Dual-image Reversible Data Hiding Based on Sudoku Block Mapping and Adaptive Embedding

Yurong Zhang[1], Ye Yao[1], Chia-Chen Lin[2], Chin-Chen Chang[3]*

[1] School of Cyberspace, Hangzhou Dianzi University, China
[2] Department of Computer Science and Information Engineering, National Chin-Yi University of Technology, Taiwan
[3] Department of Information Engineering and Computer Science, Feng Chia University, Taiwan
yurong@hdu.edu.cn, yaoye@hdu.edu.cn, ally.cclin@ncut.edu.tw, ccc@cs.ccu.edu.tw

Abstract

Dual-image reversible data hiding (RDH) schemes typically hide secret data in the cover image to generate two marked images that must be received by two trusted receivers over public transmission channels. To measure the performance of an RDH method, the two important indicators are embedding capacity and visual quality. However, existing dual-image RDH studies usually cannot keep a better trade-off between them. To address this issue and improve data security, we propose an efficient dual-image RDH scheme based on Sudoku block mapping in this paper. It is worth noting that our scheme can nicely balance the two indicators, presenting a high embedding ratio at 1.5 bits per pixel at most and better visual quality in lower payload circumstances compared with state-of-the-art dual-image RDH schemes. The average peak signal-to-noise ratio of the proposed scheme on tested images is as high as 68.70 dB for the given payload of 20,000 bits.

Keywords: Dual-image, Reversible data hiding, Sudoku block mapping, Adaptive embedding, Embedding capacity

1 Introduction

Nowadays, massive data transmission is triggering information leakage and tampering which raises a variety of network security issues and concerns. Traditional cryptography provides secret protection by encrypting and decrypting the information. But it has a drawback that the disordered encrypted text is easy for hackers to detect. In the past two decades, data hiding technology which conceals secret data into a cover image in an imperceptible way for transmission, has attracted many researchers. But it will produce some distortion that makes the cover image irreversible. In many specific areas, such as military intelligence, medical diagnosis, copyright protection, and evidence preservation, the integrity of both the secret data and the original carrier must be protected. Therefore, extracting the information and recovering the cover image are both necessary in these areas. Reversible data hiding (RDH) can fully meet the requirements.

Current image RDH schemes can be mainly classified into two categories: difference expansion (DE) and histogram shifting (HS). In 2003, Tian [1] first released the DE-based RDH scheme, which utilized the difference of two consecutive pixel values to make room for embedding through the integer wavelet transform. In 2007, Ni et al. [2] proposed an HS-based RDH scheme which modified the peak bin of the cover image histogram to embed data. These two schemes laid the foundation of RDH researches. Since then, a lot of improved schemes based on them have been proposed [3-11].

At present, there are more topics in RDH, including RDH in encrypted image [12-15], RDH in compressed image [16-21], and RDH in dual-image [22-30]. Dual-image RDH schemes tend to improve the data security on the concept of secret sharing proposed by Shamir [31] and Blakley [32] in 1979. In 1995, Naor and Shamir [33] applied secret sharing to image fields. Specifically, the sender first duplicates the cover image into t receivers individually. Only when $r(r \leq t)$ or more receivers collaborate together can they extract the data and restore the cover image completely. If the number of receivers is less than r, the data can't be extracted correctly and the cover image can't be restored. Dual-image RDH is considered as the special case of $r = 2$, $t = 2$. Figure 1 shows the schematic of dual-image RDH. On the one hand, dual-image RDH researches provide higher data security; on the other hand, they can also obtain better visual quality after hiding while maintaining higher embedding capability, compared to single-image RDH researches.

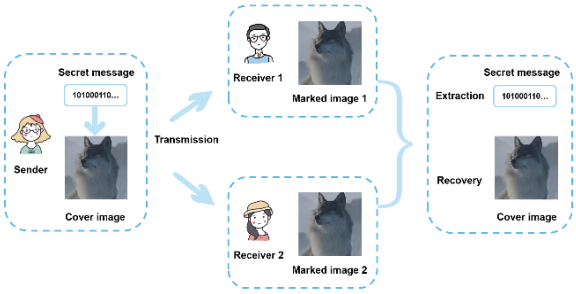


Figure 1. Schematic of dual-image RDH

Dual-image RDH schemes can be roughly divided into three types: matrix-based scheme, center folding strategy (CFS), and prediction-error scheme.

*Corresponding Author: Chia-Chen Lin; E-mail: ally.cclin@ncut.edu.tw
DOI: https://doi.org/10.70003/160792642025032602001

The first type of schemes is based on the construction of matrix. The concept of dual-image RDH was first proposed by Chang et al. [22] using an exploiting modification direction (EMD) matrix designed by Zhang and Wang [34]. The method [22] achieved the embedding ratio of 1 bit per pixel (bpp) with an average peak signal-to-noise ratio (PSNR) of 45 dB. They converted a binary secret message into base-5 digits. Two base-5 digits were embedded into two adjacent pixels of the cover image each time. The two adjacent pixels, called a "pixel pair", were located in a 5 × 5 block of the EMD matrix. In this block, the pixel pair was modified along the main diagonal direction to generate the first marked pixel pair according to the first digit and along the second diagonal direction to generate the second marked pixel pair. This was the first matrix-based method of dual-image RDH. In 2008, Chang et al. [35] came up with a single-image data hiding method based on a Sudoku matrix in which a novenary digit was embedded into a pixel pair of the cover image. To solve the irreversibility of the method [35], Huynh et al. [23] introduced an RDH scheme using the Sudoku matrix to create two marked images in 2015. Promoted by previous matrix-based schemes, Liu et al. [24] used a turtle shell matrix for dual-image RDH in 2018. In their method, secret data was concealed by shifting the location of pixel pairs in the turtle shell matrix to avoid positional collisions. The embedding ratio was 1 bpp with higher PSNR than the method [22]. In 2019, Lin et al. [25] embedded two quinary secret digits into each pixel pair of the cover image using the EMD matrix by shifting the pixel pair to appropriate positions. In 2021, Chen et al. [26] hid one base-5 secret digit into each pixel using the EMD matrix, where the secret digit was located in the horizontal or vertical direction of each pixel, achieving a higher embedding ratio of 1.56 boo with an average PSNR of 42 dB. In 2022, Chang et al. [27] proposed a novel turtle shell-based RDH hiding scheme based on the symmetric property under the guidance of position-aware. Their maximum embedding capacity was 1.25 bpp.

The second type of dual-image RDH schemes is mainly based on CFS which usually pursued high embedding capacity. Conventionally, λ bits secret data converted to a decimal digit ranging from $[0, 2^{\lambda} - 1]$ would cause a large image distortion. In 2015, Lu et al. [28] first proposed the CFS that mapped the decimal digits ranging from $[-2^{\lambda-1}, 2^{\lambda-1} - 1]$. The embedding ratio was flexible in terms of λ. The average PSNR was 46.36 dB at 1.5 bpp and 68.28 dB at 5000 bits. So the PSNR performance shows a low rate of change. In 2019, an improved CFS proposed by Shastri and Thanikaiselvan [36] realized 1.56 bpp and better visual quality than the strategy [28], however, it produced some additional information after embedding that must be transmitted to receivers in order to extract data accurately.

In addition, the third type of dual-image RDH schemes uses prediction-error of pixels. These schemes generally pursued high-fidelity images. In 2016, Jafar et al. [29] used one image as a predictor of another image. The method could embed around 1.23 bpp with image quality above 48 dB. In 2020, Yao et al. [30] proposed a dual-image RDH algorithm using Sachnev's prediction technique [4] to meet the requirements of high-fidelity application scenarios. Yao et al. [30] designed three shift modes in terms of the prediction-error values and achieved an average embedding ratio of 0.78 bpp. In 2022, Niu et al. [37] employed a new shift strategy and a pixel-value-ordering prediction to make better use of the inter-pixel correlation and image redundancy. The maximum modification of a pixel is 2, and the embedding ratio was about 0.23 bpp with the PSNR value of 53.80 dB. For most dual-image RDH schemes, the main evaluation metrics are embedding capacity and visual quality. Theoretically, if one is high, the other one is relatively low. However, based on existing schemes, it's not easy to balance them well. Considering this issue, we proposed a novel dual-image RDH scheme that takes fully account of embedding capacity and visual quality at the same time.

The main contributions of this paper are listed as follows:

a) A Sudoku block mapping (SBM) designed to restore the cover image is proposed.

b) To minimize the distortion, we construct a quantified distortion model to get an optimal distance threshold adaptively. Then, an efficient embedding strategy with minimum distortion is used to embed the secret data according to the optimal distance threshold. Our scheme is not only appropriate for high data capacity but also for low payload.

c) Not all pixel pairs of the cover image are embeddable. The optimal distance threshold determines whether the current pixel pair is capable of hiding secret data or not, bringing higher security performance than existing matrix-based schemes. If a hacker intercepted one marked image based on previous matrix-based schemes, he/she could extract half of the data. For our scheme, it is hard to know which pixel pair has hidden the secret data. It means that extracting the correct half data becomes impossible in this case.

The rest of this paper is organized as follows. In Section 2, the Sudoku matrix constructed in the method [35] and traditional Sudoku-based dual-image RDH scheme [23] are briefly introduced. The embedding, extraction and recovery procedures of our scheme are fully presented in Section 3. Experimental results and comparisons are demonstrated in Section 4. Finally, conclusions are given in Section 5.

2 Background

In this section, we first introduce the Sudoku matrix. Then, we briefly present the traditional Sudoku-based dual-image RDH scheme.

2.1 Construction of Sudoku Matrix

Sudoku is a popular number placement puzzle presented on a 9 × 9 square grid. An example of a 256 × 256 Sudoku

matrix S shown in Figure 2 is regarded as a reference matrix for our proposed dual-image RDH scheme. The horizontal axis and vertical axis represent the pixel values of a grayscale image ranging from 0 to 255. Each Sudoku grid in the Sudoku matrix which is depicted inside the brown box comprises nine 3 × 3 blocks. Each block which is depicted inside the blue box contains different digits from 0 to 8. What makes a grid unique is that each of the digits appears exactly once in every row, column, and 3 × 3 block. Due to the above fascinating properties and the number of total possible solutions of the grid, which is up to 5,472,730,538 according to the research of E. Russell et al. [38], it is hard to attack. Hence, the Sudoku matrix has higher security than other matrices.

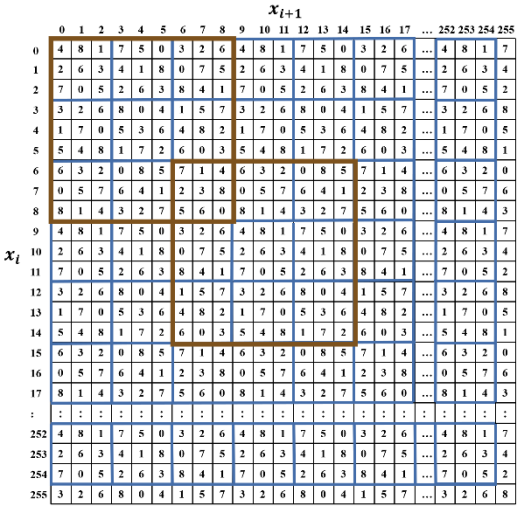


Figure 2. A 256 × 256 Sudoku matrix (The grid: the brown box with size 9 × 9; the block: the blue box with size 3 × 3)

2.2 Traditional Sudoku-based Dual-image RDH Scheme

A searching-directional algorithm using the Sudoku matrix S has been performed in Huynh et al.'s method [23]. The cover image with the size of $H \times W$ is denoted as $C = \{x_i \mid i = 0, 1, \ldots, (H \times W) - 1\}$ and every two consecutive pixels of the cover image C in a raster-scan order make up a pixel pair (x_i, x_{i+1}). In the data embedding process, each pixel of a secret image I, treated as an 8-bit secret message, is hidden into each pixel pair of C. As shown in Figure 3, x_i is mapped to the vertical axis and x_{i+1} is mapped to the horizontal axis of S. For example, if (x_i, x_{i+1}) equals (9,9), $S(9,9)$ equals the corresponding digit 4. There are four directions for each pixel pair: north, south, west and east.

To conduct data hiding, each pixel of I is converted into three base-9 numbers denoted as n_1, n_2 and n_3 that make up a number group. The value of the first number n_1 is ranging from 0 to 3. This is because the largest pixel value of a grayscale image is 255. After conversion, if the number group consists of two numbers, n_1 will be set as 0. Next, all pixel pairs of C are mapped to S in turn to embed the number group. Based on the value of n_1 , there are four situations to be considered and the searching directions SD_1 and SD_2 can be represented as:

$$(SD_1, SD_2) = \begin{cases} (LW, LN), n_1 = 0 \\ (LE, LN), n_1 = 1 \\ (LE, LS), n_1 = 2 \\ (LW, LS), n_1 = 3 \end{cases}. \quad (1)$$

where LW, LN, LE and LS denote the four directions (west, north, east and south). After the searching directions are determined, the sender searches n_2 along with SD_1 to create the first marked pixel pair (y_i, y_{i+1}) that n_2 equals $S(y_i, y_{i+1})$ and the third number n_3 along with SD_2 to create the second marked pixel pair (z_i, z_{i+1}) that n_3 equals $S(z_i, z_{i+1})$. Finally, two marked images $M^{(1)}$ and $M^{(2)}$ are generated.

To extract the secret data and recover C , the receivers have to read (y_i, y_{i+1}) of $M^{(1)}$ and (z_i, z_{i+1}) of $M^{(2)}$ orderly and map them to the same S. The second secret number n_2 and the third number n_3 are derived from the corresponding value of $S(y_i, y_{i+1})$ and $S(z_i, z_{i+1})$, respectively. Moreover, the intersection between SD_1 and SD_2 directions is the position of the original pixel pair (x_i, x_{i+1}). According to the relative location of n_2 and n_3, the first number n_1 can be recovered as:

$$n_1 = \begin{cases} 0, (y_i, y_{i+1}) \text{ in } LW \text{ and } (z_i, z_{i+1}) \text{ in } LN, \\ 1, (y_i, y_{i+1}) \text{ in } LE \text{ and } (z_i, z_{i+1}) \text{ in } LN \\ 2, (y_i, y_{i+1}) \text{ in } LE \text{ and } (z_i, z_{i+1}) \text{ in } LS \\ 3, (y_i, y_{i+1}) \text{ in } LW \text{ and } (z_i, z_{i+1}) \text{ in } LS \end{cases} \quad (2)$$

Finally, n_1, n_2 and n_3 are converted into decimal values. The original cover image C and the secret image I can be recovered.

It was the first dual-image RDH scheme that realized the reversibility of the cover image by using the Sudoku matrix. The embedding ratio is approximately up to 2 bpp but the visual quality is relatively not good. Their experimental results show that the PSNRs of two marked images are about 39.40 dB and 39.17 dB at the maximum embedding capacity, respectively.

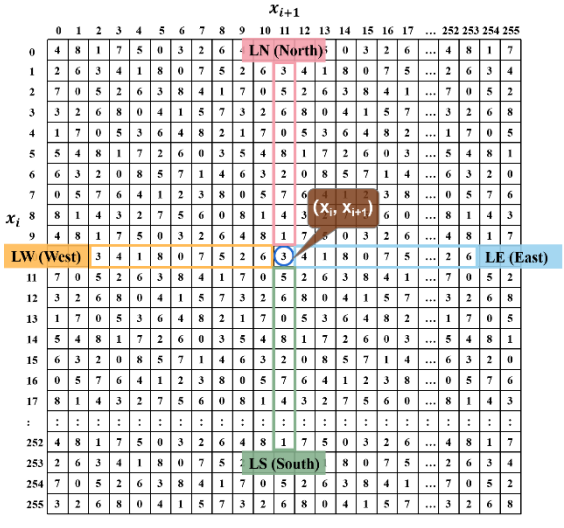


Figure 3. Pixel pair of cover image and its four directions

3 Problem Scheme

This section focuses on details of our proposed scheme. The framework of the proposed scheme consists of two phases. The embedding phase illustrated in Figure 4 contains four procedures: preprocessing of secret message, construction of SBM, optimization of distance threshold and data embedding. After embedding, two marked images are generated and delivered to the two receivers. The second phase including data extraction and image recovery will be implemented easily and simultaneously using the SBM.

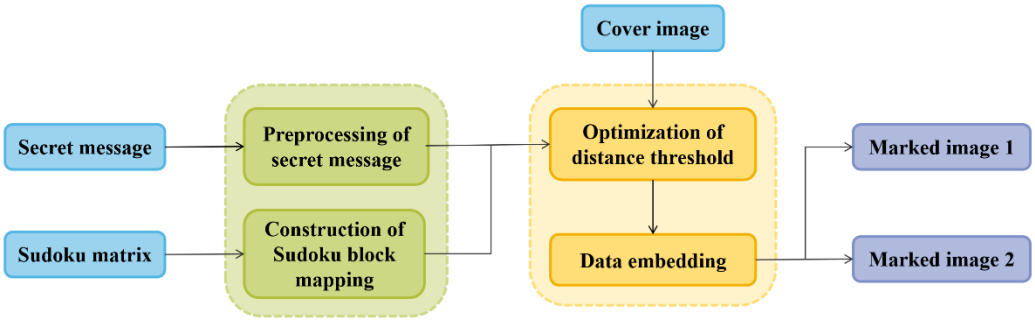


Figure 4. Embedding phase of the proposed scheme

3.1 Preprocessing of Secret Message

The secret message B with length P is a binary bit-stream $\{b_1b_2b_3 \ldots b_j \ldots b_P\}$, where b_j is a binary bit, $b_j \in \{0,1\}$. Divide B into segments denoted as s_k, where $k \in \{1, \ldots, [P/3]\}$. We convert every two segments s_k and s_{k+1} into two base-8 digits which make up a digit group denoted as g_p, where p is the index of the digit group. In this scheme, every two base-8 digits of the same digit group will be embedded into every pixel pair.

3.2 Construction of Sudoku Block Mapping

Now, we introduce the construction of SBM. As is shown in Figure 5, a Sudoku grid contains nine 3×3 blocks. The 3×3 center block is marked as the orange box, surrounded by eight 3×3 adjacent blocks. In the center block, there are nine different positions corresponding to nine Cases, and we label them as p_l, where l is the label index. The Sudoku block mapping is denoted as $\{p_l, block_l\}$. As is shown in Figure 6, we map the pixel pair (x_i, x_{i+1}) to S. There is only a center block corresponding to (x_i, x_{i+1}). Totally, nine Cases will be considered: if the position of (x_i, x_{i+1}) in its center block corresponds to label p_l, Case 1 will be matched. This means the white center block will be used to embed the first digit of the current digit group g, and the second digit will be found in the left top corner block plotted as the yellow area; if the position of (x_i, x_{i+1}) in its center block is blue, Case 2 will be chosen. The white center block will be used to embed the first digit and the blue adjacent block will be used to embed the second digit. Similarly, the rest cases are $\{p_3, block_3\}$, $\{p_4, block_4\}$, $\{p_5, block_5\}$,$\{p_6, block_6\}$, $\{p_7, block_7\}$, $\{p_8, block_8\}$ and $\{p_9, block_9\}$. It is noted that the first base-8 digit is always embedded into the center block to generate (y_i, y_{i+1}) of $M^{(1)}$ and the second digit is embedded into any blocks in the grid to generate (z_i, z_{i+1}) of $M^{(2)}$. If we match Case 5, the center block will be utilized to embed two digits.

The proposed SBM can confirm the relative position of the two embedded digits so that the reversibility for each pixel pair after data embedding is guaranteed. The details of efficient data embedding strategy with the optimal distance threshold are presented as Section 3.4.

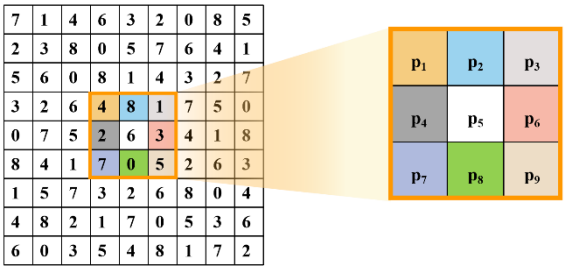


Figure 5. A grid of Sudoku and labels of center block

3.3 Optimization of Distance Threshold

At the beginning, we figure out the motivation of optimizing the distance threshold. We simulate the embedding procedure based on the SBM: two marked images $M^{(1)}$ and $M^{(2)}$ which are the same as the cover image C are generated first. Scan the pixel pair (x_i, x_{i+1}) in a raster-scan order. According to g_p to be embedded, we calculate the distance d_q and the embedding distortion ED_q of one pixel pair as the following equations:

$$d_q = \sqrt{(x_i - \hat{y}_i)^2 + (x_{i+1} - \hat{y}_{i+1})^2} + \sqrt{(x_i - \hat{z}_i)^2 + (x_{i+1} - \hat{z}_{i+1})^2}, \tag{3}$$

$$ED_q = |x_i - \hat{y}_i| + |x_{i+1} - \hat{y}_{i+1}| + |x_i - \hat{z}_i| + |x_{i+1} - \hat{z}_{i+1}| \tag{4}$$

where $(\hat{y}_i, \hat{y}_{i+1})$ and $(\hat{z}_i, \hat{z}_{i+1})$ are the coordinates of the

first digit and the second digit, respectively. Besides, q stands for the index of the pixel pair.

Take an 8 × 8 sized cover image shown in Figure 7 as an example. We embed these four digit groups: g_1 = {3,5}, g_2 = {1,4}, g_3 = {7,0}, g_4 = {2,6}. Figure 7(a) shows the traditional way to embed digits using the SBM. We scan the first four pixel pairs to embed the above digits: (9,9), (10,10), (11,13) and (10,11). The digit group g_1 needs to be embedded into the pixel pair (9,9) based on S shown in Figure 2. Due to the label of (9,9) in its center block is p_1, the block mapping is $\{p_1, block_1\}$. The pixel pair (9,9) is modified to (10,11) that equals the coordinates of the first digit of g_1 and (8,6) that equals the coordinates of the second digit of g_1. The distance d_1 calculated as Eq. 3 equals $\sqrt{5}+\sqrt{10}$. Similarly, the distances of other pixel pairs are $2\sqrt{2}$, $1+\sqrt{5}$ and $2+\sqrt{5}$ respectively. The embedding distortion is ED_1 is |9−8| + |9−6| + |10−9| + |11−9| = 7. And the total distortion of this cover image is 7 + 4 + 4 + 5 = 20. It is observed that the embedded cover pixel pairs (x_i, x_{i+1}) are concentrated on the front part of the cover image. On the contrary, if we select (10,10), (10,11), (7,11) and (9,7) orderly to embed the above digits shown in Figure 7(b), it indicates that the embeddable pixel pairs (x_i, x_{i+1}) are distributed in the cover image. The values of the distance are $1+\sqrt{2}$, 2, $\sqrt{2}$ and 1, respectively. The total embedding distortion is 3 + 2 + 2 + 1 = 8. Therefore, when the shorter distance d_q has a great impact on the visual quality of the image. The shorter distance d_q can achieve better visual quality. So, we need to set an optimal distance threshold to control which pixel pair is able to embed data.

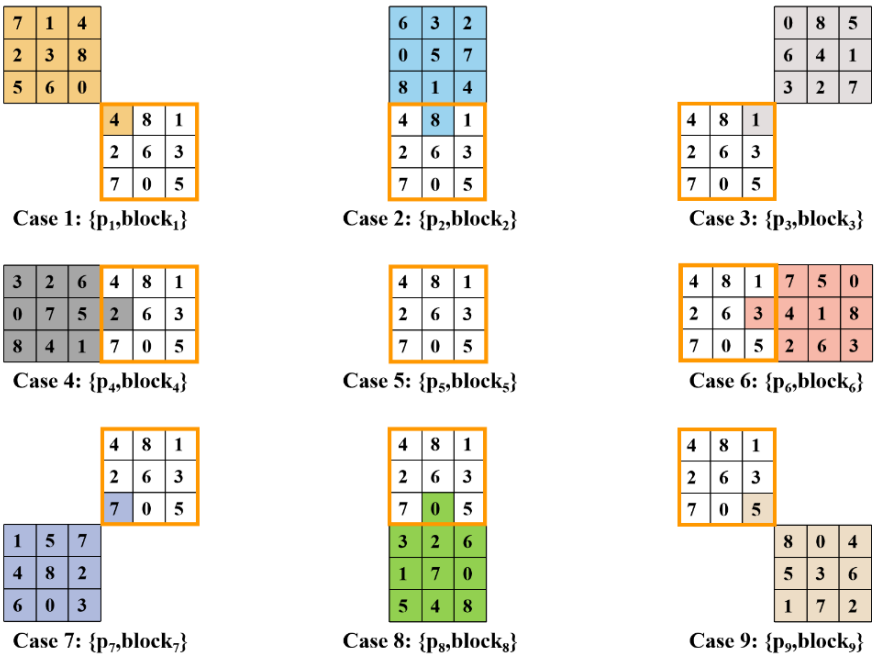


Figure 6. Sudoku block mapping (SBM)

9	9	10	10	11	13	10	11
8	9	10	11	14	12	13	14
10	12	9	8	7	11	8	8
11	13	12	12	10	12	9	7
10	9	13	13	15	14	20	19
8	8	6	8	10	11	10	10
6	6	5	6	10	9	12	12
34	10	32	22	25	26	23	22

(a) Orderly embedding

9	9	10	10	11	13	10	11
8	9	10	11	14	12	13	14
10	12	9	8	7	11	8	8
11	13	12	12	10	12	9	7
10	9	13	13	15	14	20	19
8	8	6	8	10	11	10	10
6	6	5	6	10	9	12	12
34	10	32	22	25	26	23	22

(b) Selective embedding

Figure 7. Comparison of two embedding strategy

Therefore, optimizing the distance threshold function $D(\varphi)$ is necessary, where the step φ is a parameter set. An initial value of $D(\varphi)$ is set to 1, because the minimum value of d_q is 1. According to Eq. 3, the possible values of d_q are $1, \sqrt{2}, 2, 1+\sqrt{2}, 2\sqrt{2}, \ldots, 5\sqrt{2}$, we define the parameter set $\varphi = \{0,1,2,3,4,5,6,7,8,9,10,11,12,13\}$. The two variables satisfy the function $D(\varphi) = 1/2\ \varphi+1$.

We simulate the embedding process using the SBM to find $D(\varphi)$: each pixel pair mapped to S can uniquely confirm its center block with a label. According to the label a specific case in Section 3.2 is determined. Use Eq. 3 to calculate the distance d_q. If d_q is less than or equal to $D(\varphi)$, (y_i, y_{i+1}) will equal the coordinates of the first digit and (z_i, z_{i+1}) will equal to the coordinates of the second digit, otherwise, (y_i, y_{i+1}) and (z_i, z_{i+1}) will not be modified. For each (x_i, x_{i+1}), (y_i, y_{i+1}) and (z_i, z_{i+1}) are denoted as:

$$(y_i, y_{i+1}) = \begin{cases} (\hat{y}_i, \hat{y}_{i+1}), d_q \le D(\varphi) \\ (x_i, x_{i+1}), d_q \le D(\varphi) \end{cases} \quad (5)$$

$$(z_i, z_{i+1}) = \begin{cases} (\hat{z}_i, \hat{z}_{i+1}), d_q \le D(\varphi) \\ (x_i, x_{i+1}), d_q \le D(\varphi) \end{cases} \quad (6)$$

Since the variable $D(\varphi)$ increases, the range of the embedding capacity will increase correspondingly. With the given secret message length P, the embedding capacity EC with the corresponding $D(\varphi)$ is probably not greater than P. So the optimization objective is to minimize $D(\varphi)$ with respect to the required payload. The quantified distortion model can be formulated as:

$$\begin{cases} minimize\ D(\varphi) \\ subject\ to\ EC \ge P \end{cases} \quad (7)$$

Ultimately, the minimum $D(\varphi)$ obtained by Eq.7 is the optimal distance threshold denoted as D_{opt}.

3.4 Data Embedding

The adaptive embedding procedure will be implemented after we get the D_{opt}. $(\hat{y}_i, \hat{y}_{i+1})$ will be found in the center block where $S(\hat{y}_i, \hat{y}_{i+1})$ equals the first digit; $(\hat{z}_i, \hat{z}_{i+1})$ will be found complied with the SBM where $S(\hat{z}_i, \hat{z}_{i+1})$ equals the second digit. For each pixel pair, we calculate the distance d_q, according to Eq. 3 and modify the values of two marked pixel pairs as following Equations:

$$(y_i, y_{i+1}) = \begin{cases} (\hat{y}_i, \hat{y}_{i+1}), d_q \le D_{opt} \\ (x_i, x_{i+1}), d_q > D_{opt} \end{cases} \quad (8)$$

$$(z_i, z_{i+1}) = \begin{cases} (\hat{z}_i, \hat{z}_{i+1}), d_q \le D_{opt} \\ (x_i, x_{i+1}), d_q > D_{opt} \end{cases} \quad (9)$$

In the end, $M^{(1)}$ and $M^{(2)}$ are generated. There are two special situations probably happened during the data embedding procedure.

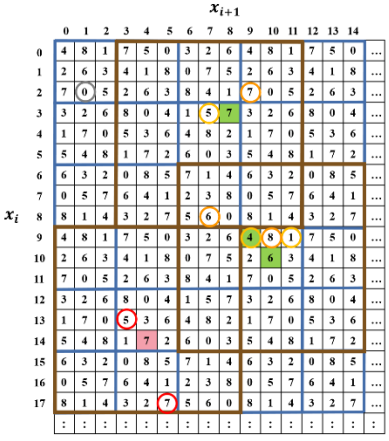
Situation 1: we regulate the pixel value ranging from 0 to 2 and 252 to 255 is the border value. If the pixel belongs to the border, we choose not to hide information and the two pixel pairs $M^{(1)}$ and $M^{(2)}$ will not be modified.

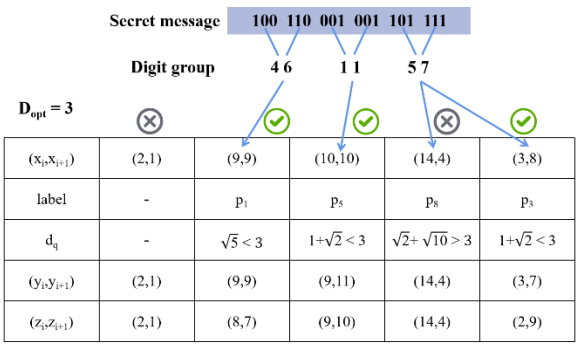
Situation 2: if (x_i, x_{i+1}) locates in label p_5 in its center block while the first digit and the second digit are equal, we find digit 8 and return its coordinate to (z_i, z_{i+1}). If we let (y_i, y_{i+1}) equals (z_i, z_{i+1}), a collision will be happened when extracting data. Because the receivers cannot confirm if this pixel pair conceals data or not.

In order to clearly illustrate the data embedding process, Figure 8 depicts a basic example using the SBM.

Hypothesis: B = {1001100010011011111}; $D_{opt} = 3$; (x_i, x_{i+1}) = {(2,1), (9,9), (10,10), (14,4), (3,8)};

Data embedding: we gain base-8 digits: {4,6,1,1,5,7} from B. (2,1) is a border pixel pair, so no secret information can be hidden. If $d_q \le 3$, two digits will be embedded into a pixel pair. Note that (10,10) matches label p_5 and the two digits are equal to 1, it is a special situation to deal with. We find digit 8 in the center block and temporarily return its coordinates (9,10) to (z_i, z_{i+1}) to calculate d_2.





(x_i, x_{i+1})	(2,1)	(9,9)	(10,10)	(14,4)	(3,8)
label	-	p_1	p_5	p_8	p_3
d_q	-	$\sqrt{5} < 3$	$1+\sqrt{2} < 3$	$\sqrt{2}+\sqrt{10} > 3$	$1+\sqrt{2} < 3$
(y_i, y_{i+1})	(2,1)	(9,9)	(9,11)	(14,4)	(3,7)
(z_i, z_{i+1})	(2,1)	(8,7)	(9,10)	(14,4)	(2,9)

Figure 8. Comparison of two embedding strategy

3.5 Data Extraction and Image Recovery

The two receivers trust each other to extract B and recover C using the SBM. They don't need to know the values of P and D_{opt}. The relationship between the two marked pixel pairs exist only two situations:

Situation 1: if (y_i, y_{i+1}) and (z_i, z_{i+1}) are the same, it indicates that there's no secret data concealed. The pixel pair (x_i, x_{i+1}) of the original image C equals (y_i, y_{i+1}) or (z_i, z_{i+1}).

Situation 2: if (y_i, y_{i+1}) and (z_i, z_{i+1}) are different, two embedded base-8 digits can be obtained by $S(y_i, y_{i+1})$ and $S(z_i, z_{i+1})$. The original pixel pair (x_i, x_{i+1}) can be retrieved by the relative location of (y_i, y_{i+1}) and (z_i, z_{i+1}) where they can determine the label of the center block.

The receivers only need to confirm if the two marked pixel pairs are equal or not. Eventually, transform every two base-8 digits into every two segments s_k and concatenate all segments together to recover the bitstream B. Note that there is a special case: if the second digit equals 8, let the value equal the value of the first digit.

Data extraction and recovery examples are as follows: According to Figure 8, we gain two marked images: (y_i, y_{i+1}) = {(2,1), (9,9), (9,11), (14,4), (3,8)}; (z_i, z_{i+1}) = {(2,1), (8,7), (9,10), (14,4), (2,9)}. Because the first pixel pairs of the two marked images are the same, there's no secret carried, and the original pixel pair is equals to (2,1). Note that the third pixel pairs are (9,11) and (9,10) in two marked images, respectively, we can easily extract the base-8 digits which are "1" and "8". This is a special case: therefore, we need to change "8" to "1". As a result, the secret digits are both equal to 1. Next, we transfer them to binary stream: 001001. According to the relative position of (9,11) and (9,10), we can determine the label of the center block is p_5. So the original pixel pair is (10,10). In the extraction phase, both marked images collaboratively contribute to data extraction and the recovery of the original image. A singular marked image is insufficient for this process. In contrast to previous matrix-based schemes where intercepting one marked image allowed a hacker to extract half of the hidden data, our scheme complicates such efforts. The obscurity of the pixel pair responsible for concealing the secret data makes it challenging to extract the correct half of the data. In our data extraction phase, both received marked images hold equal importance, with no precedence assigned to either.

4 Experimental Results

In our experiments, secret message B was generated using a pseudo-random number generator. All of the algorithms and experiments were implemented by Matlab 2022a in a PC with an Intel(R) Core™i5-1135G7 CPU @ 4.20GHz, a 16-GB RAM and Windows 11-64bit system.

4.1 Datasets

The images tested on our experiments were selected from three commonly used image databases and the Internet:

a) As is shown in Figure 9, we selected six classical 512 × 512 images from USC-SIPI database. The six images can be classified into two categories. The first three images including "Lena", "Barbara", and "Peppers", contain more smooth texture areas. The other three images contain more complex texture areas including "Baboon", "Boat' and "Lake". Moreover, six 512 × 512 sized color images which were downloaded from the Internet and trans-formed to gray-scale images were tested in our proposed scheme.

b) Kodak database consists of 24 uncompressed color images in the PNG format that were transformed to gray-scale images for testing. The sizes of these images are 768 × 512 or 512 × 768.

c) 1338 color images from the image database UCID in the TIFF format were also transformed to gray-scale images to verify the universality of our proposed scheme. The sizes of the UCID images are 512 × 384 or 384 × 512.

(a) Lena (b) Barbara (c) Peppers (d) Baboon (e) Boat (f) Lake

(g) Food (h) Bird (i) Wolf (j) Cartoon (k) Street (l) Plant

Figure 9. Twelve tested images

4.2 Experiments on Twelve Images

The PSNR indicator serves as an estimate for the visual quality of both the original cover image and the marked images. Typically, if the confidential data is securely embedded, the differences between the marked images and the cover image are nearly imperceptible to the human eye. Additionally, we employ the Structural Similarity Index Measure (*SSIM*), which aligns more closely with human visual intuition, to gauge the structural similarity between the cover image and the marked images. It's worth noting that the maximum value for *SSIM* is 1. It is computed as Eq.10:

$$SSIM(w,v)=\frac{(2\mu_w\mu_v+C_1)(2\sigma_{wv}+C_2)}{(\mu_w^2+\mu_v^2+C_1)(\sigma_w^2+\sigma_v^2+C_2)} \quad (10)$$

where μ_w and μ_v are mean, and μ_w^2 , μ_v^2 are the variance for the corresponding cover image and marked image. The constant $C_1 = (a_1e)^2$ and $C_2 = (a_2e)^2$, where $a_1 = 0.01$, $a_2 = 0.03$ and $e = 255$. Table 5 shows the results of *SSIM* for twelve tested images with different ERs, where $SSIM^1$ and $SSIM^2$ represent the *SSIM* value of $M^{(1)}$ and $M^{(2)}$, separately.

To show the performance of embedding capacity for this scheme, the embedding ratio ε is calculated as:

$$\varepsilon=\frac{P}{2\times H\times w} \quad (11)$$

Table 1 lists the PSNRs of twelve tested images of the proposed scheme with different embedding ratios, in which $PSNR^1$, $PSNR^2$ and $PSNR^{AVG}$ represent the PSNR value of $M^{(1)}$, $M^{(2)}$ and their average value, respectively. Table 2 shows the results of *SSIM* of twelve tested images with different embedding ratios, in which $SSIM^1$ and $SSIM^2$ are presented the *SSIM* value of $M^{(1)}$ and $M^{(2)}$, respectively. Thousands of Sudoku combinations have been tested to verify the effectiveness of our method, the experimental results are extremely close to the Sudoku sample given Figure 2. Specifically, the embedding capacity of each Sudoku combination also achieves up to 1.5 bits per pixel and maintains stable PSNR performance. It is proved from testing that our method can adapt to different Sudoku combinations.

Table 1. Results of SSIM with different embeddding ratios (bpp)

Image	0.2 bpp		0.5 bpp		1.0 bpp		1.5 bpp	
	$SSIM^1$	$SSIM^2$	$SSIM^1$	$SSIM^2$	$SSIM^1$	$SSIM^2$	$SSIM^1$	$SSIM^2$
Lena	0.999	0.999	0.998	0.996	0.995	0.988	0.990	0.974
Barbara	0.999	0.999	0.999	0.997	0.996	0.991	0.992	0.981
Peppers	0.999	0.999	0.998	0.996	0.995	0.989	0.990	0.976
Baboon	0.999	0.999	0.999	0.999	0.998	0.996	0.997	0.992
Boat	0.999	0.999	0.999	0.997	0.996	0.992	0.993	0.982
Lake	0.999	0.999	0.999	0.997	0.996	0.992	0.993	0.982
Food	0.999	0.999	0.998	0.996	0.994	0.987	0.989	0.972
Bird	0.999	0.999	0.998	0.996	0.993	0.982	0.985	0.974
Wolf	0.999	0.999	0.998	0.996	0.992	0.981	0.984	0.965
Cartoon	0.999	0.999	0.998	0.996	0.995	0.988	0.989	0.974
Street	0.999	0.999	0.996	0.991	0.994	0.987	0.992	0.979
Plant	0.999	0.999	0.998	0.996	0.995	0.989	0.991	0.976

Table 2. PSNR (dB) of twelve tested images with different embedding ratios (bpp)

Image	0.1 bpp			0.2 bpp			0.5 bpp			1.0 bpp			1.5 bpp		
	$PSNR^1$	$PSNR^2$	$PSNR^{AVG}$	$PSNR^1$	$PSNR^2$	$PSNR^{AVG}$	$PSNR^1$	$PSNR^2$	$PSNR^{AVG}$	$PSNR^1$	$PSNR^2$	$PSNR^{AVG}$	$PSNR^1$	$PSNR^2$	$PSNR^{AVG}$
Lena	65.14	61.73	63.44	60.35	57.27	58.81	54.74	51.20	52.97	50.02	46.42	48.22	46.87	42.92	44.90
Barbara	65.18	61.68	63.43	60.32	57.27	58.80	54.77	51.17	52.97	49.98	46.43	48.21	46.88	42.92	44.90
Peppers	65.13	61.69	63.41	60.30	57.25	58.78	54.77	51.16	52.97	50.02	46.40	48.21	46.88	42.90	44.89
Baboon	65.04	61.74	63.39	60.37	57.23	58.80	54.76	51.17	52.97	50.02	46.39	48.21	46.89	42.92	44.91
Boat	65.11	61.73	63.42	60.32	57.25	58.79	54.78	51.13	52.96	50.00	46.38	48.19	46.87	42.88	44.88
Lake	65.11	61.67	63.39	60.28	57.27	58.78	54.76	51.16	52.96	50.01	46.38	48.20	46.89	42.89	44.89
Food	65.10	61.73	63.41	60.28	57.34	58.81	54.77	51.18	52.98	50.02	46.45	48.23	46.89	42.92	44.91
Bird	65.68	62.73	64.21	59.96	57.71	58.84	54.55	52.68	53.62	50.03	46.06	48.05	47.76	42.73	45.25
Wolf	65.66	60.16	62.91	59.98	57.74	58.86	54.72	51.85	53.29	50.16	46.30	48.23	46.88	42.89	44.89
Cartoon	65.07	61.78	63.43	60.18	57.42	58.80	54.74	51.30	53.02	50.00	46.41	48.21	46.89	42.89	44.89
Street	65.08	61.72	63.40	60.31	57.28	58.80	54.72	51.16	52.94	50.00	46.41	48.21	46.89	42.89	44.89
Plant	65.10	61.71	63.41	60.25	57.33	58.79	54.75	51.23	52.99	50.03	46.40	48.22	46.85	42.90	44.88

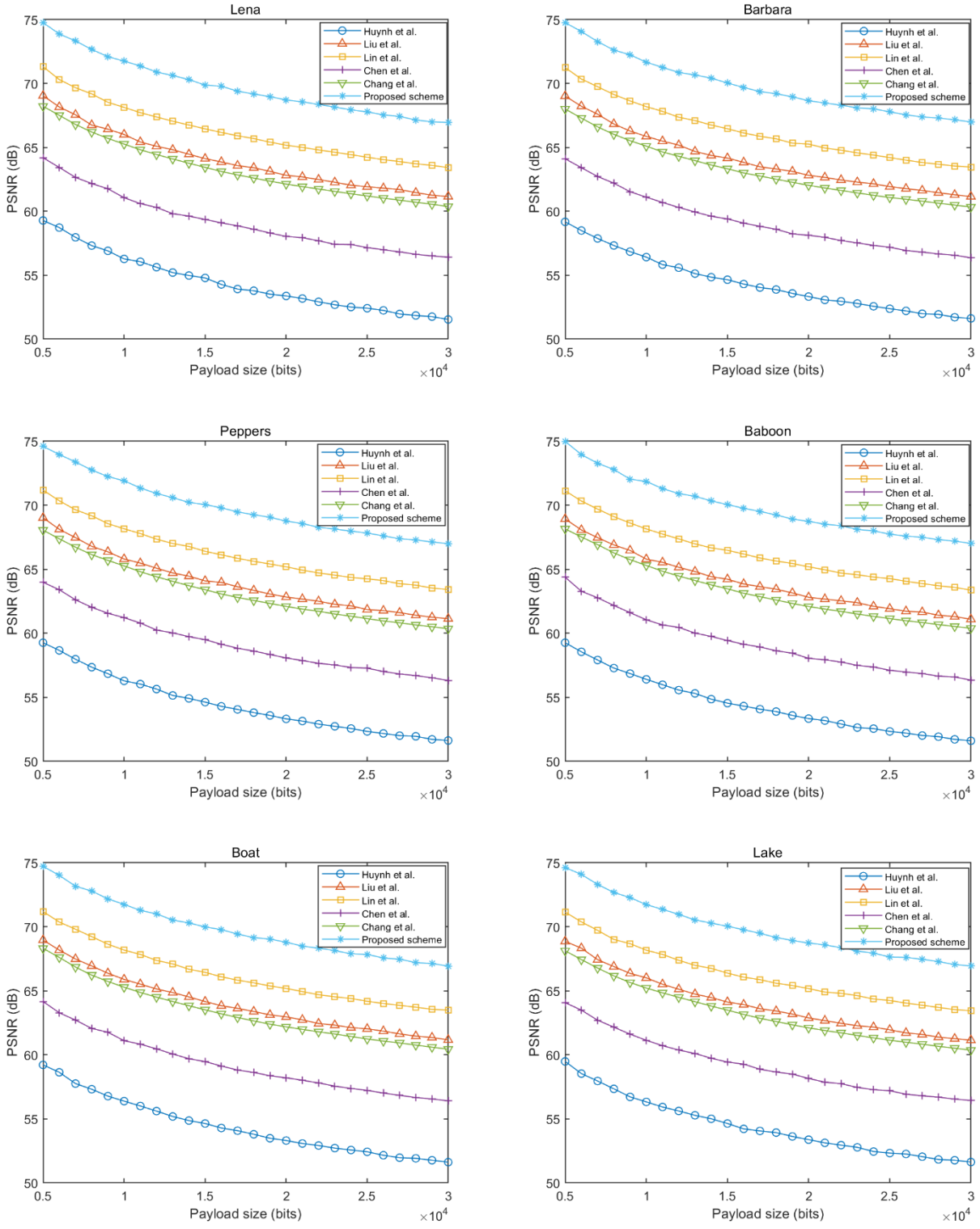


Figure 10. Average PSNR comparison with respect to different payloads between the proposed scheme and the matrix-based schemes

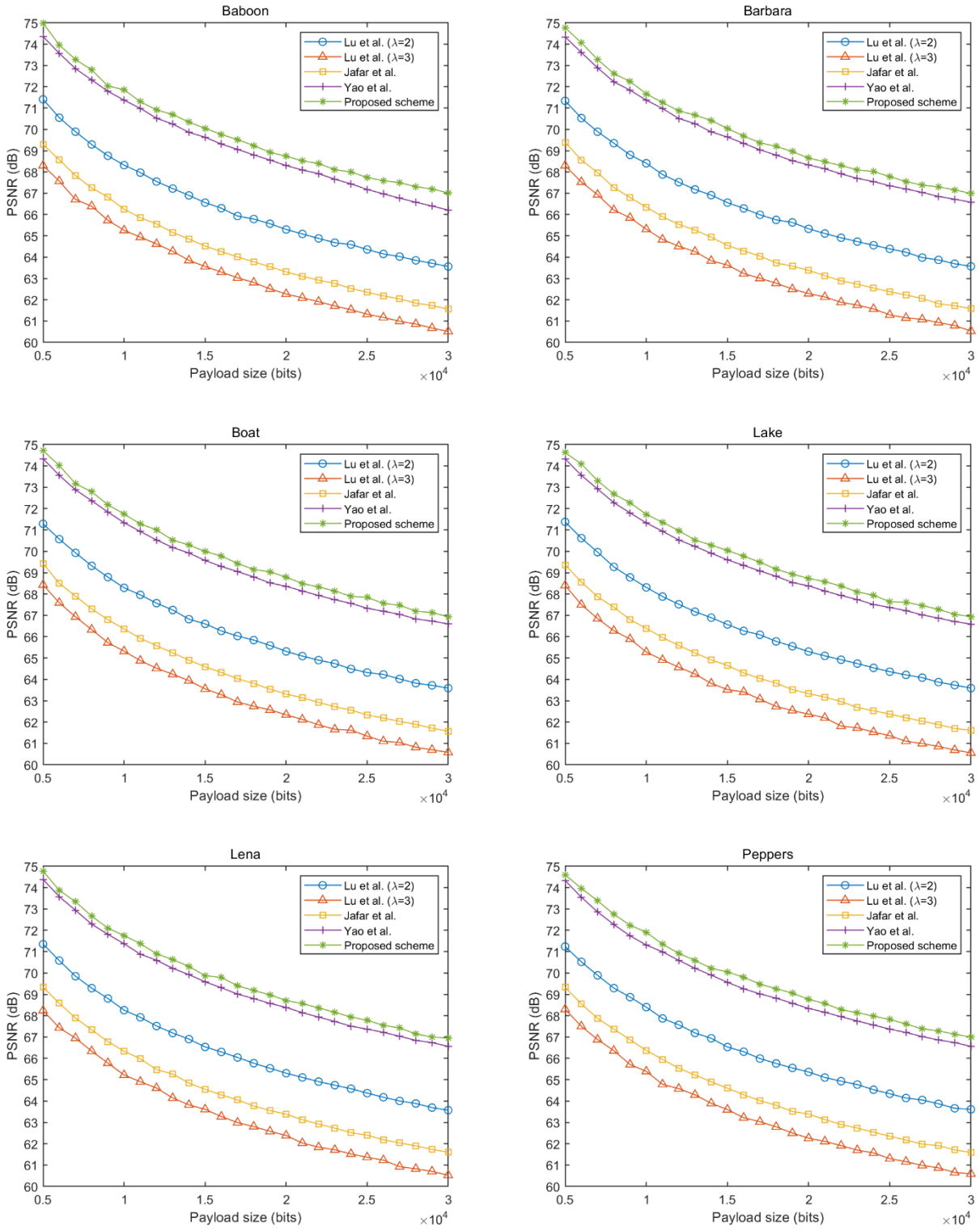


Figure 11. Average PSNR comparison with respect to different payloads between the proposed scheme and other typical schemes

Figure 10 shows $PSNR^{AVG}$ of six tested images with respect to different payload sizes. A comparison with prior matrix-based schemes [23-27], our performance stands out as superior, particularly in low-embedding rate scenarios. Notably, the enhancements achieved by our proposed scheme on image quality are substantial, measuring 15.37 dB, 5.81 dB, 3.54 dB, 6.6 dB. and 10.61 dB, respectively. Importantly, previous methods generated extensive extra data during embedding, which accompanied the marked images during delivery, posing potential security concerns and hindering convenient transmission. In contrast, our scheme streamlines extraction operations, requiring receivers to extract data and recover the image solely through the two marked images, thereby enhancing simplicity and practicality. Unlike our method, [23-26] necessitate receivers to possess additional data before extraction, resulting in less efficient data extraction operations compared to ours. Additionally, our scheme excels in self-controlling payload size, enabling receivers to easily and accurately extract secret data without prior knowledge of the payload length, and that is an adaptive capability lacking in [23-26].

Figure 11 shows $PSNR^{AVG}$ of six tested images between the proposed scheme and other typical schemes. In comparison with schemes [28] at $\lambda = 2$ and $\lambda = 3$, [29] and [30], our proposed scheme demonstrates increases in PSNR of 3.40 dB, 6.41 dB, 5.38 dB, and 0.38 dB, respectively. Notably, for the "Baboon" image with a payload of 30,000 bits, our algorithm achieves a PSNR of 67.02 dB, while the corresponding PSNR for scheme [30] decreases to 66.21 dB. This discrepancy is attributed to our algorithm maintaining stablevisual quality in complex texture images, whereas scheme [30] tends to favor smoothing texture images, resulting in limited embedding capacity.

Table 3 lists the average PSNR values of twelve tested images of previous schemes and the proposed scheme with the given payload of 5,000 bits, 10,000 bits and 20,000 bits. Compared with matrix-based schemes [23-26] and other typical schemes [28] when λ=2 and λ=3, [29-30], the average PSNR gains for our scheme are 15.5 dB, 5.87 dB, 3.4 dB, 10.59 dB, 3.5 dB, 6.38 dB, 5.3 dB and 0.45 dB on the condition of 5,000 bits. The optimization of distance threshold is the main factor for this improvement. For the given payload of 10,000 bits and 20,000 bits, the comparison results also illustrate that we achieve the best average PSNR. Overall, our scheme is superior for lower payload circumstances.

Table 4 lists the comparative results of the maximum embedding ratio of the proposed scheme with previous works. It shows that our scheme can achieve 1.5 bpp at most. The schemes [23], [26] and [28] pursue high embedding capacity. So these schemes sacrifice more visual quality, especially in low-embedding rate conditions. Table 3 and Table 4 present a comparison of average PSNR with lower payload sizes and maximum embedding ratios. Consequently, we have detailed all the methods with varied parameters in these tables toassess their performances. In contrast, Table 5 enumerates methods capable of maintaining higher embedding ratios. Notably, the method proposed by Lu et al. with λ=2 attains a maximum of 1.0 bpp, rendering it less competitive in embedding capacity. To ensure a fair comparison, methods with embedding capacities below 1.5 bpp are intentionally excluded from Table 5. Therefore, the method proposed by Lu et al. with λ=2 is excluded from Table 5. Although the scheme [23] achieves the best embedding ratio of 2 bpp, its PSNR performance is not good. Our PSNR performance keep higher than the scheme [28] when the embedding ratio is less than or equal to 1 bpp. But we fall behind at 1.5 bpp. The main reason is that the optimal distance threshold is invalid for the maximum embedding ratio. It will be our future work to improve this issue. Compared with the scheme [26], the average PSNR improvements of our scheme are 7.87 dB, 6.00 dB, 4.24 dB and 2.72 dB at 0.2 bpp, 0.5 bpp, 1.0 bpp and 1.5 bpp, respectively. All the experimental results demonstrate that our scheme gives full consideration of embedding capacity and visual quality.

Table 3. Comparision of average PSNR (dB) with different payload sizes (bits)

Scheme	5000 bits			10000 bits			20000 bits			30000 bits		
	$PSNR^1$	$PSNR^2$	$PSNR^{AVG}$	$PSNR^1$	$PSNR^2$	$PSNR^{AVG}$	$PSNR^1$	$PSNR^2$	$PSNR^{AVG}$	$PSNR^1$	$PSNR^2$	$PSNR^{AVG}$
Huynh et al. [23]	59.25	59.22	59.24	56.28	56.23	56.25	53.44	53.29	53.37	51.55	51.50	51.53
Liu et al. [24]	71.62	66.12	68.87	68.99	63.08	66.04	65.97	59.90	62.94	64.18	58.19	61.18
Lin et al. [25]	72.93	69.75	71.34	69.80	66.67	68.23	66.74	63.67	65.21	64.91	61.89	63.40
Chen et al. [26]	64.17	64.13	64.15	61.26	61.11	61.19	58.17	58.14	58.16	56.43	56.42	56.41
Chang et at. [27]	70.56	65.70	68.13	67.50	62.78	65.14	64.45	59.72	62.09	62.67	57.98	60.33
Lu et al. [28] (λ=2)	71.28	71.19	71.24	68.29	68.28	68.29	65.34	65.32	65.33	63.59	63.57	63.58
Lu et al. [28] (λ=3)	68.34	68.38	68.36	65.39	65.29	65.34	62.31	62.39	62.35	60.48	60.54	60.52
Jafar et al. [29]	71.44	67.43	69.44	68.37	64.32	66.35	65.30	61.36	63.33	63.57	59.59	61.58
Yao et al. [30]	74.31	74.26	74.29	71.36	71.36	71.36	68.40	68.29	68.35	66.53	66.59	66.56
Proposed scheme	77.22	72.26	74.74	74.15	69.26	71.71	71.18	66.21	68.70	69.38	64.48	66.93

Table 4. Comparison of maximum embedding ratio (bpp) with previous works

Scheme	Lena	Barbara	Peppers	Baboon	Boat	Lake	Food	Bird	Wolf	Cartoon	Street	Plant
Huynh et al. [23]	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00
Liu et al. [24]	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Lin et al. [25]	1.07	1.07	1.07	1.07	1.07	1.07	1.07	1.07	1.07	1.07	1.07	1.07
Chen et al. [26]	1.56	1.56	1.56	1.56	1.56	1.56	1.56	1.56	1.56	1.56	1.56	1.56
Chang et at. [27]	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25
Lu et al. [28] (λ=2)	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Lu et al. [28] (λ=3)	1.50	1.50	1.50	1.50	[30]	1.50	1.50	1.50	1.50	1.50	1.50	1.50
Jafar et al. [29]	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23
Yao et al. [30]	0.78	0.77	0.76	0.75	0.76	0.76	0.79	0.82	0.88	0.80	0.77	0.78
Proposed scheme	1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.50

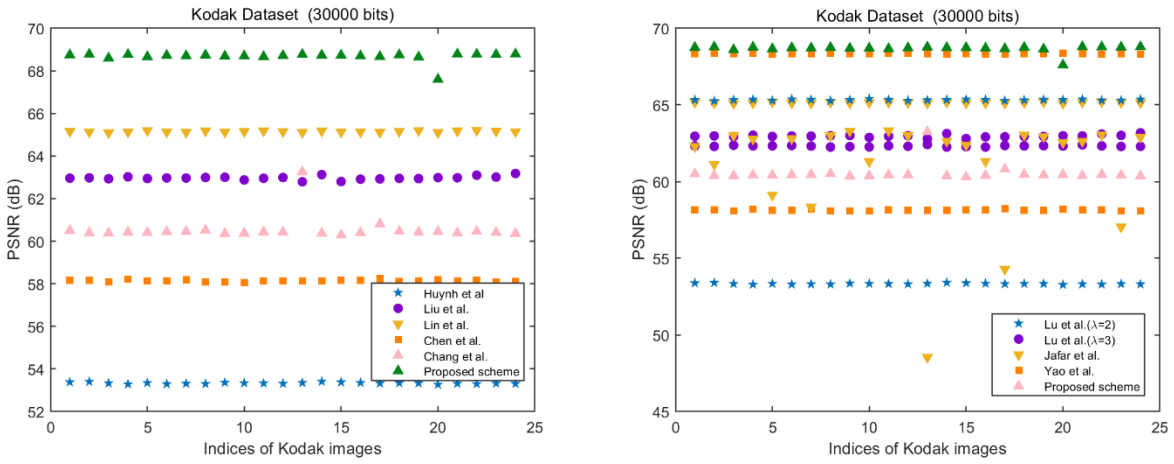
Table 5. Average PSNR (dB) comparison with high embedding ratios (bpp)

Scheme	0.5 bpp	1.0 bpp	1.5 bpp	2.0 bpp
	$PSNR^{AVG}$	$PSNR^{AVG}$	$PSNR^{AVG}$	$PSNR^{AVG}$
Huynh et al.[23]	42.16	39.14	37.39	36.13
Chen et al. [26]	46.97	43.97	42.18	-
Lu et al. [28] (λ=3)	51.15	48.10	46.36	-
Proposed scheme	53.05	48.20	44.92	-

4.3 Experiments on Diverse Images

Figure 12 shows the PSNRs on the Kodak dataset witha fixed payload size of 30,000 bits. The average PSNR ofour scheme is 68.67 dB, while the values of matrix-based schemes are 53.33 dB, 62.97 dB, 65.15 dB and 58.14 dB. The average PSNRs of other typical schemes are 68.34 dB, 65.33 dB, 62.31 dB and 61.11 dB, respectively.

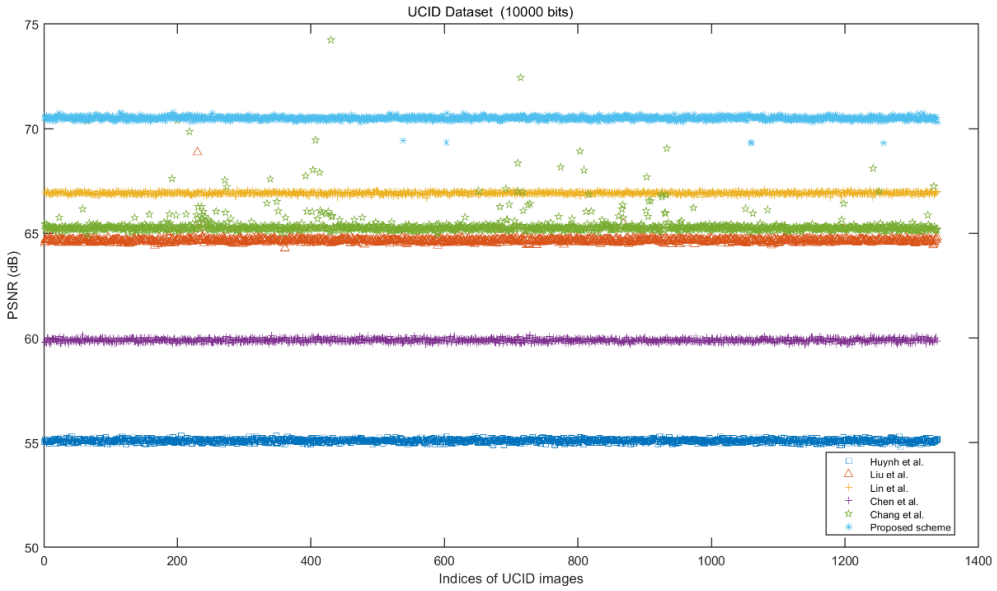
Figure 13 shows the PSNRs on the UCID dataset with a fixed payload size of 10,000 bits. It demonstrates that our scheme is also superior to others except four images which are extremely smooth. Unlike the scheme [30], experimental results find that our scheme fits complex texture images as well.



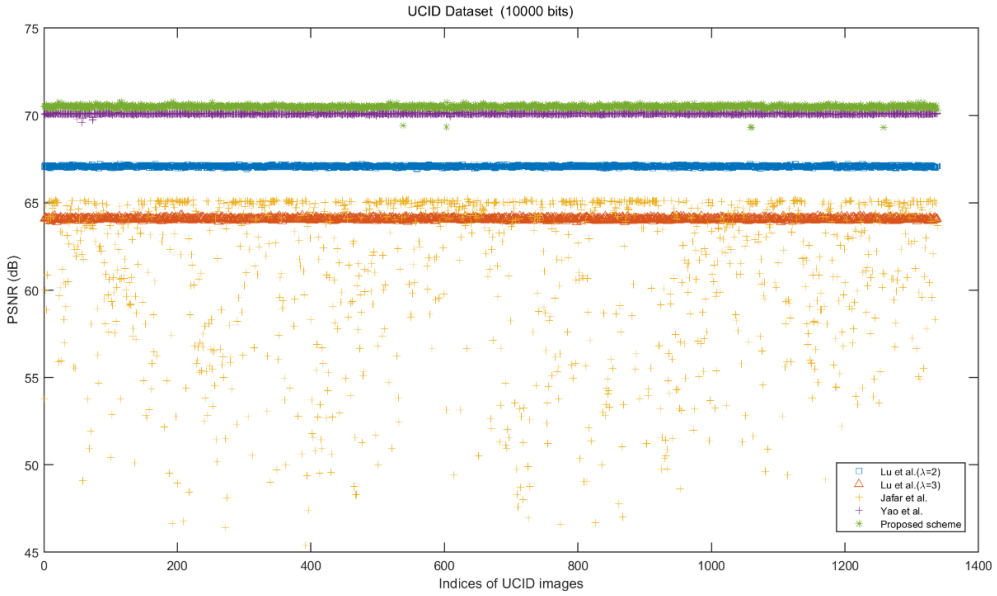
(a) Matrix-based schemes and proposed scheme

(b) Other typical schemes and proposed scheme

Figure 12. Performance comparison in terms of PSNR (dB) with the payload size of 30,000 bits in the Kodak dataset



(a) Matrix-based schemes and proposed scheme



(b) Other typical schemes and proposed scheme

Figure 13. Performance comparison in terms of PSNR (dB) with the payload size of 10,000 bits in the UCID dataset

4.4 Security Analysis

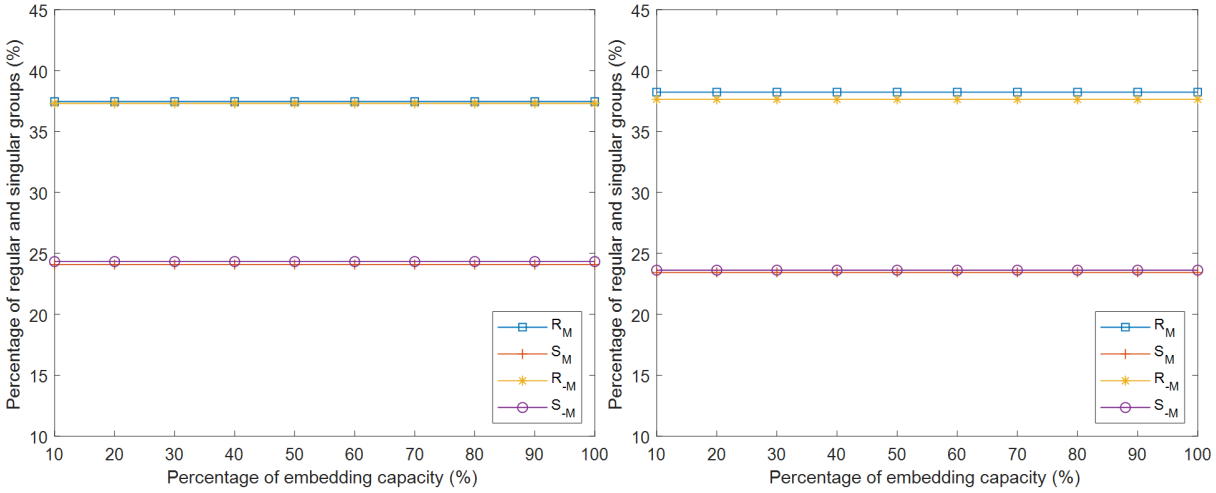
Now we test two steganalysis algorithms on the proposed scheme. The regular singular (RS) analysis [39] is a security test that considers four consecutive pixels in an image as a group, which is classified into regular, singular, and unchanged. Then, each group is flipped with a predefined mask, M or $-M$. The mask M is defined as [0,1,1,0] in our experiments. After flipping, the percentages of theregular and the singular groups with the mask M or $-M$, are calculated as R_M and S_M , or R_{-M} and S_{-M} , respectively. The values of R_M and S_M, or R_{-M} and S_{-M} should satisfy the formula:

$$R_M \cong R_{-M}, S_M \cong S_{-M} \tag{12}$$

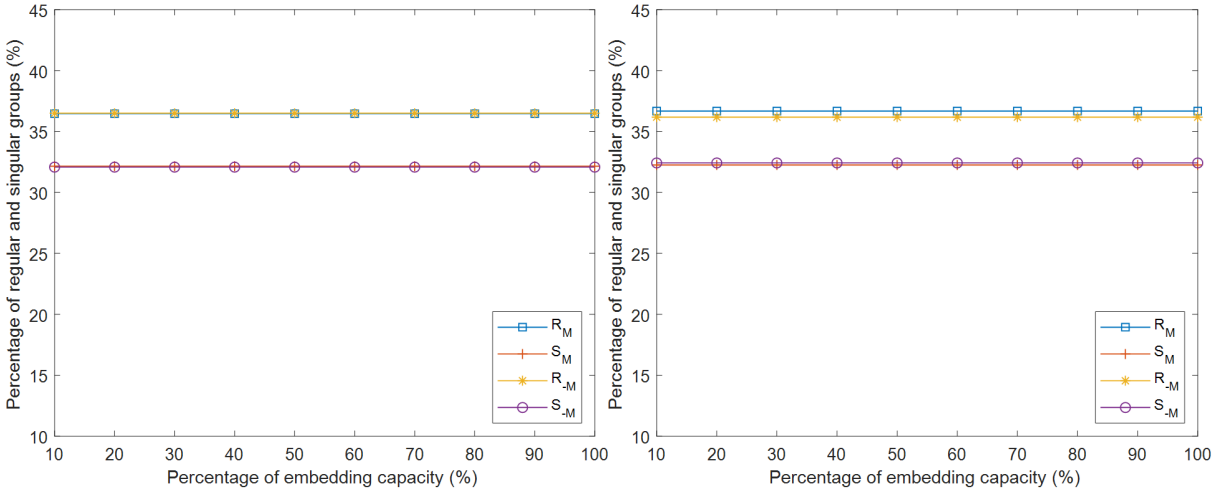
The RS analysis results for marked images of “Lena” and “Baboon” with different percentages of embedding capacityare plotted in Figure 14. As expected, each marked image follows Eq. 12, which means that two flipping operations increase the disorder of the image equally. Therefore, our method is robust to the RS analysis.

Another stegnalysis algorithm is the pixel-value differencing histogram (PDH) analysis [40]. It is a robust method that examines the similarity of the PDH between

the cover image and the marked images. It calculates the difference value of two consecutive pixels in an image and analyzes the frequency of the difference values. Due to the existing strong correlation between consecutive pixels, the PDH of a natural image is concentrated at the vicinity of the zero value. Figure 15 shows the PDH results at 0.5 bpp for four tested cover images with their dual marked images, respectively including "Lena", "Barbara", "Peppers" and "Boat". The PDH curves of the dual marked images are extremely close to that of the original cover image, which means it is difficult to distinguish between two marked images and its natural image. Our algorithm can resist the PDH analysis when the embedding ratio is less than or equal to 0.5 bpp.



(a) $M^{(1)}$ of "Lena" (b) $M^{(2)}$ of "Lena"

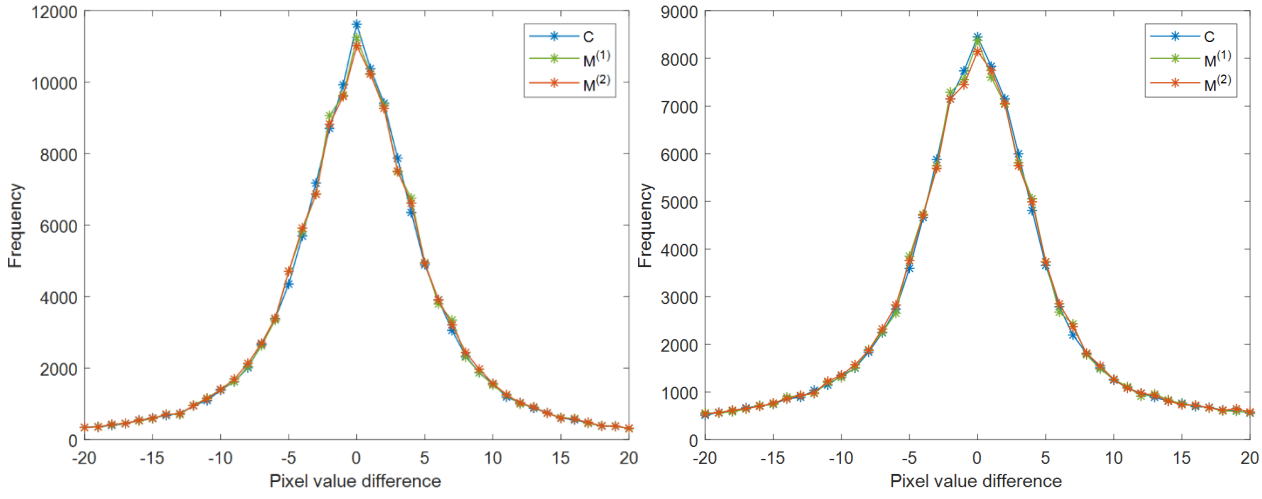


(a) $M^{(1)}$ of "Baboon" (b) $M^{(2)}$ of "Baboon"

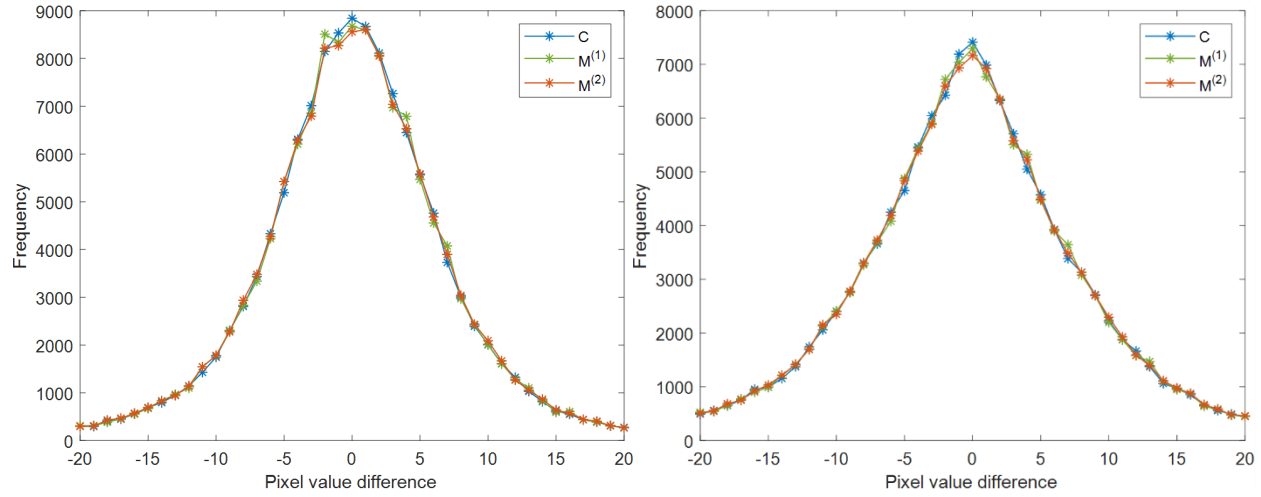
Figure 14. RS analysis for the dual marked images of "Lena" and "Baboon"

5 Conclusions

In this paper, we propose a novel dual-image RDH scheme with high capacity and low image distortion by using the SBM and the strategy of optimizing the distance threshold. The SBM achieves the reversibility of data hiding and the optimal distance threshold produces an adaptive embedding strategy to improve the visual quality, especially in low-embedding rate applications. More importantly, our scheme can balance the embedding capacity and visual quality commendably. It also provides a dynamic way to select pixel pair to enhance higher security performance than other matrix-based schemes. Some applications such as reversible data hiding based on 3D images which have left and right views will be studied for our future research. Besides, JPEG images are more widely used on the Internet we will consider more effective and simple methods based on them. The challenge to further improve the visual quality performance when the embedding ratio is greater than 1 bpp will also be our future work.



(a) "Lena" (b) "Barbara"



(a) "Lena" (b) "Barbara"

Figure 15. PDH analysis results of four cover images and their dual marked images

References

[1] J. Tian, Reversible Data Embedding Using a Difference Expansion, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 13, No. 8, pp. 890-896, August, 2003.

[2] Z. Ni, Y. Shi, N. Ansari, W. Su, Reversible Data Hiding, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 16, No. 3, pp. 354-362, March, 2006.

[3] D. M. Thodi, J. J. Rodriguez, Expansion Embedding Techniques for Reversible Watermarking, *IEEE Transactions on image Processing*, Vol. 16, No. 3, pp. 721-730, Reversible Watermarking Algorithm Using Sorting and PredictionMarch, 2007.

[4] V. Sachnev, H. Kim, J. Nam, S. Suresh, Y. Shi, Reversible Watermarking Algorithm Using Sorting and Prediction, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 19, No. 7, pp. 989-999, July, 2009.

[5] X. Li, B. Yang, T. Zeng, Efficient Reversible Watermarking Based on Adaptive Prediction-error Expansion and Pixel Selection, *IEEE Transactions on Image Processing*, Vol. 20, No. 12, pp. 3524-3533, December, 2011.

[6] X. Li, W. Zhang, X. Gui, B. Yang, A Novel Reversible Data Hiding Scheme Based on Two-dimensional Difference-Histogram Modification, *IEEE Transactions on Information Forensics and Security*, Vol. 8, No. 7, pp. 1091-1100, July, 2013.

[7] B. Ou, X. Li, Y. Zhao, R. Ni, Y. Shi, Pairwise Prediction-error Expansion for Efficient Reversible Data Hiding, *IEEE Transactions on Image Processing*, Vol. 22, No. 12, pp. 5010-5021, December, 2013.

[8] X. Li, J. Li, B. Li, B. Yang, High-fidelity Reversible Data Hiding Scheme Based on Pixel-Value-Ordering and Prediction-Error Expansion, *Signal Processing*, Vol 93, No. 1, pp. 198-205, January, 2013.

[9] X. Li, W. Zhang, X. Gui, B. Yang, Efficient Reversible Data Hiding Based on Multiple Histograms Modification, *IEEE Transactions on Information Forensics and Security*, Vol. 10, No. 9, pp. 2016-2027, September, 2015.

[10] Q. Chang, X. Li, Y. Zhao, R. Ni, Adaptive Pairwise Prediction-error Expansion and Multiple Histograms Modification for Reversible Data Hiding, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 31, No. 12, pp. 4850-4863, December, 2021.

[11] Q. Chang, X. Li, Y. Zhao, Reversible Data Hiding for Color Images Based on Adaptive Three-dimensional Histogram Modification, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 32, No. 9, pp. 5725-5735, February, 2022.

[12] Y. Fu, P. Kong, H. Yao, Z. Tang, C. Qin, Effective Reversible Data Hiding in Encrypted Image with Adaptive Encoding Strategy, *Information Sciences*, Vol. 494, pp. 21-36, August, 2019.
[13] Z. Yin, X. She, J. Tang, B. Luo, Reversible Data Hiding in Encrypted Images Based on Pixel Prediction and Multi-MSB Planes Rearrangement, *Signal Processing*, Vol 187, Article No. 108146, October, 2021.
[14] W. Lyu, L. Cheng, Z. Yin, High-capacity Reversible Data Hiding in Encrypted 3D Mesh Models Based on Multi-MSB Prediction, *Signal Processing*, Vol 201, Article No. 108686, December, 2022.
[15] H. Zou, G. Chen, Reversible Data Hiding in Encrypted Image with Local-correlation-based Classification and Adaptive Encoding Strategy, *Signal Processing*, Vol. 205, Article No. 108847, April, 2023.
[16] Z. Qian, X. Zhang, Lossless Data Hiding in JPEG Bitstream, *Journal of Systems and Software*, Vol. 85, No. 2, pp. 309-313, February, 2012.
[17] F. Huang, X. Qu, H. Kim, J. Huang, Reversible Data Hiding in JPEG Images, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 26, No. 9, pp. 1610-1621, September, 2016.
[18] D. Hou, H. Wang, W. Zhang, N. Yu, Reversible Data Hiding in JPEG Image Based on DCT Frequency and Block Selection, *Signal Processing*, Vol. 148, pp. 41-47, July, 2018.
[19] N. Li, F. Huang, Reversible Data Hiding for JPEG Images Based on Pairwise Nonzero AC Coefficient Expansion, *Signal Processing*, Vol. 171, Article No. 107476, June, 2020.
[20] Y. Du, Z. Yin, X. Zhang, High Capacity Lossless Data Hiding in JPEG Bitstream Based on General VLC Mapping, *IEEE Transactions on Dependable and Secure Computing*, Vol. 19, No. 2, pp. 1420-1433, March-April, 2022.
[21] X. Yang, T. Wu, F. Huang, Reversible Data Hiding in JPEG Images Based on Coefficient-first Selection, *Signal Processing*, Vol. 200, Article No. 108639, November, 2022.
[22] C. Chang, T. Kieu, Y. Chou, Reversible Data Hiding Scheme Using Two Steganographic Images, *TENCON 2007-2007 IEEE Region 10 Conference*, Taipei, Taiwan, 2007, pp. 1-4.
[23] N. Huynh, K. Bharanitharan, C. Chang, Quadri-directional Searching Algorithm for Secret Image Sharing Using Meaningful Shadows, *Journal of Visual Communication and Image Representation*, Vol. 28, pp. 105-112, April, 2015.
[24] Y. Liu, C. Chang, A Turtle Shell-based Visual Secret Sharing Scheme with Reversibility and Authentication, *Multimedia Tools and Applications*, Vol. 77, No. 19, pp. 25295-25310, October, 2018.
[25] J. Lin, Y. Chen, C. Chang, Y. Hu, Dual-image-based Reversible Data Hiding Scheme with Integrity Verification Using Exploiting Modification Direction, *Multimedia Tools and Applications*, Vol. 78, No. 18, pp. 25855-25872, September, 2019.
[26] X. Chen, C. Hong, An Efficient Dual-image Reversible Data Hiding Scheme Based on Exploiting Modification Direction, *Journal of Information Security and Applications*, Vol. 58, Article No. 102702, May, 2021.
[27] C. Chang, G. Su, C. Lin, Y. Li, Position-aware Guided Hiding Data Scheme with Reversibility and Adaptivity for Dual Images, *Symmetry*, Vol. 14, No. 3, Article No. 509, March, 2022.
[28] T. Lu, J. Wu, C. Huang, Dual-image-based Reversible Data Hiding Method Using Center Folding Strategy, *Signal Processing*, Vol. 115, pp. 195-213, October, 2015.
[29] I. Jafar, K. Darabkh, R. Al-Zubi, R. Saifan, An Efficient Reversible Data Hiding Algorithm Using Two Steganographic Images, *Signal Processing*, Vol. 128, pp. 98-109, November, 2016.
[30] H. Yao, F. Mao, Z. Tang, C. Qin, High-fidelity Dual-image Reversible Data Hiding via Prediction-error Shift, *Signal Processing*, Vol. 170, Article No. 107447, May, 2020.
[31] A. Shamir, How to Share a Secret, *Communications of the ACM*, Vol. 22, No. 11, pp. 612-613, November, 1979.
[32] G. Blakley, Safeguarding Cryptographic Keys, *1979 International Workshop on Managing Requirements Knowledge (MARK)*, New York, USA, 1979, pp. 313-318.
[33] M. Naor, A. Shamir, Visual Cryptography, *Advances in Cryptology-EUROCRYPT'94*, Perugia, Italy, 1994, pp. 1-12.
[34] X. Zhang, S. Wang, Efficient Steganographic Embedding by Exploiting Modification Direction, *IEEE Communications Letters*, Vol. 10, No. 11, pp. 781-783, November, 2006.
[35] C. Chang, Y. Chou, T. Kieu, An Information Hiding Scheme Using Sudoku, *2008 3rd International Conference on Innovative Computing Information and Control*, Dalian, China, 2008, pp. 17-17.
[36] S. Shastri, V. Thanikaiselvan, Dual Image Reversible Data Hiding Using Trinary Assignment and Centre Folding Strategy with Low Distortion, *Journal of Visual Communication and Image Representation*, Vol. 61, pp. 130-140, May, 2019.
[37] Y. Niu, S. Shen, A Novel Pixel Value Ordering Reversible Data Hiding Based on Dual-image, *Multimedia Tools and Applications*, Vol. 81, No. 10, pp. 13751-13771, April, 2022.
[38] E. Russel, F. Jarvis, Mathematics of Sudoku II, *Mathematical Spectrum*, Vol. 39, No. 2, pp. 54-58, January, 2006.
[39] J. Fridrich, M. Goljan, Practical Steganalysis of Digital Images: State of the Art, *Security and Watermarking of Multimedia Contents IV*, California, USA, 2002, pp. 1-13.
[40] X. Zhang, S. Wang, Vulnerability of Pixel-value Differencing Steganography to Histogram Analysis and Modification for Enhanced Security, *Pattern Recognition Letters*, Vol. 25, No. 3, pp. 331-339, February, 2004.

Biographies

Yurong Zhang received the B.S. degree in Chongqing College of Mobile Communication, Chongqing, China, in 2016. She is currently pursuing the master's degree with School of Cyberspace, Hangzhou Dianzi University. Her current research interests include image processing and information hiding.

Ye Yao received the M.S. degree in computer science and the Ph.D. degree in communication and information systems from Wuhan University, Wuhan, China, in 2005 and 2008, respectively. He is currently an Associate Professor with the School of Cyberspace, Hangzhou Dianzi

University, Hangzhou. His research interests include multimedia forensics and information security.

Chia-Chen Lin received the M.S. degree and the Ph.D degree in information management from Chiao Tung University, Hsinchu, Taiwan, in 1994 and 1998, respectively. She is currently a Professor in the Department of Computer and Information Management, Providence University, Sha-Lu, Taiwan. Her research interests include image and signal processing, image data hiding.

Chin-Chen Chang received the Ph.D degree in computer engineering from National Chiao Tung University, Hsinchu, in 1982. From July 1998 to June 2000, he was Director of the Advisory Office, Ministry of Education, R.O.C. From 2002 to 2005, he was a Chair Professor at National Chung Cheng University. From February 2005, he has been a Chair Professor at Feng Chia University. In addition, he was severed as a consultant to several research institutes and government departments. His current research interests include database design, computer cryptography, image compression, and data structures.