

Received November 6, 2021, accepted December 15, 2021, date of publication December 20, 2021, date of current version January 4, 2022.

Digital Object Identifier 10.1109/ACCESS.2021.3136883

A Multilayer Steganographic Method Using Improved Exploiting Modification Directions Scheme

HUI-SHIH LENG¹, CHENG-JUNG TSAI^{1,2}, AND TIEN-JUNG WU²

¹Department of Mathematics, National Changhua University of Education, Changhua 500, Taiwan

²Graduate Institute of Statistics and Information Science, National Changhua University of Education, Changhua 500, Taiwan

Corresponding author: Cheng-Jung Tsai (cjtsai@cc.ncue.edu.tw)

This work was supported in part by the Ministry of Science and Technology of the Republic of China under Grant MOST 110-2622-E-018-002, Grant MOST 110-2118-M-018-001, and Grant MOST 108-2622-E-018-002-CC3.

ABSTRACT The exploiting modification direction scheme is a well-known irreversible steganographic method because of its high embedding efficiency and imperceptibility. The shortcomings of the exploiting modification direction scheme are the use of a $(2n + 1)$ -ary notational system secret digit and low payload. Some studies transform secret data into a sequence of secret digits in the 8-ary notational system to overcome the disadvantage of using a non-binary secret data problem, which also increases the payload. Additionally, some studies have proposed a two-layer embedding method to enhance data security, which also increases the payload. Besides, most of the EMD-based scheme has the fall of boundary problems when the pixel group is located in the boundary area, which increase the distortion of the stego-image. In this study, we proposed a multilayer steganographic method using an improved exploiting modification direction scheme, which includes the abovementioned advantages, using binary secret data, increasing the payload, enhance data security and preventing the fall of boundary problem. In addition, we demonstrate a three-layer EMD-based data hiding example. The experimental results show that the proposed method achieves a high payload (approximately 4.5 bpp) and the quality of the stego-image satisfying the human vision system sensitivity (PSNR value is greater than 30 dB).

INDEX TERMS Exploiting modification direction scheme, 8-ary notational system, multilayer embedding.

I. INTRODUCTION

The popularity of computers and the fascinating development of the Internet have made digital data widely used in the contemporary world. It is convenient to share large amounts of multimedia data through ubiquitous channels. However, this has also caused consumers to worry about security and privacy issues. Digital steganography is a technology in which the sender conceals secret data in the cover media (e.g., text, sound, images, videos) imperceptibly to obtain the stego-media. After being transmitted to the communication channel, secret data with a specific algorithm can be extracted accurately and completely. Owing to its imperceptible characteristics, stego-media can defend against aggressive and rude attackers during transmission, which, in some ways, makes differences from cryptographic encryption, another

The associate editor coordinating the review of this manuscript and approving it for publication was Ramakrishnan Srinivasan ¹.

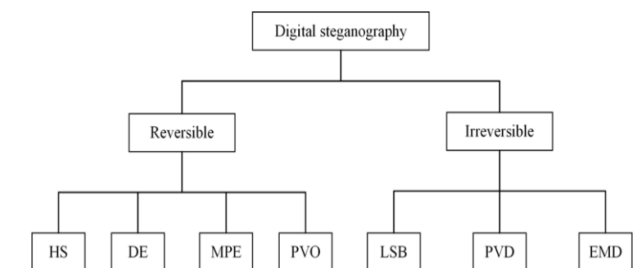


FIGURE 1. Two types of digital steganographic method.

communication security technology. In this study, grayscale images were used as cover media.

Digital steganography can be divided into two categories: reversible and irreversible. The two types of digital steganography methods are shown in Fig. 1. Reversible steganographic technology can restore the cover image after the recipient extracts the secret data from the stego-image, while irreversible steganographic technology cannot. The most

commonly used steganographic schemes include the histogram shifting (HS) scheme, the difference expansion (DE) scheme, modification of prediction errors (MPE) scheme, and pixel-value-ordering (PVO) scheme. On the other hand, for the irreversible part, the least significant bit substitution (LSB) scheme, pixel-value differencing (PVD) scheme, and exploiting modification direction (EMD) scheme are common.

Recently, dual-image reversible data hiding (DRDH) has been an active research area and attract attention because of its unique advantages in security, embedding capacity and visual quality [1]–[6]. Chang, Kieu, and Chou first proposed the DRDH scheme [1], [2]. A location-based DRDH is proposed by Lee *et al.* [3]. Lee and Huang [4] proposed a orientation combinations DRDH. Biswapati, Debasis and Kumar [5], [6] proposed the DRDH by combine the PVD and EMD methods and using pixel value differencing expansion.

The embedding capacity and the quality of the stego-image are the primary indicators of steganographic technology performance. However, many researchers have focused on irreversible steganographic techniques owing to their high payload and simple implementation. In this study, we focus on irreversible steganographic technologies.

The LSB scheme is widely used and was first proposed by Bender *et al.* [7]. It directly substitutes n secret bits and makes them embedded into n LSBs of each pixel in the cover image, providing a high payload and obtaining a good visual quality stego-image. To improve visual quality, two critical strategies are required for the LSB scheme. One is the optimal adjustment process (OPAP) proposed by Chan and Cheng [8], and the other is the modulus function proposed by Thien and Lin [9]. OPAP was proposed to enhance the image quality of the stego-image obtained using the simple LSB substitution method. The modulus function achieves good image vision quality without the need for post-processing. To overcome the asymmetry of the LSB scheme, Li *et al.* [10] proposed an LSB matching (LSBM) scheme. It reduces the asymmetric effects by randomly adding ± 1 to each pixel of the cover image, but this addition depends on the secret bits and pixels of the cover image. The LSB matching revised (LSBMR) scheme, proposed by Mielikinen [11], was used to raise the stego-image quality. A binary function and four hiding rules were used to embed secret bits inside two consecutive cover pixels, and the value of one cover pixel at most will vary by one. Note that the LSBMR scheme embeds the same payload as the LSBM scheme but has fewer distortions. Sahu and Swain [12] proposed a high fidelity reversible data hiding scheme using modified LSBM and pixel difference. However, the LSB-based scheme can be simply detected using statistical analysis techniques, such as RS-steganalysis [13].

Based on human vision system sensitivity, Wu and Tsai [14] proposed the PVD method to conceal the secret data. The values of two consecutive cover pixels are calculated and replaced with a new difference value based on the sum of the lower bound of an interval range and the secret data. Because human eyes are more sensitive to

changes in pixels in smooth areas than in edge areas, the PVD method can be embedded more secret data in smooth areas than in edge areas. A steganographic method called side matching was proposed by Chang and Tseng [15]. The side information of neighboring pixels is used to estimate the embedding capacity for reaching the target pixel. Additionally, to enhance the capacity and imperceptibility, Tseng and Leng [16] took advantage of the PVD-based methodology. Swain [17] proposed an adaptive PVD-based approach that considers vertical and horizontal pixel pairs. Two steganographic schemes were proposed by Hussain *et al.* [18]. One is based on the parity-bit pixel value difference (PBPVD), and the other improves the rightmost digit replacement (iRMDR) technique. The latter pays the cost of low stego-image quality to reach a high capacity. However, both cases in the cover image are chosen sequentially to hide the secret message. However, the PVD-based scheme has two problems: (1) The error block problem and (2) The fall of boundary problem. Besides, the PVD-based scheme can be detected by the pixel difference histogram (PDH) analysis.

Sahu and Swain [19] using adaptive LSB and PVD technique resisting PDH and RS analysis. Sahu *et al.* [20] using multi-directional block based PVD and modulus function resisting PDH and RS-steganalysis. Sahu and Swain [21] proposed an optimal information hiding approach based on PVD method and modulus function. Various start-of-the-art digital image steganography and steganalysis is introduced in [22].

Zhang and Wang [23] proposed an EMD scheme to completely exploit the modification of a pixel group. Each secret digit in a $(2n + 1)$ -ary notational system is carried by n cover pixels, where n is a system parameter, and one pixel is increased or decreased by one at most. In the EMD scheme, when $n = 2$, the payload is about 1.26 bpp with high quality of the stego-image (approximately 51.14 dB). Many researchers are interested in the EMD method, improving the use of non-binary notational system secret digits, or increasing the embedding capacity [24]–[39]. Based on the embedding and extraction procedure, EMD-based data hiding methods can be classified into two categories: extraction function-based [24]–[30] and reference matrix-based [31]–[40].

To increase the payload, many researchers [24]–[30] used two pixels as a group and defined different extraction functions for the embedding and extraction procedures. Chang *et al.* [24] showed that the embedding rate is bounded by $(\log_2 5)/2$. Additionally, using a two-stage embedding technology to embed n pixels twice at a time, an improved EMD-hiding method increases the payload by 2.32 bpp. Chao *et al.* [25] defined a brand-new extraction function: $f(x_1, x_2) = ((2k + 1) \times x_1 + x_2) \bmod N$ (where N is the range of component of neighboring vectors of (x_1, x_2)), which can embed secret digits in $(2k^2 + 2k + 1)$ -ary notational system, and the payload increased up to 2.68 bpp when $k = 4$. Chen *et al.* [26] proposed a flexible EMD-based technique that considers the overflow problem. To address this issue, the danger pixel components that may result in overflow

TABLE 1. Comparison among the extraction function-based EMD method.

Method	Extraction function	Notational system	Payload
Chang et al. [24]	$f(x_1, x_2, \dots, x_n) = \left[\sum_{i=1}^n (i \times x_i) \right] \bmod (2n + 1)$	$2n + 1$	2.32 bpp
Chao et al. [25]	$f(x_1, x_2) = ((2k + 1)x_1 + x_2) \bmod N$	$2k^2 + 2k + 1$	2.68 bpp
Chen et al. [26]	$f(x_1, x_2) = (x_1 \times n^0 + x_2 \times n^1) \bmod n^2$	n^2	2.32 bpp
Kim et al. [27]	$f(x_1, x_2, \dots, x_n) = \left[\sum_{i=1}^n (x_i \times \pi_i) \right] \bmod (2^{n+x} - 1)$	$2^{n+x} - 1$	3.22 bpp
Kieu and Chang [28]	$f(x_1, x_2) = ((s - 1) \times x_1 + s \times x_2) \bmod s^2$	$2n + 1$	4.5 bpp
Leng and Tseng [29]	$f(x_1, x_2, \dots, x_n) = \left[\sum_{i=1}^n (x_i \times n^{i-1}) \right] \bmod (w^n)$	w^n	4.75 bpp
Leng et al. [30]	$f(x_1, x_2) = (c_1 \times x_1 + c_2 \times x_2) \bmod 2^k$	2^k	4 bpp

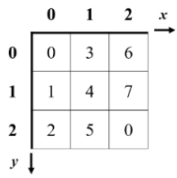
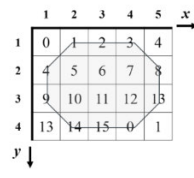
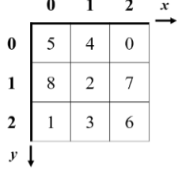
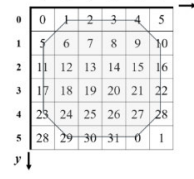
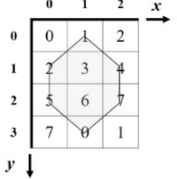
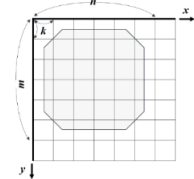
problems are identified and moved to a safe area. Then, the EMD scheme is used for the embedding and extraction processes to improve the stego-image quality. Kim *et al.* [27] proposed 2-EMD and EMD-2 schemes based on $(2^{n+x} - 1)$ -ary notational system secret digits to increase the payload and achieve maximum payload $\log_2 10$ or 3.22 bpp. Furthermore, Kieu and Chang [28] used eight directions to improve the EMD scheme, which is called the fully exploiting modification direction (FEMD) scheme. The extraction function is defined as $f(x_1, x_2) = ((s - 1) \times x_1 + s \times x_2) \bmod s^2$, where s is an integer and $s \geq 2$. It achieves a higher payload (up to 4.5 bpp when the parameter $s = 23$), and the quality of the stego-image is acceptable (approximately 31 dB). Leng and Tseng [29] used an n -pixels-group in a grayscale cover image as a group that is based on an n -dimensional hypercube to increase the payload and achieve a maximum payload 4.75 bpp. The disadvantage of the abovementioned methods is the use of non-binary notational system secret digits. To overcome this shortcoming, Leng *et al.* [30] proposed a high-payload EMD-based steganographic scheme based on a binary notational system secret digit. The embedding and extraction procedure combines the mapping matrix, which is generated by two extraction functions. With the diversity of the definitions of these two extraction functions, the proposed method achieves a higher payload of up to 4 bpp, and the quality of the stego-image satisfies human vision system sensitivity. The abovementioned methods use the extraction function in the embedding and extraction procedures. According to the function, the corresponding mapping matrix is constructed using an extraction-function-based EMD method. A comparison of the EMD method [24]–[30] based on extraction function-based methods is shown in Table 1.

Another shortcoming of the EMD scheme is that the secret data must be converted into a non-binary notational system. To solve this problem, Lee *et al.* [31] proposed an improved exploiting modification direction (IEMD) scheme, which

defines an extraction function $f(x_1, x_2) = (x_1 + 3x_2) \bmod 2^3$ to embed three binary secret bits simultaneously. Its payload is 1.5 bpp and achieves high stego-image quality (PSNR value of approximately 50 dB). Although the IEMD technique is represented as a mapping matrix similar to the reference matrix, it essentially uses the extraction function to embed the secret data. Chang *et al.* [32] presented a Sudoku-based data hiding scheme to improve security, in which a secret digit in 9-ary notational system is embedded in each pair of pixels; however, their scheme has a large search set and can only embed a 9-ary notational system, which is inflexible to multiple embedded requirements. Chang *et al.* [33] proposed a novel turtle shell-based scheme for data hiding. They used a reference matrix based on hexagon-shaped turtle shells instead of the extraction function of the EMD scheme in the embedding and extraction procedure. Each turtle shell has 8 distinct digits (from 0 to 7), which means that three binary secret bits can be embedded simultaneously, with a payload of 1.5 bpp. Liu *et al.* [34] also conducted research on high-capacity turtle shell-based data hiding by adding an extra location table to increase the payload (2 bpp). Kurup *et al.* [35] proposed a reference matrix-based data hiding scheme based on octagon-shaped shells to achieve a higher payload (2 bpp). It enlarges the reference matrix from hexagon to octagon, each octagon-shaped shell has 16 distinct digits (from 0 to 15) and can embed four binary secret bits simultaneously. Leng [36] improved Kurup *et al.*'s scheme by using regular octagon-shaped shells to achieve a higher payload (2.5 bpp) and stego-image quality in 2017.

Furthermore, Leng and Tseng [37] extended the scheme of Leng [36] to search for all possible solutions to maximize the payload of the octagon-shaped shell reference matrix. It achieves 3.5 bpp payloads, and the stego-image quality is suitable (approximately 30.60 dB). In 2019, Leng [38] generalized a data hiding scheme based on octagon-shaped shells. The octagon-shaped shell reference matrix is built

TABLE 2. Comparison among the reference matrix-based method.

Method	Reference Matrix	Payload	Method	Reference Matrix	Payload
Lee <i>et al.</i> [31]		1.5 bpp	Kurup <i>et al.</i> [35]		2 bpp
Chang <i>et al.</i> [32]		1.5 bpp	Leng [36]		2.5 bpp
Chang <i>et al.</i> [33] Liu <i>et al.</i> [34]		1.5 bpp 2 bpp	Leng and Tseng [37] Leng [38]		3.5 bpp 4 bpp

Note. $m \cdot n = 2^k + 4$ in [29]; $m \cdot n - 4 \cdot \frac{k(k+1)}{2} = 2^w$ in [33]

using four parameters: m , n , k , and w , where m and n are the width and height of the reference matrix, k is the width and height of the corner of the octagon shell, and w is the number of secret bits to embed. However, at $(m, n, \text{ and } k) = (14, 20, \text{ and } 3)$, a high payload (4 bpp) with good visual quality (34.70 dB) was achieved. The comparison among the methods [31]–[38] based on the reference matrix-based method is shown in Table 2.

To enhance security, a two-layer EMD scheme was proposed by [39]–[42]. Xie *et al.* [39] proposed a two-layer turtle shell matrix method to achieve 2.5 bpp payloads. Wang *et al.* [40] proposed an improved method that embeds secret bits in the two-dimensional space of the reference matrix twice and implied two 5-ary digits at a time. Under a two-layer magic matrix, Shen *et al.* [41] proposed a method that can embed secret bits in various notational systems by combining many signet sizes and achieving 3.5 bpp payloads. Lee *et al.* [42] proposed a new scheme based on a two-layer octagon-shaped shell matrix in which each cover pixel pair can carry 7-bit sub-streams of secret data and achieve an embedding capacity of 3.5 bpp. These schemes [39]–[42] based on the two-layer EMD scheme are described in detail in Section 2. The above methods use the reference matrix instead of the extraction function in the embedding and extraction procedures. Reference matrix-based schemes provide a balance between the payload and stego-image quality. The comparison among the methods [39]–[42] based on the two-layer reference matrix-based method is shown in Table 3.

In this study, we propose a multilayer steganographic method using an improved exploiting modification direction scheme, which includes the abovementioned advantages,

TABLE 3. Comparison among the two-layer’s reference matrix-based method.

Method	Notational system	Payload
Xie <i>et al.</i> [39]	binary	2.5 bpp
Wang <i>et al.</i> [40]	5-ary	2.32 bpp
Shen <i>et al.</i> [41]	binary	3.5 bpp
Lee <i>et al.</i> [42]	binary	3.5 bpp

using binary secret data, increasing the payload, and enhancing data security.

The remainder of this paper is organized as follows. Section 2 briefly reviews the introduction of Lee *et al.*’s IEMD scheme, Xie *et al.*’s two-layer turtle shell matrix scheme, Wang *et al.*’s scheme, Shen *et al.*’s two-layer magic matrix scheme, and Lee *et al.*’s two-layer octagon-shaped shell matrix scheme. Section 3 describes the proposed data-hiding scheme, and the experimental results are presented in Section 4. Finally, conclusions are presented in Section 5.

II. RELATED WORKS

In this section, five schemes [31], [39]–[42] are described in detail. Lee *et al.* [31] showed the main idea of the proposed method, as well as how to reduce the distortion when the embedding procedure. The current two-layer embedding schemes are demonstrated in [39]–[42].

A. IEMD SCHEME

To achieve a high payload and solve the problem using non-binary notational system secret data of the EMD method,

Lee et al. [31] increased the notational system state from five to eight, which conforms to the binary notational system. In the embedding procedure, every pixel pair can embed three binary secret bits at a time. The extraction function f and embedding procedure are as follows:

$$f(x, y) = (x + 3y) \bmod 8. \quad (1)$$

- Step 1: The cover image C is divided into non-overlapping pixel pairs (p_i, p_{i+1}) .
- Step 2: Convert the binary secret bits B into one 8-ary secret digit s .
- Step 3: Denote the pixel pair (p_i, p_{i+1}) using extraction function f .
- Step 4: Adjust the pixel pair (p_i, p_{i+1}) such that the secret digits match the following condition:
- (1) If $f(p_i, p_{i+1}) = s$, the cover pixel pair (p_i, p_{i+1}) remains unchanged.
 - (2) If $f(p_i + 1, p_{i+1}) = s$, the cover pixel pair (p_i, p_{i+1}) is changed to $(p_i + 1, p_{i+1})$.
 - (3) If $f(p_i - 1, p_{i+1}) = s$, the cover pixel pair (p_i, p_{i+1}) is changed to $(p_i - 1, p_{i+1})$.
 - (4) If $f(p_i, p_{i+1} + 1) = s$, the cover pixel pair (p_i, p_{i+1}) is changed to $(p_i, p_{i+1} + 1)$.
 - (5) If $f(p_i, p_{i+1} - 1) = s$, the cover pixel pair (p_i, p_{i+1}) is changed to $(p_i, p_{i+1} - 1)$.
 - (6) If $f(p_i + 1, p_{i+1} + 1) = s$, the cover pixel pair (p_i, p_{i+1}) is changed to $(p_i + 1, p_{i+1} + 1)$.
 - (7) If $f(p_i + 1, p_{i+1} - 1) = s$, the cover pixel pair (p_i, p_{i+1}) is changed to $(p_i + 1, p_{i+1} - 1)$.
 - (8) If $f(p_i - 1, p_{i+1} + 1) = s$, the cover pixel pair (p_i, p_{i+1}) is changed to $(p_i - 1, p_{i+1} + 1)$.
- Step 5: Repeat Steps 3 and 4 to embed all secret digits and obtain the stego-image. The embedding process is complete.

In the extraction procedure, each stego pixel pair can extract three binary secret bits at a time. The extraction procedure is described as follows:

- Step 1: Divide stego-image I into non-overlapping pixel pairs (p'_i, p'_{i+1}) .
- Step 2: Extract one 8-ary secret digit s by denoting the extraction function, where $f(p'_i, p'_{i+1}) = s$.
- Step 3: Convert the secret digit s into three secret bits B in binary notational system.
- Step 4: Repeat steps 2 and 3 to extract all the secret digits until the extraction process is complete.

For example, in Fig. 2, the cover pixel pair (p_i, p_{i+1}) is $(5, 6)$, while the to-be-embedded secret bit is $(110)_2$. First, the secret data are converted to secret bit $s = (6)_8$ in the 8-ary notational system. Second, the extraction function, $f(5, 6) = 7$. According to step 3, we find that $f(4, 6) = 6$ satisfies the requirement. Hence, replacing the cover pixel pair $(5, 6)$ with $(4, 6)$, the stego pixel pair (p'_i, p'_{i+1}) is successfully obtained. On the receiver side, denotes the stego pixel pair $(p'_i, p'_{i+1}) = (4, 6)$ by the extraction function $f(4, 6) = (6)_8$, which is converted into a binary notational

	0	1	2	3	4	5	6	7	8	...	x_i
0	0	1	2	3	4	5	6	7	0		
1	3	4	5	6	7	0	1	2	3		
2	6	7	0	1	2	3	4	5	6		
3	1	2	3	4	5	6	7	0	1		
4	4	5	6	7	0	1	2	3	4		
5	7	0	1	2	3	4	5	6	7		
6	2	3	4	5	6	7	0	1	2		
7	5	6	7	0	1	2	3	4	5		
8	0	1	2	3	4	5	6	7	0		
⋮											
x_2											

FIGURE 2. Mapping matrix of the IEMD method.

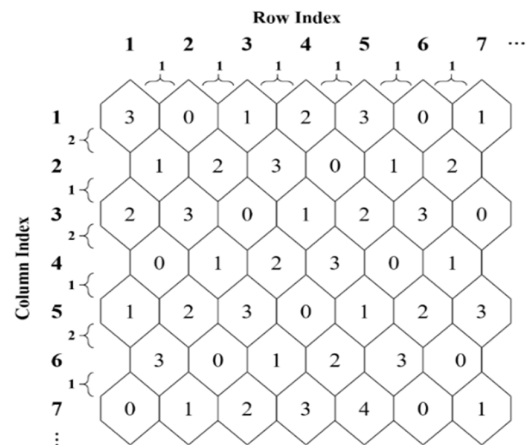


FIGURE 3. An example of the turtle-shell matrix.

system as $(110)_2$. Therefore, the confidential data were successfully extracted. The main advantage of the IEMD scheme is that the notational system is binary and the payload is 1.5 bpp, which is compared with the EMD method without loss of visual quality and security.

B. TWO-LAYER TURTLE SHELL MATRIX SCHEME

Xie et al. [39] constructed a turtle shell matrix with the extra attribute of a 4-ary digit, as shown in Fig. 3, which is the first two-layer scheme applied to the EMD scheme. The construction rules are as follows: first, the values of an element in a row change with the ascent or descent of the gradient, and the value is one, and the value of a column of elements changes with the ascent or descent of the gradient, and the variable values are two and three.

Furthermore, an additional attribute is assigned to each pixel pair $m(x, y)$, called $t_m(x, y)$. The algorithm for calculating $t_m(x, y)$ can be described as follows: conduct an exclusive

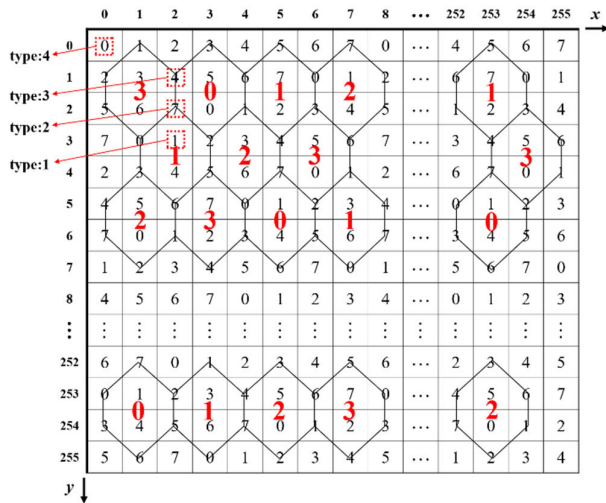


FIGURE 4. Four type attributes of $m(x, y)$.

operation on all types of turtle shell involved. The algorithm categorizes all pixel pairs into one of the following four cases, as shown in Fig. 4.

- C1se 1: No turtle shells were involved, such as $m(0, 0)$. The type attribute of the specified element was set to 4, for example, $t_m(0, 0) = 4$.
- C2se 2: Only one turtle shell is involved, such as in $m(2, 3)$. The type attribute of the specified element is set to the type of turtle shell, for example, $t_m(2, 3) = 1$.
- C3se 3: Two turtle shells were involved, such as $m(2, 1)$. Then, the type attribute of a specified element is the result of the operation of binary representations of the two turtle shell types, for example, $t_m(2, 1) = 3$.
- C4se 4: Three turtle shells are involved, such as $m(2, 2)$. The type attribute of a specified element is the result of an operation of binary representations of the three turtle shell types, for example, $t_m(2, 2) = 2$.

According to Fig. 4, $m(x, y)$ in the turtle shell matrix M is assigned to a corresponding type attribute $t_m(x, y)$. In other words, a two-layer turtle shell matrix is constructed. The first layer is the matrix $M = [m(x, y)]_{256 \times 256}$, and the second layer matrix is $T_m = [t_m(x, y)]_{256 \times 256}$. Fig. 5 shows a part of the two-layer turtle shell matrix.

In the embedding procedure, every pixel pair can embed five binary secret bits at a time. The embedding procedure is described as follows:

- Step 1: The cover image C is divided into non-overlapping pixel pairs (p_i, p_{i+1}) . Consider (p_i, p_{i+1}) as the coordinates of matrix M to specify the value $m(p_i, p_{i+1})$ with the corresponding type attribute $t_m(p_i, p_{i+1})$.
- Step 2: Divide a secret binary stream B into five binary bit sub-streams b . For each sub-stream, the first two bits are converted into a 4-ary digit s_1 and the last three bits are converted into an octal digit s_2 , viz., $s = s_1 || s_2$, where “||” denotes the string concatenation operator.

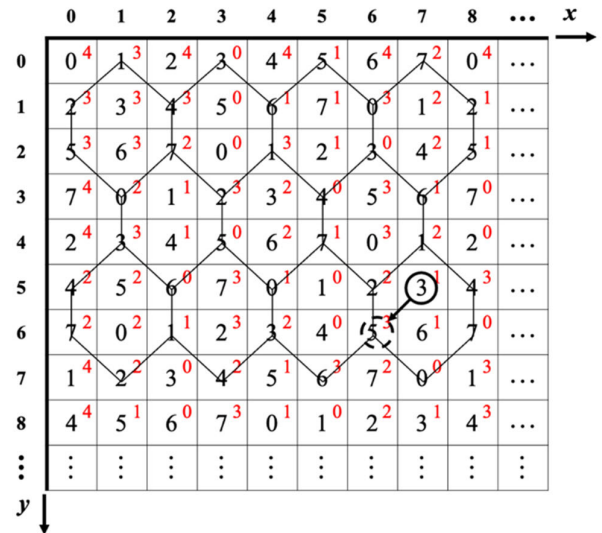


FIGURE 5. Part of the two-layer turtle shell matrix.

- Step 3: A sub-stream s is embedded in a pixel pair (p_i, p_{i+1}) according to the following rule: Find the closest element $m(p'_i, p'_{i+1})$, where $m(p'_i, p'_{i+1}) = s_2$ and $t_m(p'_i, p'_{i+1}) = s_1$, by spiral scanning from $m(p_i, p_{i+1})$. Replace (p_i, p_{i+1}) with (p'_i, p'_{i+1}) in the cover image to embed the sub-stream s consisting of s_1 and s_2 .
- Step 4: Repeat Step 3 until all sub-streams are embedded.

In the extraction procedure, each stego pixel pair can extract five binary secret bits at a time. The extraction procedure is described as follows:

- Step 1: Divide stego-image I into non-overlapping pixel pairs (p'_i, p'_{i+1}) .
- Step 2: Extract a sub-stream s ($s = s_1 || s_2$) by mapping the pixel pair (p'_i, p'_{i+1}) on the two-layer turtle shell matrix (M and T_m), where $s_1 = t_m(p'_i, p'_{i+1})$ and $s_2 = m(p'_i, p'_{i+1})$, respectively.
- Step 3: Convert a sub-stream s ($s = s_1 || s_2$) into five binary-bit sub-streams b .
- Step 4: Finally, when all sub-streams b have been extracted, they are combined to form a secret binary stream B .

For example, in Fig. 5, the cover pixel pair is $(7, 5)$, the to-be-embedded bit stream is $(11, 101)_2$, which is divided into two sub-streams, namely, $s_1 = (11)_2$ and $s_2 = (101)_2$. Convert s_1 and s_2 into $(3)_4$ and $(5)_8$, respectively. Find the closest satisfactory element by spiral scanning the surrounding elements, that is, $(6, 6)$, where $m(6, 6) = 5$ and $t_m(6, 6) = 3$ (Fig. 5). Then, replace the cover pixel pair $(7, 5)$ with $(6, 6)$ as the stego-image. On the receiver side, the stego-image is divided into non-overlapping pixel pairs, which can be used to extract five binary secret bits. Two secret digits are extracted, that is, $(6, 6)$, where $t_m(6, 6) = (3)_4$ and $m(6, 6) = (5)_8$, and then converted into binary streams $(11)_2$ and $(101)_2$, respectively. Finally, the embedded binary secret data $(11, 101)_2$ are extracted completely. Therefore, each pixel pair can embed five binary secret bits using a two-layer turtle shell matrix at a time.

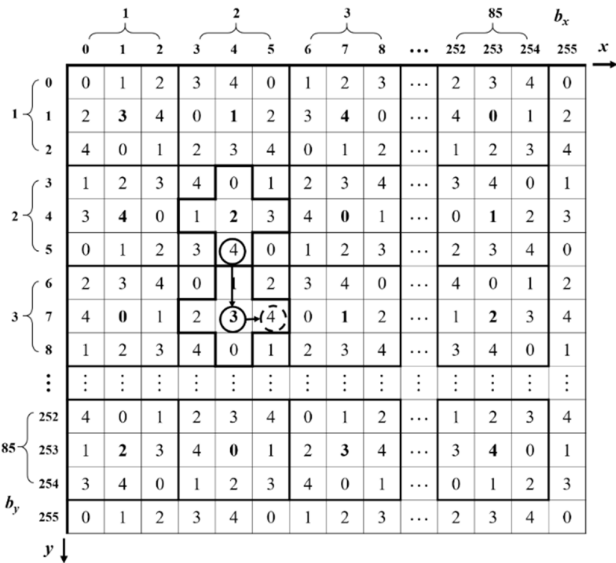


FIGURE 6. The reference matrix M sized 256×256 and blocks sized by 3×3 .

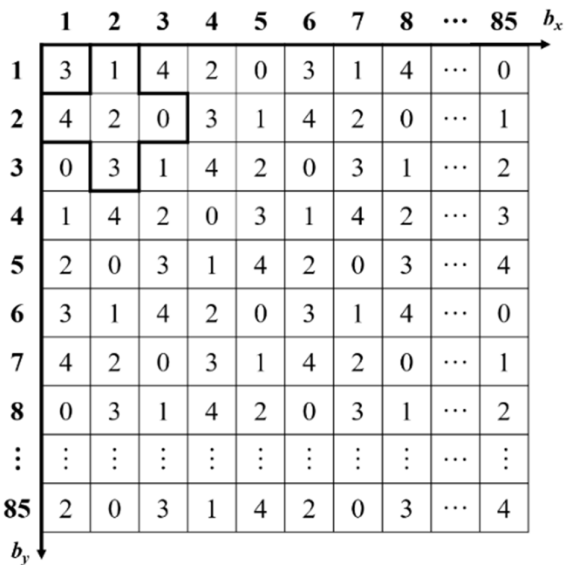


FIGURE 7. The sub-matrix B .

C. WANG et al.'s SCHEME

Wang et al. [40] proposed a new scheme for data hiding based on a reference matrix to achieve a greater payload. Compared with the published scheme, their research embedded two 5-ary secret digits in a two-layer reference matrix at a time. The reference matrix $M = [m(x, y)]_{256 \times 256}$ was constructed using Eq. (2) and divided into 3×3 blocks, as shown in Fig. 6.

$$f(x, y) = (x + 2y) \bmod 5. \tag{2}$$

According to Eq. (3), for every pixel pair (p_i, p_{i+1}) , we have an element $M(p_i, p_{i+1})$, which belongs to $B(b_x, b_y)$, and b_x, b_y can be calculated using Eq. (3) as follows: Construct a sub-matrix B on the reference matrix M by extracting the center value of each 3×3 blocks, as shown in Fig. 7. This causes the value of one cover pixel to be increased or

decreased by 1.

$$\begin{cases} b_x = \lceil (p_i + 1)/3 \rceil, \\ b_y = \lceil (p_{i+1} + 1)/3 \rceil. \end{cases} \tag{3}$$

$$B(x, y) = M(3x - 2, 3y - 2). \tag{4}$$

In the embedding procedure, every pixel pair can embed two 5-ary secret bits at a time. The embedding procedure is described as follows:

- Step 1: The cover image C is divided into non-overlapping pixel pairs (p_i, p_{i+1}) .
- Step 2: Convert a secret binary stream B into a secret digit stream S in 5-ary notational system.
- Step 3: Select a pixel pair (p_i, p_{i+1}) and calculate the blocks $B(b_x, b_y)$ containing the located element $M(p_i, p_{i+1})$, where b_x and b_y can be calculated using Eq. (3).
- Step 4: Embed two 5-ary secret bits $(s_i, s_{i+1})_5$ by selecting the most suitable replacement element $B(b'_x, b'_y)$ and $M(p'_i, p'_{i+1})$, where $s_{i+1} = B(b'_x, b'_y)$ and $s_i = M(p'_i, p'_{i+1})$.
- Step 5: Based on the principle of the Euclidean distance, select the minimum distance.
- Step 6: Repeat Step 3–6 until all secret digits are embedded.

In the extraction procedure, each stego pixel pair can extract two 5-ary secret digits at a time. The extraction procedure is described as follows:

- Step: 1 Divide stego-image I into non-overlapping pixel pairs (p'_i, p'_{i+1}) .
- Step: 2 Extract a secret digit s_{i+1} by mapping each pixel pair (p'_i, p'_{i+1}) on the reference matrix M , where $s_{i+1} = M(p'_i, p'_{i+1})$.
- Step: 3 Calculate $B(b'_x, b'_y)$ using Eq. (4), and extract the embedded secret bit s_i using reference matrix B , where $s_i = B(b'_x, b'_y)$.
- Step: 4 Convert 5-ary sub-stream s into binary sub-stream b .
- Step: 5 Finally, when all sub-streams are extracted, they are combined into a secret binary stream B .

For example, in Fig. 6, the cover pixel pair (4, 5) and the to-be-embedded bit stream $(43)_5$ are divided into two sub-streams, namely $s_i = (4)_5$ and $s_{i+1} = (3)_5$. First, $(b_x, b_y) = (2, 2)$ is calculated using Eq. (3). Because $B(2, 2) = 2 \neq 3$, we search the eight adjacent blocks in the reference matrix M , where $B(1, 1) = B(2, 3) = 3$, which can be chosen as candidate blocks. Using Eq. (4), we obtain $M(2, 1) = M(0, 2) = 4$ in $B(1, 1)$ and $M(3, 8) = M(5, 7) = 4$ in $B(2, 3)$. Finally, according to the principle of the minimum Euclidean distance, the most suitable replacement $M(5, 7)$ is selected. Then, replace the cover pixel pair (4, 5) with (5, 7) as the stego pixel pair.

On the receiver side, the stego-image is divided into non-overlapping pixel pairs, which can be extracted from two 5-ary secret digits. $B(2, 3)$ contains the element $M(5, 7)$. Then, the secret digits $s_i = M(5, 7) = (4)_5$ and $s_{i+1} = B(2, 3) = (3)_5$. Finally, the embedded secret data $(43)_5$ were extracted completely. Therefore, each pixel pair can extract two 5-ary secret digits using a two-layer reference

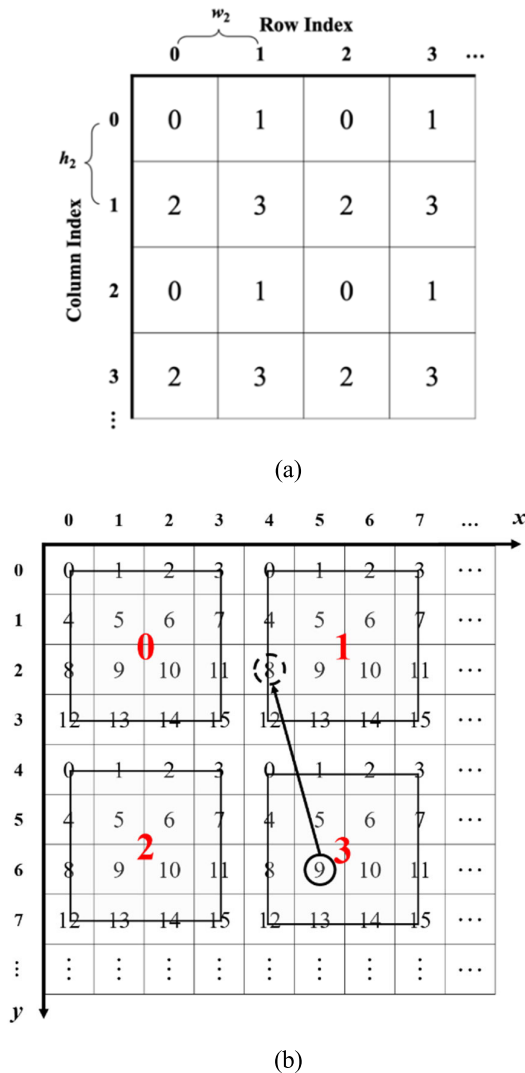


FIGURE 8. Construction of two-layer magic matrix: (a) The second-layer matrix T , (b) Two-layer magic matrix.

matrix at a time. The disadvantage of this second two-layer scheme used in the EMD scheme is that it employs a 5-ary notational system.

D. TWO-LAYER MAGIC MATRIX SCHEME

Shen et al. [41] proposed a two-layer magic matrix by improving the magic matrix. The first-layer matrix $M = [m(x, y)]_{256 \times 256}$ in grayscale images. The second-layer matrix $T = [t(x, y)]$ contains $(256/w_1) \times (256/h_1)$ magic signets, which are shown as an example in Fig. 8(a), where magic signets are 64×64 and each signet is 4×4 . For example, in the data embedding procedure, we take $w_1 = h_1 = 4$, $w_2 = h_2 = 2$ and assign $t(x, y)$ to the type attribute in $\{0, 1, 2, 3\}$, for example, $t(0, 0) = 2$. Therefore, a two-layer magic matrix was constructed, as shown in Fig. 8(b).

In the embedding procedure, every pixel pair can embed six binary secret bits at a time. The embedding procedure is described as follows:

- Step 1: The cover image C is divided into non-overlapping pixel pairs (p_i, p_{i+1}) . Consider (p_i, p_{i+1}) as the coordinates of matrix M to specify the value $m(p_i, p_{i+1})$ with the corresponding type attribute $t_m(p_i, p_{i+1})$.
- Step 2: Divide a binary secret stream B into six binary bit sub-streams b . For each sub-stream, convert the first two bits into a 4-ary digit s_1 and the last four bits into a 16-ary digit s_2 , viz., $s = s_1 || s_2$, where “||” denotes the string concatenation operator.
- Step 3: A sub-stream s is embedded in the pixel pair (p_i, p_{i+1}) according to the following rules: Find the closest element $m(p'_i, p'_{i+1})$, where $m(p'_i, p'_{i+1}) = s_2$ and $t_m(p'_i, p'_{i+1}) = s_1$, by spiral scanning from $m(p_i, p_{i+1})$. Replace (p_i, p_{i+1}) with (p'_i, p'_{i+1}) in the cover image to embed the sub-stream s consisting of s_1 and s_2 .
- Step 4: Repeat Step 3 until all sub-streams are embedded.

In the extraction procedure, each stego pixel pair can extract six binary secret bits at a time. The extraction procedure is described as follows:

- Step 1: Divide stego-image I into non-overlapping pixel pairs (p'_i, p'_{i+1}) .
- Step 2: Extract a subsecret stream $s (s = s_1 || s_2)$ by mapping each pixel pair (p'_i, p'_{i+1}) on the reference matrix M , where $s_1 = t_m(p'_i, p'_{i+1})$ and $s_2 = m(p'_i, p'_{i+1})$, respectively.
- Step 3: Convert sub-stream $s (s = s_1 || s_2)$ into binary sub-streams b .
- Step 4: Finally, when all sub-streams b have been extracted, they are combined into a secret binary stream B .

For example, in Fig. 8(b), the cover pixel pair (p_i, p_{i+1}) is $(5, 6)$, and the to-be-embedded secret bit is $(01, 1000)_2$, which is divided into two sub-streams, namely, $s_1 = (01)_2$ and $s_2 = (1000)_2$. Convert s_1 and s_2 into $(1)_4$ and $(8)_{16}$, respectively. Find the closest satisfactory element by spiral scanning the surrounding elements, that is, $(4, 2)$, where $m(4, 2) = 8$ and $t_m(4, 2) = 1$. Then, replace the cover pixel pair $(5, 6)$ with $(4, 2)$ as the stego-image. On the receiver side, the stego-image is divided into non-overlapping pixel pairs, which can be extracted from six binary bits. Two secret digits are extracted, that is, $(4, 2)$, where $t_m(4, 2) = (1)_4$ and $m(4, 2) = (8)_{16}$, and then converted into binary streams $(01)_2$ and $(1000)_2$, respectively. Finally, the embedded confidential data $(01, 1000)_2$ were completely extracted. Therefore, each pixel pair can extract six binary bits using a two-layer magic matrix at a time.

E. TWO-LAYER OCTAGON-SHAPED SHELL MATRIX SCHEME

Lee et al. [42] proposed a two-layer octagon-shaped shell matrix in which each cover pixel pair can embed seven bits at a time. The first-layer matrix $M = [m(x, y)]_{256 \times 256}$ based on the regular octagon-shaped shells was proposed by Leng and Tseng [37]. The second-layer octagon-shaped shell matrix $T = [t(x, y)]_{51 \times 51}$ is shown in Fig. 9(a). For each element $m(x, y)$ in the bottom layer of the reference matrix M ,

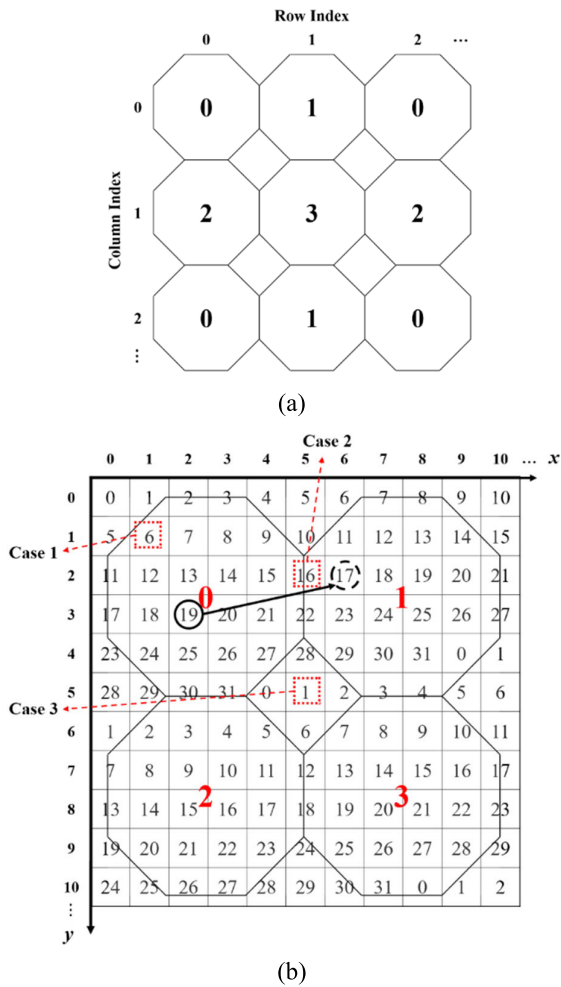


FIGURE 9. Construction of a two-layer octagon-shaped shell matrix: (a) The second-layer octagon-shaped shell matrix T , (b) Two-layer octagon-shaped shell matrix.

we assign $t_m(x, y)$ to the type attribute, which corresponds to each element $m(x, y)$. Each type attribute $t_m(x, y)$ is categorized into three cases, as shown in Fig. 9(b).

- Case 1: One octagon involved, for example, $m(1, 1) = 6$. The type attribute of the specified element is assigned as 0, for example, $t_m(0, 0) = 0$.
- Case 2: Two octagons, $m(5, 2) = 16$. The type attribute of the two octagons can be calculated using Eq. (5), i.e., $t(0, 0) = 0$ and $t(1, 0) = 1$. Therefore, we obtain $t_m(5, 2) = 0$, as shown in Fig. 9(b).
- Case 3: Not involved in any octagon, e.g., $t_m(5, 5) = 0$.

$$t_m(x, y) = \begin{cases} t(x, y) + t(i, j) - 1, & \text{if } t(x, y) + t(i, j) < 3, \\ t(x, y) + t(i, j) - 2, & \text{if } t(x, y) + t(i, j) > 3. \end{cases} \quad (5)$$

In the embedding procedure, every pixel pair can embed seven binary secret bits at a time. The embedding procedure is described as follows:

Step 1: The cover image C is divided into non-overlapping pixel pairs (p_i, p_{i+1}) . Consider (p_i, p_{i+1}) as the

coordinates of matrix M to specify the value $m(p_i, p_{i+1})$ with the corresponding type attribute $t_m(p_i, p_{i+1})$.

- Step 2: Divide a binary secret stream B into seven binary bit sub-streams b . For each sub-stream, convert the first two bits into a 4-ary digit s_1 and the last five bits into a 32-ary digit s_2 , viz., $s = s_1 || s_2$, where “||” denotes the string concatenation operator.
- Step 3: A sub-stream s is embedded in the pixel pair (p_i, p_{i+1}) according to the following rule: Find the closest element $m(p'_i, p'_{i+1})$ by searching in a square of 25×25 centered on $m(p_i, p_{i+1})$, where $m(p'_i, p'_{i+1}) = s_2$ and $t_m(p'_i, p'_{i+1}) = s_1$. Replace (p_i, p_{i+1}) with (p'_i, p'_{i+1}) in the cover image to embed the sub-stream s consisting of s_1 and s_2 .
- Step 4: Repeat Step 3 until all sub-streams are embedded.

In the extraction procedure, each stego pixel pair can extract seven binary secret bits at a time. The extraction procedure is described as follows:

- Step 1: Divide stego-image I into non-overlapping pixel pairs (p'_i, p'_{i+1}) .
- Step 2: Extract a secret sub-stream s ($s = s_1 || s_2$) by mapping each pixel pair (p'_i, p'_{i+1}) on the reference matrix M , where $s_1 = t_m(p'_i, p'_{i+1})$ and $s_2 = m(p'_i, p'_{i+1})$, respectively.
- Step 3: Convert sub-stream s ($s = s_1 || s_2$) into binary sub-streams b .
- Step 4: Finally, when all sub-streams b have been extracted, they are combined into a secret binary stream B .

For example, in Fig. 9(b), the cover pixel pair $(p_i, p_{i+1}) = (2, 3)$, and the to-be-embedded secret bit is $(01, 10001)_2$, which is divided into two sub-streams, namely, $s_1 = (01)_2$ and $s_2 = (10001)_2$. Convert s_1 and s_2 into $(1)_4$ and $(17)_{32}$, respectively. Find the closest satisfactory element by spiral scanning the surrounding elements, that is, $(6, 2)$, where $m(6, 2) = 17$ and $t_m(6, 2) = 1$. Then, replace the cover pixel pair $(2, 3)$ with $(6, 2)$ as the stego-image. On the receiver side, the stego-image is divided into non-overlapping pixel pairs, which can be extracted from seven binary bits. Two secret digits are extracted, that is, $(6, 2)$, where $t_m(6, 2) = (1)_4$ and $m(6, 2) = (17)_{32}$, and then converted into binary streams $(01)_2$ and $(10001)_2$, respectively. Finally, the embedded secret data $(01, 10001)_2$ were completely extracted. Therefore, each pixel pair can extract seven binary secret bits using a two-layer octagon-shaped shell matrix at a time.

III. THE PROPOSED METHOD

This study proposes a multilayer steganographic method based on the IEMD scheme [31], which includes the advantages of multilayer embedding, using binary secret data, high payload, and stego-image quality that satisfies the human vision system sensitivity (PSNR value greater than 30 dB). The proposed scheme can be divided into five phases: (1) multilayer reference matrix construction, (2) embedding, (3) data embedding, (4) extraction, and (5) data extraction.

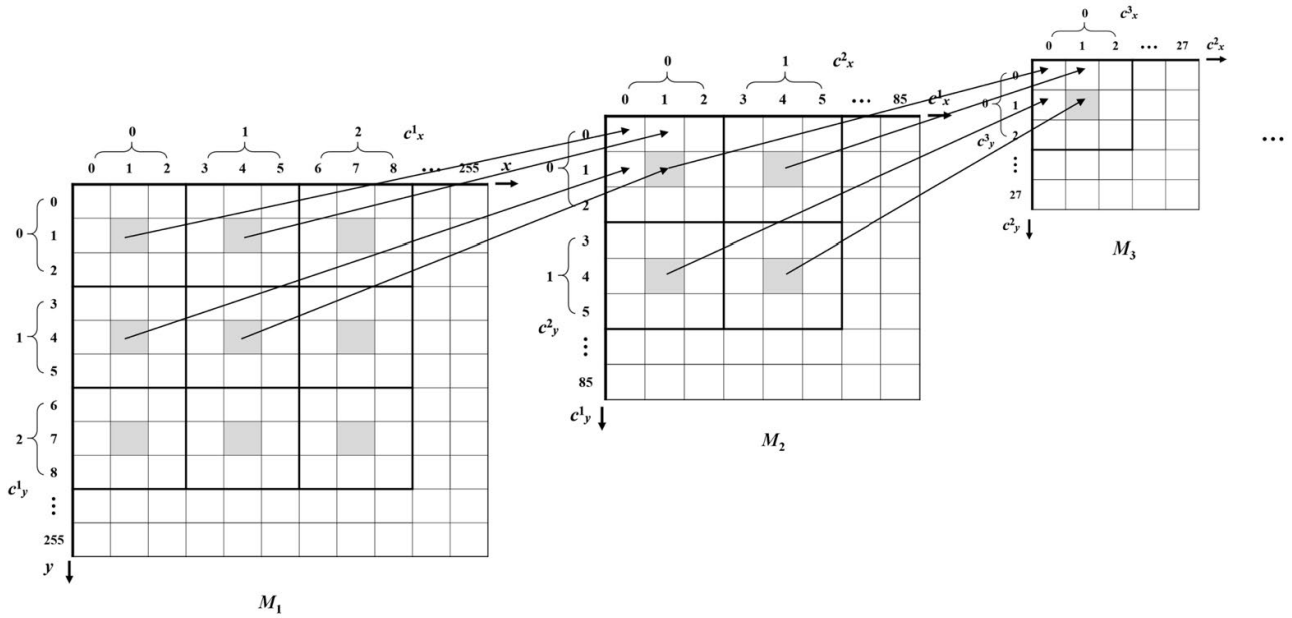


FIGURE 10. An example of multilayer reference matrix.

		0			1			2			...			84			c^1_x
		0	1	2	3	4	5	6	7	8	...	252	253	254	255	x	
0	0	0	1	2	3	4	5	6	7	0	...	4	5	6	7		
	1	3	4	5	6	7	0	1	2	3	...	7	0	1	2		
	2	6	7	0	1	2	3	4	5	6	...	2	3	4	5		
1	3	1	2	3	4	5	6	7	0	1	...	5	6	7	0		
	4	4	5	6	7	0	1	2	3	4	...	0	1	2	3		
	5	7	0	1	2	3	4	5	6	7	...	3	4	5	6		
2	6	2	3	4	5	6	7	0	1	2	...	6	7	0	1		
	7	5	6	7	0	1	2	3	4	5	...	1	2	3	4		
	8	0	1	2	3	4	5	6	7	0	...	4	5	6	7		
84	252	4	5	6	7	0	1	2	3	4	...	0	1	2	3		
	253	7	0	1	2	3	4	5	6	7	...	3	4	5	6		
	254	2	3	4	5	6	7	0	1	2	...	6	7	0	1		
255	5	6	7	0	1	2	3	4	5	...	1	2	3	4			
c^1_y	y																

FIGURE 11. The first-layer reference matrix M_1 .

For convenience, the definition of symbols is showed in appendix. An example of a multilayer reference matrix is presented in Fig. 10.

A. MULTILAYER REFERENCE MATRIX CONSTRUCTION PROCEDURE

In this section, we construct a three-layer reference matrix for data hiding. First, we generated 256 × 256 sizes of the first-layer reference matrix M_1 using Eq. (6) and divided it into 3 × 3 blocks, as shown in Fig. 11. The center of each 3 × 3 blocks (marked in dark color) is the second-layer reference matrix M_2 .

$$f(x, y) = (x + 3y) \bmod 8. \tag{6}$$

$$M_1 = \{m_1(x, y) | x, y = 0, 1, \dots, 255\}. \tag{7}$$

Second, for every pixel pair (p_i, p_{i+1}) in M_1 , we have an element $M_1(p_i, p_{i+1})$, which belongs to $M_2(c^1_x, c^1_y)$, and c^1_x, c^1_y calculated using Eq. (8), respectively: Then, we generated 85 × 85-size of second-layer reference matrix M_2 and divided it into blocks of 3 × 3-size, as shown in Fig. 12. The center of each 3 × 3 blocks (marked in dark color) is the third-layer reference matrix M_3 . In other words, the second-layer reference matrix M_2 is the center of each of the 3 × 3 blocks (marked in dark color) in Fig. 11.

$$\begin{cases} c^n_x = \lfloor \frac{p_i}{3^n} \rfloor \\ c^n_y = \lfloor \frac{p_{i+1}}{3^n} \rfloor \end{cases} \text{ for } n = 1, 2. \tag{8}$$

$$M_2 = \{m_2(c^1_x, c^1_y) | c^1_x, c^1_y = 0, 1, \dots, 84\}. \tag{9}$$

Finally, for every pixel pair (p_i, p_{i+1}) in M_1 , we have an element $M_1(p_i, p_{i+1})$, which belongs to $M_3(c^2_x, c^2_y)$, and c^2_x, c^2_y calculated using Eq. (8), respectively: Then, we generated 28 × 28-size of third-layer reference matrix M_3 , which is shown in Fig. 13. In other words, the third-layer reference matrix M_3 is the center of each 9 × 9 blocks in Fig. 11 and the center of each 3 × 3 blocks (marked in dark color) in Fig. 12.

$$M_3 = \{m_3(c^2_x, c^2_y) | c^2_x, c^2_y = 0, 1, \dots, 27\}. \tag{10}$$

The corresponding of the three-layer reference matrix in Figs. 11–13 satisfies Eq. (11), e.g., $M_3(0, 0) = M_2(1, 1) = M_1(4, 4)$ as shown in Fig. 14.

$$M_n(x, y) = M_{n-1}(3x + 1, 3y + 1). \tag{11}$$

Furthermore, a three-layer reference matrix can be created, which can embed nine binary secret digits in most of the non-overlapping pixel pairs.

		0	1	2	3	4	5	6	7	8	...	82	83	84	85	c^2_x
0	0	4	7	2	5	0	3	6	1	4	...	5	0	3	6	c^1_x
	1	5	0	3	6	1	4	7	2	5	...	6	1	4	7	
	2	6	1	4	7	2	5	0	3	6	...	7	2	5	0	
1	3	7	2	5	0	3	6	1	4	7	...	0	3	6	1	c^1_y
	4	0	3	6	1	4	7	2	5	0	...	1	4	7	2	
	5	1	4	7	2	5	0	3	6	1	...	2	5	0	3	
2	6	2	5	0	3	6	1	4	7	2	...	3	6	1	4	c^2_y
	7	3	6	1	4	7	2	5	0	3	...	4	7	2	5	
	8	4	7	2	5	0	3	6	1	4	...	5	0	3	6	
...
27	82	5	0	3	6	1	4	7	2	5	...	0	3	6	1	c^1_x
	83	6	1	4	7	2	5	0	3	6	...	1	4	7	2	
	84	7	2	5	0	3	6	1	4	7	...	2	5	0	3	
85	0	3	6	1	4	7	2	5	0	...	3	6	1	4	c^1_y	

FIGURE 12. The second-layer reference matrix M_2 .

		0	1	2	3	4	5	6	7	...	27	c^2_x
0		0	1	2	3	4	5	6	7	...	3	c^2_y
1		3	4	5	6	7	0	1	2	...	6	
2		6	7	0	1	2	3	4	5	...	1	
3		1	2	3	4	5	6	7	0	...	4	
4		4	5	6	7	0	1	2	3	...	7	
5		7	0	1	2	3	4	5	6	...	2	
6		2	3	4	5	6	7	0	1	...	5	
7		5	6	7	0	1	2	3	4	...	0	
...		
27		1	2	3	4	5	6	7	0	...	4	

FIGURE 13. The third-layer reference matrix M_3 .

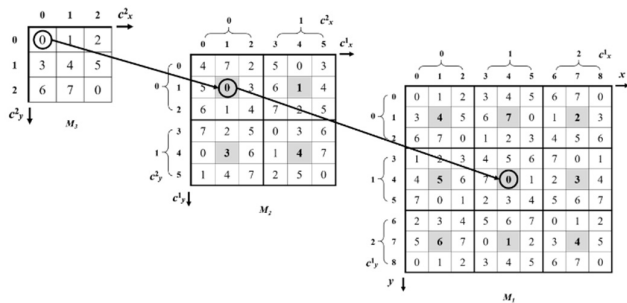


FIGURE 14. The corresponding of three-layer reference matrix.

B. THE EMBEDDING PROCEDURE

In this section, the data embedding procedure of the proposed method is presented. Fig. 15 shows a flowchart of the data embedding procedure. Assume that a secret binary stream B with length N is embedded in the cover image C with

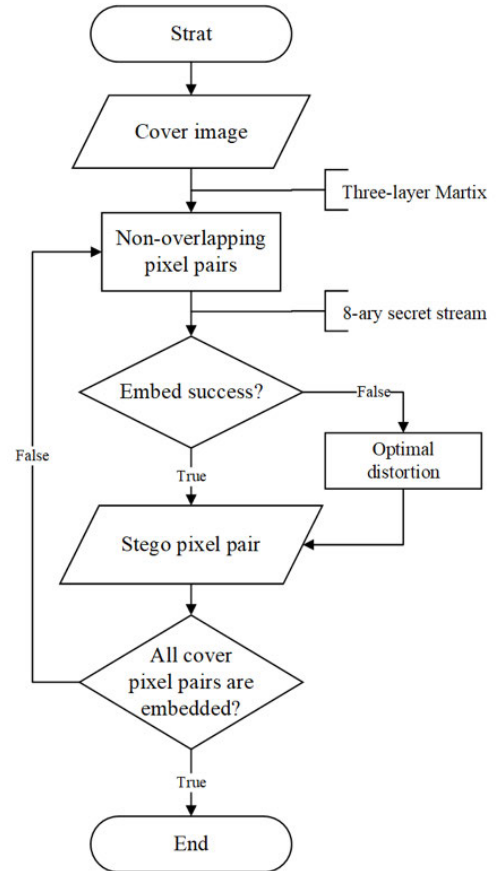


FIGURE 15. Flowchart of data embedding procedure.

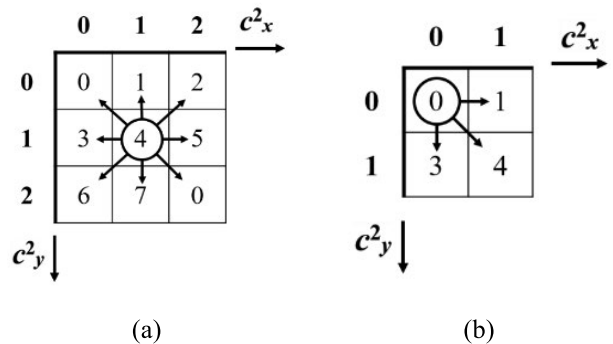


FIGURE 16. Two types of embedding algorithm: (a) Embed success, (b) Embed fault.

size $W \times H$ using the three-layer reference matrix. Initially, the cover image C is transferred into non-overlapping pixel pairs; the binary secret stream B is converted into 8-ary secret stream S . Then, three 8-ary secret digits are embedded in each pixel pair using the constructed three-layer reference matrix. Because the third secret digit is embedded first, the embedding algorithm is classified into one of the two cases shown in Fig. 16. One is embedding three 8-ary secret digit success, and the other is embedding false. It depends on whether the third secret digit of the 8-ary secret stream S can be embedded. Finally, we obtained the stego-image after all pixel pairs are modified.

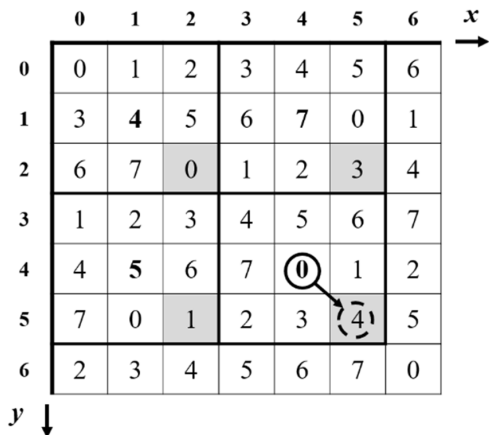


FIGURE 17. An example of embedding false.

C1se 1: Embed success, for example, in Fig. 16(a), assumes that $M_3(c_x^2, c_y^2) = M_3(1, 1)$, a part of the third-layer reference matrix is shown. Regardless of the third 8-ary secret digit, we can find the corresponding value around $M_3(1, 1)$. In Fig. 16(b), we assume that $M_3(c_x^2, c_y^2) = M_3(0, 0)$, the third 8-ary secret digit in $\{1, 3, 4\}$, we can find the corresponding value around $M_3(0, 0)$, which means that the third secret digit can be embedded. Therefore, three 8-ary secret digits are embedded in the pixel pair success.

C2se 2: Embed false, for example, in Fig. 16(b), it is assumed that $M_3(c_x^2, c_y^2) = M_3(0, 0)$, a part of the third-layer reference matrix is shown. When the third 8-ary secret digit in $\{2, 5, 6, 7\}$, we cannot find the corresponding value around $M_3(0, 0)$, which means that the third secret digit cannot be embedded. Therefore, in Fig. 17, the area marked with a dark color in M_1 does not represent secret digits. After calculating $M_1(4, 4)$, using Eq. (11), that is, $M_3(0, 0) = M_2(1, 1) = M_1(4, 4)$, modifies the pixel pair to the closest area.

In the embedding procedure, most of the pixel pairs can embed nine binary secret digits at a time. The embedding procedure is described as follows:

- Step 1: Generate a three-layer reference matrix (M_1, M_2 , and M_3), as shown in Figs. 10–13 in Section 3.1.
- Step 2: The cover image C is divided into non-overlapping pixel pairs (p_i, p_{i+1}) , and the coordinate of the reference matrix M_1 is considered to specify the value $m_1(p_i, p_{i+1})$.

$$C = \{(p_i, p_{i+1}) | i = 1, 3, \dots, W \times H - 1\}. \quad (12)$$

- Step 3: Convert a binary secret stream $B = b_1 b_2 \dots b_N$ into an 8-ary secret stream S . Three binary digits can be completely converted into one 8-ary digit.

$$S = \{(s_i, s_{i+1}, s_{i+2}) | i = 1, 4, \dots, \frac{N}{3} - 2\}. \quad (13)$$

- Step 4: Using Eq. (8), calculate a pixel pair (p_i, p_{i+1}) to find the center pixel pair (c_x^2, c_y^2) of 9×9 blocks in M_3 .

- Step 5: Three secret digits s_i, s_{i+1} , and s_{i+2} are embedded in the pixel pair (c_x^2, c_y^2) according to the following rules: first, the third secret digits s_{i+2} are embedded in a pixel pair (c_x^2, c_y^2) in M_3 . Find the closest element $m_3(c_x^2, c_y^2)$ by searching in a square of 3×3 centered on $M_3(c_x^2, c_y^2)$. The algorithm was categorized into two cases.

- Case 1: $m_3(c_x^2, c_y^2) = s_{i+2}$, which indicates embed success. The current pixel pair (c_x^2, c_y^2) can embed the third secret digit s_{i+2} . Then, using Eq. (11), embed two secret digits s_i, s_{i+1} , where $m_2(c_x^1, c_y^1) = s_{i+1}$ and $m_1(p'_i, p'_{i+1}) = s_i$ by searching in a square of 3×3 blocks. Replace the original pixel pair (p_i, p_{i+1}) with (p'_i, p'_{i+1}) as the stego pixel pair.
- Case 2: $m_3(c_x^2, c_y^2) \neq s_{i+2}$, which means that it embeds false. The pixel pair (c_x^2, c_y^2) cannot embed the third secret digit s_{i+2} ; therefore, sign the pixel pair (p_i, p_{i+1}) to represent, which is not an embedded secret digit, as follows: Calculate the d value of the pixel pair (p_i, p_{i+1}) using Eq. (14), and replace the original pixel pair (p_i, p_{i+1}) with (q_i, q_{i+1}) using Eq. (15) as a stego-pixel pair.

$$d = (p_i + 1) \bmod 3. \quad (14)$$

$$q_i = \begin{cases} p_i & \text{if } d = 0 \\ p_i - 1 & \text{if } d = 1 \\ p_i + 1 & \text{if } d = 2 \end{cases} \quad (15)$$

Fig. 18 shows an example illustrating Eqs. (14) and (15), the area marked with a dark color represents not embedding secret digits. The d value of pixel p_i shows the distance from the area marked dark color. When $d = 1$, it shows that pixel p_i needs to be increased by two but decreasing by one has the same effect and reduces the distortion, for example, $(3, 4)$, $d_i = 1$ and $d_{i+1} = 2$. Using Eq. (15), replace $(3, 4)$ with $(2, 5)$ as the stego pixel pair.

- Step 6: Repeat Steps 4-5 until all pixel pairs are modified. Finally, the stego-image I is obtained.

C. EXAMPLE OF DATA EMBEDDING PROCEDURE

For a better understanding, three cases of data embedding are presented in this section, and we take three layers as an example. The corresponding three-layer reference matrix is shown in Figs. 10–13, and the first case of data embedding is shown in Fig. 19. Taking the first original pixel pair $(13, 13)$ and the to-be-embedded bit stream is $(111, 001, 100)_2$. The bit stream $(111, 001, 100)_2$ is first converted into three 8-ary digit streams $(714)_8$, and the center pixel pair $M_3(1, 1)$ of the 9×9 blocks is calculated using Eq. (8). To embed the third secret digit '4' in M_3 , find the closest satisfactory element by searching in a square of 3×3 centered on $M_3(1, 1)$, that is, $(1, 1)$, where $M_3(1, 1) = 4$. Second, the pixel pair $M_3(1,1) = M_2(4, 4)$ is calculated using Eq. (11). To embed the second secret digit '1', find a satisfactory element by searching in a square of 3×3 centered on

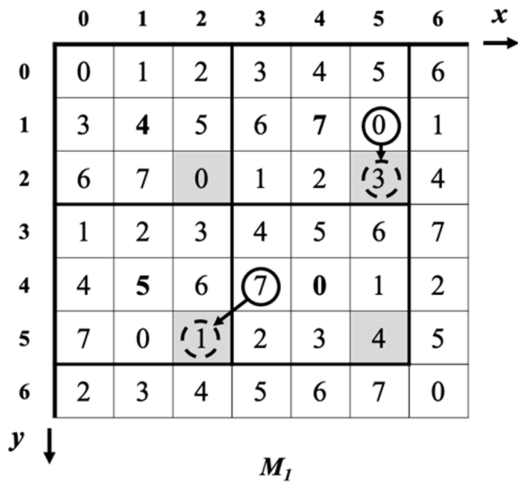


FIGURE 18. An example of embed false.

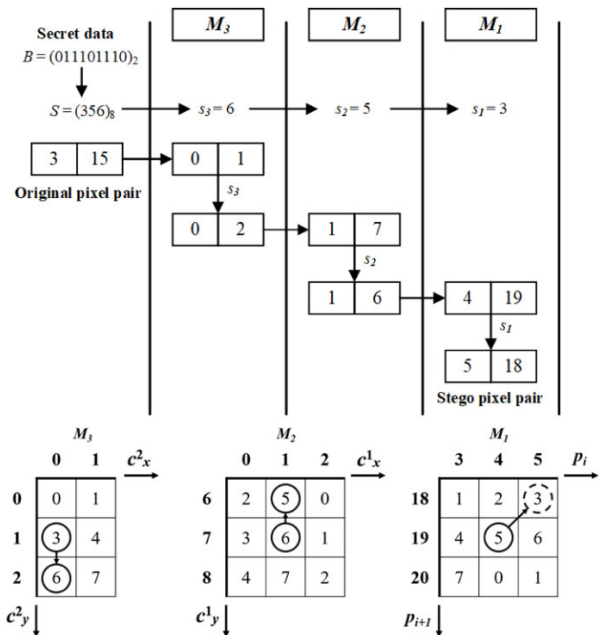


FIGURE 20. Second case to illustrate the embedding procedure.

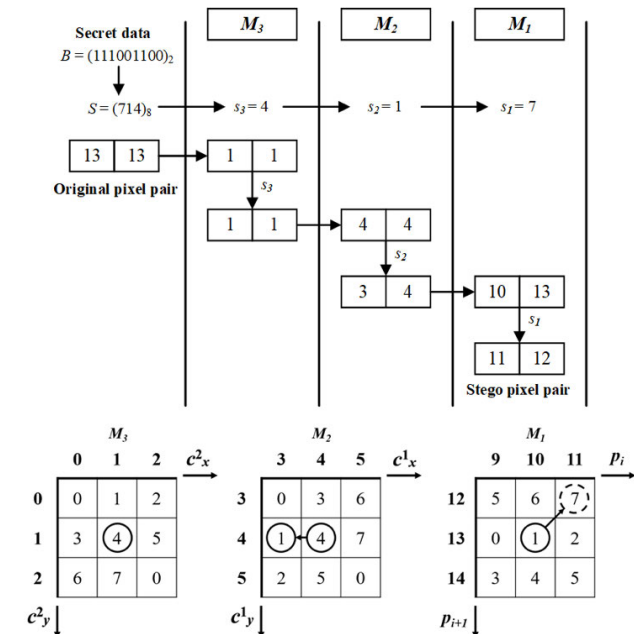


FIGURE 19. The first case illustrates the embedding procedure.

$M_2(4, 4)$, that is, $(3, 4)$, where $M_2(3, 4) = 1$. Finally, the pixel pair $M_2(3, 4) = M_1(10, 13)$ is calculated using Eq. (11). To embed the first secret digit ‘7’, find the satisfactory element by searching in a square of 3×3 centered on $M_1(10, 13)$, that is, $(11, 12)$ where $M_1(11, 12) = 7$. Then, replace the original pixel pair $(13, 13)$ with $(11, 12)$ as the stego pixel pair.

For the second case, Fig. 20 shows the illustration of embedding procedure. The second original pixel pair $(3, 15)$ and the to-be-embedded bit stream are $(011, 101, 110)_2$. The bit stream $(011, 101, 110)_2$ is first converted to three 8-ary digit streams $(356)_8$ and calculates the center pixel pair $M_3(0, 1)$ of the 9×9 blocks using Eq. (8). To embed the third secret digit ‘6’, find the closest satisfactory element by searching in a square of 3×3 centered on $M_3(0, 1)$, that is, $(0, 2)$ where $M_3(0, 2) = 6$. Second, calculate the pixel

pair $M_3(0, 2) = M_2(1, 7)$ using Eq. (11). To embed the second secret digit ‘5’, find a satisfactory element by searching in a square of 3×3 centered on $M_2(1, 7)$, that is, $(1, 6)$ where $M_2(1, 6) = 5$. Finally, we convert the pixel pair $M_2(1, 6) = M_1(4, 19)$ using Eq. (11). To embed the first secret digit ‘3’, find a satisfactory element by searching in a square of 3×3 centered on $M_1(4, 19)$, that is, $(5, 18)$ where $M_1(5, 18) = 3$. Then, replace the original pixel pair $(3, 15)$ with $(5, 18)$ as the stego pixel pair.

For the third case, the only special circumstance that needs to be mentioned is that the third secret digit cannot be embedded. Fig. 21 illustrates the embedding procedure. The third original pixel pair and the to-be-embedded bit stream are $(6, 2)$ and $(100, 000, 010)_2$, respectively. The bit stream $(100, 000, 010)_2$ is first converted to three 8-ary digit streams $(402)_8$ and calculates the center pixel pair $M_3(0, 0)$ of the 9×9 blocks using Eq. (8). To embed the third secret digit ‘2’, it was difficult to find a satisfactory element by searching in a 3×3 square centered on $M_3(0, 0)$. Therefore, sign the pixel pair $(6, 2)$ to represent, which is not an embedded secret digit, as follows: Calculate the d value of the original pixel pair $(6, 2)$ using Eq. (14), respectively, i.e., $d_1 = 1$ and $d_2 = 0$. Using Eq. (15), replace $(6, 2)$ with $(5, 2)$ as the stego pixel pair.

D. THE EXTRACTION PROCEDURE

In this section, the data extraction procedure of the proposed method is presented. Fig. 22 shows a flowchart of the data extraction procedure. According to the three-layer reference matrix, stego-image I with size $W \times H$ is an embedded secret digit. First, stego-image I is divided into non-overlapping stego pixel pairs. In the embedding procedure, most of the

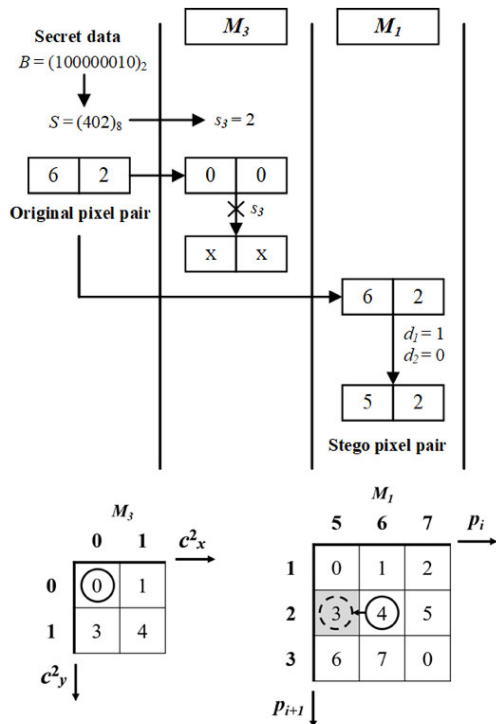


FIGURE 21. The third case illustrates the embedding procedure.

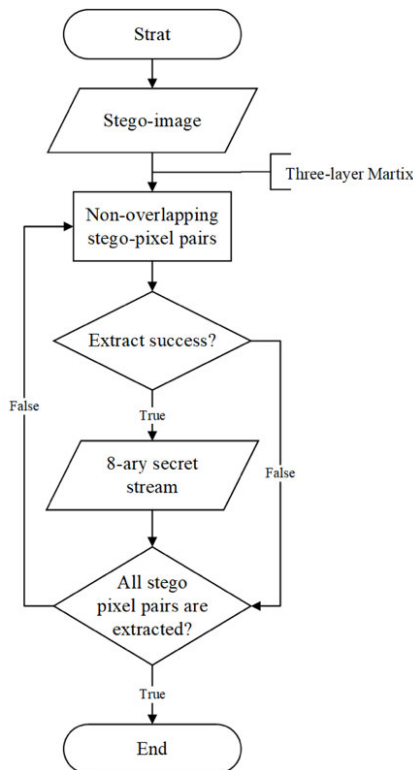


FIGURE 22. Flowchart of the data extraction procedure.

pixel pairs can extract nine binary secret digits at a time. The extraction procedure is described as follows:

Step 1: Generate the three-layer reference matrix (M_1 , M_2 , and M_3) using the shared construction information in the same way as described in Section 3.2.

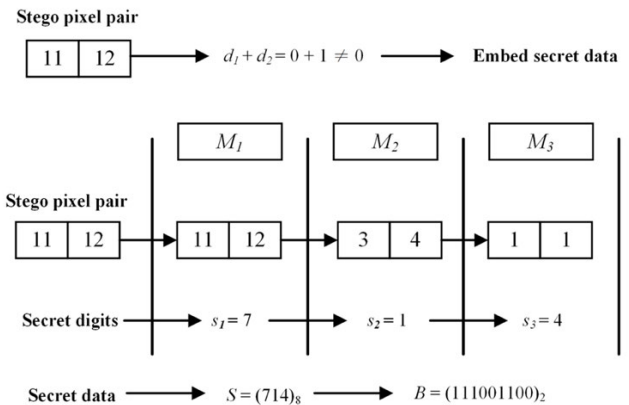


FIGURE 23. First case illustrating the extraction procedure.

Step 2: The stego -image I is divided into non-overlapping pixel pairs (p'_i, p'_{i+1}) , and consider (p'_i, p'_{i+1}) as the coordinates of reference matrix M_1 to specify the value $m_1(p'_i, p'_{i+1})$.

$$I = \{(p'_i, p'_{i+1}) | i = 1, 3, \dots, W \times H - 1\}. \quad (16)$$

Step 3: Distinguish whether each stego pixel pair (p'_i, p'_{i+1}) embeds the secret data. Calculate the d value of the stego pixel pair (p'_i, p'_{i+1}) using Eq. (14), respectively. The algorithm was categorized into two cases.

Case 1: $d_i + d_{i+1} \neq 0$, which means that the stego pixel pair (p'_i, p'_{i+1}) is embedded in three secret digits; therefore, Step 4 is used to extract three 8-ary secret digits.

Case 2: $d_i + d_{i+1} = 0$, which means that the stego pixel pair (p'_i, p'_{i+1}) is not an embedded secret digit; therefore, go to Step 3 to select the next stego pixel pair.

Step 4: Extract the first secret digit s_i by mapping the pixel pair (p'_i, p'_{i+1}) using the reference matrix M_1 , $s = m_1(p'_i, p'_{i+1})$. Using Eq. (8), calculate (c_x^1, c_y^{21}) and (c_x^2, c_y^2) , respectively. Extract the second and third secret digits, that is, $m_2(c_x^1, c_y^{21}) = s_{i+1}$ and $m_3(c_x^2, c_y^2) = s_{i+2}$.

Step 5: Convert the value of these three 8-ary secret digits into nine binary secret bits.

Step 6: Repeat Steps 3–5 until all stego pixel pairs are extracted. Combine them into the secret binary stream B .

E. EXAMPLE OF DATA EXTRACTION PROCEDURE

On the receiver side, two cases of data extraction are presented in this section, and we take three layers as an example. The corresponding three-layer reference matrix is shown in Figs. 11-13, and the first case illustrating the extraction procedure is shown in Fig. 23.

Consider the stego pixel pair (11, 12), for example. First, calculate the d value of the stego pixel pair (11, 12) using Eq. (14), respectively, i.e., $d_1 = 0$ and $d_2 = 1$. Then, $d_1 + d_2 = 0 + 1 \neq 0$ means that the stego pixel pair (11, 12) is embedded in three secret digits; therefore, the first secret

TABLE 4. Comparison between the proposed method and two-layer's schemes.

Images		Lena	Peppers	Baboon	Average
Xie et al. [39]	PSNR	47.13	47.13	47.09	47.11
	EC(bpp)	2.5	2.5	2.5	2.5
Wang et al. [40]	PSNR	43.7	43.7	43.7	43.7
	EC(bpp)	2.32	2.32	2.32	2.32
Shen et al. [41]	PSNR	36.87	36.87	36.86	36.87
	EC(bpp)	3.5	3.5	3.5	3.5
Lee et al. [42]	PSNR	36.92	36.93	36.91	36.92
	EC(bpp)	3.5	3.5	3.5	3.5
Proposed scheme	PSNR	30.85	30.88	30.86	30.86
	EC(bpp)	4.50	4.48	4.50	4.49

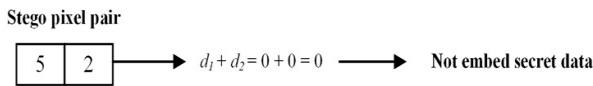


FIGURE 24. An example to illustrate the extraction procedure.

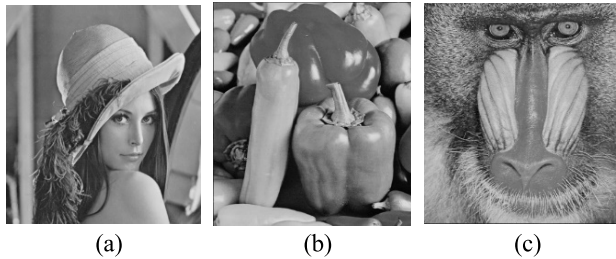


FIGURE 25. Three 512 × 512 grayscale images: (a) Lena, (b) Peppers, (c) Baboon.

digit s_i is extracted by mapping the pixel pair (p'_i, p'_{i+1}) by the reference matrix M_1 , $M_1(11, 12) = 7$. Using to Eq. (8), calculate the pairs (3, 4) and (1, 1), respectively. Extract the second and third secret digits, that is, $M_2(3, 4) = 1$ and $M_3(1, 1) = 4$. Finally, three 8-ary secret digits $(714)_8$ are extracted and converted into nine binary secret bits $(111, 001, 100)_2$.

In another case, take the stego pixel pair (5, 2) as an example. Fig. 24 shows the case illustrating the extraction procedure. The d value of the stego pixel pair (5, 2) was calculated using Eq. (14), that is, $d_1 = 0$ and $d_2 = 0$. Then, $d_1 + d_2 = 0 + 0 = 0$, which means that the stego pixel pair (5, 2) is not an embedded secret digit.

IV. EXPERIMENTAL RESULTS

We used MATLAB R2018a software to execute the algorithm on WINDOWS 10 64-bit system equipped with an i5-4210 2.6 GHz CPU and 4 GB RAM. In this section, we used three 512 × 512 test images from the USC-SIPI image database, that is, Lena, Peppers, and Baboon. As shown in Fig. 25, three 512 × 512 RGB color images were converted into grayscale images before testing. A pseudo-random number generator was used to generate the required secret dataset. Based on the embedding capacity, PSNR, and security under RS-steganalysis, performance comparisons were evaluated.

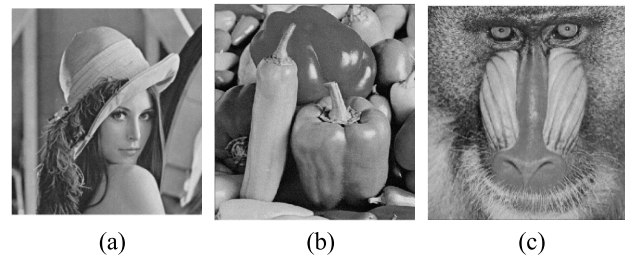


FIGURE 26. Three 512 × 512 grayscale stego-images: (a) Lena, (b) Peppers, (c) Baboon.

Furthermore, these experimental results were analyzed and shown in Sections 4.1, 4.2, and 4.3.

A. EMBEDDING CAPACITY AND VISUAL QUALITY ANALYSIS

This section analyzes the performance of the proposed visual quality and embedding capacity. The performance of the stego-image was measured using the peak signal-to-noise ratio (PSNR). The PSNR, mean square error (MSE), and embedding capacity (EC) are defined as follows:

$$PSNR = 10 \cdot \log_{10} \left(\frac{255^2}{MSE} \right). \tag{17}$$

$$MSE = \frac{1}{W \times H} \sum_{i=1}^W \sum_{j=1}^H |x_{ij} - x'_{ij}|^2. \tag{18}$$

$$EC = \frac{N_S}{W \times H} (bpp). \tag{19}$$

In Eqs. (17)-(19), W and H are the length and width of the cover image, respectively; x_{ij} and x'_{ij} are the pixel values of the cover image and stego-image, respectively; N_S represents the number of bits hidden in the secret data. Fig. 26 shows the stego-images obtained using the proposed scheme. Obviously, the proposed scheme will not reduce the visual quality of the image, even though a large amount of secret data is embedded (up to 4.5 bpp). Table 4 shows a comparison between the proposed method and the other two-layer schemes [39]–[42] in Section 2.

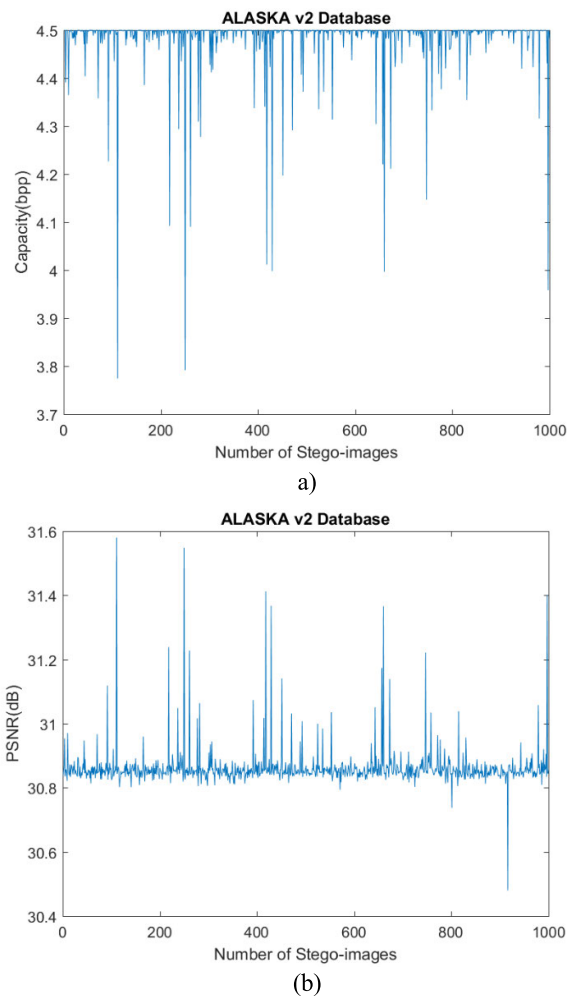


FIGURE 27. Performance of the proposed method on 1000 raw images of ALASKA v2 database (a) The embedding capacity (b) The stego-image quality.

Note that with the embedding capacity, the proposed scheme is significantly superior to the other two-layer schemes included in this experiment. More specifically, a higher embedding capacity (up to 4.5 bpp) and proper image quality decline are obtained, which shows a vivid difference from the methods in [39]–[42]. While other studies [39]–[42] obtained have higher average PSNR values of 47.11, 43.7, 36.87, and 36.92 dB, respectively, the embedding capacity of our proposed scheme is higher. The experimental results show that the proposed method in the three-layer reference matrix achieves a high payload (approximately 4.5 bpp) with good stego-image quality (PSNR value greater than 30 dB).

In order to evaluate the performance of the proposed method, we select 1000 RAW images from “ALASKA v2” database as the target images to repeat the experiment again. The embedding capacity and the stego-image quality is shown in Fig. 27.

The embedding capacity is falling into the range between 3.7 bpp and 4.5 bpp. The average embedding capacity is 4.49 bpp. The stego-image quality is falling into the range between 30.4 db and 31.6 db. The average stego-image

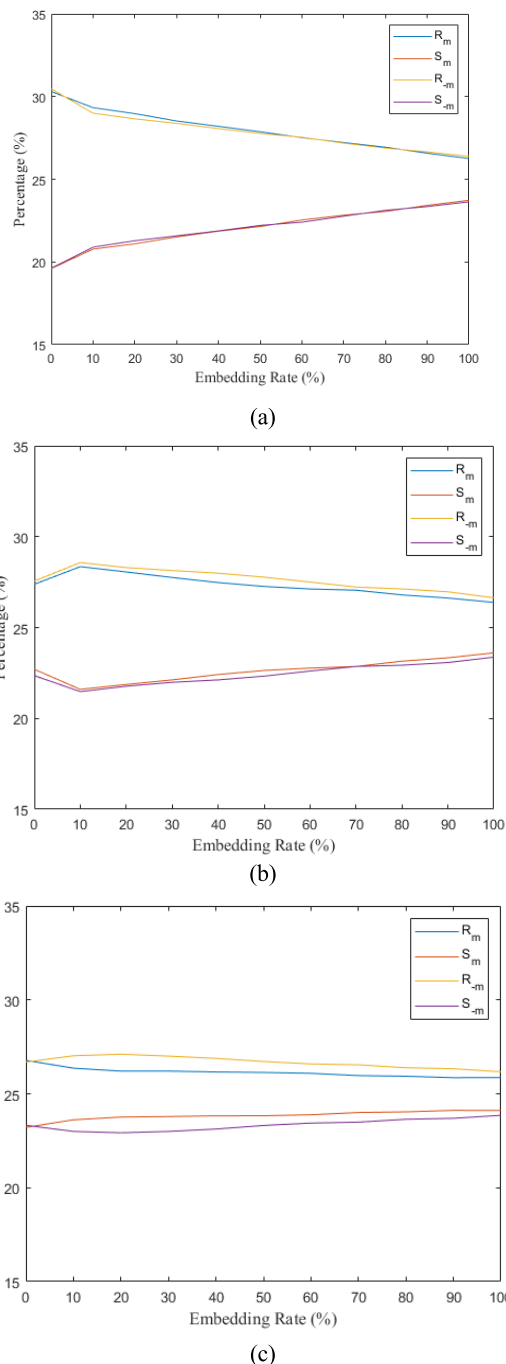


FIGURE 28. RS-steganalysis by the proposed method: (a) Lena, (b) Peppers, (c) Baboon.

quality is 30.86 dB. Clearly, the data of experimental results is very table. Because it only depends on the number of “embedding false” pixel group, which is located in the boundary area. Therefore, the proposed method can prevent the fall of boundary problem.

B. SECURITY AGAINST STATISTICAL RS-STEAGANALYSIS

The human eye cannot judge what is conducted by steganalysis on the stego-image. As the quality of stego-images is more than 30 dB, it is difficult for human eyes to determine whether

TABLE 5.

W	Weight of an image.
H	Height of an image.
Cover image C	Original grayscale image.
Stego-image I	Grayscale stego-image.
(p_i, p_{i+1})	Pixel pairs of consecutive cover pixels.
B	Secret stream in binary notational system.
b	Secret sub-stream in binary notational system.
S	Secret stream in 8-ary notational system.
s	Secret sub-stream in 8-ary notational system.
M_n	N -layer reference matrix.
M_1	First-layer reference matrix size of 256×256 .
M_2	Second-layer reference matrix size of 85×85 .
M_3	Third-layer reference matrix size of 28×28 .
$m_n(i, j)$	Value of reference matrix M_n .
(c_x^1, c_y^1)	Center pixel pair of 3×3 blocks.
(c_x^2, c_y^2)	Center pixel pair of 9×9 blocks.
(p_i^1, p_{i+1}^1)	Stego pixel pairs, which embedded three secret digits.
(c_x^1, c_y^1)	Stego pixel pairs, which embedded two secret digits.
(c_x^2, c_y^2)	Stego pixel pairs, which embedded one secret digit.
d	Value to determine whether the pixel pair is embedded secret digit or not.
(q_i, q_{i+1})	Pixel pair, which is not embedded secret digit.

secrets are hidden in stego-images. Thus, an effective method is needed, such as the security of the proposed method against statistical RS-steganalysis [13], which efficiently detects hidden data inside the stego-image. The RS-steganalysis was used as a detection method in our study, proving good security under our proposed scheme. In RS-steganalysis, n adjacent pixels (x_1, x_2, \dots, x_n) are selected as a pixel group. Then, the discrimination function, DF , defined by Eq. (20), is applied to quantify the smoothness or regularity of each pixel group. R_M , S_M , R_{-M} , and S_{-M} denote the proportion of blocks, where the magnitude of DF varies when applying each block. If the parameters satisfy $R_M \approx R_{-M} > S_M \approx S_{-M}$, there are no hidden data in the respective images. When an image has hidden data, R_{-M} and S_M increase, whereas R_M and S_{-M} decreases and are exposed by RS-analysis.

$$DF(x_1, x_2, \dots, x_n) = \sum_{i=1}^n |x_{i+1} - x_i|. \quad (20)$$

Fig. 27 shows the RS detection analysis of the Lena, Peppers, and Baboon images with the proposed method. Even with the addition of hidden data bits, stego-images have singular and regular parameters close to each other in the RS-analysis graphs. This shows that the proposed method has a high level of protection against statistical RS-analysis.

V. CONCLUSION

In this study, the proposed method is based on the concept of using a reference matrix in the embedding and extraction procedures. We proposed a multilayer reference matrix that can multi-layer embed binary secret bits, increase payload, enhance data security, and prevent the fall of boundary

problem. In addition, we demonstrate a three-layer EMD-based data hiding example. The payload and image quality depend on the number of layers in the reference matrix. In the three-multilayer reference matrix, most of the cover pixel pairs can embed nine binary secret bits in a time and lead to a high payload approximately 4.5 bpp. The quality of the stego-image satisfies the human vision system sensitivity (PSNR value greater than 30 dB). The characteristics of the media limit the number of embedding layers. For grayscale images, only three layers can be embedded; if applied to color images (RGB), the capacity can reach three times.

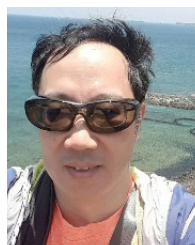
APPENDIX

See Table 5.

REFERENCES

- [1] C.-C. Chang, T. Kieu, and Y.-C. Chou, "Reversible data hiding scheme using two steganographic images," in *Proc. IEEE Region Conf.*, Oct. 2007, pp. 1–4.
- [2] C.-C. Chang, Y.-C. Chou, and T. D. Kieu, "Information hiding in dual images with reversibility," in *Proc. 3rd Int. Conf. Multimedia Ubiquitous Eng.*, Jun. 2009, pp. 145–152.
- [3] C.-F. Lee and Y.-L. Huang, "Reversible data hiding scheme based on dual stego-images using orientation combinations," *Telecommun. Syst.*, vol. 52, no. 4, pp. 2237–2247, Apr. 2013.
- [4] C.-F. Lee, K.-H. Wang, C.-C. Chang, and Y.-L. Huang, "A reversible data hiding scheme based on dual steganographic images," in *Proc. 3rd Int. Conf. Ubiquitous Inf. Manage. Commun. (ICUIMC)*, 2009, p. 228.
- [5] J. Biswapati, G. Debasis, and M. S. Kumar, "Dual-image based reversible data hiding scheme through pixel value differencing with exploiting modification direction," in *Proc. 1st Int. Conf. Intell. Comput. Commun.*, 2017, pp. 549–557.
- [6] J. Biswapati, G. Debasis, and M. S. Kumar, "Dual-image based reversible data hiding scheme using pixel value differencing expansion," *Int. J. Netw. Secur.*, vol. 18, no. 4, pp. 633–643, Jul. 2016.

- [7] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, "Techniques for data hiding," *IBM Syst. J.*, vol. 35, nos. 3–4, pp. 313–336, 1996, doi: [10.1147/sj.353.0313](https://doi.org/10.1147/sj.353.0313).
- [8] C.-K. Chan and L. M. Cheng, "Hiding data in images by simple LSB substitution," *Pattern Recognit.*, vol. 37, no. 3, pp. 469–474, Mar. 2004, doi: [10.1016/j.patcog.2003.08.007](https://doi.org/10.1016/j.patcog.2003.08.007).
- [9] C. Thien and J. Lin, "A simple and high-hiding capacity method for hiding digit-by-digit data in images based on modulus function," *Pattern Recognit.*, vol. 36, no. 12, pp. 2875–2881, 2003, doi: [10.1016/s0031-3203\(03\)00221-8](https://doi.org/10.1016/s0031-3203(03)00221-8).
- [10] X. Li, B. Yang, D. Cheng, and T. Zeng, "A generalization of LSB matching," *IEEE Signal Process. Lett.*, vol. 16, no. 2, pp. 69–72, Feb. 2009, doi: [10.1109/lsp.2008.2008947](https://doi.org/10.1109/lsp.2008.2008947).
- [11] J. Mielikainen, "LSB matching revisited," *IEEE Signal Process. Lett.*, vol. 13, no. 5, pp. 285–287, May 2006, doi: [10.1109/lsp.2006.870357](https://doi.org/10.1109/lsp.2006.870357).
- [12] A. Sahu and G. Swain, "High fidelity based reversible data hiding using modified LSB matching and pixel difference," *J. King Saud Univ.-Comput. Inf. Sci.*, Jul. 2019, doi: [10.1016/j.jksuci.2019.07.004](https://doi.org/10.1016/j.jksuci.2019.07.004).
- [13] J. Fridrich, M. Goljan, and R. Du, "Reliable detection of LSB steganography in color and grayscale images," in *Proc. Workshop Multimedia Secur. New Challenges*, 2001, pp. 27–30.
- [14] D.-C. Wu and W.-H. Tsai, "A steganographic method for images by pixel-value differencing," *Pattern Recognit. Lett.*, vol. 24, pp. 1613–1626, Jun. 2003.
- [15] C.-C. Chang and H.-W. Tseng, "A steganographic method for digital images using side match," *Pattern Recognit. Lett.*, vol. 25, no. 12, pp. 1431–1437, Sep. 2004.
- [16] H.-W. Tseng and H.-S. Leng, "A steganographic method based on pixel-value differencing and the perfect square number," *J. Appl. Math.*, vol. 2013, pp. 1–8, Jan. 2013.
- [17] G. Swain, "Adaptive pixel value differencing steganography using both vertical and horizontal edges," *Multimedia Tools Appl.*, vol. 75, no. 21, pp. 13541–13556, 2016.
- [18] M. Hussain, A. W. A. Wahab, A. T. S. Ho, N. Javed, and K. H. Jung, "A data hiding scheme using parity-bit pixel value differencing and improved rightmost digit replacement," *Signal Process., Image Commun.*, vol. 50, pp. 44–57, Feb. 2017.
- [19] A. Sahu and G. Swain, "Data hiding using adaptive LSB and PVD technique resisting PDH and RS analysis," *Int. J. Electron. Secur. Digit. Forensics*, vol. 11, no. 4, p. 458, 2019.
- [20] A. Sahu, G. Swain, M. Sahu, and J. Hemalatha, "Multi-directional block based PVD and modulus function image steganography to avoid FOBP and IEP," *Journal of Information Security and Applications*, 58, Apr. 2014, Art. no. 102808.
- [21] A. Sahu and G. Swain, "An optimal information hiding approach based on pixel value differencing and modulus function," *Wireless Pers. Commun.*, vol. 108, no. 1, pp. 159–174, 2019.
- [22] A. Sahu and M. Sahu, "Digital image steganography and steganalysis: A journey of the past three decades," *Open Comput. Sci.*, vol. 10, no. 1, pp. 296–342, 2020.
- [23] X. Zhang and S. Wang, "Efficient steganographic embedding by exploiting modification direction," *IEEE Commun. Lett.*, vol. 10, no. 11, pp. 781–783, Nov. 2006.
- [24] C.-C. Chang, W.-L. Tai, and K.-N. Chen, "Improvements of EMD embedding for large payloads," in *Proc. 3rd Int. Conf. Intell. Inf. Hiding Multimedia Signal Process. (IIH-MSP)*, Nov. 2007, pp. 473–476.
- [25] R.-M. Chao, H.-C. Wu, C.-C. Lee, and Y.-P. Chu, "A novel image data hiding scheme with diamond encoding," *EURASIP J. Inf. Secur.*, vol. 2009, pp. 1–9, Dec. 2009.
- [26] K.-N. Chen, C.-C. Chang, and H.-C. Lin, "A large payload EMD embedding scheme with high stego-image quality," in *Proc. Int. Conf. Comput. Aspects Social Netw.*, Sep. 2010, pp. 126–130.
- [27] C. Kim, "Data hiding by an improved exploiting modification direction," *Multimedia Tools Appl.*, vol. 69, no. 3, pp. 569–584, Apr. 2014.
- [28] T. D. Kieu and C.-C. Chang, "A steganographic scheme by fully exploiting modification directions," *Expert Syst. Appl.*, vol. 38, no. 8, pp. 10648–10657, Aug. 2011.
- [29] H.-S. Leng and H.-W. Tseng, "Generalize the EMD scheme on an N-dimensional hypercube with maximum payload," *Multimedia Tools Appl.*, vol. 78, no. 13, pp. 18363–18377, Jul. 2019.
- [30] H. S. Leng, J. F. Lee, and H. W. Tseng, "A high payload EMD-based steganographic method using two extraction functions," *Digit. Signal Process.*, vol. 113, Oct. 2021, Art. no. 103026.
- [31] C.-F. Lee, Y.-R. Wang, and C.-C. Chang, "A steganographic method with high embedding capacity by improving exploiting modification direction," in *Proc. 3rd Int. Conf. Intell. Inf. Hiding Multimedia Signal Process. (IIH-MSP)*, Nov. 2007, pp. 497–500.
- [32] C.-C. Chang, Y.-C. Chou, and T. D. Kieu, "An information hiding scheme using sudoku," in *Proc. 3rd Int. Conf. Innov. Comput. Inf. Control*, 2008, p. 17.
- [33] C. C. Chang, Y. Liu, and T. S. Nguyen, "A novel turtle shell based scheme for data hiding," in *Proc. 10th Int. Conf. Intell. Inf. Hiding Multimedia Signal Process.*, Aug. 2014, pp. 89–93.
- [34] Y. Liu, C.-C. Chang, and T.-S. Nguyen, "High capacity turtle shell-based data hiding," *IET Image Process.*, vol. 10, no. 2, pp. 130–137, Feb. 2016.
- [35] S. Kurup, A. Rodrigues, and A. Bhise, "Data hiding scheme based on octagon shaped shell," in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Aug. 2015, pp. 1982–1986.
- [36] H. S. Leng, "Data hiding scheme based on regular octagon-shaped shells," in *Proc. Int. Conf. Intell. Inf. Hiding Multimedia Signal Process.*, 2017, pp. 29–35.
- [37] H.-S. Leng and H.-W. Tseng, "Maximizing the payload of the octagon-shaped shell-based data hiding scheme," in *Proc. IEEE 8th Int. Conf. Awareness Sci. Technol. (iCAST)*, Nov. 2017, pp. 45–49.
- [38] H.-S. Leng, "Generalized scheme based on octagon-shaped shell for data hiding in steganographic applications," *Symmetry*, vol. 11, no. 6, p. 760, Jun. 2019.
- [39] 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.
- [40] F. Wang, Y. Guo, Z. Yin, X. Zhou, and Q. Zhao, "Data hiding method based on reference matrix," *Proc. Comput. Sci.*, vol. 131, pp. 800–809, Oct. 2018.
- [41] J. J. Shen, C. F. Lee, Y. H. Li, and S. Agrawal, "Image steganographic scheme based on doublelayer magic matrix," in *Proc. IEEE 10th Int. Conf. Awareness Sci. Technol. (iCAST)*, Feb. 2019, pp. 1–6.
- [42] C.-F. Lee, J.-J. Shen, S. Agrawal, and Y.-H. Li, "High-capacity embedding method based on double-layer octagon-shaped shell matrix," *Symmetry*, vol. 13, no. 4, p. 583, Apr. 2021.



HUI-SHIH LENG is currently an Associate Professor with the Department of Mathematics, National Changhua University of Education, Chang-Hua, Taiwan. His research interests include information hiding and web technologies.



CHENG-JUNG TSAI is currently a Professor with the Graduate Institute of Statistics and Information Science, National Changhua University of Education, Chang-Hua, Taiwan. His research interests include data mining, big data analysis, information security, e-learning, and digital image processing.



TIEN-JUNG WU received the M.S. degree from the Graduate Institute of Statistics and Information Science, National Changhua University of Education, Chang-Hua, Taiwan. His research interests include information hiding, big data analysis, and digital image processing.