



Article

Position-Aware Guided Hiding Data Scheme with Reversibility and Adaptivity for Dual Images

Chin-Chen Chang ^{1,*} , Guo-Dong Su ^{1,2}, Chia-Chen Lin ^{3,*} and Yung-Hui Li ⁴ 

¹ Department of Information Engineering and Computer Science, Feng Chia University, Taichung 40724, Taiwan; gdsu0206@gmail.com

² School of Big Data and Artificial Intelligence, Fujian Polytechnic Normal University, Fuzhou 350300, China

³ Department of Computer Science and Information Engineering, National Chin-Yi University of Technology, Taichung 41170, Taiwan

⁴ AI Research Center, Hon Hai Research Institute, Taipei 114699, Taiwan; yunghui.li@foxconn.com

* Correspondence: alan3c@gmail.com (C.-C.C.); ally.cclin@ncut.edu.tw (C.-C.L.)

Abstract: Reversible data hiding (RDH) in dual images is a technique that shares secret messages into two similar shadow images, while the secret messages and the cover image can be restored only when those two shadows are gathered simultaneously. In this paper, a novel turtle shell-based RDH hiding scheme based on the symmetric property is presented in order to increase the embedding capacity and maintain good visual quality in dual images under the guidance of position-aware. First, we classify each pixel pair into one of four types according to their locations and then determine a sunflower area centered around it in order to construct the combination of positions and the embedding table. Using the embedding table, the secret messages are concealed into a cover image by generating two shadow images. At the decoder's side, the complete restoration of the secret messages and the cover image can be accomplished by identifying the position relationship between the two stego-pixel pairs. The experimental results confirmed that the proposed position-aware guided RDH scheme is superior to some of the relevant works on the aspects of embedding capacity or image quality. In addition, the proposed scheme provides a secure communication that can effectively resist attacks on the pixel value difference histogram, relative entropy, and regular singular analysis.

Keywords: position-aware; reversible data hiding; position combination; turtle shell; embedding capacity



Citation: Chang, C.-C.; Su, G.-D.; Lin, C.-C.; Li, Y.-H. Position-Aware Guided Hiding Data Scheme with Reversibility and Adaptivity for Dual Images. *Symmetry* **2022**, *14*, 509. <https://doi.org/10.3390/sym14030509>

Academic Editor: Dumitru Baleanu

Received: 6 December 2021

Accepted: 23 February 2022

Published: 2 March 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the development of computer science and tele-communication, the acquisition and sharing of messages have become easy and fast. However, due to the openness and the transparent nature of the Internet, the messages that users transmit can be stolen easily by criminals. This real-world issue motivated this case study of secure covert communication. In order to address this issue, a scheme using data hiding (DH) [1–3], also named information hiding, to imperceptibly transmit secret messages has been explored over the past decades. Those DH algorithms generally focus on carrying a larger embedding capacity, but do not consider its reversibility. Simultaneously, reversible data hiding (RDH) [4–6], as another branch of DH, is used extensively in many applications because of its reversibility, such as the secure transmission of military images or medical images.

RDH is one technique that can be used to embed secret messages into a cover image, and, at the decoder's side, the cover image and the secret messages can be restored losslessly. Without loss of generality, RDH schemes can be grouped roughly into five classes, i.e., (1) difference expansion [7,8], (2) prediction-error expansion [9,10], (3) histogram shifting [11,12], (4) pixel-value ordering [13,14], and (5) machine learning-based steganography [15,16]. One of the key points of these RDH schemes is how to maintain the

visual quality of stego-images, even at the expense of the partial amount of the embedded secret messages.

In recent years, RDH schemes in dual images have attracted a lot of attention because they can be used to share secret messages [17–25]. These RDH schemes insert the secret messages into a cover image so that two similar shadow images are generated. Subsequently, the lossless restoration of the secret messages and the cover image only can be accomplished when the decoder has access to those two shadow images. According to our best knowledge, the first RDH scheme in two shadow images was pioneered by Chang et al. [17] in 2007. In this scheme, two 5-ary digits are carried by two shadow images according to the exploiting modification direction (EMD)-based reference matrix. The embedding capacity (EC) of their scheme is approximately 1.0 bpp, and the peak signal-to-noise ratio (*PSNR*) of the shadow images reaches 45.33 dB, on average. In 2009, Lee et al.'s scheme [18] exploited four directions of each pixel pair in order to represent two 4-ary secret digits. In order to ensure reversibility, the orientation relationship between the pixel pairs of the two shadow images is used to determine whether or not the second 4-ary secret digit could be concealed in the latter pixel pair. Although the average *PSNR* of the shadow images is up to 52.30 dB, their *ER* is no more than 0.75 bpp. Considering the disadvantages of the schemes [17,18], Lee and Huang's scheme [19] increases the *ER* up to 1.04 bpp, while maintaining a better image quality through the joint use of improved orientation combinations and the upgraded 5-ary numeral system.

In 2018, with the guidance of the turtle shell-based DH scheme published by Chang et al. [3], Liu and Chang [20] proposed a novel RDH scheme in two shadow images where the variable secret digit is carried using a pixel pair that is duplicated from a pixel of the cover image. Concretely, an 8-ary secret digit will be concealed when the pixel pair is identified to be a back element; conversely, for a pixel pair that belongs to the edge element, only a 2-ary secret digit will be carried. In 2019, Lin et al. [22] observed that there was room for improvement in the design of the turtle shell-based reference matrix, and their scheme succeeded in embedding a 2-ary secret digit into the pixel pair of the first shadow image and a 16-ary secret digit into the pixel pair of the second shadow image. On average, this scheme provides an *ER* of up to 1.25 bpp, and its shadow images have considerable quality, with *PSNR*s of 49.38 dB and 45.55 dB, respectively. In order to increase the *ER* further, Xie et al. [23] used a newly-designed, turtle shell-based reference matrix to implement a high-capacity RDH scheme. Their scheme classifies a pixel pair duplicated from a pixel into an upper type or a lower type. According to the type of the pixel pair, a specific four \times five block is selected, and this pixel pair is used as the bottom-right corner or the top-left corner so that each pixel pair can carry a 16-ary secret digit, thus, an *ER* of 2.0 bpp is obtained. However, in some cases, a large modification of the shadow images must be performed in order to avoid the problem of the location conflict. Thus, the visual quality of shadow images does not yield complete satisfaction, i.e., they have an average *PSNR* of 40.80 dB. In order to achieve good visual quality, Chen and Guo [24] presented a new RDH scheme based on fully exploiting the combination of the orientations of each pixel pair in the two shadow images. They took a pixel pair as a pivot point and drew a three \times three block around it. Within this block, there were a total of 25 combinations of orientations that could uniquely determine the center pixel pair. They labelled those 25 orientation combinations from 0 to 24, with the assistance of a mark matrix. Thus, each orientation combination can be used to represent a 25-ary secret digit. As a result, the *ER* of scheme [24] is approximately 1.14 bpp, and the *PSNR* is around 49.92 dB. In 2021, Chen and Hong [25] exploited an improved EMD in order to implement the design of an RDH scheme for dual images without any overhead message. By this way, this scheme not only provides the better performance but also can resist some attacks.

Among those, Chen and Guo [24] presented a new RDH scheme based on fully exploiting the combination of the orientations and obtained an average *ER* of 1.14 bpp. In order to pursue a larger *ER* while keeping the amount of distortion as low as possible, in this paper, we propose a position-aware guided RDH scheme in dual images with the

combination usage of the symmetric property and turtle shell. In our algorithm, first, a pixel pair is identified into four types, i.e., upper back element (UBE), lower back element (LBE), upper edge element (UEE), and lower edge element (LEE). Concerning one of the types, a specific sunflower flower area, consisting of seven (or four) neighbor turtle shells, is selected in order to create a series of position combinations, which make positive contributions to increasing the *ER*. Using the position combinations, we adaptively constructed an embedding table (ET) according to their respective loss values. While constructing the ET, if any pixel value inside a position combination exceeds the range [0, 255], it will be abandoned. This mechanism ensures that the problem of overflow will be avoided, so there is no overhead information required in this paper. Finally, according to the ET, each position combination represents the embedding of a secret digit to generate the shadow images. Extensive experimental results have proved that the proposed position-aware guided RDH scheme achieves a considerable *ER*, i.e., up to 1.25 bpp, while maintaining the good visual quality of the shadow images. In addition, three attacks, i.e., pixel value difference histogram, relative entropy, and regular singular analysis, were used in order to evaluate whether or not the proposed scheme can provide a secure, covert communication.

The rest of this paper is organized as follows. Section 2 reviews the related works. Section 3 introduces the proposed scheme. Experiments are given in Section 4, and Section 5 presents our conclusions.

2. Related Works

This section briefly states the turtle shell-based reference matrix [4], which is essential for the proposed scheme. The RDH scheme, which is a related, dual images scheme presented by Chen and Guo [24], is also reviewed in this section.

2.1. Turtle Shell-Based Reference Matrix

Over the past decade or so, a series of DH or RDH schemes were published that applied the idea of the turtle shell, and Chang et al.’s scheme [1] is a representative among those schemes. In Chang et al.’s work, a 256×256 reference matrix, i.e., where *R*, is constructed in advance according to the specific rule, as shown in Figure 1. It means that the reference matrix *R* is easy to regenerate, so it is not necessary to send it to the recipient.

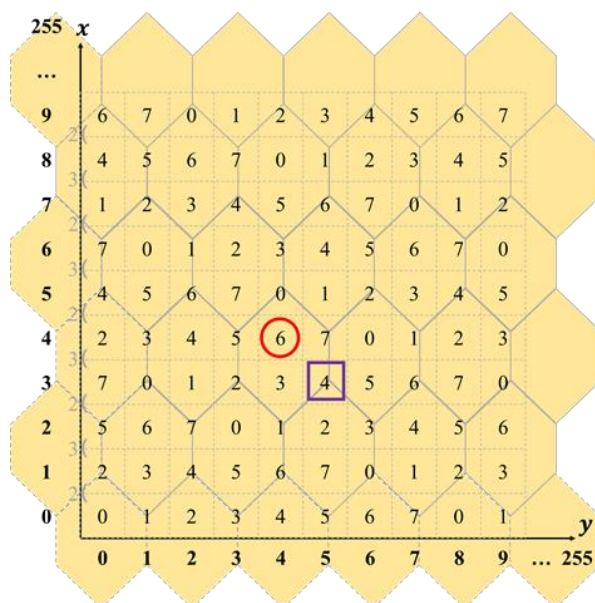


Figure 1. Part of the reference matrix.

Figure 1 shows that the reference matrix *R* is decomposed into a series of non-overlapping turtle shells. Each turtle shell covers eight distinct digits, ranging from 0

to 7. In view of the location of those digits, we can classify them into four types, i.e., UBE, LBE, UEE, and LEE. Without loss of generality, given a digit located at (x,y) in R , i.e., $R(x,y)$, its type can be identified as follows:

$$\text{Type is } \begin{cases} \text{UBE if } (y \bmod 2 = 1 \ \& \ x \bmod 4 = 2) \parallel (y \bmod 2 = 0 \ \& \ x \bmod 4 = 0) \\ \text{LBE if } (y \bmod 2 = 1 \ \& \ x \bmod 4 = 1) \parallel (y \bmod 2 = 0 \ \& \ x \bmod 4 = 3) \\ \text{UEE if } (y \bmod 2 = 1 \ \& \ x \bmod 4 = 0) \parallel (y \bmod 2 = 0 \ \& \ x \bmod 4 = 2) \\ \text{LEE if } (y \bmod 2 = 1 \ \& \ x \bmod 4 = 3) \parallel (y \bmod 2 = 0 \ \& \ x \bmod 4 = 1) \end{cases} \quad (1)$$

For example, $R(4,4)$ belongs to UBE, as shown by the red circle in Figure 1, and $R(3,5)$ belongs to LEE, as shown by the purple quadrangle in Figure 1.

2.2. Review of Chen and Guo’s Scheme

In 2020, Chen and Guo [24] proposed a dual-image-based RDH scheme in which each pixel pair (x,y) is processed to carry a 25-ary secret digit and produces two stego-pixel pairs, i.e., a primary pixel pair (x_p,y_p) and a foreign pixel pair (x_f,y_f) . In order to increase the embedding capacity and achieve a relatively high visual quality, this scheme fully exploits the combinations of pixel pair orientations.

Given a cover pixel pair (x,y) , they take the (x,y) as a pivot point and draw a three \times three block, as shown in Figure 2a. They mark each location inside of the block to clarify their priority, as shown in Figure 2b. For reversibility, 25 combinations of orientations are utilized, and they can uniquely determine the center pixel pair (x,y) . Subsequently, those 25 combinations of orientations are ordered by their marks to adaptively construct the embedding table, as shown in Table 1. The combination of orientations that has a smaller primary mark is ordered ahead. If two of the combinations of orientations have the same primary marks, the one that has the smaller foreign mark has priority to be ordered ahead. Table 1 presents all of the 25 cases, and each one occupies a row, i.e., $(t, b^t, (x_p^t, y_p^t), (x_f^t, y_f^t))$, which means to alter the cover pixel pair (x,y) to the stego-primary-pixel pairs (x_p^t, y_p^t) and the stego-foreign-pixel pair (x_f^t, y_f^t) and to embed a 25-ary secret digit, t , the binary representation of which is b^t .

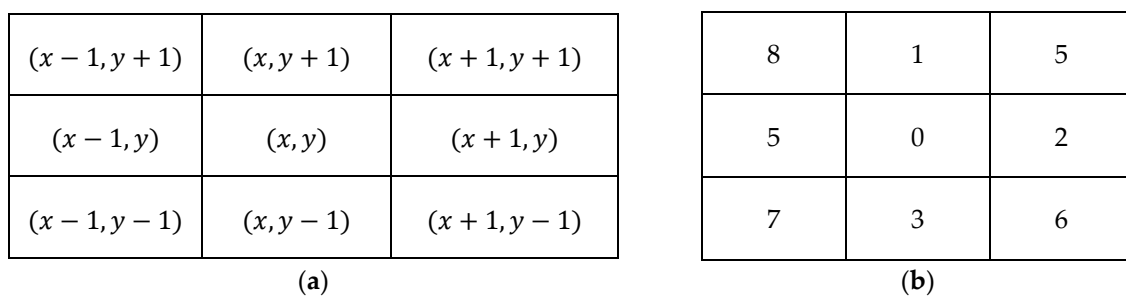


Figure 2. Definition of a 3×3 block. (a) The value of each location in a 3×3 block, (b) mark matrix.

Table 1. Embedding rules of Chen and Guo’s scheme [24].

t	b^t	(x_p^t, y_p^t)	Primary Mark	(x_f^t, y_f^t)	Foreign Mark
0	00000			(x, y)	0
1	00001			$(x, y + 1)$	1
2	00010			$(x + 1, y)$	2
3	00011			$(x, y - 1)$	3
4	00100	(x, y)	0	$(x - 1, y)$	4
5	00101			$(x + 1, y + 1)$	5
6	00110			$(x + 1, y - 1)$	6
7	00111			$(x - 1, y - 1)$	7
8	01000			$(x - 1, y + 1)$	8

Table 1. Cont.

t	b^t	(x_p^t, y_p^t)	Primary Mark	(x_f^t, y_f^t)	Foreign Mark
9	01001			$(x, y - 1)$	3
10	01010	$(x, y + 1)$	1	$(x + 1, y - 1)$	6
11	01011			$(x - 1, y - 1)$	7
12	01100			$(x - 1, y)$	4
13	01101	$(x + 1, y)$	2	$(x - 1, y - 1)$	7
14	01110			$(x - 1, y + 1)$	8
15	01111			$(x - 1, y + 1)$	1
16	10000	$(x, y - 1)$	3	$(x + 1, y + 1)$	5
17	10001			$(x, y + 1)$	8
18	1001			$(x + 1, y)$	2
19	1010	$(x - 1, y)$	4	$(x + 1, y - 1)$	5
20	1011			$(x + 1, y + 1)$	6
21	1100	$(x + 1, y + 1)$	5	$(x - 1, y - 1)$	7
22	1101	$(x + 1, y - 1)$	6	$(x - 1, y + 1)$	8
23	1110	$(x - 1, y - 1)$	7	$(x + 1, y + 1)$	5
24	1111	$(x - 1, y + 1)$	8	$(x + 1, y - 1)$	6

3. Proposed Scheme

In this section, we describe a position-aware guided RDH scheme based on turtle shell to insert the secret messages into a cover image by generating two shadow images. To begin with, the description of the position combinations in a sunflower area is given in Section 3.1. Then, the details of the shadow construction are introduced in Section 3.2. Section 3.3 illustrates the extraction of the secret messages and the recovery of the cover image. The overall flowchart of the proposed scheme is shown in Figure 3.

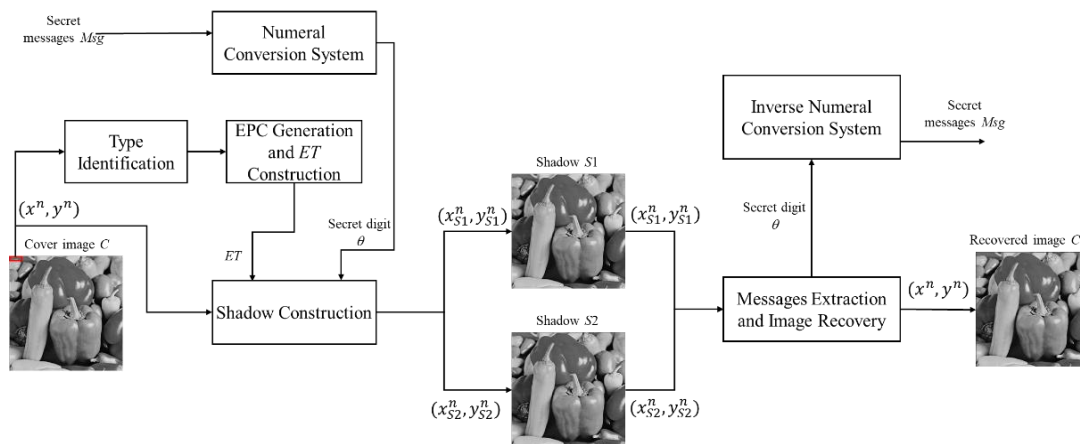


Figure 3. The overall flowchart of the proposed scheme.

3.1. Position Combinations in a Sunflower Area

In a pixel-by-pixel manner, each two adjacent pixels, i.e., x and y , selected from the cover image, are grouped into a pixel pair (x, y) . Then, we take the x as the index of the x -axis and y as the index of the y -axis and project this information in the reference matrix R , thereby locating at $R(x, y)$. According to the location of $R(x, y)$, the sunflower area is determined as shown in the following four cases below:

Case 1: $R(x, y)$ belongs to UBE. The sunflower area consists of seven turtle shells, and the value of the location of each element is shown in Figure 4a.

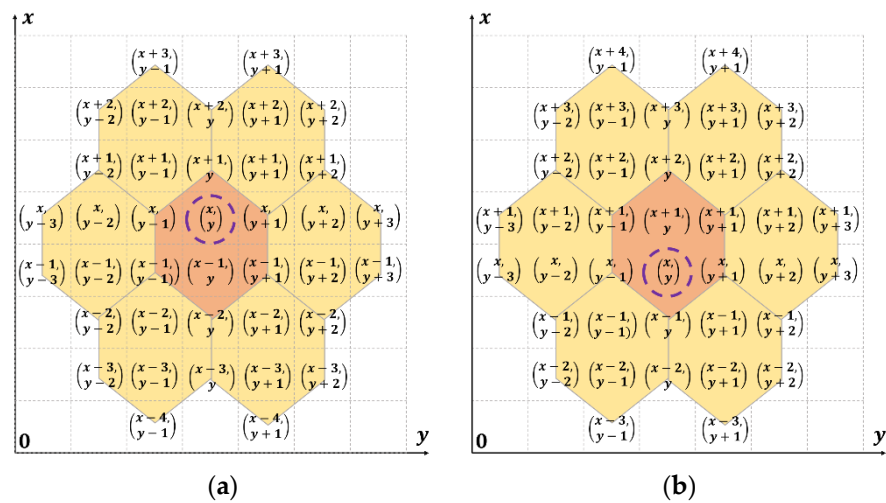


Figure 4. The value of each element’s location in the sunflower area for (a) UBE, (b) LBE.

Case 2: $R(x, y)$ belongs to LBE. The sunflower area consists of seven turtle shells, and the value of the location of each element is shown in Figure 4b.

Case 3: $R(x, y)$ belongs to UEE. The sunflower area consists of four turtle shells, and the value of the location of each element is shown in Figure 5a.

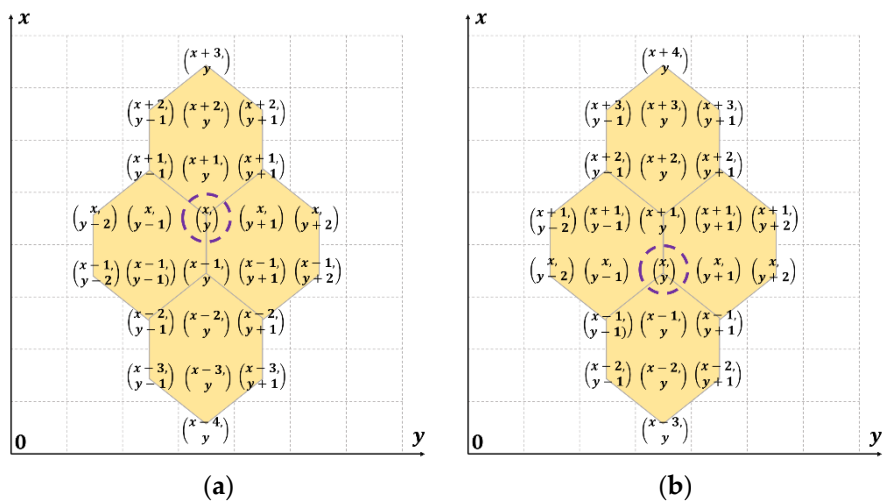


Figure 5. The value of each element’s location in the sunflower area for (a) UEE, (b) LEE.

Case 4: $R(x, y)$ belongs to LEE. The sunflower area consists of four turtle shells, and the value of the location of each element is shown in Figure 5b.

After determining the sunflower area, we can embed a secret digit into the pixel pair in order to obtain two stego-pixel pairs, i.e., the primary pixel pair in shadow S1 and the foreign pixel pair in shadow S2. Concretely, to reversibility, the effective position combinations (EPCs) of the primary and foreign pixel pairs are exploited and can be generated by the rules that are described in Algorithm 1. To the best of our knowledge, there are 62 combinations of positions for Case 1 or Case 2 and 16 combinations of positions for Case 3 or Case 4 at most that can uniquely determine the cover pixel pair (x, y) . It is noted that, while generating the EPCs, any single pixel value inside a position combination that exceeds the range $[0, 255]$ will be abandoned. Thus, it is certain that there will be a smaller number of combinations of positions in the corresponding EPC for the pixel pairs that are located in the boundary area. The advantage is that this mechanism ensures that the problem of overflow will be avoided, so no overhead information is required. It also

implies that it is not required for the *ET*, that is derived adaptively from EPC, to be sent to the decoder.

Algorithm 1: Generation of effective position combinations for (x, y)

Input: $(x, y), R$
Output: Effective position combinations EPC

```

1  EPC = ∅
2  If  $R(x, y)$  belongs to UBE/LBE:
3      Traverse each element  $R(x', y')$  in the sunflower area that corresponds to  $R(x, y)$ .
4      If  $R(x', y')$  is inside R and belongs to UBE/LBE:
5          Add the position combination  $((x, y), (x', y'))$  into EPC.
6      Else, if  $R(x', y')$  is inside R and belongs to UEE/LEE:
7          Add the position combinations  $((x, y), (x', y'))$  and  $((x', y'), (x, y))$  into EPC.
8      End if
9  Else if  $R(x, y)$  belongs to UEE/LEE
10     Traverse each element  $R(x', y')$  in the sunflower area that corresponds to  $R(x, y)$ .
11     If  $R(x', y')$  is inside R and belongs to UEE/LEE
12         Add the position combination  $((x, y), (x', y'))$  into EPC
13     End if
14 End if
    
```

After generating EPC, *ET* can be constructed. For ease of illustration, we assume that $EPC = \{(x_p^t, y_p^t), (x_f^t, y_f^t)\}$, where $0 \leq t \leq T - 1$, and T is the number of combinations of positions in EPC. First, all of the position combinations in EPC are sorted in the ascending order of the loss value, l^t . Among which, l^t can be represented as follows:

$$l^t = (x_p^t - x)^2 + (y_p^t - y)^2 + (x_f^t - x)^2 + (y_f^t - y)^2, \tag{2}$$

Then, we label those ordered position combinations arranging from 0 to $T - 1$. Tables 2 and 3 show the detailed *ET*s for Case 1 and Case 3, respectively. Note that the *ET* of Case 2 (or Case 4) is similar to that of Case 1 (or Case 3).

Table 2. Embedding table for Case 1 (UBE).

d^t	$(x_p^t, y_p^t), (x_f^t, y_f^t)$	l^t	d^t	$(x_p^t, y_p^t), (x_f^t, y_f^t)$	l^t	d^t	$(x_p^t, y_p^t), (x_f^t, y_f^t)$	l^t
0	$(x, y), (x, y)$	0	21	$(x + 1, y - 2), (x, y)$	5	42	$(x, y), (x - 3, y)$	9
1	$(x, y), (x, y - 1)$	1	22	$(x, y), (x + 1, y + 2)$	5	43	$(x - 3, y), (x, y)$	9
2	$(x, y - 1), (x, y)$	1	23	$(x + 1, y + 2), (x, y)$	5	44	$(x, y), (x + 3, y - 1)$	10
3	$(x, y), (x, y + 1)$	1	24	$(x, y), (x + 2, y - 1)$	5	45	$(x + 3, y - 1), (x, y)$	10
4	$(x, y + 1), (x, y)$	1	25	$(x, y), (x + 2, y + 1)$	5	46	$(x, y), (x + 3, y + 1)$	10
5	$(x, y), (x + 1, y)$	1	26	$(x, y), (x - 1, y - 2)$	5	47	$(x + 3, y + 1), (x, y)$	10
6	$(x + 1, y), (x, y)$	1	27	$(x, y), (x - 1, y + 2)$	5	48	$(x, y), (x - 1, y - 3)$	10
7	$(x, y), (x - 1, y)$	1	28	$(x, y), (x - 2, y - 1)$	5	49	$(x - 1, y - 3), (x, y)$	10
8	$(x, y), (x + 1, y - 1)$	2	29	$(x, y), (x - 2, y + 1)$	5	50	$(x, y), (x - 1, y + 3)$	10
9	$(x, y), (x + 1, y + 1)$	2	30	$(x, y), (x + 2, y - 2)$	8	51	$(x - 1, y + 3), (x, y)$	10
10	$(x, y), (x - 1, y - 1)$	2	31	$(x + 2, y - 2), (x, y)$	8	52	$(x, y), (x - 3, y - 1)$	10
11	$(x - 1, y - 1), (x, y)$	2	32	$(x, y), (x + 2, y + 2)$	8	53	$(x, y), (x - 3, y + 1)$	10
12	$(x, y), (x - 1, y + 1)$	2	33	$(x + 2, y + 2), (x, y)$	8	54	$(x, y), (x - 3, y - 2)$	13
13	$(x - 1, y + 1), (x, y)$	2	34	$(x, y), (x - 2, y - 2)$	8	55	$(x - 3, y - 2), (x, y)$	13
14	$(x, y), (x + 2, y)$	4	35	$(x - 2, y - 2), (x, y)$	8	56	$(x, y), (x - 3, y + 2)$	13
15	$(x + 2, y), (x, y)$	4	36	$(x, y), (x - 2, y + 2)$	8	57	$(x - 3, y + 2), (x, y)$	13
16	$(x, y), (x, y - 2)$	4	37	$(x - 2, y + 2), (x, y)$	8	58	$(x, y), (x - 4, y - 1)$	17
17	$(x, y), (x, y + 2)$	4	38	$(x, y), (x, y - 3)$	9	59	$(x - 4, y - 1), (x, y)$	17

Table 2. Cont.

d^t	$(x_p^t, y_p^t), (x_f^t, y_f^t)$	l^t	d^t	$(x_p^t, y_p^t), (x_f^t, y_f^t)$	l^t	d^t	$(x_p^t, y_p^t), (x_f^t, y_f^t)$	l^t
18	$(x, y), (x - 2, y)$	4	39	$(x, y - 3), (x, y)$	9	60	$(x, y), (x - 4, y + 1)$	17
19	$(x - 2, y), (x, y)$	4	40	$(x, y), (x, y + 3)$	9	61	$(x - 4, y + 1), (x, y)$	17
20	$(x, y), (x + 1, y - 2)$	5	41	$(x, y + 3), (x, y)$				

d^t : The t -th to-be-embedded secret digit; l^t : The loss value if d^t is embedded; (x_p^t, y_p^t) : The stego-primary-pixel pair after embedding d^t ; (x_f^t, y_f^t) : The stego-foreign-pixel pair after embedding d^t .

Table 3. Embedding table for Case 3 (UEE).

d^t	$(x_p^t, y_p^t), (x_f^t, y_f^t)$	l^t	d^t	$(x_p^t, y_p^t), (x_f^t, y_f^t)$	l^t
0	$(x, y), (x, y)$	0	8	$(x, y), (x + 2, y - 1)$	5
1	$(x, y), (x - 1, y)$	1	9	$(x, y), (x - 2, y + 1)$	5
2	$(x, y), (x + 1, y - 1)$	2	10	$(x, y), (x - 1, y + 2)$	5
3	$(x, y), (x + 1, y + 1)$	2	11	$(x, y), (x + 2, y + 1)$	5
4	$(x, y), (x, y - 2)$	4	12	$(x, y), (x + 3, y)$	9
5	$(x, y), (x, y + 2)$	4	13	$(x, y), (x - 3, y + 1)$	10
6	$(x, y), (x - 1, y - 2)$	5	14	$(x, y), (x - 3, y - 1)$	10
7	$(x, y), (x - 2, y - 1)$	5	15	$(x, y), (x - 4, y)$	16

Take Table 2 for example. For Case 1 (UBE), there is a total of 62 position combinations, and each occupies a row of a quaternion sequence, $(d^t, (x_p^t, x_p^t), (x_f^t, x_f^t), l^t)$, which indicates to modify the cover pixel pair (x, y) to the dual stego-pixel pairs, (x_p^t, x_p^t) and (x_f^t, x_f^t) , to carry a 62-ary secret digit, d^t , under a total loss l^t . Taking a specific case as an example, suppose that the cover pixel pair is (176, 70), which belongs to UBE, and it needs to carry a secret digit, i.e., 11. First, it is easy to find the row in Table 2, where $d^t = 11$. Next, the pixel pair (176, 70) is modified to (175, 69) in the shadow S1, and it is (176, 70) in shadow S2. Finally, the secret digit, 11, is carried.

3.2. Shadow Construction

Assume that we want to embed the secret messages, i.e., Msg , into a grayscale cover image, C , with a size of $H \times W$ in order to generate two shadow images, S1 and S2. The detailed construction of the shadow images is described in Algorithm 2.

Algorithm 2: Construction of the shadow images

Input: C, Msg, R

Output: S1, S2

- 1 Separate C into a set of non-overlapping pixel pairs in order from top to bottom, left to right, and denoted as $C = \{(x^n, y^n)\}$, where $1 \leq n \leq N$ and $N = \frac{H \cdot W}{2}$.
- 2 Read an unvisited pixel pair (x^n, y^n) from C and project it into the reference matrix R , i.e., $R(x^n, y^n)$.
- 3 Identify $R(x^n, y^n)$'s type into one of {UBE, LBE, UEE, LEE} according to Equation (1).
- 4 Use Algorithm 1 to generate EPC for $R(x^n, y^n)$ and then construct the corresponding ET .
- 5 Convert Msg into a T -ary numeral system to derive a T -ary secret digit, θ .
- 6 Use ET to embed θ into (x^n, y^n) :
 - 6.1 Find the quaternion sequence $(d^t, (x_p^t, y_p^t), (x_f^t, y_f^t), l^t)$ in ET where $t = \theta$.
 - 6.2 Modify two pixel pairs, i.e., $(x_{S1}^n, y_{S1}^n) = (x_p^t, y_p^t)$ in shadow S1 and $(x_{S2}^n, y_{S2}^n) = (x_f^t, y_f^t)$ in shadow S2.
- 7 Repeat Steps 2 through 7 until all cover pixel pairs and secret message Msg have been dealt with.
- 8 Output two shadow images, i.e., S1 and S2.

After the two shadow images $S1$ and $S2$ have been generated, we send shadow image $S1$ to one recipient and send shadow image $S2$ to another recipient.

3.3. Extraction of Secret Messages and Recovery of the Cover Image

Retrieving the secret messages, Msg , and restoring the cover image, C , can be accomplished only if both of the recipients release their own shadow image. The details of extracting secret messages, Msg , and restoring image C , are described in Algorithm 3.

Algorithm 3: Extracting secret messages, Msg , and restoring the cover image, C

Input: $S1, S2, R$

Output: Msg, C

- 1 Separate shadow images $S1$ and $S2$ into a set of non-overlapping pixel pairs in order from top to bottom and from left to right, respectively. To ease the discussion, denote $S1$ as $\{(x_{S1}^n, y_{S1}^n)\}$ and $S2$ as $\{(x_{S2}^n, y_{S2}^n)\}$, where $1 \leq n \leq N$ and $N = \frac{H \cdot W}{2}$.
 - 2 Read a couple of pixel pairs (x_{S1}^n, y_{S1}^n) and (x_{S2}^n, y_{S2}^n) and restore the original cover pixel pair (x^n, y^n) as shown below:
 - 2.1 Project (x_{S1}^n, y_{S1}^n) and (x_{S2}^n, y_{S2}^n) into R , i.e., $R(x_{S1}^n, y_{S1}^n)$ and $R(x_{S2}^n, y_{S2}^n)$.
 - 2.2 If $R(x_{S1}^n, y_{S1}^n)$ belongs to UBE/LBE, set $(x^n, y^n) = (x_{S1}^n, y_{S1}^n)$.
 - 2.3 If $R(x_{S1}^n, y_{S1}^n)$ belongs to UEE/LEE, then:
 - 2.3.1 If $R(x_{S2}^n, y_{S2}^n)$ belongs to UBE/LBE, set $(x^n, y^n) = (x_{S2}^n, y_{S2}^n)$;
 - 2.3.2 Else set $(x^n, y^n) = (x_{S1}^n, y_{S1}^n)$.
 - 3 Project (x^n, y^n) into R , i.e., $R(x^n, y^n)$.
 - 4 Identify the type of $R(x^n, y^n)$ as one of {UBE, LBE, UEE, LEE} according to Equation (1).
 - 5 Use Algorithm 1 to generate EPC for $R(x^n, y^n)$ and construct the corresponding ET .
 - 6 Use ET to extract a secret digit, θ , from (x_{S1}^n, y_{S1}^n) and (x_{S2}^n, y_{S2}^n) :
 - 6.1 Find the quaternion sequence $(d^t, (x_p^t, x_p^t), (x_f^t, x_f^t), l^t)$ in ET to meet

$$(x_p^t, x_p^t) = (x_{S1}^n, x_{S1}^n) \text{ and } (x_f^t, x_f^t) = (x_{S2}^n, x_{S2}^n).$$
 - 6.2 Extract $\theta = d^t$.
 - 7 Convert θ in a T -ary inverse numeral system to derive the sequence of the binary codes and concatenate it into Msg .
 - 8 Repeat Steps 2 through 8 until all pixel pairs have been processed.
 - 9 Output Msg and restore cover image C .
-

4. Experiments

In order to demonstrate the effectiveness of our approach, in this section we present and discuss the simulation results of the proposed scheme and other dual-image-based RDH schemes [17,19,22–24]. All of the experiments were performed under MATLAB R2019b on a personal computer with an Intel® Core™ i5-1035G1 CPU @1.00GHz 1.19GHz, 16 GB RAM, and a Windows 10 operating system. In the following experiments, eight commonly used grayscale images with sizes of 512×512 (i.e., Lena, Airplane, Peppers, Baboon, Goldhill, Elaine, Barbara, and Wine) were downloaded from the USC-SIPI image database and used as test images, as listed in Figure 6.

In order to evaluate the secrecy of the hidden data offered by the generated shadow images, the image distortion between the cover image C and the generated shadow image S (i.e., $S1$ or $S2$) was measured using the peak signal-to-noise ratio ($PSNR$) [26], and it can be represented as follows:

$$PSNR = 10 \log_{10} \left(\frac{255^2}{MSE} \right), \quad (3)$$

$$MSE = \frac{1}{H \cdot W} \sum_{i=1}^H \sum_{j=1}^W (C_{i,j} - S_{i,j})^2, \quad (4)$$

where $C_{i,j}$ and $S_{i,j}$ represent the pixel values located at the i th row and j th column in images C and S (i.e., $S1$ or $S2$), respectively. In general, the larger the $PSNR$ value is, the better the quality of the shadow image will be.

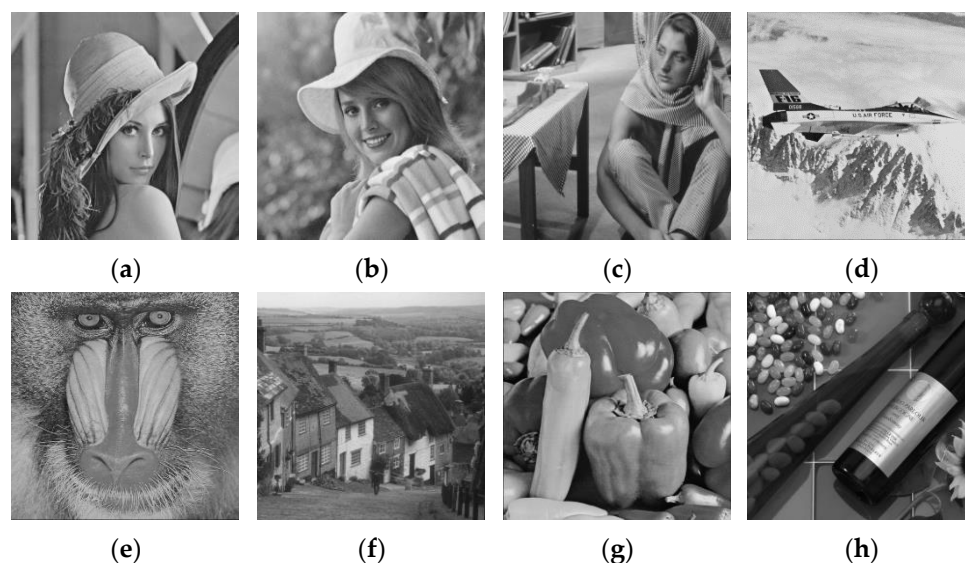


Figure 6. Eight test images: (a) Lena, (b) Elaine, (c) Barbara, (d) Airplane, (e) Baboon, (f) Goldhill, (g) Peppers, and (h) Wine.

In addition to *PSNR*, *ER* provides an objective metric for the evaluation of the embedding capacity, and *ER* is computed as follows:

$$ER = \frac{L_B}{2 \cdot H \cdot W}, \quad (5)$$

where L_B represents all of the embedded secret messages. In our experiment, both H and W are equal to 512.

4.1. Security Analysis

In order to prove that the proposed scheme can provide a secure, covert communication, security analyses, such as pixel value difference (PVD) histogram [3], relative entropy (RE) analysis, and regular-singular (RS) steganalysis [24], are evaluated experimentally in this section.

4.1.1. PVD Histogram

Usually, a general image is considered to have the feature of local smoothness, so the difference between two neighboring pixels is very likely to be slight. Subsequently, the PVD histogram is created by gathering the frequencies of these differences, and its shape will be a Laplace distribution. If a shadow image is distorted seriously, the shape of its PVD histogram will be very different from that of its original PVD histogram. Figure 6 demonstrates the PVD histograms of four cover images and their corresponding shadows. Figure 7 shows that the shape of each cover PVD histogram is well preserved on the two shadows.

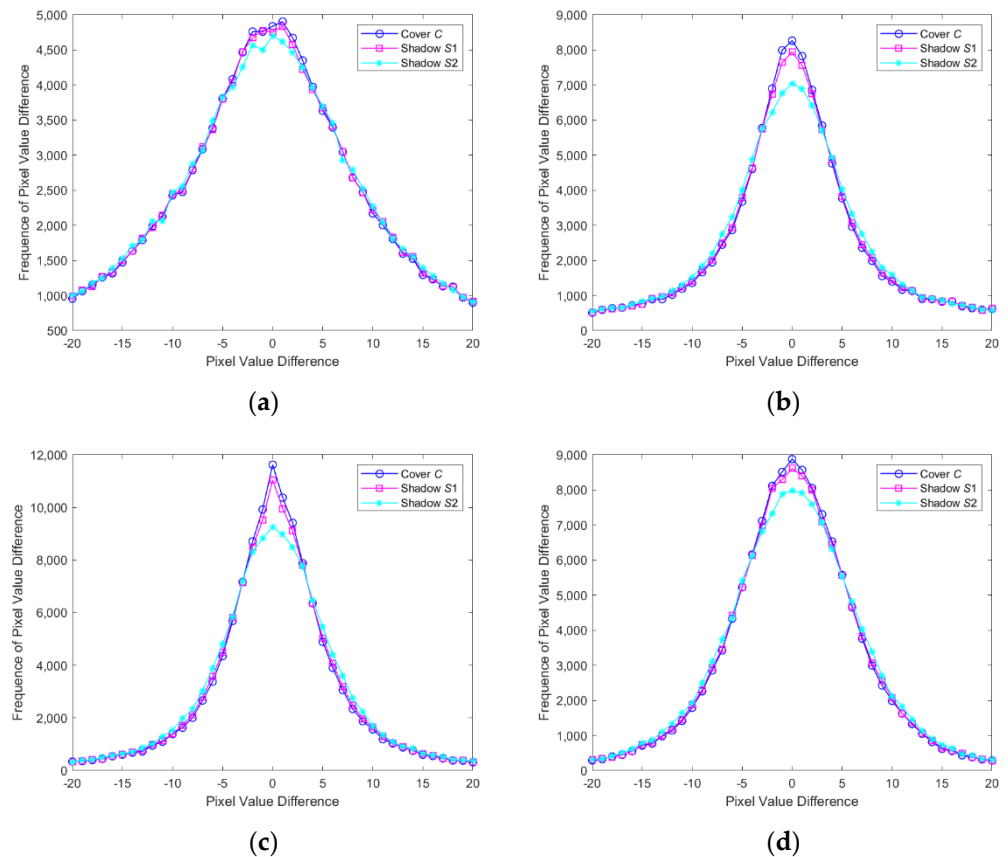


Figure 7. Four PVD histograms of the cover images and the corresponding shadows: (a) Baboon, (b) Barbara, (c) Lena, and (d) Peppers.

4.1.2. Relative Entropy Analysis

Also, we analyzed the performance of the proposed scheme in terms of the relative entropy, which is used to measure the divergence of the shadow image S (i.e., $S1/S2$) from the cover image (C). The relative entropy can be represented mathematically as follows:

$$RE(C, S) = \sum_{z=0}^{255} C(z) \cdot \log\left(\frac{C(z)}{S(z)}\right), \quad (6)$$

where $C(z)$ and $S(z)$ represent the probability distributions of the pixel values for images C and S (i.e., $S1/S2$), respectively. When the value of RE approaches 0, it is considered that these two images are nearly coincident, meaning that the system is more secure. Table 4 elaborates the results of RE among the cover image and the two shadow images and it shows that the entropies of all of the images are very close to having the same values, and the RE s also are near zero. This implies that our approach is quite secure.

Table 4. Entropy and relative entropy among C , $S1$, and $S2$.

Test Images	Entropy			Relative Entropy		
	C	$S1$	$S2$	($C, S1$)	($C, S2$)	($S1, S2$)
Airplane	6.7059	6.7129	6.7286	0.0007	0.0054	0.0039
Baboon	7.1391	7.1404	7.1425	0.0003	0.0008	0.0009
Goldhill	7.4778	7.4811	7.4869	0.0006	0.0033	0.0024
Barbara	7.6321	7.6341	7.6377	0.0002	0.0014	0.0013

Table 4. Cont.

Test Images	Entropy			Relative Entropy		
	C	$S1$	$S2$	$(C, S1)$	$(C, S2)$	$(S1, S2)$
Elaine	7.4980	7.4991	7.5011	0.0003	0.0012	0.0012
Lena	7.4455	7.4477	7.4513	0.0003	0.0011	0.0011
Peppers	7.5944	7.5964	7.6006	0.0002	0.0016	0.0016
Wine	7.4649	7.4681	7.4747	0.0010	0.0038	0.0043
Average	7.3697	7.3725	7.3779	0.0004	0.0023	0.0021

4.1.3. RS Steganalysis

In addition to the PVD histogram and the RE analysis, the RS steganalysis [27,28] is introduced in order to identify whether or not an image carries secret messages by detecting the degree of LSB modifications.

For any two shadows, we divide them into several non-overlapping blocks with four consecutive pixels, and we classify them into three categories, i.e., regular type (R_M or R_{-M}), singular type (S_M or S_{-M}), and the unchanged type (U). Here, M is a mask, and it is defined as $[1\ 0\ 1\ 0]$. R_M and R_{-M} represent the percentages of the regular blocks when the flipping function with M and the shifting function with $-M$ are applied, respectively. Similarly, S_M and S_{-M} represent the percentages of the singular blocks when the flipping function with M and the shifting function with $-M$ are applied, respectively. If an algorithm deployed on an image can defend against RS steganalysis, the relationship derived from the stego-image should meet Equation (7).

$$R_M \cong R_{-M} \text{ and } S_M \cong S_{-M}. \quad (7)$$

Figure 8 depicts the graphs of the RS steganalysis for the four shadow images that were generated from the Lena and Baboon images. Figure 8 indicates that every shadow of the two test images follows Equation (7), indicating that our approach is robust for defending against the RS steganalysis.

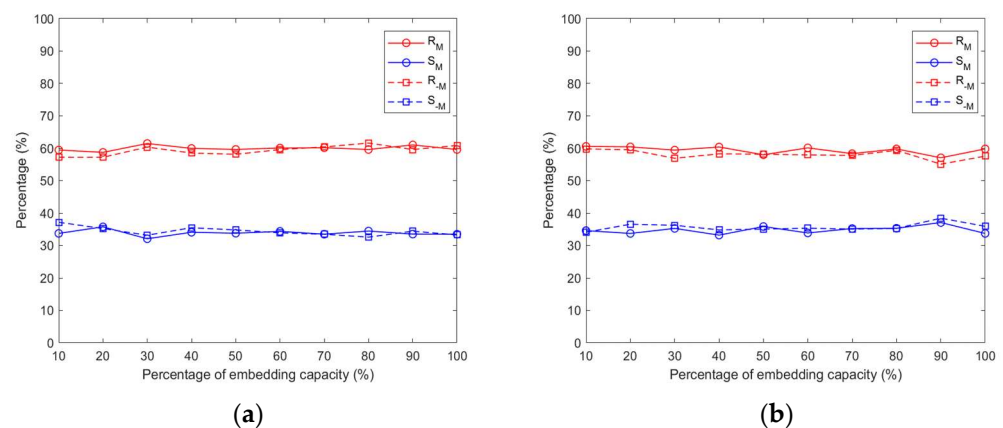


Figure 8. Cont.

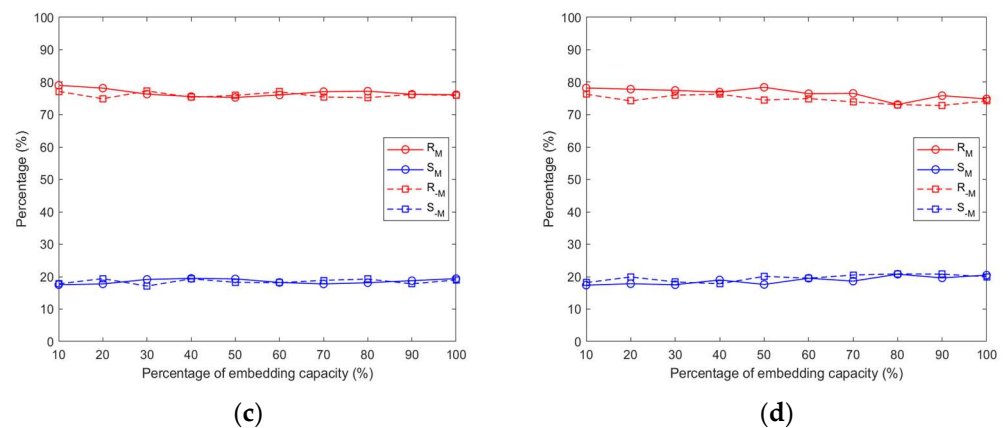


Figure 8. Graphs of RS steganalysis for shadow images: (a) Shadow S1 for Lena, (b) Shadow S2 for Lena, (c) Shadow S1 for Baboon, and (d) Shadow S2 for Baboon.

4.2. Results of the Proposed Scheme

Table 5 lists the experimental results in terms of the ER and $PSNR/SSIM$ values of the two shadow images provided by the proposed scheme. It is not surprising that the $PSNR$ values of the images decrease as ER increases, since more secret messages were carried. Moreover, the $SSIM$ values of each shadow image are higher than 0.9900, indicating the high structure similarity between the cover and shadow images. Also, it can be seen from Table 5 that the proposed scheme achieves a high ER with a value of 1.25 bpp, and the average $PSNR$ values of shadow images S1 and S2 are 49.46 dB and 44.49 dB, respectively. Obviously, the $PSNR$ values provided by our approach are significantly larger than 30 dB, which implies that the human vision system has difficulty identifying the shadow images from the corresponding cover image.

Table 5. $PSNRs/SSIMs$ under different ERs in the proposed scheme.

Test Images	ER	$PSNR/SSIM$ (S1)	$PSNR/SSIM$ (S2)	ER	$PSNR/SSIM$ (S1)	$PSNR/SSIM$ (S2)	ER	$PSNR/SSIM$ (S1)	$PSNR/SSIM$ (S2)
Airplane	0.20	57.38/0.9997	52.47/0.9992	1.00	50.41/0.9989	45.47/0.9965	1.25	49.44/0.9986	44.49/0.9957
Baboon	0.20	57.38/1.0000	52.48/0.9999	1.00	50.42/0.9996	45.46/0.9986	1.25	49.46/0.9994	44.49/0.9983
Goldhill	0.20	57.42/0.9998	52.44/0.9994	1.00	50.39/0.9994	45.47/0.9981	1.25	49.44/0.9992	44.50/0.9976
Barbara	0.20	57.38/0.9999	52.48/0.9996	1.00	50.42/0.9994	45.47/0.9981	1.25	49.45/0.9992	44.50/0.9976
Elaine	0.20	57.42/0.9998	52.44/0.9994	1.00	50.45/0.9991	45.45/0.9972	1.25	49.49/0.9989	44.48/0.9966
Lena	0.20	57.33/0.9997	52.47/0.9991	1.00	50.39/0.9990	45.47/0.9969	1.25	49.43/0.9988	44.50/0.9963
Peppers	0.20	57.47/0.9998	52.45/0.9994	1.00	50.45/0.9990	45.45/0.9970	1.25	49.48/0.9988	44.48/0.9963
Wine	0.20	57.39/0.9999	52.44/0.9995	1.00	50.43/0.9991	45.46/0.9973	1.25	49.47/0.9988	44.50/0.9964
Average	0.20	57.40/0.9998	52.46/0.9994	1.00	50.42/0.9992	45.46/0.9975	1.25	49.46/0.9990	44.49/0.9968

4.3. Comparison and Analysis

In order to further prove the excellent performance of the proposed scheme, we compared the results provided by the proposed scheme with the results provided by some relevant works, including Chang et al.'s scheme [17], Lee and Huang's scheme [19], Lin et al.'s scheme [22], Xie et al.'s scheme [23], Chen and Guo's scheme [24], and Chen and Hong's scheme [25].

First, we compared the maximum ER provided by various schemes [17,19,22–25] and our approach, all of which are RDH schemes in dual images. The results are shown in Table 6, and they show that the average maximum ER provided by the proposed scheme is identical to that of Lin et al.'s scheme [22]. However, it is higher than other three schemes [17,19,24] with differences of 0.25 bpp, 0.21 bpp, and 0.11 bpp, respectively. This means that our approach has the ability to carry more secret messages. Also, Xie et al.'s scheme [23] and Chen and Hong's scheme [25] achieved the greater ERs of 2.0 bpp and 1.56 bpp, respectively, which is better than our approach and Lin et al.'s scheme [22].

However, all of their *PSNR* values of two shadow images were only 40.80 dB and 43.43 dB on average, as shown in Table 7. That is mainly because some pixels have to be shifted into a larger area in order to avoid the location conflict problem during the embedding of the secret messages. Concerning the quality of the image, both our approach and schemes of Lee and Huang’s [19] and Lin et al.’s [22] have their merits and their downsides. Our approach provides a higher *PSNR* value than the other two approaches for shadow image *S1*, but it provided a lower *PSNR* value of shadow image *S2*. Obviously, Chen and Guo’s scheme [24] has a better *PSNR* value than our approach. The main reason is that they carried a smaller amount of secret message than ours. Moreover, they constructed orientation combinations in a three \times three square block, while our approach generates position combinations in a turtle shell-based sunflower area. Conversely, our approach achieves a higher *ER* when compared to Chen and Guo’s scheme [24]. Thus, it is clear that there is always a trade-off between the quality of the image and the embedding capacity.

Table 6. Comparisons of maximum *ER* (bpp) among the different schemes.

Test Images	Chang et al. [17]	Lee and Huang [19]	Lin et al. [22]	Xie et al. [23]	Chen and Guo [24]	Chen and Hong [25]	Proposed Scheme
Airplane	1.00	1.04	1.25	2.00	1.14	1.56	1.25
Baboon	1.00	1.04	1.25	2.00	1.14	1.56	1.25
Goldhill	1.00	1.04	1.25	2.00	1.14	1.56	1.25
Barbara	1.00	1.04	1.25	2.00	1.14	1.56	1.25
Elaine	1.00	1.04	1.25	2.00	1.14	1.56	1.25
Lena	1.00	1.04	1.25	2.00	1.14	1.56	1.25
Peppers	1.00	1.04	1.25	2.00	1.14	1.56	1.25
Wine	1.00	1.04	1.25	2.00	1.14	1.56	1.25
Average	1.00	1.04	1.25	2.00	1.14	1.56	1.25

Table 7. Comparisons of *PSNR* (dB) with maximum *ER* among the different schemes.

Test Images	Chang et al. [17]		Lee and Huang [19]		Lin et al. [22]		Xie et al. [23]		Chen and Guo [24]		Chen and Hong [25]		Proposed Scheme	
	S1	S2	S1	S2	S1	S2	S1	S2	S1	S2	S1	S2	S1	S2
Airplane	45.32	45.34	49.38	49.38	49.39	45.54	40.80	40.81	49.91	49.92	43.40	43.45	49.44	44.49
Baboon	45.34	45.34	49.38	49.38	49.38	45.55	40.80	40.78	49.91	49.92	43.39	43.42	49.46	44.49
Goldhill	45.35	45.34	49.38	49.38	49.38	45.55	40.79	40.79	49.91	49.92	43.39	43.43	49.44	44.50
Barbara	45.32	45.32	49.38	49.38	49.39	45.55	40.79	40.78	49.91	49.92	43.42	43.44	49.45	44.50
Elaine	45.33	45.34	49.38	49.38	49.38	45.55	40.79	40.79	49.91	49.92	43.40	43.43	49.49	44.48
Lena	45.32	45.32	49.38	49.38	49.38	45.54	40.80	40.80	49.91	49.92	43.41	43.41	49.43	44.50
Peppers	45.32	45.35	49.38	49.38	49.38	45.55	40.80	40.80	49.91	49.92	43.42	43.41	49.48	44.48
Wine	45.33	45.34	49.38	49.38	49.38	45.55	40.80	40.80	49.91	49.92	43.41	43.42	49.47	44.50
Average	45.33	45.33	49.38	49.38	49.38	45.55	40.80	40.80	49.91	49.92	43.40	43.43	49.46	44.49

In order to obtain a fair comparison, Table 8 demonstrates a comparison of *PSNR*s among the different schemes under the same *ER* = 1.25 bpp. As can be seen, our approach and Lin et al.’s scheme [22] provide a better *PSNR* compared to Chen and Hong’s scheme [25]. Furthermore, Figure 9 demonstrates the *ER*–*PSNR* curves of shadow images between our approach and schemes [17,19,22,23]. The left column in Figure 9 indicates that our approach always achieved higher *PSNR* values than the other schemes when *ER* varied from 0.2 bpp to 1.25 bpp. Similarly, we can see from the right column in Figure 9 that the *PSNR* value of our approach is the third, after Lee and Huang’s scheme [19] and Lin et al.’s scheme [22], when the same *ER* is set. In summary, our approach is an excellent scheme that obtains both a higher *ER* and a higher *PSNR*, as Lin et al.’s scheme achieved. However, no confusion occurs by referring EPC, which is built up based on our Algorithm 1. It means the extraction and recovery procedure is relatively more efficient than that of Lin et al.’s scheme.

Table 8. Comparisons of PSNR (dB) among the different schemes under $ER = 1.25$ bpp.

Test Images	Lin et al. [22]		Chen and Hong [25]		Proposed Scheme	
	S1	S2	S1	S2	S1	S2
Airplane	49.39	45.54	44.36	44.41	49.44	44.49
Baboon	49.38	45.55	44.34	44.38	49.46	44.49
Goldhill	49.38	45.55	44.35	44.40	49.44	44.50
Barbara	49.39	45.55	44.38	44.40	49.45	44.50
Elaine	49.38	45.55	44.37	44.40	49.49	44.48
Lena	49.38	45.54	44.37	44.38	49.43	44.50
Peppers	49.38	45.55	44.39	44.37	49.48	44.48
Wine	49.38	45.55	44.37	44.39	49.47	44.50
Average	49.38	45.55	44.37	44.39	49.46	44.49

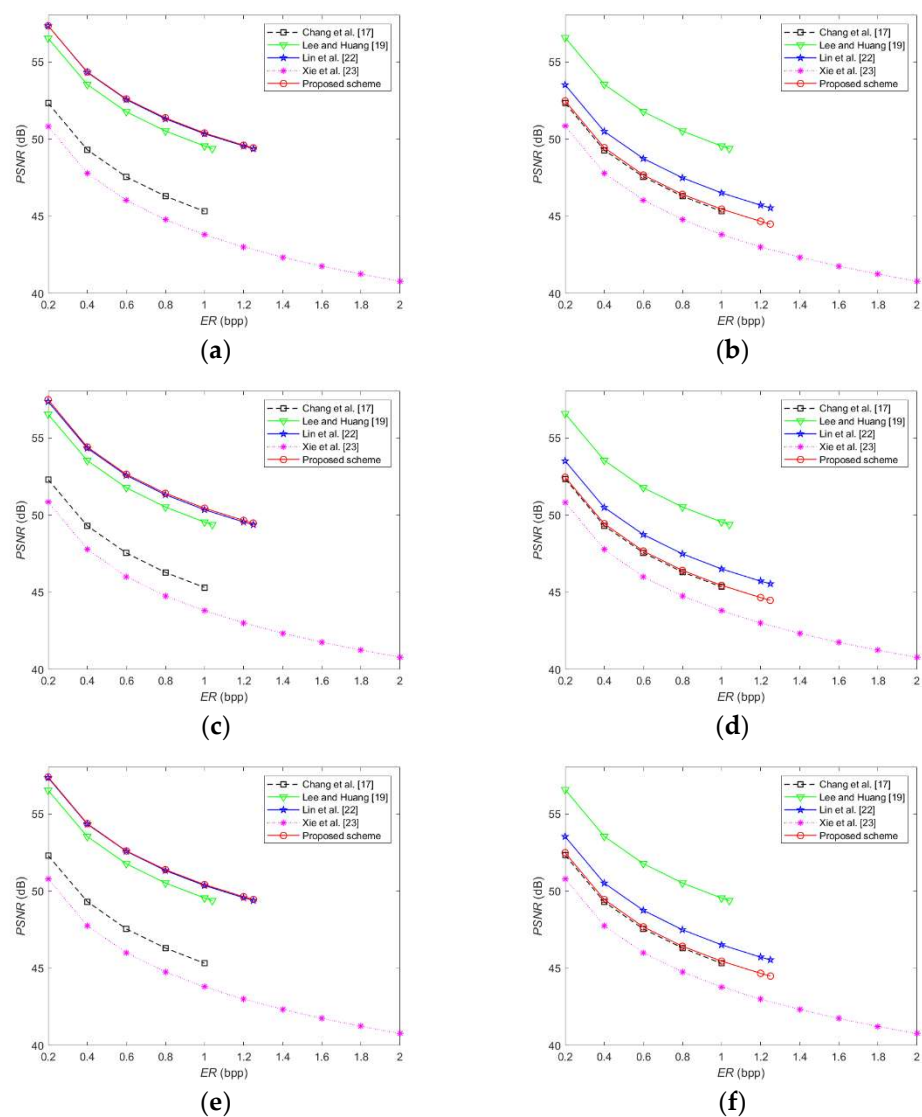


Figure 9. Cont.

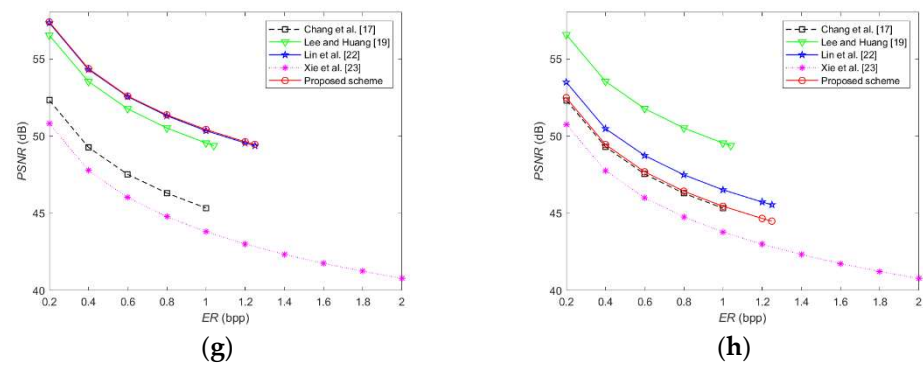


Figure 9. Comparisons various ER s and $PSNR$ s among the different schemes: (a) Shadow S1 for Lena, (b) Shadow S2 for Lena, (c) Shadow S1 for Peppers, (d) Shadow S2 for Peppers, (e) Shadow S1 for Barbara, (f) Shadow S2 for Barbara, (g) Shadow S1 for Baboon, and (h) Shadow S2 for Baboon.

Finally, we have given the comparisons in term of the execution times between our approach and some related works [17,19,22,23,25]. The results are listed in Table 9. Therein, each statistical value is a mean of 10 experimental results. As to our approach, on average, the execution time is around 0.5522 s, which means that our approach is more efficient. Moreover, it is obvious to find that those systems are capable of implementing experiments in less than 1s, indicating that they can be applied in most real-time scenarios.

Table 9. Comparisons of execution time (s) among the different schemes.

Test Images	Chang et al. [17]	Lee and Huang [19]	Lin et al. [22]	Xie et al. [23]	Chen and Hong [25]	Proposed Scheme
Airplane	0.0533	0.7550	0.0445	0.1320	0.0800	0.5217
Baboon	0.0543	0.7655	0.0511	0.1203	0.0781	0.5230
Goldhill	0.0557	0.7475	0.0458	0.1198	0.0609	0.4869
Barbara	0.0530	0.7458	0.0493	0.1196	0.0605	0.5118
Elaine	0.0527	0.7429	0.0461	0.1213	0.0600	0.5338
Lena	0.0527	0.7327	0.0459	0.1205	0.0597	0.5036
Peppers	0.0523	0.7359	0.0459	0.1199	0.0660	0.5479
Wine	0.0566	0.7581	0.0432	0.1221	0.0720	0.5485
Average	0.0538	0.7479	0.0465	0.1219	0.0672	0.5222

4.4. Discussions

The above experiments have confirmed that our approach has a good ER , while keeping the image distortion as low as possible. Nevertheless, we provide some discussions to guide the future improvements, in terms of the limited capabilities and the potential failures.

4.4.1. Limited Capabilities Analysis

- The to-be-embedded secret messages, i.e., Msg , are too large. The larger the Msg is, the larger the sunflower area should be. In doing so, the visual quality of shadow images will be seriously distorted. This indicates that the system is not capable of keeping the trade-off between $PSNR$ and ER ;
- Cost of execution time. In order to construct the ET adaptively, both in data embedding and extraction stages, each pixel pair should determine the sunflower area and select the EPCs according to their types. This requires the additional cost of execution time.

4.4.2. Potential Failures Analysis

- Without the prior knowledge of the rule of constructing EPC and ET, in this case, the receiver cannot extract the secret messages or carry out image recovery;

- Only owning one shadow image. In this paper, the secret messages and image recovery was correctly performed only if both of the recipients release their own shadow image. Therefore, for any one receiver alone, there will be a failure to extract the hidden secret messages.

5. Conclusions

In this paper, we presented an effective RDH scheme based on a turtle shell-based reference matrix and a position-aware strategy in order to achieve shadow images that had considerable quality, as well as a higher embedding capacity. First, a sunflower area based on turtle shell was designed, and then the corresponding embedding table was constructed by fully considering the combinations of positions. Using an embedding table, each pixel pair is used to carry the specific secret messages in order to generate two shadows. Due to the flexible and adaptive position combination strategy, we can process our system without any overhead information. As a consequence, an *ER* value of 1.25 bpp was obtained while the visual quality of the shadow images was maintained at around 49.46 dB. The experimental results confirmed that our approach outperformed some of the relevant works, in terms of both embedding capacity and image quality. In addition, we conducted security analyses in order to prove that our approach is effective against attacks on the PVD, RE, and RS analyses.

Due the reversibility and high capacity, our system can be used to develop a software on WeChat, etc., which aims to transmit the personal data. In the future, we will explore the novel strategy in order to solve the potential failures in our system. Also, we will try to investigate more applications by using our algorithm, such as image watermarking, image ownership verification, and intellectual property protection.

Author Contributions: Conceptualization, C.-C.C. and C.-C.L.; methodology, G.-D.S. and C.-C.L.; software, G.-D.S.; validation, G.-D.S. and C.-C.L.; formal analysis, Y.-H.L.; investigation, G.-D.S. and C.-C.L.; writing—original draft preparation, G.-D.S. and C.-C.L.; writing, C.-C.C., G.-D.S. and C.-C.L.; visualization, G.-D.S.; supervision, C.-C.C., Y.-H.L. and C.-C.L.; project administration, Y.-H.L.; funding acquisition, Y.-H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Hon Hai Research Institute, Foxconn Technology Group.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The source code will be provided as requested.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chang, C.C.; Liu, Y.; Nguyen, T.S. A novel turtle shell based scheme for data hiding. In Proceedings of the 2014 Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Kitakyushu, Japan, 27–29 August 2014; pp. 89–93.
2. Duan, X.; Li, B.; Xie, Z.; Yue, D.; Ma, Y. High-capacity information hiding based on residual network. *IETE Tech. Rev.* **2021**, *38*, 172–183. [[CrossRef](#)]
3. Ge, X.; Yu, J.; Hao, R.; Lv, H. Verifiable Keyword search supporting sensitive information hiding for the cloud-based healthcare sharing system. *IEEE Trans. Ind. Inform.* **2021**. [[CrossRef](#)]
4. Chang, C.C.; Li, C.T.; Shi, Y.Q. Privacy-aware reversible watermarking in cloud computing environments. *IEEE Access* **2018**, *6*, 70720–70733. [[CrossRef](#)]
5. Chang, C.C. Adversarial learning for invertible steganography. *IEEE Access* **2020**, *8*, 198425–198435. [[CrossRef](#)]
6. Chang, C.C.; Li, C.T.; Chen, K. Privacy-preserving reversible information hiding based on arithmetic of quadratic residues. *IEEE Access* **2019**, *7*, 54117–54132. [[CrossRef](#)]
7. Hu, Y.; Lee, H.K.; Chen, K.; Li, J. Difference expansion based reversible data hiding using two embedding directions. *IEEE Trans. Multimed.* **2008**, *10*, 1500–1512. [[CrossRef](#)]
8. Lee, C.F.; Chen, H.L.; Tso, H.K. Embedding capacity raising in reversible data hiding based on prediction of difference expansion. *J. Syst. Softw.* **2010**, *83*, 1864–1872. [[CrossRef](#)]

9. Ou, B.; Li, X.; Zhao, Y.; Ni, R.; Shi, Y.Q. Pairwise prediction-error expansion for efficient reversible data hiding. *IEEE Trans. Image Process.* **2013**, *22*, 5010–5021. [[CrossRef](#)]
10. Chang, Q.; Li, X.; Zhao, Y.; Ni, R. Adaptive pairwise prediction-error expansion and multiple histograms modification for reversible data hiding. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *31*, 4850–4863. [[CrossRef](#)]
11. Wang, J.; Ni, J.; Zhang, X.; Shi, Y.Q. Rate and distortion optimization for reversible data hiding using multiple histogram shifting. *IEEE Trans. Cybern.* **2016**, *47*, 315–326. [[CrossRef](#)]
12. Peng, F.; Zhao, Y.; Zhang, X.; Long, M.; Pan, W.Q. Reversible data hiding based on RSBEMD coding and adaptive multi-segment left and right histogram shifting. *Signal Process. Image Commun.* **2020**, *81*, 115715. [[CrossRef](#)]
13. Qu, X.; Kim, H.J. Pixel-based pixel value ordering predictor for high-fidelity reversible data hiding. *Signal Process.* **2015**, *111*, 249–260. [[CrossRef](#)]
14. Chang, J.; Ding, F.; Li, X.; Zhu, G. Hybrid prediction-based pixel-value-ordering method for reversible data hiding. *J. Vis. Commun. Image Represent.* **2021**, *77*, 103097. [[CrossRef](#)]
15. Chang, C.C. Cryptospace invertible steganography with conditional generative adversarial networks. *Secur. Commun. Netw.* **2021**, *2021*, 5538720. [[CrossRef](#)] [[PubMed](#)]
16. Chang, C.C. Neural reversible steganography with long short-term memory. *Secur. Commun. Netw.* **2021**, *2021*, 5580272. [[CrossRef](#)]
17. Chang, C.C.; Kieu, T.D.; Chou, Y.C. Reversible data hiding scheme using two steganographic images. In Proceedings of the Tencon 2007–IEEE Region 10 Conference, Taipei, Taiwan, 30 October–2 November 2007; pp. 1–4.
18. Lee, C.F.; Wang, K.H.; Chang, C.C.; Huang, Y.L. A reversible data hiding scheme based on dual steganographic images. In Proceedings of the 3rd International Conference on Ubiquitous Information Management and Communication, Suwon, Korea, 15–16 January 2009; pp. 228–237.
19. Lee, C.F.; Huang, Y.L. Reversible data hiding scheme based on dual stego-images using orientation combinations. *Telecommun. Syst.* **2013**, *52*, 2237–2247. [[CrossRef](#)]
20. Liu, Y.; Chang, C.C. A turtle shell-based visual secret sharing scheme with reversibility and authentication. *Multimed. Tools Appl.* **2018**, *77*, 25295–25310. [[CrossRef](#)]
21. Su, G.D.; Liu, Y.; Chang, C.C. A square lattice oriented reversible information hiding scheme with reversibility and adaptivity for dual images. *J. Vis. Commun. Image Represent.* **2019**, *64*, 102618. [[CrossRef](#)]
22. Lin, J.Y.; Liu, Y.; Chang, C.C. A real-time dual-image-based reversible data hiding scheme using turtle shells. *J. Real-Time Image Process.* **2019**, *16*, 673–684. [[CrossRef](#)]
23. Xie, X.Z.; Chang, C.C. Hiding data in dual images based on turtle shell matrix with high embedding capacity and reversibility. *Multimed. Tools Appl.* **2021**, *80*, 36567–36584. [[CrossRef](#)]
24. Chen, X.; Guo, W. Reversible data hiding scheme based on fully exploiting the orientation combinations of dual stego-images. *Int. J. Netw. Secur.* **2020**, *22*, 126–135.
25. Chen, X.; Hong, C. An efficient dual-image reversible data hiding scheme based on exploiting modification direction. *J. Inf. Secur. Appl.* **2021**, *58*, 102702. [[CrossRef](#)]
26. Chang, C.C.; Li, C.T. Algebraic secret sharing using privacy homomorphisms for IoT-based healthcare systems. *Math. Biosci. Eng.* **2019**, *16*, 3367–3381. [[CrossRef](#)] [[PubMed](#)]
27. Fridrich, J.; Goljan, M.; Du, R. Detecting LSB steganography in color, and gray-scale images. *IEEE Multimed.* **2001**, *8*, 22–28. [[CrossRef](#)]
28. Lou, D.C.; Hu, C.H. LSB steganographic method based on reversible histogram transformation function for resisting statistical steganalysis. *Inf. Sci.* **2012**, *188*, 346–358. [[CrossRef](#)]