Signal Processing 236 (2025) 110056

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

A puzzle matrix oriented secret sharing scheme for dual images with reversibility

Yijie Lin [a], Jui-Chuan Liu [a,*], Ching-Chun Chang [b], Chin-Chen Chang [a,*]

[a] Department of Information Engineering and Computer Science, Feng Chia University, Taichung 40724, Taiwan
[b] Information and Communication Security Research Center, Feng Chia University, Taichung 40724, Taiwan

ARTICLE INFO

Keywords:
Dual images
Reference matrix
Reversible data hiding
Steganography

ABSTRACT

Dual image reversible data hiding is a specialized research direction in the field of information security. Secret messages are embedded into the original image using an image signal processing algorithm for reversible data hiding. This process generates two share images, indistinguishable from the original image, and each distributed to different receivers. Only through collaboration between the receivers can the original image and secret message be fully recovered. A dual image reversible data hiding scheme is proposed based on a reference matrix. The pixel pairs from the two share images collaborate and result in embedding 6 bits using the puzzle matrix; thus, achieving an embedding rate of up to 1.5 bits per pixel. Experimental results demonstrate that the proposed scheme offers a high embedding capacity of up to 786,432 bits, good visual quality, and fast execution time. Compared to other state-of-the-art dual image reversible data hiding schemes, our advantage mainly lies in the larger embedding capacity. Compared with a previously proposed scheme with the same embedding capacity, our proposed scheme provides better visual quality and faster execution time.

1. Introduction

In recent decades, information technology has developed rapidly making the Internet an indispensable part of people's lives. With a large amount of data is generated every day, while bringing convenience, it also raises significant security concerns. To prevent criminals from stealing data during transmission or storage, data hiding (DH) [1,2] has become an effective information security mechanism.

Data hiding protects information by embedding it into carriers such as text, audio, images, and videos. With the rapid advancements in image processing technologies [3], the use of images as carriers has become increasingly widespread in various fields. Using images as carriers is widely applied in data hiding technology and data is embedded into cover images to produce stego images. Data hiding is categorized into irreversible data hiding [4–7] and reversible data hiding (RDH) [8–13]. Since irreversible data hiding cannot restore the cover image, it concentrates on maintaining visual quality so that alterations are difficult to detect with naked eyes. Reversible data hiding, on the other hand, becomes particularly important because it allows for both data extraction and lossless restoration of the cover image. Due to its ability of maintaining data integrity and security, RDH is widely used in fields such as confidential communication, privacy protection, and copyright authentication.

Thien et al. [14] introduced the concept of reversible secret image sharing in 2002, which involved embedding secret data into multiple shares. A specified number of these shares could then be used to extract the secret data and reconstructed the original image. Building on this concept, secret image sharing [15–17] has been extensively explored, with dual-image techniques emerging as a distinct research area. For instance, dual images are employed in watermarking technology [18–19] for purposes such as authentication and anti-tampering, as well as in reversible data hiding [20–29] for the secure transmission and storage of secret messages. The dual image schemes utilize two cover images for reversible data hiding, enabling them to cooperate in both extracting the hidden data and restoring the original images. This scheme is particularly suitable for secret sharing applications involving two participants, such as spouses, where the data must be securely shared and extracted collaboratively. It addresses the challenge of ensuring data confidentiality and security while allowing the original image to be fully restored without distortion, thereby maintaining both data integrity and reversibility.

We propose a dual image reversible data hiding scheme based on a

* Corresponding authors.
E-mail addresses: p1263670@o365.fcu.edu.tw (Y. Lin), p1200318@o365.fcu.edu.tw (J. Liu), ccc@fcu.edu.tw (C.-C. Chang), ccc@o365.fcu.edu.tw (C. Chang).

https://doi.org/10.1016/j.sigpro.2025.110056
Received 5 October 2024; Received in revised form 6 February 2025; Accepted 14 April 2025
Available online 15 April 2025

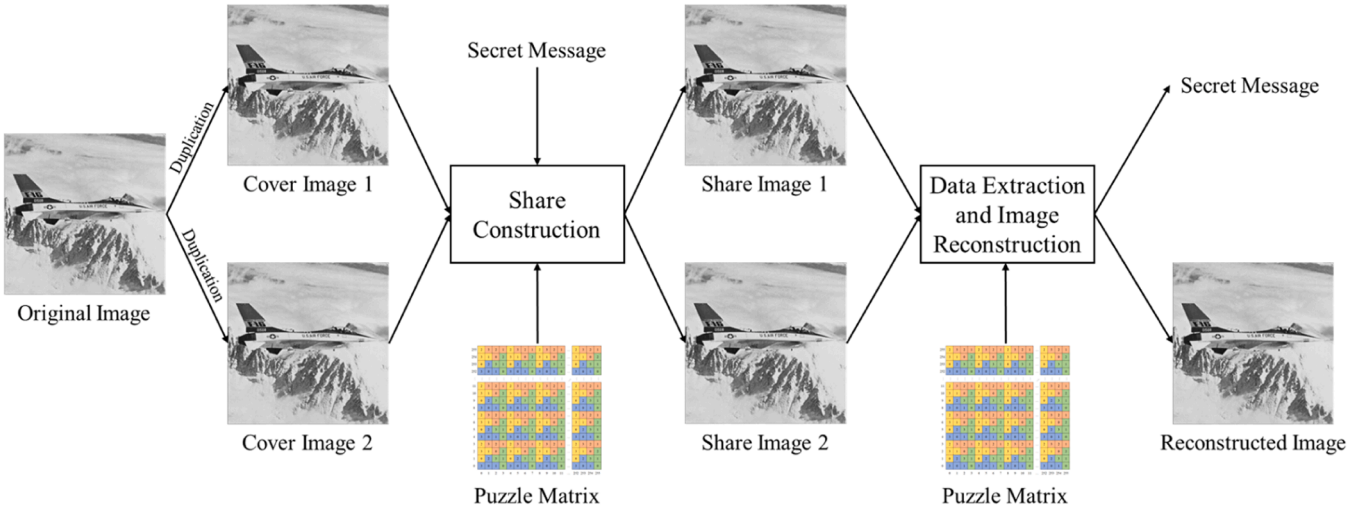

Fig. 1. Flow of our proposed scheme.

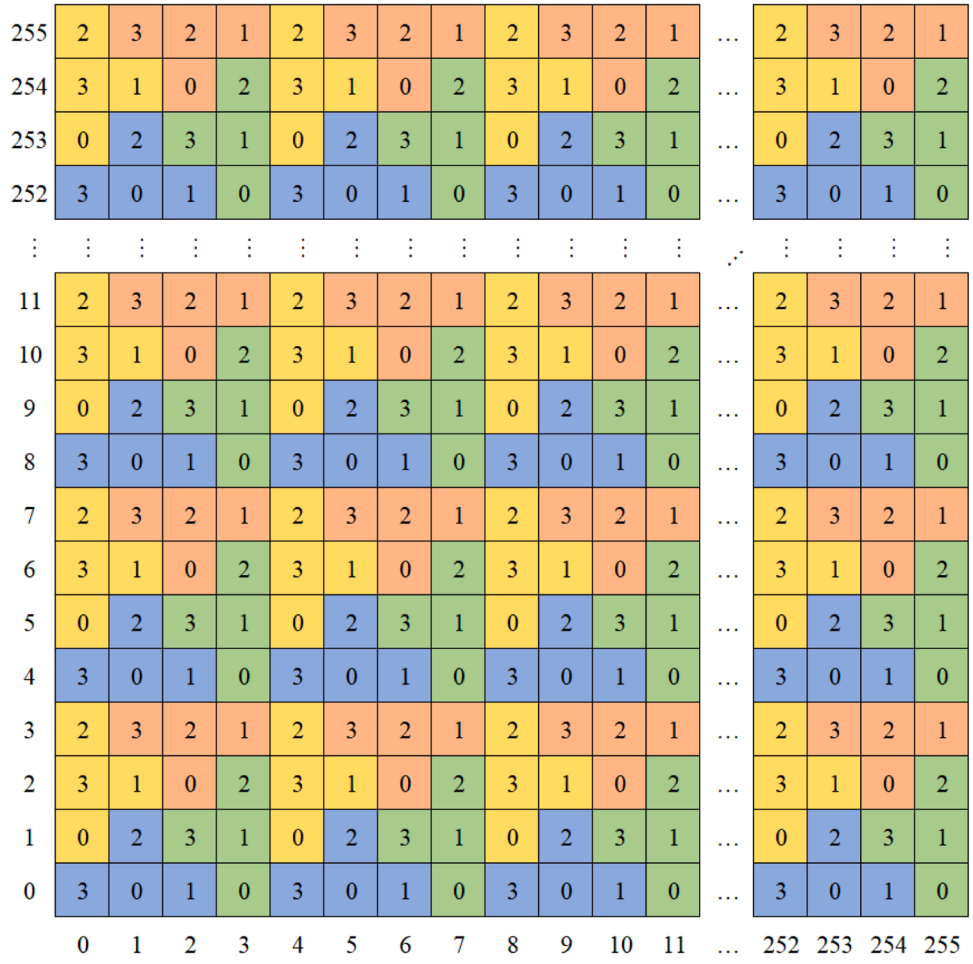

Fig. 2. Our proposed puzzle matrix.

reference matrix. Most data hiding schemes that use reference matrix [30–33] are irreversible. Our proposed scheme combines the characteristics of dual images and utilizes a proposed puzzle matrix to embed the necessary auxiliary information into the shares, thereby achieving the reversibility of the original image. The contributions of our proposed scheme are as follows:

1. We propose a dual image reversible data hiding scheme, using a puzzle matrix, which achieves a high embedding rate of 1.5 bits per pixel.
2. The visual quality is high; the difference between the share image and the original image is barely noticeable to the naked eye. The peak signal-to-noise ratio can reach approximately 39 dB.

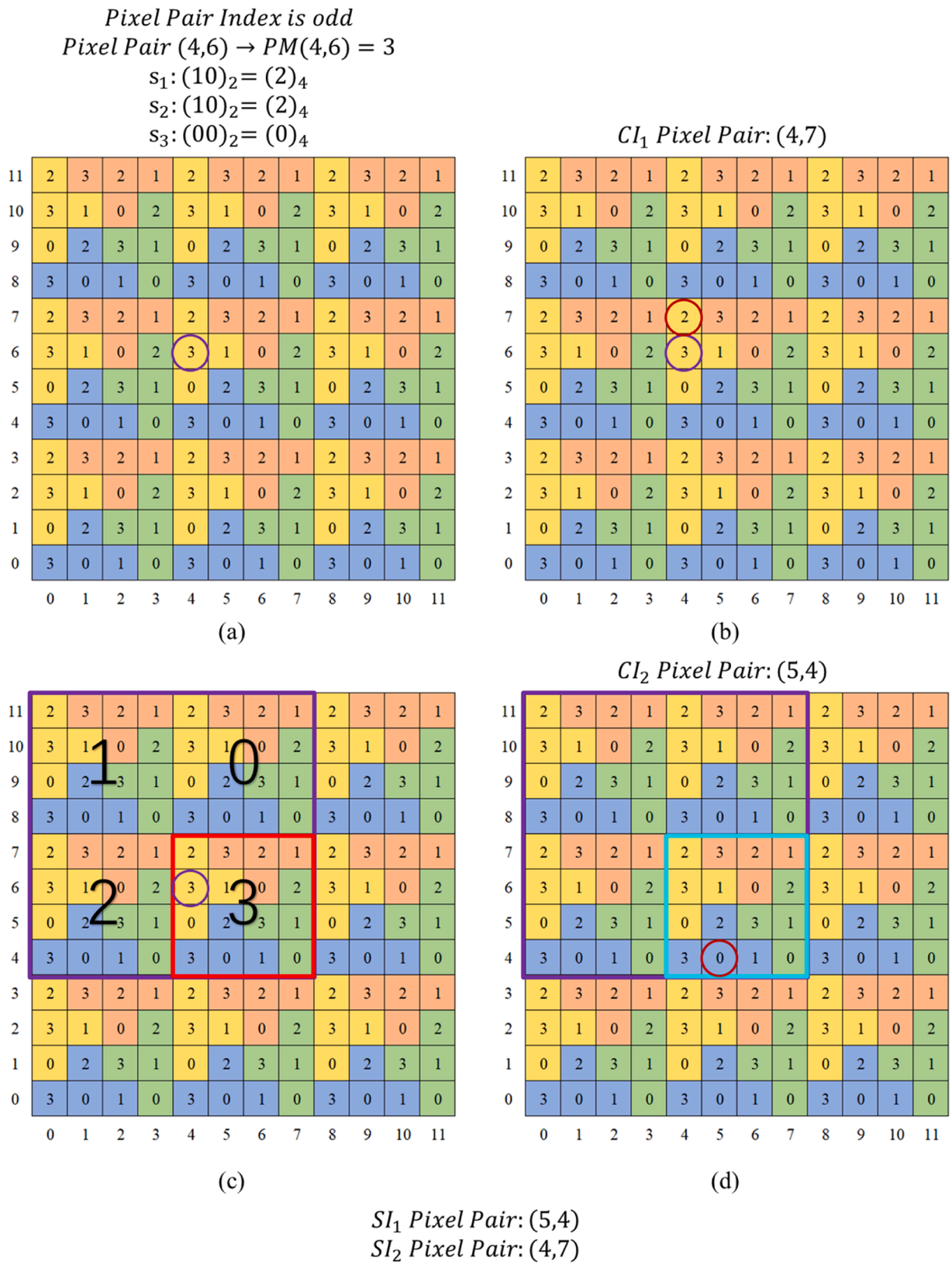

Fig. 3. A pixel pair example of share construction phase.

3. The execution time is very fast; for both the share construction phase and the data extraction and image reconstruction phase, it needs approximately 0.1 s.

The rest of this paper is organized as follows: Section 2 introduces the related works of our proposed scheme, Section 3 elaborates on our proposed dual image reversible data hiding scheme based on a designed puzzle matrix, and Section 4 presents the experimental evaluations and

Algorithm 1
Share construction.

Input	An $m \times n$ original image OI, the secret message S, and a puzzle matrix PM.
Output	Share images SI_1 and SI_2.
Step 1	Duplicate the original image OI into two cover images CI_1 and CI_2.

Step 2	Split S into $\frac{m \times n}{2}$ 6-bit segments. Each segment can be further split into three 2-bit segments, denoted as $s_{1,i,j}$, $s_{2,i,j}$ and $s_{3,i,j}$.
	for $i \in \{1, 2, \ldots, m\}$
	for $j \in \left\{1, 2, \ldots, \frac{n}{2}\right\}$
	$x = OI(i, 2j-1)$;
	$y = OI(i, 2j)$;
	// Step 2.1: Find the position of the element corresponding to $s_{1,i,j}$ in the puzzle piece of $PM(x,y)$
	$(x_1, y_1) = FindPosition(PM(x,y), s_{1,i,j})$;
	// Step 2.2: Expand 4 × 4 block to 16 × 16 block
	$expandedBlock = ExpandBlock(PM(x, y), (x1, y1))$;
	// Step 2.3: Select the block and element based on $s_{2,i,j}$ and $s_{3,i,j}$
	$blockIndex = PM(x, y)$;
	$puzzlePiece = Select(expandedBlock, blockIndex, s_{2,i,j}, s_{3,i,j})$;
	$(x_2, y_2) = puzzlePiece.Position$;
	// Step 2.4: Assign values to cover images
	$CI_1(i, 2j-1) = x_1$;
	$CI_1(i, 2j) = y_1$;
	$CI_2(i, 2j-1) = x_2$;
	$CI_2(i, 2j) = y_2$;
	// Step 2.5: Assign to share images
	if j mod 2 == 0
	$SI_1(i, 2j-1) = CI_1(i, 2j-1)$;
	$SI_1(i, 2j) = CI_1(i, 2j)$;
	$SI_2(i, 2j-1) = CI_2(i, 2j-1)$;
	$SI_2(i, 2j) = CI_2(i, 2j)$;
	else
	$SI_1(i, 2j-1) = CI_2(i, 2j-1)$;
	$SI_1(i, 2j) = CI_2(i, 2j)$;
	$SI_2(i, 2j-1) = CI_1(i, 2j-1)$;
	$SI_2(i, 2j) = CI_1(i, 2j)$;
	end if
	end for
	end for

Step 3	Output the share images SI_1 and SI_2.

Algorithm 2
Data extraction and image reconstruction.

Input	$m \times n$ Share images SI_1 and SI_2, and a puzzle matrix PM.
Output	A reconstructed image RI, and the secret message S.

Step 1	**for** $i \in \{1, 2, \ldots, m\}$
	for $j \in \left\{1, 2, \ldots, \frac{n}{2}\right\}$
	// Step 1.1: Recover the cover images
	if j mod 2 == 0
	$CI_1(i, 2j-1) = SI_1(i, 2j-1)$;
	$CI_1(i, 2j) = SI_1(i, 2j)$;
	$CI_2(i, 2j-1) = SI_2(i, 2j-1)$;
	$CI_2(i, 2j) = SI_2(i, 2j)$;
	else
	$CI_1(i, 2j-1) = SI_2(i, 2j-1)$;
	$CI_1(i, 2j) = SI_2(i, 2j)$;
	$CI_2(i, 2j-1) = SI_1(i, 2j-1)$;
	$CI_2(i, 2j) = SI_1(i, 2j)$;
	end if
	// Step 1.2: Extract values from cover images
	$x_1 = CI_1(i, 2j-1)$;
	$y_1 = CI_1(i, 2j)$;
	$x_2 = CI_2(i, 2j-1)$;
	$y_2 = CI_2(i, 2j)$;
	// Step 1.3: Extract parts of the secret message $s_{1,i,j}$ and $s_{3,i,j}$
	$s_{1,i,j} = dec2bin(PM(x_1, y_1))$;
	$s_{3,i,j} = dec2bin(PM(x_2, y_2))$;
	// Step 1.4: Expand a 4 × 4 block to a 16 × 16 block
	$expandedBlock = ExpandBlock(PM(x_1, y_1), (x_1, y_1))$;
	// Step 1.5: Identify the block value at $PM(x_2, y_2)$
	$blockIndex = PM(x_2, y_2)$;
	// Step 1.6: Determine the original pixel values
	$(x, y) = Reconstruct(expandedBlock, PM(x_1, y_1), PM(x_2, y_2))$;
	// Step 1.7: Extract $s_{2,i,j}$ and convert it to binary
	$s_{2,i,j} = dec2bin(blockIndex)$;
	// Step 1.8: Update the reconstructed image RI
	$RI(i, 2j-1) = x$;
	$RI(i, 2j) = y$;
	end for
	end for

Step 2	Output a reconstructed image RI, and the secret message S.

comparisons with state-of-the-art dual image reversible data hiding schemes. Finally, Section 5 summarizes and discusses our work.

2. Related work

As mentioned in the previous section, dual image reversible data hiding is an effective technique for information protection. By embedding a secret message into the original image using a specific image signal processing algorithm, two share images are generated and distributed to different receivers. When the receivers collaborate, the secret message can be extracted and the original image can be reconstructed using the reverse algorithm.

In recent years, dual image reversible data hiding has emerged as an active research field. While earlier studies primarily focused on enhancing visual quality with a fixed embedding capacity, recent efforts have been shifted toward increasing embedding capacity. Chang et al. [20] introduced the exploiting modification direction (EMD) algorithm [30], embedding secret messages to generate two share images and achieving an embedding capacity of 1 bits per pixel (bpp), in 2007. By 2015, Lu et al. [21] implemented seven rules based on least significant bit (LSB) matching to modify pixels. It ensured both high embedding quantities of secret messages and good visual quality of embedded images. On top of these methods, Sahu and Swain [25] applied an enhanced LSB matching algorithm which led to even better visual quality in 2019. In 2022, Tseng and Leng [26] refined the approach further by combining the improved LSB matching with the EMD algorithm and resulted in notable visual improvements. The following year, Liu et al. [28] introduced a matrix coding based scheme, successfully increasing embedding capacity to 1.5 bpp while preserving acceptable visual quality. Most recently, Kim et al. [29] further optimized this line of research by using the EMD algorithm alongside an optimal least significant bit technique to achieve an embedding capacity of approximately 1.1 bpp in 2024.

These advancements encouraged the continuous innovations in balancing embedding capacity and image quality in the field of dual image reversible data hiding.

3. Proposed scheme

This section elaborates on the dual image reversible data hiding scheme based on a puzzle matrix that we proposed. Fig. 1 shows the flow of the proposed scheme. First, the original image is duplicated into two cover images; then, the secret message is encrypted using a secret key and embedded into the two cover images by referring to the preset puzzle matrix to obtain share image 1 and share image 2. During the data extraction and image reconstruction phase, both share image 1 and share image 2 must cooperate, refer to the puzzle matrix, use the secret key to extract the secret message and reconstruct the original image.

3.1. Puzzle matrix generation

Using a designed puzzle matrix as the reference matrix to embed data is the key step of the proposed scheme. The construction rule for the puzzle matrix is that each 4 × 4 block contains four puzzle pieces, represented by red, yellow, blue, and green in the Fig. 2. Each element in a puzzle piece is represented by a non-repeating number from 0 to 3. This same rule is repeatedly applied to the entire 256 × 256 puzzle matrix, where the numbers 0–255 for rows and columns are

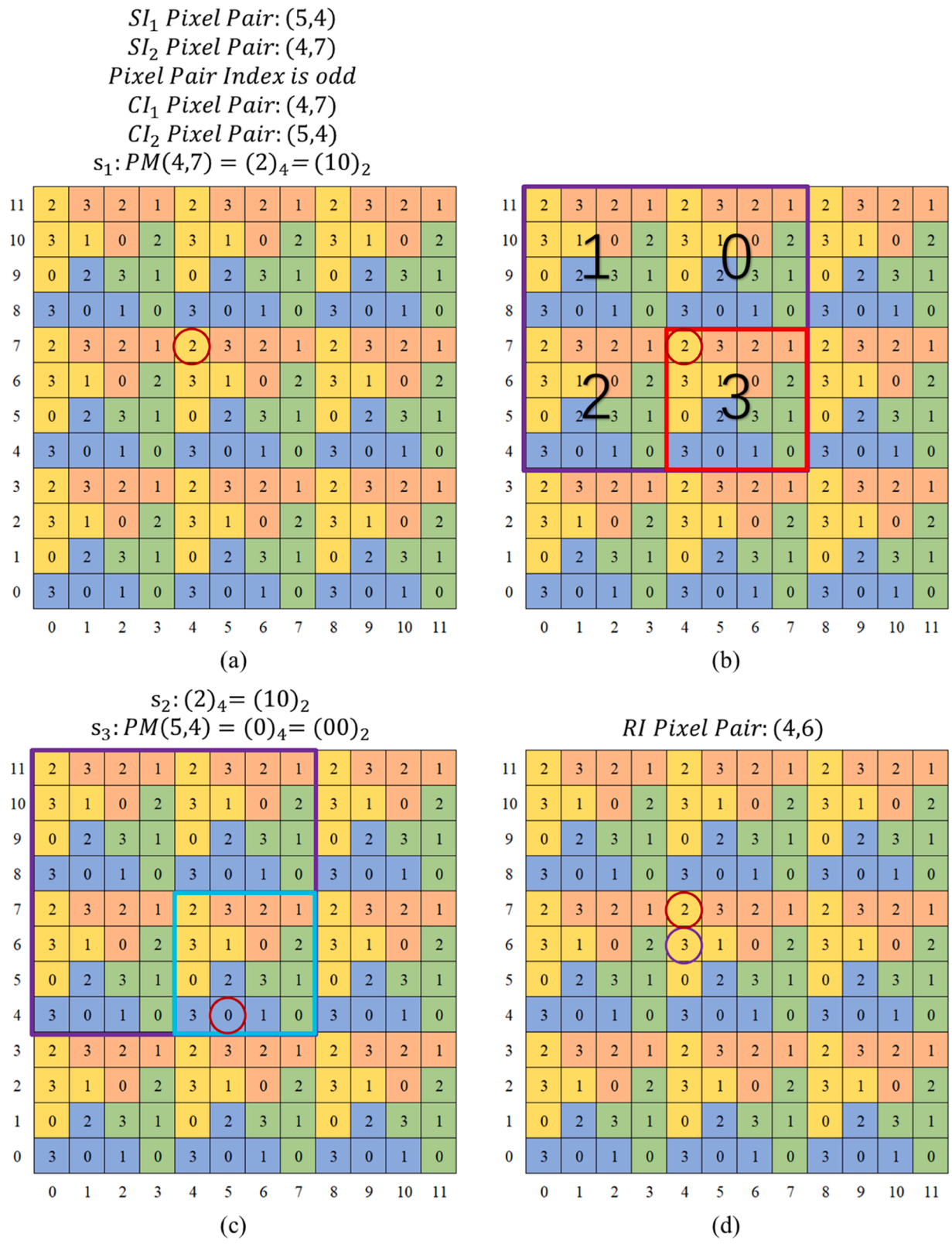

Fig. 4. A pixel pair example of data extraction and image reconstruction phase.

corresponded to pixel values.

According to the above construction rules, the puzzle matrix has $4!^{\frac{256 \times 256}{4}} = 24^{16384}$ variations, making it difficult to crack by brute force. Therefore, the puzzle matrix can be used as a reference matrix for embedding data and effectively served as a data hiding key.

Although the puzzle matrix has a vast number of combination possibilities, an example of an ideal puzzle matrix is shown in Fig. 2 for clarification. Its construction rules are simple: for a 4 × 4 block, first fill the 2 × 2 block at the center with the numbers 0 to 3 in sequence corresponding to the four puzzle pieces: red, yellow, blue, and green.

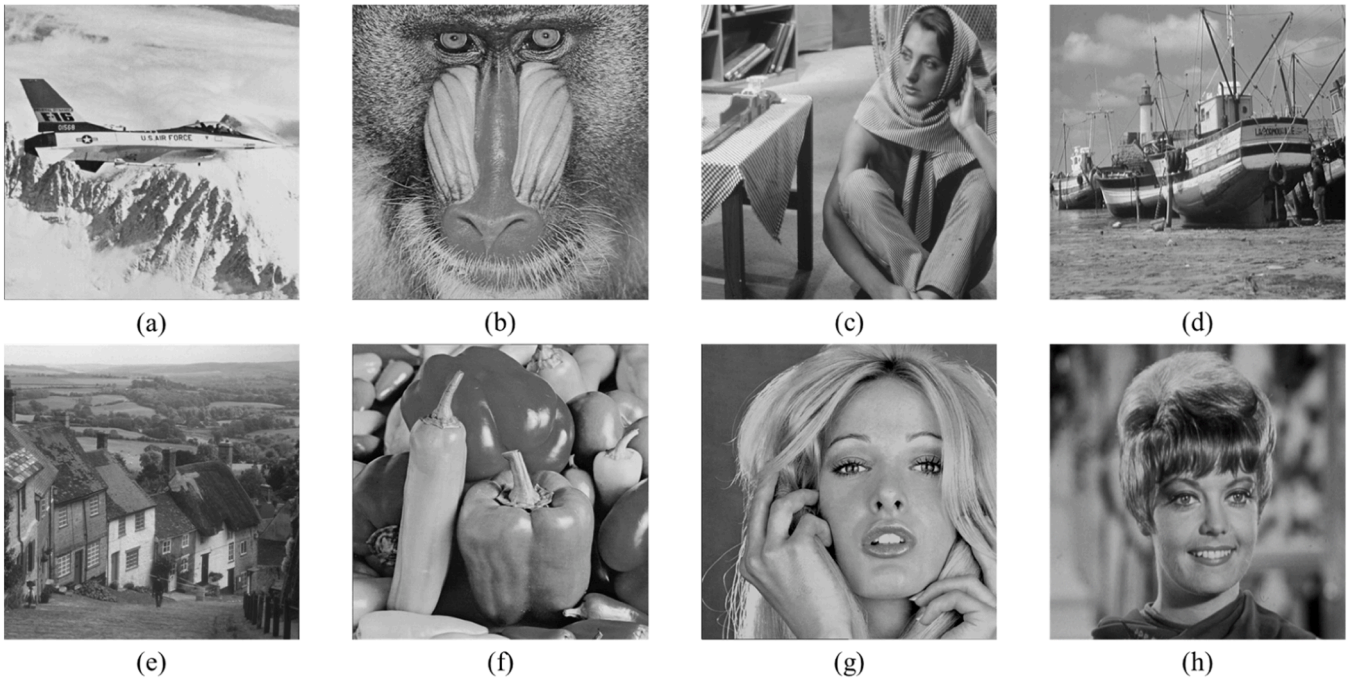

Fig. 5. Test images: (a) Airplane; (b) Baboon; (c) Barbara; (d) Boat; (e) Goldhill; (f) Peppers; (g) Tiffany; (h) Zelda.

Then, fill in the remaining elements of each puzzle piece in sequence to complete the 4×4 block. This block is then copied and expanded to a size of 256×256 to form an ideal puzzle matrix. It is called the ideal puzzle matrix because this construction scheme maintains regularity and aesthetic appeal. The distances between identical elements are relatively similar, preventing any pixel modification from being excessively large or small.

3.2. Share construction phase

We first duplicate the original image into two cover images and group two adjacent pixels from the original image into non-overlapped pixel pairs. Assuming the size of the original image is $m \times n$, there will be $\frac{m \times n}{2}$ pixel pairs. The secret message S, which is encrypted using a secret key, is then divided into $\frac{m \times n}{2}$ segments, each containing 6 bits. These segments are further divided into three 2-bit parts: s_1, s_2, and s_3.

For each pixel pair, first find the location in the puzzle matrix based on the pixel values. Then, identify the corresponding element in the puzzle piece, where the location is determined by s_1, and store the element's location as a pixel pair in the first cover image. Next, depending on the location of the puzzle piece within the block, expand it in the same direction into a 16×16 block. If the puzzle is at the upper left corner, expand it to a 16×16 block at the upper left corner. Apply the same logic for other directions. If the block exceeds the matrix boundaries, translate it back to the matrix. This 16×16 block consists of four 4×4 blocks, represented sequentially by 0–3 as seen in Fig. 3(c). The selected block is determined by the value at the puzzle matrix position corresponding to the pixel pair, and the puzzle piece in the selected block is selected based on s_2. Finally, the element in this puzzle piece is selected based on s_3, and the position corresponding to the element is stored as a pixel pair in the second cover image.

Depending on whether the pixel pair index is odd or even, the cover image pixel pairs are used to construct the share image pixel pairs in an alternating manner. The purpose is to balance the distortion between the two share images and maintain a relatively close difference from the original image. The two share images can eventually encode the same amount of secret information.

Algorithm 1 describes the detailed steps of share construction. In it, "==" means equal and "=" means assign.

Fig. 3 shows a pixel pair example going through the share construction phase. Assume the pixel pair index is odd, the pixel pair processed is $(4,6)$; segmented secrets s_1 is $(10)_2$, s_2 is $(10)_2$, and s_3 is $(00)_2$. As shown in Fig. 3(a), first locate the position of the pixel pair in the puzzle matrix PM and it falls on the yellow puzzle piece. Since s_1 is $(10)_2$ which equals $(2)_4$, find the location with the value of $(2)_4$ in the yellow puzzle piece and use the coordinates of the location as the pixel pair of the cover image CI_1. It results in $(4,7)$ as shown in Fig. 3(b). Because the yellow puzzle piece is located at the upper left corner of the 4×4 block within the red box, it's expanded in the upper left direction into a 16×16 block marked by the purple box as seen in Fig. 3(c). There are four 4×4 blocks represented by 0 to 3 within the 16×16 purple block. In Fig. 3(d), since $PM(4,6)$ equals $(3)_4$, the 4×4 block represented $(3)_4$ is selected and marked by the light blue box. Since s_2 is $(10)_2$ which equals $(2)_4$, it corresponds to the blue puzzle piece in the selected block. With s_3 being $(00)_2$, select the location with a value of $(0)_4$ in the blue puzzle piece, use its coordinates as the pixel pair of cover image CI_2, and result in $(5,4)$. Finally, because the pixel pair index is odd, SI_1 is represented by CI_2, and SI_2 is represented by CI_1 to yield the share image pairs SI_1 and SI_2.

3.3. Data extraction and image reconstruction phase

When two shares are combined, data can be extracted and the original image can be reconstructed. First, the share image pixel pairs are used to construct the embedded cover image pixel pairs in an interleaved manner based on the parity of the pixel pair index. It results in two embedded cover images CI_1 and CI_2. From the value of CI_1 pixel pair and its corresponding puzzle piece, s_1 can be determined. From the value of CI_2 pixel pair and its corresponding puzzle piece in the puzzle matrix, s_3 can be obtained similarly. After analyzing the puzzle piece corresponding to CI_1 pixel pair, we can determine its location within the 4×4 block. This direction, determined by the location of the puzzle piece, is then used to expand the block into a 16×16 area. The area consisting of four 4×4 blocks and blocks are numbered by 0–3. The value of the original pixel pair in the puzzle matrix can be determined based on the block corresponding to CI_2 pixel pair. By combining this

Fig. 6. Share images: (a)-(b) Airplane; (c)-(d) Baboon; (e)-(f) Barbara; (g)-(h) Boat; (i)-(j) Goldhill; (k)-(l) Peppers; (m)-(n) Tiffany; (o)-(p) Zelda.

with the puzzle piece corresponding to CI_1 pixel pair, the original pixel pair can be obtained. From the puzzle piece corresponding to CI_2 pixel pair, s_2 can be derived. After processing all pixel pairs, the secret message, encrypted with the secret key, can be extracted and the original image can be reconstructed to image RI.

Algorithm 2 details the process of data extraction and image reconstruction. In it, "==" means equal and "=" means assign.

Fig. 4 shows an example of the data extraction and image reconstruction. Given that the pixel pair index is odd, the share image pixel pairs SI_1 and SI_2 are (5,4) and (4,6) respectively. Since the pixel pair index is odd, the cover image pixel pairs CI_1 and CI_2 are represented by SI_2 and SI_1 after embedding. By substituting CI_1 into the puzzle matrix, we can directly obtain the value of s1 as indicated by the red circle in Fig. 4(a). After converting to binary, s_1 is $(10)_2$. As shown in Fig. 4(b), the position corresponding to CI_1 is in the yellow puzzle piece which is located at the upper left of the 4 × 4 block in red. It then expands to the upper left to form the 16 × 16 block marked by the purple box. The purple block consists of four 4 × 4 blocks represented by numbers from 0 to 3. In Fig. 4(c), the position of the 4 × 4 light blue block corresponding to CI_2 is equivalent to $(3)_4$, the blue puzzle piece is corresponding to $(2)_4$, and the value at location (5,4) is $(0)_4$. This indicates that the value of the reconstructed image RI pixel pair in the puzzle matrix is $(3)_4$, s_2 is $(2)_4$, and s_3 is $(0)_4$. Converting s_2 and s_3 to binary gives $(10)_2$ and $(00)_2$ respectively. As shown in Fig. 4(d), we find the position with the value $(3)_4$ in the yellow puzzle piece corresponding to the pixel pair in CI_1 as well as the reconstructed image. The red circle

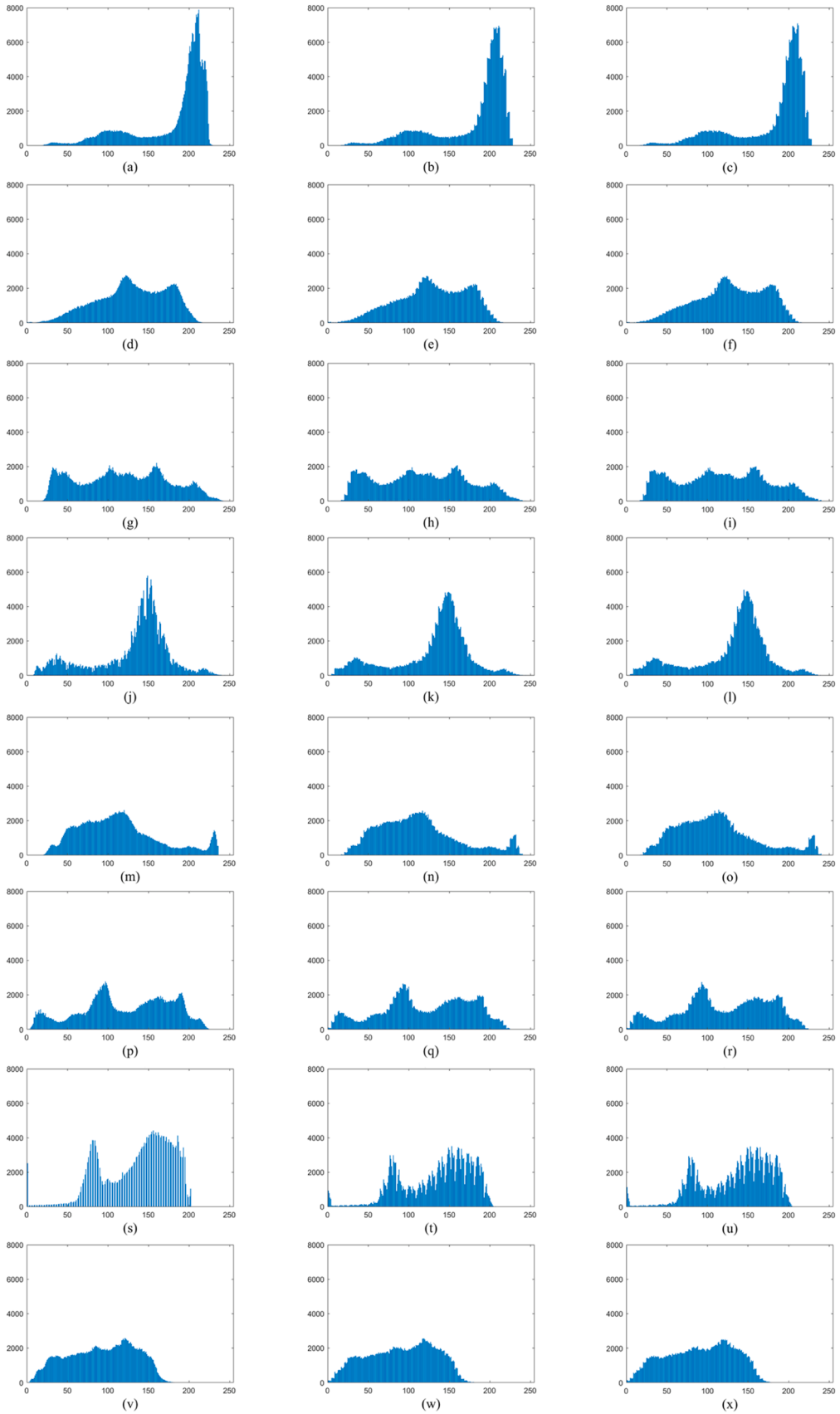

Fig. 7. Histograms of original image, share image 1, and share image 2: (a)-(c) Airplane; (d)-(f) Baboon; (g)-(i) Barbara; (j)-(l) Boat; (m)-(o) Goldhill; (p)-(r) Peppers; (*s*)-(u) Tiffany; (v)-(x) Zelda.

Table 1
Horizontal and vertical correlation coefficients of the original image, share image 1, and share image 2.

Images	Horizontal			Vertical		
	Original	$Share_1$	$Share_2$	Original	$Share_1$	$Share_2$
Airplane	0.936	0.926	0.926	0.959	0.956	0.956
Baboon	0.836	0.832	0.832	0.715	0.712	0.712
Barbara	0.866	0.864	0.864	0.951	0.949	0.949
Boat	0.860	0.851	0.851	0.963	0.960	0.960
Goldhill	0.943	0.932	0.932	0.970	0.967	0.967
Peppers	0.974	0.971	0.971	0.972	0.969	0.969
Tiffany	0.935	0.931	0.931	0.919	0.911	0.911
Zelda	0.974	0.969	0.969	0.990	0.985	0.985

marks the position corresponding to CI_1; the purple circle marks $(4, 6)$ which is the position corresponding to RI. This is the position used as a pixel pair to reconstruct the original image.

4. Experimental results

There were eight 512 × 512 standard grayscale images were selected, as shown in Fig. 5, for experiments. The experiments were conducted on a laptop with a 3.20 GHz AMD Ryzen 7 CPU and 16GB of RAM, running MATLAB R2024a under Window 11. We compared various dual image reversible data hiding schemes to evaluate the performance of the proposed scheme. The results show that the proposed scheme achieves a high embedding capacity up to 1.5 bpp while ensuring visual quality.

The peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) are commonly utilized for assessing visual quality in data hiding experiments. PSNR evaluates the difference in visual fidelity between two images, as outlined in Eqs. (1) and (2). In Eq. (2), the variables m and n represent the image dimensions, while i and j correspond to individual pixel coordinates. I stands for the original image, and K denotes the share image. A lower mean square error (MSE) indicates less image distortion which correlates with better visual quality. Since PSNR and MSE are inversely related, a higher PSNR value signifies improved image fidelity. SSIM, on the other hand, measures the resemblance between two images, referred to as X and Y. Eq. (3) illustrates how SSIM independently examines the luminance, contrast, structure of the images, and then combines these aspects into a single value using weighted products. In this equation, μ_X and μ_Y represent the mean luminance; σ_X and σ_Y denote the luminance variance; σ_{XY} refers to the covariance between X and Y; C_1 and C_2 are constants added to prevent instability. A value of SSIM closed to 1 implies that the two images are highly similar.

$$PSNR = 10 \times log_{10}\frac{255^2}{MSE}. \tag{1}$$

$$MSE = \frac{1}{m \times n}\sum_{i=1}^{m}\sum_{j=1}^{n}[I(i,j) - K(i,j)]^2. \tag{2}$$

$$SSIM(X, Y) = \frac{(2\mu_X\mu_Y + C_1)(2\sigma_{XY} + C_2)}{(\mu_X^2 + \mu_Y^2 + C_1)(\sigma_X^2 + \sigma_Y^2 + C_2)}. \tag{3}$$

Fig. 6 shows the share images generated by the proposed scheme.

Table 2
Performance comparison with other dual image reversible data hiding schemes.

Images	Metrics	Chang et al. [20]	Lu et al. [21]	Sahu et al. [25]	Tseng et al. [26]	Liu et al. [28]	Kim et al. [29]	Proposed
Airplane	EC	524,208	524,288	524,288	524,288	786,432	576,558	786,432
	bpp	1.00	1.00	1.00	1.00	1.50	1.10	1.50
	$PSNR_1$	45.11	49.39	51.13	51.13	36.36	50.34	38.97
	$PSNR_2$	45.11	49.03	49.73	51.13	36.35	51.13	38.98
	$PSNR_{AVG}$	45.11	49.21	50.43	51.13	36.35	50.74	38.98
Baboon	EC	522,888	522,996	524,288	524,210	786,432	576,203	786,432
	bpp	1.00	1.00	1.00	1.00	1.50	1.10	1.50
	$PSNR_1$	45.18	47.95	51.16	51.14	36.33	50.34	39.03
	$PSNR_2$	45.19	49.15	49.41	51.14	36.34	51.13	39.03
	$PSNR_{AVG}$	45.19	48.55	50.29	51.14	36.33	50.74	39.03
Barbara	EC	524,288	524,288	524,288	524,288	786,432	577,040	786,432
	bpp	1.00	1.00	1.00	1.00	1.50	1.10	1.50
	$PSNR_1$	45.20	49.14	51.12	51.13	36.36	50.34	39.01
	$PSNR_2$	45.21	49.11	49.73	51.13	36.36	51.12	39.00
	$PSNR_{AVG}$	45.21	49.13	50.43	51.13	36.36	50.73	39.01
Boat	EC	524,208	524,208	524,288	524,288	786,432	576,521	786,432
	bpp	1.00	1.00	1.00	1.00	1.50	1.10	1.50
	$PSNR_1$	45.20	49.00	51.18	51.14	36.30	50.35	38.93
	$PSNR_2$	45.21	49.07	49.41	51.14	36.27	51.15	38.93
	$PSNR_{AVG}$	45.21	49.04	50.30	51.14	36.28	50.75	38.93
Goldhill	EC	524,288	524,288	524,288	524,288	786,432	576,399	786,432
	bpp	1.00	1.00	1.00	1.00	1.50	1.10	1.50
	$PSNR_1$	45.20	49.17	51.12	51.13	36.32	50.35	39.03
	$PSNR_2$	45.17	49.09	49.72	51.12	36.32	51.15	39.03
	$PSNR_{AVG}$	45.19	49.13	50.42	51.13	36.32	50.75	39.03
Peppers	EC	523,356	524,192	524,288	524,240	786,432	576,659	786,432
	bpp	1.00	1.00	1.00	1.00	1.50	1.10	1.50
	$PSNR_1$	45.21	49.11	51.14	51.14	36.36	50.35	38.99
	$PSNR_2$	45.21	49.08	49.72	51.14	36.36	51.13	39.00
	$PSNR_{AVG}$	45.21	49.10	50.43	51.14	36.36	50.74	38.99
Tiffany	EC	524,288	524,288	524,288	524,288	786,432	576,947	786,432
	bpp	1.00	1.00	1.00	1.00	1.50	1.10	1.50
	$PSNR_1$	45.18	49.38	51.15	51.16	36.39	50.35	39.03
	$PSNR_2$	45.19	49.06	49.73	51.16	36.39	51.16	39.03
	$PSNR_{AVG}$	45.19	49.22	50.44	51.16	36.39	50.76	39.03
Zelda	EC	524,288	524,288	524,288	524,288	786,432	576,705	786,432
	bpp	1.00	1.00	1.00	1.00	1.50	1.10	1.50
	$PSNR_1$	45.66	49.14	51.14	51.14	36.34	50.36	39.02
	$PSNR_2$	45.20	49.09	49.71	51.14	36.33	51.14	39.02
	$PSNR_{AVG}$	45.43	49.12	50.43	51.14	36.34	50.75	39.02

Table 3
Comparison of visual quality between our proposed scheme and Liu et al.'s scheme [28].

Images	PSNR						SSIM					
	Liu et al. [28]			Proposed			Liu et al. [28]			Proposed		
	SI_1	SI_2	Average	SI_1	SI_2	Average	SI_1	SI_2	Average	SI_1	SI_2	Average
Airplane	36.36	36.35	36.35	38.97	38.98	38.98	0.8990	0.8984	0.8987	0.9468	0.9468	0.9468
Baboon	36.33	36.34	36.33	39.03	39.03	39.03	0.9645	0.9644	0.9645	0.9839	0.9839	0.9839
Barbara	36.36	36.36	36.36	39.01	39.00	39.01	0.9313	0.9313	0.9313	0.9657	0.9658	0.9658
Boat	36.30	36.27	36.28	38.93	38.93	38.93	0.9259	0.9255	0.9257	0.9648	0.9648	0.9648
Goldhill	36.32	36.33	36.32	39.03	39.03	39.03	0.9274	0.9279	0.9277	0.9659	0.9658	0.9658
Peppers	36.36	36.36	36.36	38.99	39.00	38.99	0.9053	0.9053	0.9053	0.9534	0.9533	0.9534
Tiffany	36.39	36.39	36.39	39.03	39.03	39.03	0.9150	0.9149	0.9150	0.9585	0.9585	0.9585
Zelda	36.34	36.33	36.34	39.02	39.02	39.02	0.8916	0.8915	0.8916	0.9476	0.9475	0.9475
Average	36.34	36.34	36.34	39.00	39.00	39.00	0.9200	0.9199	0.9200	0.9608	0.9608	0.9608

Table 4
Comparison of execution time for different phases between our proposed scheme and Liu et al.'s scheme [28].

Phase	Liu et al. [28]	Proposed	Improvement Rate
Share Construction Phase	0.5242 s	0.0804 s	84.66 %
Data Extraction and Image Reconstruction Phase	0.9158 s	0.1236 s	86.50 %

The differences between the original image, share image 1, and share image 2 are almost imperceptible to the naked eye, indicating that the share images have good visual quality.

From the perspective of security analysis, Fig. 7 presents the histograms of the original image, share image 1, and share image 2. The histograms of the share images are not significantly different from that of the original image, maintaining a similar pixel distribution. It suggests that the share pixels generated by the proposed scheme exhibit limited changes, which is crucial for ensuring the security of the data embedding process. The similarity in histograms indicates that the visual quality is preserved after hiding the information, making the scheme resistant to detection through statistical analysis of pixel values. Table 1 demonstrates the performance of different images, including the original image and share images, in terms of horizontal and vertical correlation coefficients. Typically, the original image exhibits high horizontal and vertical correlation coefficients, reflecting a strong similarity between adjacent pixels. In contrast, the share images show only a slight decrease in both horizontal and vertical correlation coefficients compared to the original image. The minimal reduction in correlation further suggests that the proposed scheme effectively embeds data while maintaining the structural integrity of the image. This indicates that the scheme offers good security and ensures that the embedded data does not compromise the overall statistical properties of the image.

Table 2 compares the performance of the proposed scheme with other dual image reversible data hiding schemes. We primarily evaluate the embedding capacity, bpp, and PSNR. The calculation of bpp is shown in Eq. (4). The bpp indicates the average number of bits that can be embedded per pixel, EC represents the embedding capacity, and $2 \times M \times N$ represents the two images of size $M \times N$. Experimental results show that both Liu et al.'s scheme [28] and our proposed scheme achieve an embedding capacity up to 786,432 bits, or 1.5 bpp. Other dual image reversible data hiding schemes [20,21,25,26] offer embedding capacities of 1 bpp, while Kim et al.'s scheme [29] achieves 1.1 bpp. These results highlight the higher capacity advantage of our proposed scheme. As expected, a larger embedding capacity results in a greater image distortion. The PSNR values of our proposed scheme are around 39 dB. When a PSNR value exceeds 30 dB, the distortion is imperceptible to the human eye and it meets visual quality standards. Therefore, even though the PSNR values of our proposed scheme are slightly lower than those of other schemes with smaller capacities, they remain within acceptable limits. It is unlikely to achieve the best in both embedding capacity and visual quality simultaneously, as this is a natural tradeoff.

$$bpp = \frac{EC}{2 \times M \times N}. \tag{4}$$

To further discuss the performance of the dual image reversible data hiding scheme on high embedding capacity, we compare our results with results from Liu et al.'s scheme [28] which can also achieve 1.5 bpp. Table 3 presents the visual quality comparison between the two schemes. The difference between share image 1 SI_1 and share image 2 SI_2 in both schemes is negligible. This indicates balanced visual quality and contributes to enhanced security. The PSNR values of Liu et al.'s scheme are around 36 dB, while the values of our proposed scheme are approximately 39 dB. The SSIM values of Liu et al.'s scheme ranges from 0.8915 to 0.9645, while the values of our proposed scheme range from 0.9468 to 0.9839. These results demonstrate that, under the same embedding capacity of 1.5 bpp, our proposed scheme offers better visual quality.

We also compared the execution time across different phases between the two schemes. As show in Table 4, the execution time for the share construction phase in Liu et al.'s scheme is 0.5242 s, while the time for our proposed scheme is 0.0804 s. It represents an improvement of 84.66 %. For the data extraction and image reconstruction phase, Liu et al.'s scheme takes 0.9158 s, whereas our proposed scheme takes 0.1236 s which shows an improvement of 86.50 %. These results demonstrate that our proposed scheme has superior efficiency in execution time.

5. Conclusions

Data hiding is a crucial technique for ensuring information security. In this paper, we propose a dual-image reversible data hiding scheme using a puzzle matrix. The novelty of our proposed scheme lies in the introduction of the puzzle matrix, which allows for the embedding of 6 bits into pixel pairs of two cover images and achieves an embedding capacity of up to 1.5 bpp. Experimental results reveal several key findings: first, the proposed scheme achieves an embedding capacity of 786,432 bits which surpasses most of the state-of-the-art schemes. Second, our approach demonstrates superior visual quality with PSNR values around 39 dB compared to other schemes with similar embedding capacities. Additionally, the execution time for each stage is approximately 0.1 s demonstrating the scheme's efficient processing. These findings collectively highlight that the proposed scheme offers higher embedding capacity, faster execution time, and good visual quality. It makes our proposed scheme well-suited for secret sharing between two receivers, while ensuring the lossless reversibility of both the secret message and the original image.

CRediT authorship contribution statement

Yijie Lin: Writing – review & editing, Writing – original draft,

Software, Data curation. **Jui-Chuan Liu:** Writing – original draft, Visualization, Software, Project administration. **Ching-Chun Chang:** Validation, Resources, Investigation, Formal analysis, Conceptualization. **Chin-Chen Chang:** Validation, Supervision, Resources, Project administration, Investigation, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

I have shared the link to my data.

References

[1] W. Bender, D. Gruhl, N. Morimoto, A. Lu, Techniques for data hiding, IBM Syst. J. 35 (3.4) (1996) 313–336.
[2] M. Wu, B. Liu, Data hiding in image and video. I. Fundamental issues and solutions, IEEE Trans. Image Process. 12 (6) (2003) 685–695.
[3] Y. Fan, C. Hong, G. Zeng, L. Liu, A deep convolutional encoder–Decoder–Restorer architecture for image deblurring, Neural Process. Lett. 56 (1) (2024) 27.
[4] C.K. Chan, L.M. Cheng, Hiding data in images by simple LSB substitution, Pattern. Recognit. 37 (3) (2004) 469–474.
[5] A.D. Ker, Steganalysis of LSB matching in grayscale images, IEEE Signal Process Lett. 12 (6) (2005) 441–444.
[6] J. Mielikainen, LSB matching revisited, IEEE Signal Process Lett. 13 (5) (2006) 285–287.
[7] X. Zhang, S. Wang, Efficient steganographic embedding by exploiting modification direction, IEEE Commun. Lett. 10 (11) (2006) 781–783.
[8] Z. Ni, Y.Q. Shi, N. Ansari, W. Su, Reversible data hiding, IEEE Trans. Circuits Syst. Video Technol. 16 (3) (2006) 354–362.
[9] D. Hou, W. Zhang, Z. Zhan, R. Jiang, Y. Yang, N. Yu, Reversible image processing via reversible data hiding, in: 2016 IEEE International Conference on Digital Signal Processing (DSP), IEEE, 2016, pp. 427–431.
[10] F. Peng, X. Li, B. Yang, Adaptive reversible data hiding scheme based on integer transform, Signal Process. 92 (1) (2012) 54–62.
[11] X. Gao, Z. Pan, E. Gao, G. Fan, Reversible data hiding for high dynamic range images using two-dimensional prediction-error histogram of the second time prediction, Signal Process. 173 (2020) 107579.
[12] L. Li, Y. Yao, N. Yu, High-fidelity video reversible data hiding using joint spatial and temporal prediction, Signal Process. 208 (2023) 108970.
[13] Z. Fu, X. Chai, Z. Tang, X. He, Z. Gan, G. Cao, Adaptive embedding combining LBE and IBBE for high-capacity reversible data hiding in encrypted images, Signal Process. 216 (2024) 109299.
[14] C.C. Thien, J.C. Lin, Secret image sharing, Comput. Graph. 26 (5) (2002) 765–770.
[15] T.H. Chen, C.S. Wu, Efficient multi-secret image sharing based on boolean operations, Signal Process. 91 (1) (2011) 90–97.
[16] X. Yan, Y. Lu, L. Liu, X. Song, Reversible image secret sharing, IEEE Trans. Inf. Forensics Secur. 15 (2020) 3848–3858.
[17] L. Xiong, X. Zhong, C.N. Yang, DWT-SISA: a secure and effective discrete wavelet transform-based secret image sharing with authentication, Signal Process. 173 (2020) 107571.
[18] P. Pal, B. Jana, J. Bhaumik, Watermarking scheme using local binary pattern for image authentication and tamper detection through dual image, Secur. Priv. 2 (2) (2019) e59.
[19] A. Dey, P. Pal, P. Chowdhuri, B. Jana, S. Jana, A. Singha, Dual image based watermarking scheme using quorum function, in: Advanced Techniques for IoT Applications: Proceedings of EAIT 2020, Springer Singapore, 2022, pp. 114–123.
[20] C.C. Chang, T.D. Kieu, Y.C. Chou, Reversible data hiding scheme using two steganographic images, in: TENCON 2007-2007 IEEE Region 10 Conference, IEEE, 2007, pp. 1–4.
[21] T.C. Lu, C.Y. Tseng, J.H. Wu, Dual imaging-based reversible hiding technique using LSB matching, Signal Process. 108 (2015) 77–89.
[22] G. Debasis, J. Biswapati, M.S. Kumar, Dual image based reversible data hiding scheme using three pixel value difference expansion, in: Information Systems Design and Intelligent Applications: Proceedings of Third International Conference INDIA 2016, Springer India 2, 2016, pp. 403–412.
[23] B. Jana, Dual image based reversible data hiding scheme using weighted matrix, Int. J. Electron. Inf. Eng. 5 (1) (2016) 6–19.
[24] B. Jana, D. Giri, S. Kumar Mondal, Dual image based reversible data hiding scheme using (7, 4) hamming code, Multimed. Tools Appl. 77 (2018) 763–785.
[25] A.K. Sahu, G. Swain, Dual stego-imaging based reversible data hiding using improved LSB matching, Int. J. Intell. Eng. Syst. 12 (5) (2019) 63–73.
[26] H.W. Tseng, H.S. Leng, A reversible modified least significant bit (LSB) matching revisited method, Signal Process. 101 (2022) 116556.
[27] S. Midya, P. Chowdhuri, P. Pal, P.K. Singh, B. Jana, A dual image based data hiding scheme based on difference of pixel values in integer wavelet transform domain, in: International Conference on Computational Intelligence in Communications and Business Analytics, Cham: Springer International Publishing, 2022, pp. 76–84.
[28] J.C. Liu, C.C. Chang, Y. Lin, C.C. Chang, J.H. Horng, A matrix coding-oriented reversible data hiding scheme using dual digital images, Mathematics 12 (1) (2023) 86.
[29] C. Kim, L. Cavazos Quero, K.H. Jung, L. Leng, Advanced dual reversible data hiding: a focus on modification direction and enhanced least significant bit (LSB) approaches, Appl. Sci. 14 (6) (2024) 2437.
[30] Xinpeng Zhang, Shuozhong Wang, Efficient steganographic embedding by exploiting modification direction, IEEE Commun. Lett. 10 (11) (2006) 781–783.
[31] H.J. Kim, C. Kim, Y. Choi, S. Wang, X. Zhang, Improved modification direction methods, Comput. Math. Appl. 60 (2) (2010) 319–325.
[32] C.C. Chang, Y. Liu, T.S. Nguyen, A novel turtle shell based scheme for data hiding, in: 2014 tenth international conference on intelligent information hiding and multimedia signal processing, IEEE, 2014, pp. 89–93.
[33] Y. Lin, C.C. Lin, J.C. Liu, C.C. Chang, Verifiable (t, n) secret image sharing scheme based on slim turtle shell matrix, J. Inf. Secur. Appl. 80 (2024) 103679.