00	3.00	3.00
	PSNR	19.78	19.77	19.89	20.08	19.69	19.84
Zhang *et al*. [14]	CAP	524,288	524,288	524,288	524,288	524,288	524,288
	E_R	2.00	2.00	2.00	2.00	2.00	2.00
	PSNR	20.00	20.24	19.94	20.39	20.03	20.12
Cao *et al*. [15]	CAP	786,432	786,432	786,432	786,432	786,432	786,432
	E_R	3.00	3.00	3.00	3.00	3.00	3.00
	PSNR	37.94	37.92	37.90	37.95	37.92	37.93
Wu *et al*. [26]	CAP	408,911	457,170	407,257	409,819	477,836	432,199
	E_R	1.56	1.74	1.55	1.56	1.82	1.65
	PSNR	41.53	37.44	41.39	40.70	36.33	39.48
Yang *et al*.[27]	CAP	765,905	717,849	770,272	770,464	661,223	737,143
	E_R	2.92	2.73	2.94	2.94	2.52	2.81
	PSNR	34.63	30.53	33.87	34.02	32.76	33.16
Khodaei *et al*. [28]	CAP	806,948	851,311	803,184	805,809	851,309	823,712
	E_R	3.07	3.25	3.06	3.07	3.25	3.14
	PSNR	36.02	32.70	34.03	35.69	31.77	34.04
Hussain *et al*. [29]	CAP	800,673	825,881	798,636	795,304	865,094	817,118
	E_R	3.05	3.15	3.04	3.03	3.30	3.11
	PSNR	35.76	33.57	35.64	35.99	34.03	35.00
Shukla *et al*. [30]	CAP	815,085	927,623	813,268	822,622	894,744	854,668
	E_R	3.11	3.54	3.10	3.14	3.41	3.26
	PSNR	37.32	33.18	37.47	36.83	35.51	36.06
Ours	CAP	786,432	786,432	786,432	786,432	786,432	786,432
	E_R	3.00	3.00	3.00	3.00	3.00	3.00
	PSNR	38.59	38.50	38.52	38.68	38.48	38.56

our method, since our distortion is superior to the former. Our approach is optimal in term of *PSNR* among five schemes, as proven by the experimental results above. Table 2 also shows that the performance of our method is independent of the cover images, i.e., the *PSNR* is almost the same in each of the nine images, with at most a 0.04 dB difference, which is only affected by embedding capacity. In other words, we can take advantage of this characteristic to expand the application scenario of our method by adjusting the value of E_R.

Finally, through four typical grayscale test images, which include the two smooth images of "Lena" and "Airplane", as well as two complex images "Baboon" and "Pepper", we compare data hiding performance under the same embedding capacity between the proposed ABMC-DH method and four representative studies: Crandall [13], Zhang *et al.* [14], Cao *et al.* [15], and Shukla *et al.* [30]. The experimental results are plotted on the coordinate plane using line charts, and the graph is shown in Fig. 8, in which the x-axis represents hiding capability rate, the y-axis represents image visual quality, and the points on the coordinate plane represent a set of experimental data. Our results are represented by a red curve with circle marks, Crandall's scheme [13] is represented by a blue curve with cross marks, Zhang *et al.*'s scheme [14] is represented by a yellow curve with triangle marks, Cao *et al.*'s [15] results are represented by an green curve with cross marks, and Shukla *et al.*'s [30] results are presented by a black curve with square marks.

Our performance outperforms other methods as shown in Fig. 8 (a)-(d), e.g., $PSNR = 61.26$, when $E_R = 0.1$ and $PSNR = 38.67$, when $E_R = 3$. Although the schemes of Crandall [13] and Zhang *et al.* [14] have little higher results than ours when $E_R \leq 0.5$, they decline sharply with an increase in hiding capacity, whereas our result is always ahead when $E_R > 0.5$, and Zhang *et al.*'s scheme [14] is invalid when $E_R > 2$. Shukla *et al.*'s scheme [30] is always behind compared to ours, especially in Fig. 8 (b), where the difference is obvious. Cao *et al.*'s scheme [15] and ours have good visual quality under various embedding capacities, e.g., the average $PSNR > 38$, and the gradient is increasingly flat with increasing of E_R. Compared with the other two schemes, Cao *et al.*'s scheme [15] and our scheme have a comparative advantage especially when the embedding capacity is greater.

TABLE 2. Comparisons of PSNR in different E_R between ours and existing four related works [13]–[15], [26].

Method	Lena	Baboon	Airplane	Peppers	Boats	Barbara	Goldhill	Tiffany	Zelda
E_R=1									
Crandall [13]	47.01	47.03	47.03	47.00	47.00	47.04	47.03	47.00	47.02
Zhang *et al*. [14]	44.93	44.92	44.92	44.92	44.93	44.92	44.93	44.92	44.92
Cao *et al*. [15]	51.13	51.14	51.14	51.14	51.15	51.13	51.13	51.14	51.14
Shukla *et al*. [30]	44.14	37.61	44.46	43.04	43.07	42.19	44.03	43.61	44.44
Ours	51.25	51.23	51.23	51.23	51.20	51.22	51.24	51.23	51.22
E_R=1.5									
Crandall [13]	39.92	39.91	39.90	39.89	39.94	39.93	39.92	39.88	39.91
Zhang *et al*. [14]	32.63	32.63	32.63	32.53	32.59	32.63	32.62	32.62	32.56
Cao *et al*. [15]	46.36	46.36	46.32	46.37	46.28	46.36	46.39	46.37	46.37
Shukla *et al*. [30]	42.02	36.14	42.36	41.31	40.64	40.24	41.63	41.82	42.20
Ours	46.88	46.86	46.90	46.88	46.84	46.84	46.87	46.87	46.86
E_R=2									
Crandall [13]	33.06	33.07	33.18	33.04	33.07	33.08	33.09	33.15	33.08
Zhang *et al*. [14]	20.00	20.24	20.39	19.94	20.03	19.84	19.99	20.46	19.63
Cao *et al*. [15]	41.63	41.60	41.62	41.59	41.58	41.59	41.56	41.57	41.60
Shukla *et al*. [30]	40.42	35.26	40.13	40.01	38.82	38.34	39.99	40.39	40.85
Ours	42.24	42.19	42.46	42.20	42.13	42.20	42.13	42.49	42.17
E_R=3									
Crandall [13]	19.78	19.77	20.08	19.89	19.69	19.86	19.80	19.85	19.71
Zhang *et al*. [14]	-	-	-	-	-	-	-	-	-
Cao *et al*. [15]	37.94	37.92	37.95	37.90	37.92	37.93	37.90	37.92	37.92
Shukla *et al*. [30]	38.31	34.10	37.72	38.39	36.93	35.74	37.60	38.62	39.28
Ours	38.61	38.51	38.67	38.52	38.48	38.53	38.50	38.68	38.53

Our method exceeds the former by 0.6 in average *PSNR* under the same payload when $E_R \geq 1$ and has a comparable advantage when $E_R < 1$. As mentioned in Subsection III-A2 2.1), our method achieves relatively optimal image visual quality by adjusting percentages of high, middle and low levels when E_R is fixed. If amounts of to-be-embedded bits is few, e.g., $E_R < 1$, only data hiding procedure for low level is conducted to obtain higher visual quality of stego image. When E_R is increasing, e.g., $E_R = 2$, data hiding procedures for middle and high level are conducted to minimize the distortion of stego image. When $E_R = 3$, only data hiding procedure for high level is conducted. In this case, although the execution cost may increase, but *PSNR* is still satisfactory due to our proposed pixel value difference optimization strategy.

In summary, the comprehensive performance of our proposed ABMC-DH method is significantly superior to other four related works based on matrix coding especially in the term of image visual quality.

The RS-steganalysis method [45] is used to verify the security of our method, which includes a discrimination function with flipping mask M and $-M$ correspoding to two matrix [0,1;1,0] and [0,-1;-1,0] respectively. Futhermore, the results of function R_M, R_{-M}, S_M and S_{-M} are utilized to find out whether an image contian hiding content or not. If these results satisfy $R_M \approx R_{-M} > S_M \approx S_{-M}$, the image is considered as normal. Vice versa, the image is considered to be having hidden data in its least significant bits.

TABLE 3. The RS-steganalysis of our method when $E_R = 1$ and $E_R = 3$.

Images	E_R=1		E_R=3	
	$\lvert R_M - R_{-M} \rvert$	$\lvert S_M - S_{-M} \rvert$	$\lvert R_M - R_{-M} \rvert$	$\lvert S_M - S_{-M} \rvert$
Lena	0.1109	0.0720	0.2821	0.2553
Baboon	0.0387	0.0332	0.0533	0.0534
Airplane	0.0664	0.0575	0.2832	0.2594
Peppers	0.0949	0.0604	0.2473	0.2125
Boats	0.0693	0.0500	0.1252	0.1392
Barbara	0.0634	0.0529	0.1539	0.1577
Goldhill	0.0863	0.0531	0.1646	0.1472
Tiffany	0.0764	0.0776	0.2638	0.2543
Zelda	0.1608	0.1046	0.3069	0.2922
Average	0.0852	0.0624	0.2089	0.1968

Firstly, the ABMC-DH method is used to test images shown in Fig. 6 under 1 bpp and 3 bpp, respectively. And then, two sets of stego images are achieved. Finally, the RS-steganalysis method is used for these marked images. The statistical results indicate that the differences in the our method are extremely close to each other between R_M and R_{-M} or between S_M and S_{-M}. The average value of

$|R_M - R_{-M}|$ and $|S_M - S_{-M}|$ is equal to 0.08% and 0.06%, respectively, when $E_R = 1$, and equal to 0.20% and 0.19%, respectively, when $E_R = 3$. Although the difference has increased slightly with the increase in hiding capacity, there is still good performance. In other words, our method can resist against RS-steganalysis detection. Detailed results are shown in Table 3.

V. CONCLUSION

This paper proposed a novel adaptive data hiding scheme based on matrix coding. Firstly, we constructed a reference data set by classifying all possible 7-bit binary number combinations. Subsequently, through the user predefined proportion of the embedding rate, pixels of the cover image were classified into three levels: low, middle and high. Each level is treated with a different and separate strategy. For the low level, a triple pixel combination is used to hide 3-bit secret data by choosing a suitable alternative from the reference data. For the middle level, a pair of pixels is used to hide 3-bit secret data by choosing a suitable alternative from the reference data. For the high level, a single pixel is used to hide 3-bit secret data by choosing a suitable alternative from the reference data. The experimental results show that the data hiding capacity and image visual quality are influenced by the proportion of E_R, where with an increase of the E_R, our method has more data hiding capacity, but the quality of the visual image declines and vice versa. From another perspective, we can expand the scenario of our method by adjusting the predefined proportion of the embedding capacity. The experimental results and analysis imply that our approach outperforms other existing related methods in terms of embedding capacity and visual quality. In the future, a more efficient and lossless hiding method coworking with different kinds of matrix coding or turtle shelling matrix [41], [42] could be further explored.

REFERENCES

[1] D. Zou, C. W. Wu, G. Xuan, and Y. Shi, "A content-based image authentication system with lossless data hiding," in *Proc. Int. Conf. Multimedia Expo. (ICME)*, Baltimore, MD, USA, 2003, pp. 213–216.

[2] C. C. Chang, Y. C. Chou, and T. D. Kieu, "High capacity data hiding for grayscale images," in *Proc. Int. Conf. Ubiquitous Inf. Manage. Commun.*, Seoul, South Korea, 2007, pp. 139–148.

[3] J. Mielikainen, "LSB matching revisited," *IEEE Signal Process. Lett.*, vol. 13, no. 5, pp. 285–287, May 2006.

[4] Z.-L. Zhao and F. Gao, "An improved steganographic method of product perfect codes," in *Proc. IEEE Int. Conf. Signal Process., Commun. Comput. (ICSPCC)*, Xi'an, China, Sep. 2011, pp. 1–5.

[5] A. Sanchez-Macian, P. Reviriego, and J. A. Maestro, "Enhanced detection of double and triple adjacent errors in Hamming codes through selective bit placement," *IEEE Trans. Device Mater. Rel.*, vol. 12, no. 2, pp. 357–362, Jun. 2012.

[6] J. Fridrich and D. Soukal, "Matrix embedding for large payload," *IEEE Trans. Inf. Forensics Security*, vol. 1, no. 3, pp. 390–395, Aug. 2006.

[7] T. Filler, J. Judas, and J. Fridrich, "Minimizing additive distortion in steganography using syndrome-trellis codes," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 3, pp. 920–935, Sep. 2011.

[8] Y. Liu and C.-C. Chang, "A turtle shell-based visual secret sharing scheme with reversibility and authentication," *Multimed Tools Appl*, vol. 77, no. 19, pp. 25295–25310, Oct. 2018.

[9] G. Liu, W. Liu, Y. Dai, and S. Lian, "An adaptive matrix embedding for image steganography," in *Proc. 3rd Int. Conf. Multimedia Inf. Netw. Secur.*, Shanghai, China, Nov. 2011, pp. 642–646.

[10] W. Zhang, X. Zhang, and S. Wang, "Near-optimal codes for information embedding in gray-scale signals," *IEEE Trans. Inf. Theory*, vol. 56, no. 3, pp. 1262–1270, Mar. 2010.

[11] J. Chen, Y. Zhu, Y. Shen, and W. Zhang, "Efficient matrix embedding based on random linear codes," in *Proc. Int. Conf. Multimedia Inf. Netw. Secur.*, Nanjing, China, 2010, pp. 880–883.

[12] X. Zhang and S. Wang, "Efficient steganographic embedding by exploiting modification direction," *IEEE Commun. Lett.*, vol. 10, no. 11, pp. 781–783, Nov. 2006.

[13] R. Crandall. (1998). *Some Notes on Steganography, Posted on Steganography Mailing*. [Online]. Available: http://os.inf.tu-dresden.de/west-feld/crandall.pdf

[14] W. Zhang, S. Wang, and X. Zhang, "Improving embedding efficiency of covering codes for applications in steganography," *IEEE Commun. Lett.*, vol. 11, no. 8, pp. 680–682, Aug. 2007.

[15] Z. Cao, Z. Yin, H. Hu, X. Gao, and L. Wang, "High capacity data hiding scheme based on (7, 4) Hamming code," *SpringerPlus.*, vol. 5, no. 1, pp. 175–187, Feb. 2016.

[16] C.-C. Chang, T. D. Kieu, and Y.-C. Chou, "A high payload steganographic scheme based on (7, 4) Hamming code for digital images," in *Proc. Int. Symp. Electron. Commerce Secur.*, Guangzhou, China, 2008, pp. 16–21.

[17] S. Arjun, A. Negi, C. Kranthi, and D. Keerthi, "An approach to adaptive steganography based on matrix embedding," in *Proc. TENCON IEEE Region Conf.*, Taipei, Taiwan, Oct. 2007, pp. 1–4.

[18] C.-C. Chang, T. D. Kieu, and Y.-C. Chou, "Using nearest covering codes to embed secret information in grayscale images," in *Proc. 2nd Int. Conf. Ubiquitous Inf. Manage. Commun. (ICUIMC)*, Suwon-Si, South Korea, 2008, pp. 315–320.

[19] L. Zhang and H. Li, "A product code in steganography with improved embedding rate," in *Proc. IEEE Inf. Technol., Netw., Electron. Autom. Control Conf.*, Chongqing, China, May 2016, pp. 246–250.

[20] B. Jana, D. Giri, and S. K. Mondal, "Partial reversible data hiding scheme using (7, 4) Hamming code," *Multimedia Tools Appl.*, vol. 76, no. 20, pp. 21691–21706, Oct. 2017.

[21] R. W. Hamming, "Error detecting and error correcting codes," *Bell Syst. Tech. J.*, vol. 29, no. 2, pp. 147–160, Apr. 1950.

[22] K.-H. 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. 127–136, Jan. 2018.

[23] S. Weng, Y. Chen, B. Ou, C.-C. Chang, and C. Zhang, "Improved K-pass pixel value ordering based data hiding," *IEEE Access*, vol. 7, pp. 34570–34582, 2019.

[24] W. Hong, "Efficient data hiding based on block truncation coding using pixel pair matching technique," *Symmetry*, vol. 10, no. 2, p. 36, Jan. 2018.

[25] J. Chen, W. Hong, T.-S. Chen, and C.-W. Shiu, "Steganography for BTC compressed images using no distortion technique," *Imag. Sci. J.*, vol. 58, no. 4, pp. 177–185, Aug. 2010.

[26] H.-C. Wu, N.-I. Wu, C.-S. Tsai, and M.-S. Hwang, "Image steganographic scheme based on pixel-value differencing and LSB replacement methods," *IEE Proc., Vis. Image Process.*, vol. 152, no. 5, p. 611, 2005.

[27] C.-H. Yang, C.-Y. Weng, S.-J. Wang, and H.-M. Sun, "Varied PVD+LSB evading detection programs to spatial domain in data embedding systems," *J. Syst. Softw.*, vol. 83, no. 10, pp. 1635–1643, Oct. 2010.

[28] M. Khodaei and K. Faez, "New adaptive steganographic method using least-significant-bit substitution and pixel-value differencing," *IET Image Process.*, vol. 6, no. 6, pp. 677–686, Aug. 2012.

[29] M. Hussain, A. W. A. Wahab, N. Javed, and K.-H. Jung, "Recursive information hiding scheme through LSB, PVD shift, and MPE," *IETE Tech. Rev.*, vol. 35, no. 1, pp. 53–63, Jan. 2018.

[30] A. K. Shukla, A. Singh, B. Singh, and A. Kumar, "A secure and high-capacity data-hiding method using compression, encryption and optimized pixel value differencing," *IEEE Access*, vol. 6, pp. 51130–51139, 2018.

[31] J. Fridrich, M. Goljan, and R. Du, "Reliable detection of LSB steganography in color and grayscale images," in *Proc. Workshop Multimedia Secur. New Challenges (MM Sec)*, Ottawa, ON, Canada, 2001, pp. 27–30.

[32] P. Bas, T. Filler, and T. Pevný, "'Break our steganographic system': The ins and outs of organizing BOSS," in *Proc. 13th Int. Workshop Inf. Hiding*, Prague, Czech Republic, 2011, pp. 59–70.

[33] C.-C. Lin, X.-L. Liu, and S.-M. Yuan, "Reversible data hiding for VQ-compressed images based on search-order coding and state-codebook mapping," *Inf. Sci.*, vol. 293, pp. 314–326, Feb. 2015.

[34] Z. Yu, C.-C. Lin, C.-C. Chang, and G.-D. Su, "HBF-DH: An enhanced payload hybrid data hiding method based on a hybrid strategy and block features," *IEEE Access*, vol. 7, pp. 148439–148452, 2019.
[35] A. Malik, R. Kumar, and S. Singh, "Reversible data hiding scheme for LZW codes using LSB flipping strategy," in *Proc. Int. Conf. Adv. Inf. Commun. Technol. Comput. (AICTC)*, Bikaner, India, 2016, p. 58.
[36] A. Malik, S. Singh, and R. Kumar, "Recovery based high capacity reversible data hiding scheme using even-odd embedding," *Multimedia Tools Appl.*, vol. 77, no. 12, pp. 15803–15827, Jun. 2018.
[37] R. Kumar, S. Chand, and S. Singh, "An improved histogram-shifting-imitated reversible data hiding based on HVS characteristics," *Multimedia Tools Appl.*, vol. 77, no. 11, pp. 13445–13457, Jun. 2018.
[38] R. Kumar, S. Chand, and S. Singh, "A reversible data hiding scheme using pixel location," *Int. Arab J. Inf. Technol.*, vol. 15, no. 4, pp. 763–768, Jul. 2018.
[39] S. Chand, R. Kumar, and S. Singh, "A reversible high capacity data hiding scheme using combinatorial strategy," *Int. J. Multimedia Intell. Secur.*, vol. 3, no. 2, p. 146, 2018.
[40] R. Kumar, S. Chand, and S. Singh, "An optimal high capacity reversible data hiding scheme using move to front coding for LZW codes," *Multimedia Tools Appl.*, vol. 78, no. 16, pp. 22977–23001, Aug. 2019.
[41] C.-C. Chen, C.-C. Chang, C.-C. Lin, and G.-D. Su, "TSIA: A novel image authentication scheme for AMBTC-based compressed images using turtle shell based reference matrix," *IEEE Access*, vol. 7, pp. 149515–149526, 2019.
[42] X.-Z. Xie, C.-C. Lin, and C.-C. Chang, "Data hiding based on a two-layer turtle shell matrix," *Symmetry*, vol. 10, no. 2, p. 47, Feb. 2018.

ZHAN YU received the B.S. degree from the Lanzhou University of Technology, in 2004, and the M.S. degree in computer and technology from Fuzhou University, in 2017. He is currently a Lecturer with the School of Electronics and Information Engineering, Fuqing Branch of Fujian Normal University. He also chaired and participated in the research of several provincial projects. His research interests include the Internet of Things, image and signal processing, and computer network technology.

CHIA-CHEN LIN (MIN-HUI LIN) received the Ph.D. degree in information management from National Chiao Tung University, in 1998. She is currently a Professor with the Department of Computer Science and Information Management, Providence University. Since 2018, she has been a Fellow of IET and the School Counselor for Providence University. In additions, from 2009 to 2012, she served as the Vice Chairman of the Tainan Chapter IEEE Signal Processing Society. She also serves as an Associate Editor and Editor for several representative EI and SCIE journals. Her research interests include image and signal processing, information hiding, mobile agent, and electronic commerce.

CHIN-CHEN CHANG (Fellow, IEEE) received the B.Sc. degree in applied mathematics and the M.Sc. degree in computer and decision sciences, and the Ph.D. degree in computer engineering from National Chiao Tung University. He was with National Chung Cheng University, from 1989 to 2005. Prior to joining Feng Chia University, he was an Associate Professor with Chiao Tung University, a Professor with National Chung Hsing University, and the Chair Professor of National Chung Cheng University. He has also been a Visiting Researcher and a Visiting Scientist with Tokyo University and Kyoto University, Japan. He is currently the Chair Professor of the Department of Information Engineering and Computer Science, Feng Chia University, from February 2005. During his service in Chung Cheng University, he served as the Chairman of the Institute of Computer Science and Information Engineering, the Dean of College of Engineering, a Provost and an Acting President of Chung Cheng University, and the Director of Advisory Office, Ministry of Education, Taiwan. His current research interests include database design, computer cryptography, image compression, and data structures. He has won many research awards and honorary positions by and in prestigious organizations both nationally and internationally. He is also a Fellow of IEE, U.K. Since his early years of career development, he consecutively won the Outstanding Talent in Information Sciences of the R.O.C., the AceR Dragon Award of the Ten Most Outstanding Talents, the Outstanding Scholar Award of the R.O.C., the Outstanding Engineering Professor Award of the R.O.C., the Distinguished Research Awards of National Science Council of the R.O.C., and the Top Fifteen Scholars in Systems and Software Engineering of the *Journal of Systems and Software*. He received the award in National Tsing Hua University for his B.Sc. and M.Sc. degrees. On numerous occasions, he was invited to serve as a Visiting Professor, the Chair Professor, an Honorary Professor, an Honorary Director, an Honorary Chairman, a Distinguished Alumnus, a Distinguished Researcher, and a Research Fellow by universities, such as Hangzhou Dianzi University, and research institutes.