Low-Computation-Cost Data Hiding Scheme Based on Turtle Shell

Yu Chen[1], Jiang-Yi Lin[2,3], Chin-Chen Chang[3], and Yu-Chen Hu[4]

(Corresponding author: Chin-Chen Chang)

School of Information Science and Engineering, Fujian University of Technology, Fuzhou 350118, China[1]
33 Xuefu South Road, Fuzhou 350118, China
Department of Computer Science, Xiamen University of Technology, Xiamen 361024, China[2]
600 Ligong Road, Xiamen 361024, China
Department of Information Engineering and Computer Science, Feng Chia University, Taichung 40724, Taiwan[3]
100 Wenhua Road,Taichung 40724, Taiwan
Department of Computer Science and Information Management, Providence University, Taichung 43301, Taiwan[4]
200, Sec. 7, Taiwan Boulevard, Shalu Dist., Taichung 43301, Taiwan
(Email: alan3c@gmail.com)

(Received May 16, 2018; Revised and Accepted Nov. 2, 2018; First Online July 16, 2019)

Abstract

Data-hiding technology is to study how to embed secret data into digital media such as images, audio, and video. Chang *et al.* adopted a novel turtle shell-based reference matrix to hide secret data, which resulted in better visual effects and higher embedding capacities. By changing the range of searching for elements, our proposed scheme improves the data hiding scheme of Chang *et al.* in terms of computational complexity and image quality. Experimental results verify that the proposed scheme improves the image quality of the stego images, accelerates the speed of the embedding operations, and maintains the same hiding capacity as the comparative method.

Keywords: Data Hiding; Exploiting Modification Direction; Turtle Shell

1 Introduction

Data hiding mainly studies how to hide secret data in public digital media. Usually, a certain method is designed to embed secret data into digital carriers, such as texts, audios, images, and videos. Among these digital media, digital images are used extensively as the cover medium for data hiding. Hiding secret data into the cover image makes the steganographic image slightly different from the original image, so that it will not attract attention when it is transmitted through a public network. This is one of the objectives of the data hiding scheme, and another objective is to increase the embedding capacity.

Many researchers have proposed various data hiding schemes [1, 2, 5, 10, 13, 16, 20, 21]. The aim of some of these schemes is to provide a good quality image, some focus on achieving high embedding capacity, and others focus on providing low computational cost. Chan *et al.* (2004) designed a data hiding scheme based on the simple least-significant-bit (LSB) substitution technique [1]. In their scheme, the simple LSB substitution method was for the initial hiding, and, then, the pixel values of the stego-image were modified appropriately according to the embedding error of the stego image and the original cover image. Thus, the optimization adjustment of the pixels was made, and the quality of the image was improved. Mielikainen (2006) presented a novel data hiding method named the LSB matching revisited scheme [16]. In his proposed scheme, two secret bits were embedded in a pair of cover pixels by modifying their directions. However, there is a weakness in this scheme in that exploitation is incomplete. Zhang and Wang proposed a novel steganographic method that they called exploiting modification directions (EMD) [24] in digital images, which overcame the weakness of Mielikainen's method. In Zhang *et al.*'s scheme, different embedded secret digits were represented by modifications in different directions. And each $(2k+1)$-based secret digit could be embedded by k cover pixels.

Chang and others proposed a novel information hiding method named Sudoku-S [3]. In their method, a reference matrix was generated by using a certain Sudoku solution. According to the reference matrix, each pixel pair can carry a 9-ary secret digit. Also, using the Sudoku solution, Hong and others presented an improved data hiding scheme named Sudoku-SR [9], which eliminated the shortcomings of the Sudoku-S. They proposed a search algorithm based on the nearest distance to determine where the secret digit was located, which further reduced the distortion of the image compared to the scheme of Chang *et* *al.* Kim *et al.* introduced two new data hiding methods, EMD-2 and 2-EMD [12]. Compared to EMD proposed by Zhang and Wang, EMD-2 and 2-EMD improve the embedding rate and easily can be extended to EMD-K and K-EMD.

In recent years, some researchers have proposed some effective data hiding schemes [7,8,11,17,23]. Yang *et al.* proposed a scheme for embedding data using pixel-value differencing (PVD) [23]. In their scheme, the secret data were embedded by changing the value of the difference between two pairs of pixels instead of one pair of pixels. Their scheme increased the embedding capacity by using the more flexible method of searching the edge area. Chen proposed a PVD-based data hiding method that could embed secret information with a variable number of bits [7]. In his method, how many bits of secret information a pair of pixels could embed is determined by the complexity of the pixels in the area. Some reversible data hiding schemes have also been proposed [6,15,18,19].

In addition, Chang and others proposed a new turtle shell-based data hiding scheme (TDH) [4]. In their scheme, the octal digits valued from 0 to 7 are arranged aptly in each hexagonal area in a constructed reference matrix, that is, in a turtle shell. In the TDH scheme, each cover pixel pair can be used to embed three bits of secret data, and the embedding capacity is improved compared to some previous schemes [2,10].

The novelty of the turtle shell-based matrix has attracted some scholars to use it to conduct more research on data hiding. Liu *et al.* [14] improved Chang *et al.*'s scheme [4] by improving the hiding capacity. They used a positional relationship between the elements and the turtle shells to create a location table, which enabled each pixel pair to embed four bits. Xie *et al.* proposed a two-layer turtle shell-based data hiding scheme [22] in 2018. In their proposed scheme, the turtle shell-based reference matrix was considered as a layer. And different types of relationships were defined between the elements and the number of turtle shells involved, which constituted another layer, the type matrix. The proposed two-layer scheme can represent more cases than when only the turtle shell matrix is used. In this scheme, up to five bits of secret data can be embedded in each pixel pair. The above two schemes [4,14] provided better embedding capacity than Chang *et al.*'s scheme [4], but their performances on the quality of the stego image and search time were not as good.

Inspired by Chang *et al.*'s scheme [4], we propose an improved turtle shell-based data hiding scheme that reduces the time required to generate the stego image and increases its image quality. In our scheme, first, a reference matrix is built and its internal elements are arranged in the form of turtle shells, which is the same as arrangement design in the TDH scheme. Then, when searching for the secret digit in the reference matrix, a 3×3 block is simply used as a search area instead of dealing with many turtle shell-related rules as in the TDH scheme.

The following content of this paper is arranged as follows. The TDH scheme proposed by Chang *et al.* is reviewed in Section 2. Section 3 specifies the improved scheme we proposed. Section 4 provides our experimental results, and Section 5 presents our conclusions.

2 Review of Chang and others' Scheme

In 2014, Chang *et al.* proposed the turtle shell-based data hiding scheme (TDH) [4]. In their scheme, a hexagonal area is named the turtle shell, and its range has exactly eight points, which can be used to represent the numbers 0 to 7. The reference matrix, R, is composed of many such turtle shells. Figure 1 shows the data distribution of a part of R. In R, the value of the adjacent elements in the horizontal direction increases by 1 in order, and the value of the increase between the adjacent elements in the longitudinal direction is alternating 2 and 3. Such turtle shells are arranged continuously until a reference matrix is obtained.

To generate the same reference matrix for embedding and extracting data, the element with the coordinates of (0,0) is set to 0. The other elements are generated by using the steps mentioned above. The resultant reference matrix, R, is shown in Figure 1, where the identifier p_i represents a selected pixel, the identifier p_{i+1} represents the pixel adjacent to the selected pixel, and the values of the horizontal and vertical coordinates from 0 to 255 indicate the gray-scale pixel values.

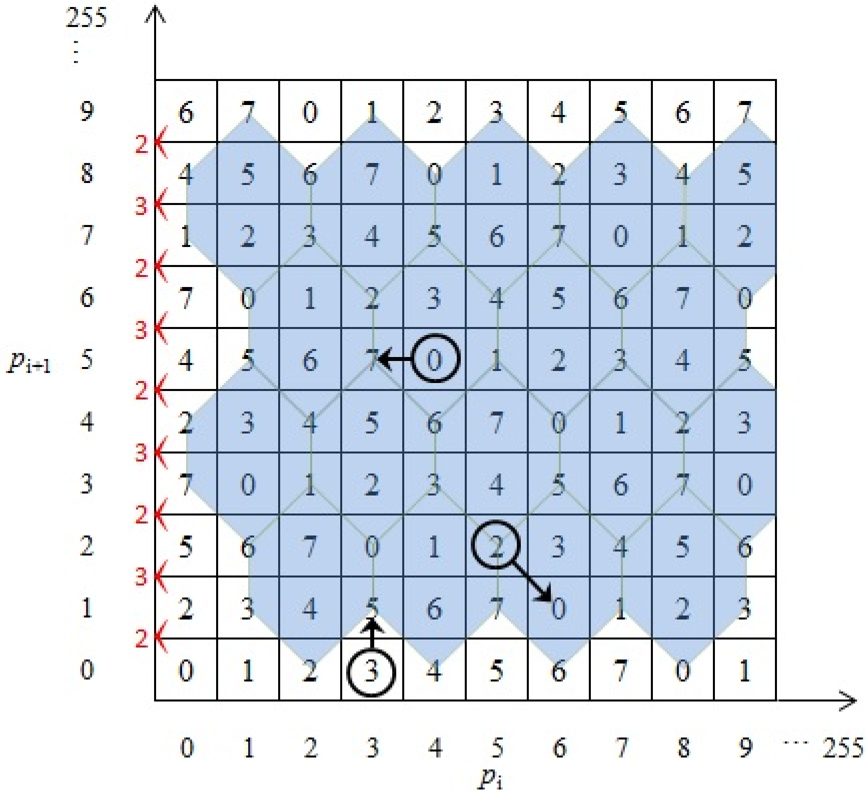
2.1 Data Hiding Procedure of TDH Scheme

Assume that a cover image I has a size $W \times H$. A cover pixel pair (p_i, p_{i+1}) will be mapped to the position (p_i, p_{i+1}) of R, where i=1, 3, ..., $(W \times H)-1$. $R(p_i, p_{i+1})$ represents the element at (p_i, p_{i+1}) in R. Chang *et al.* classified all the elements of R into normal elements and special elements. The elements within the turtle shell are classified as normal elements, and the remaining elements are classified as special elements. The normal elements are divided further into back elements and edge elements. In their proposed scheme, there are three cases to deal with for different categories of elements. Let S be the set of the area that contains the secret digit to be embedded. The specific processing cases are as follows:

Case **1:** If $R(p_i, p_{i+1})$ is a back element, S is the turtle shell where $R(p_i, p_{i+1})$ is located.

Case **2:** If $R(p_i, p_{i+1})$ is an edge element, there is at least one turtle shell that contains $R(p_i, p_{i+1})$, and S is the set of these turtle shells.

Case **3:** If $R(p_i, p_{i+1})$ is a special element, S is a set of all 3×3 blocks that contain $R(p_i, p_{i+1})$.



	0	1	2	3	4	5	6	7	8	9
9	6	7	0	1	2	3	4	5	6	7
8	4	5	6	7	0	1	2	3	4	5
7	1	2	3	4	5	6	7	0	1	2
6	7	0	1	2	3	4	5	6	7	0
5	4	5	6	7	0	1	2	3	4	5
4	2	3	4	5	6	7	0	1	2	3
3	7	0	1	2	3	4	5	6	7	0
2	5	6	7	0	1	2	3	4	5	6
1	2	3	4	5	6	7	0	1	2	3
0	0	1	2	3	4	5	6	7	0	1

Figure 1: Example of the turtle shell-based reference matrix

Let d be the secret digit to be embedded. Because of the structural characteristics of the turtle shell-based matrix, in *Case* 2 and *Case* 3, there may be, at most, three turtle shells in S that contain d. Among all of the candidate elements included in S, the element that is the shortest distance from $R(p_i, p_{i+1})$ is selected and set as $R(p'_i, p'_{i+1})$. Then, the cover pixel pair (p_i, p_{i+1}) is modified to (p'_i, p'_{i+1}) to ensure the smallest distortion while also embedding d. After all of the pixel pairs have been processed, the stego image, I', is generated. Next is an example of embedding secret digits using the TDH scheme.

Example 1.

Assume that a binary secret data stream SD_2 is denoted as $SD_2 = (111000101)_2$, and the three stego pixel pairs used to embed secret data are (4, 5), (5, 2), and (3, 0). First, SD_2 is converted to an octal stream $SD_8 = (705)_8$. Then, each stego pixel pair is separately embedded with one digit in SD_8. The detailed embedding process is as follows. Figure 1 shows the relevant flags for the secret digits, pixel pairs, and embedding results.

1) Embed digit $(7)_8$ into the pixel pair (4, 5)

   Mapping the pixel pair (4, 5) to R, the corresponding element, $R(4, 5)$, is a back element. It is *Case* 1, and the only candidate element for digit $(7)_8$ is $R(3, 5)$, so the cover pixel pair (4, 5) is replaced by (3, 5) in I' to embed $(7)_8$.

2) Embed digit $(0)_8$ into the pixel pair (5, 2)

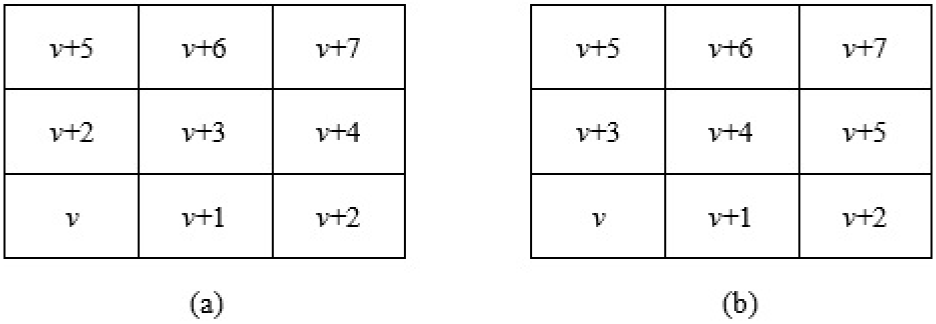
   Mapping the pixel pair (5, 2) to R, the corresponding $R(5, 2)$ is an edge element. It is *Case* 2, and there are three candidate turtle shells that involve $R(5, 2)$. The secret digits $(0)_8$ in the three candidate turtle shells are located at (3, 2), (6, 1), and (6, 4) in R, respectively. The squared distances between the above three elements and $R(5, 2)$ are 4, 2 and 5, respectively, where the value 2 is the smallest. Thus, the cover pixel pair (5, 2) is replaced by (6, 1) in I' to embed $(0)_8$.

3) Embed digit $(5)_8$ into the pixel pair (3, 0)

   Mapping the pixel pair (3, 0) to R, the corresponding $R(3, 0)$ is a special element. It is *Case* 3, and there are three candidate 3×3 blocks that involve $R(3, 0)$, and these three blocks contain two secret digits $(5)_8$, located at (3, 1) and (5, 0) in R, respectively. The squared distances between the above two elements and $R(3, 0)$ are 1 and 4, respectively, where the value 1 is the smallest. So, the cover pixel pair (3, 0) is replaced to (3, 1) in I' to embed $(5)_8$.

2.2 Extracting Procedure of TDH Scheme

In the TDH scheme proposed by Chang *et al.*, the reference matrix R used in the data embedding procedure also



(a)

v+5	v+6	v+7
v+2	v+3	v+4
v	v+1	v+2

(b)

v+5	v+6	v+7
v+3	v+4	v+5
v	v+1	v+2

Figure 2: Two digital distributions in the 3 × 3 blocks

is used for the extraction of the embedded secret data. If we assume that (p'_i, p'_{i+1}) is a pixel pair of the stego image, it clearly can be mapped to $R(p'_i, p'_{i+1})$ in R. The element $R(p'_i, p'_{i+1})$ is just the embedded secret digit. The embedded secret data is obtained exactly from all of the extracted secret digits.

3 Proposed Secret Image Sharing Scheme

After studying the TDH scheme [4] proposed by the Chang *et al.*, we propose an improved turtle shell-based scheme with better image quality and faster data embedding process for its shortcomings. First, a reference matrix is constructed by the same process used in Chang *et al.*'s scheme. Both schemes are constructed based on hexagonal turtle shells. Then, each pixel pair of the cover image is used to carry three bits through the reference matrix to obtain the stego image. Unlike the TDH scheme, the method of locating the secret digit in the reference matrix of our scheme is simple and efficient, and it improves the quality of the stego image.

3.1 Secret Data Embedding

Let I be the cover image of size $M \times N$ and E be the binary secret data stream. A pixel pair of I is represented as (p_i, p_{i+1}), where $i = 1, 3, \ldots,\ M \times N - 1$. The reference matrix R used in our proposed scheme is shown in Figure 1. A pixel pair, (p_i, p_{i+1}), is simultaneously used as a coordinate in R corresponding to an element $R(p_i, p_{i+1})$. For example, in Figure 1, the pixel pair (2, 3) is mapped to the element $R(2, 3)$, and its value is 1. The detailed process steps for embedding secret data into each pixel pairs are shown below:

Step 1. Sequentially, read a 3-bit secret data from E and convert it to an octal secret digit n_j, where $n_j \in [0, 7]$.

Step 2. Read a pixel pair (p_i, p_{i+1}) of I and map it to a 3×3 block, which contains $R(p_i, p_{i+1})$. There are two cases to deal with.

Case 1: If the coordinate (p_i, p_{i+1}) can be the central point of a 3×3 block, then the block is set to the candidate block B.

Case 2: If the coordinate (p_i, p_{i+1}) cannot be the central point of a 3×3 block, then it will be subdivided into two cases.

Case 2.1: If $(0 < p_i < 255, p_{i+1} = 0)$ or $(0 < p_i < 255, p_{i+1} = 255)$ or $(p_i = 0, 0 < p_{i+1} < 255)$ or $(p_i = 255, 0 < p_{i+1} < 255)$, the 3×3 block whose center point coordinate of one of its edges is (p_i, p_{i+1}) is the candidate block B.

Case 2.2: If $(p_i = 0,\ p_{i+1} = 0)$ or $(p_i = 0,\ p_{i+1} = 255)$ or $(p_i = 255,\ p_{i+1} = 0)$ or $(p_i = 255,\ p_{i+1} = 255)$, the 3×3 block containing coordinate (p_i, p_{i+1}) is the candidate block B.

Step 3. Search for the secret digit, n_j, in B. If the element $R(p'_i, p'_{i+1})$ equal to n_j, change the cover pixel pair (p_i, p_{i+1}) to (p'_i, p'_{i+1}), which is a pixel pair of the stego image I'.

Step 4. Repeat Steps 1-3 until all pixel pairs of I are processed.

Step 5. Output I'.

In our scheme, any secret digit n_j can be found in the 3×3 block B. Our proof is given as below. Let v be the digit in the lower left corner of B. According to the construction of the turtle shell we are using, in general, the form of the other digits will be arranged as shown in Figures 2(a) and 2(b). Since the range of n_j is from 0 to 7, the numbers from v to $v + 7$ are to be executed by module 8 when they are greater than 7. Therefore, the numbers represented from v to $v + 7$ certainly contain eight numbers from 0 to 7. After the above process is completed, a stego image I' is produced.

Example 2. *Assume that the secret data in binary form is* $(101000110)_2$*, it can be converted to octal stream* $(506)_8$*. And assume that (3, 5), (5, 0), and (0, 0) are the three cover pixel pairs that will be used to embed the three octal digits. The following processing steps show the processes of embedding three octal digits into the pixel pairs, and the embedded results are shown in Figure 3.*

1) Embed $(5)_8$ into pixel pair (3, 5)

   The pixel pair (3, 5) is mapped to $R(3,5)$ and a 3×3 block B centered on $R(3,5)$ is determined. It is *Case* 1, and the octal digit $(5)_8$ in B is found at (3, 4). Then, the pixel pair (3, 5) in I is changed to (3, 4) in I' for embedding $(5)_8$.

2) Embed $(0)_8$ into pixel pair (5, 0)

   The pixel pair (5, 0) is mapped to $R(5,0)$, but any 3×3 block centered on $R(5,0)$ cannot be found. The element $R(5,0)$ is at the boundary of R. It is *Case* 2.1, and the corresponding 3×3 block B is marked in Figure 3. The secret digit $(0)_8$ in B is found at (6, 1). Therefore, the pixel pair (5, 0) in I is changed to (6, 1) in I' for embedding $(0)_8$.

3) Embed $(6)_8$ into pixel pair (0, 0)

   The pixel pair (0, 0) is mapped to $R(0,0)$, which is located at the corner of R. It is *Case* 2.2, and the corresponding 3×3 block B also is marked in Figure 3. The secret digit $(6)_8$ in B is found at (1, 2). Thus, the pixel pair (0, 0) in I is changed to (1, 2) in I' for embedding $(6)_8$.

3.2 Extraction of Secret Data

When a stego image I' of size $M\times N$ is received, the proposed scheme can accurately extract the secret data embedded therein. First, the receiver constructs a reference matrix, R, used in the secret data embedding process. Second, all pixel pairs in I' are read sequentially and mapped as coordinates to R. Let (p'_i,p'_{i+1}) be a pixel pair in I', and $R(p'_i,p'_{i+1})$ is the corresponding mapping element in R, where $i=1,3,\ldots,\ M\times N-1$. Then, the element $R(p'_i,p'_{i+1})$ is the octal secret digit embedded in the pixel pair (p'_i,p'_{i+1}). The extracted secret digits will be sequentially concatenated together to form an octal stream. After all the pixel pairs in I' have been processed in this way, a complete octal secret stream is obtained. The octal secret stream is then converted to a binary stream, and the receiver successfully extracts the secret data from I'.

Example 3. *Assume that the pixel pairs (3, 4), (6, 1), and (1, 2) are consecutive pixel pairs in the stego image I' generated in Example 2. We now extract the secret data embedded in them. In R, the elements corresponding to coordinates (3, 4), (6, 1), and (1, 2) are* $R(3,4)=(5)_8$, $R(6,1)=(0)_8$, *and* $R(1,2)=(6)_8$, *respectively. The obtained octal digits are connected one by one to form an octal stream* $(506)_8$. *Finally, the octal stream is converted to a binary stream* $(101000110)_2$, *which is the embedded secret data.*

4 Experimental Results

Some experiments were conducted on some test images to illustrate the correctness of the proposed scheme. Our experiments were conducted in MATLAB 8.0 software in a personal computer configured Intel(R) Core (TM) i7-3770 @ 3.40 GHZ and 8 GB of memory, and the operating system was installed is Windows 10 Education 64 bits.

4.1 Experiment Design

Eight original grayscale images, namely, Airplane, Baboon, Boat, House, Elaine, Lena, Man, and Peppers, were tested as cover images in our experiment. All test images had 512×512 pixels. The secret data used in the experiment consisted of a binary stream formed by randomly-generated, binary bits. The original grayscale test images are shown in Figure 4.

In the proposed scheme, the peak signal-to-noise ratio ($PSNR$) score is used to evaluate the quality of the stego image, which is defined as Equation (1):

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

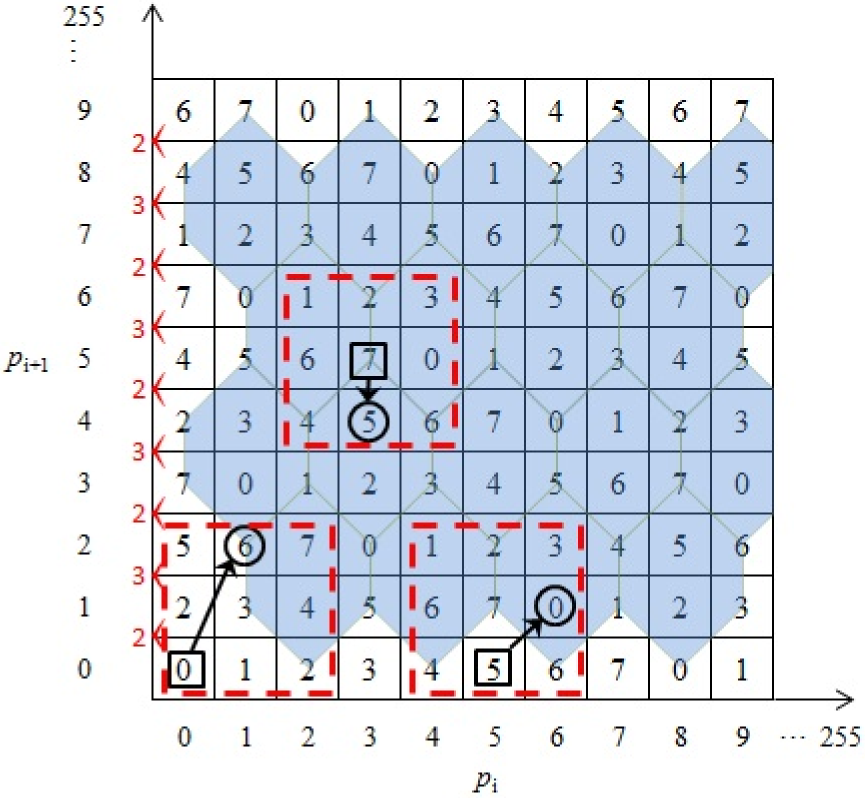
where MSE represents the mean square error of the stego image and the cover image, which is defined in Equation (2).

$$MSE = \frac{1}{M\times N}\sum_{i=1}^{M}\sum_{j=1}^{N}(X_{ij}-Y_{ij}). \tag{2}$$

where M and N represent the height and width of the image, respectively, and X_{ij} and Y_{ij} are the values of the pixels of the stego image and the cover image, respectively. Equations (1) and (2) indicate that there is an inverse relation between $PSNR$ and MSE, *i.e.*, the lower the MSE value is, the higher the $PSNR$ value becomes. A higher $PSNR$ value indicates that there is a smaller difference between the two images.

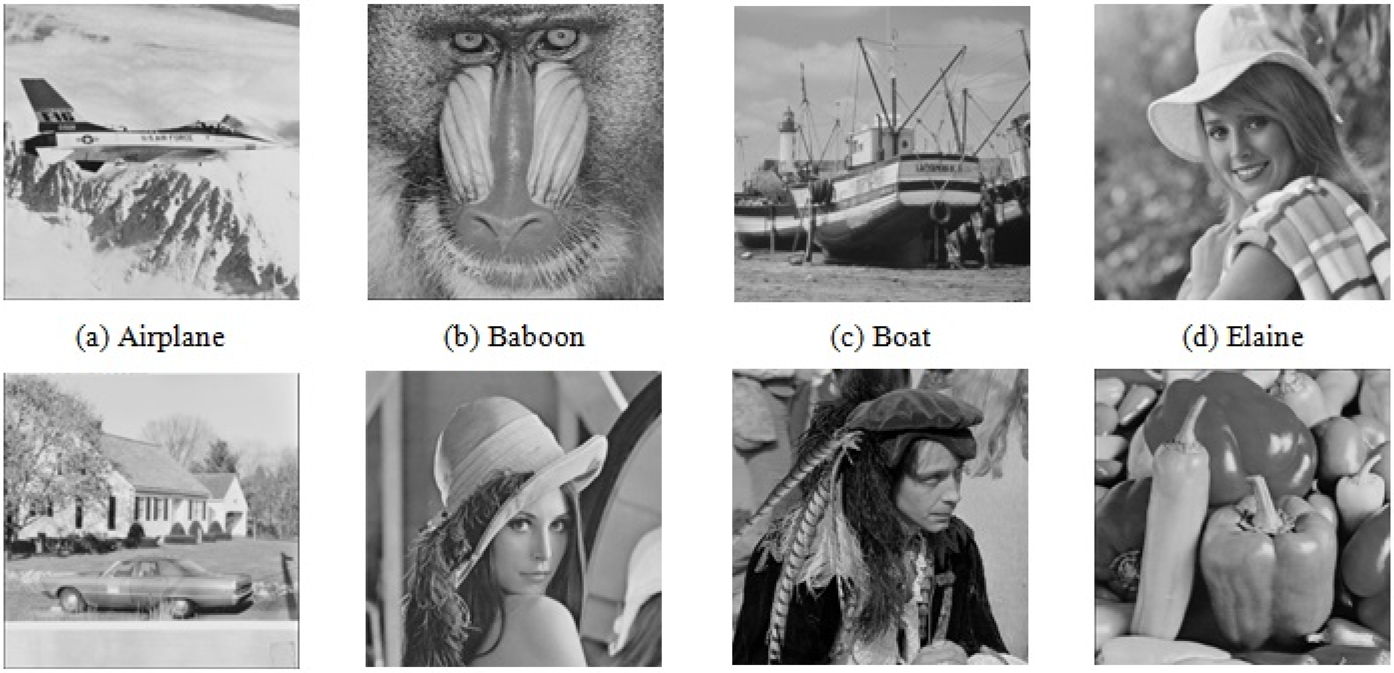
Another performance measurement we used to evaluate the data hiding capacity is the embedding capacity (EC), which represents the number of binary bits that can be embedded in a cover image. The EC of our scheme also is compared with the EC of TDH scheme, and Table 1 shows the results. Table 1 indicates that the EC of the proposed scheme had the same value as that of the TDH scheme, reaching 1.5 bpp. Average image qualities of 49.72 dB and 50.14 dB are achieved by the TDH scheme and the proposed scheme, respectively. In addition, the average execution time of the TDH scheme is 0.67 second, and for the proposed scheme, it is 0.58 second. Obviously, the proposed scheme gains better quality of stego images than the TDH scheme proposed by Chang *et al.*, and its computational cost for embedding data is less than that of Chang *et al.*'s TDH scheme.

To understand the visual qualities of the stego images of the TDH scheme and the proposed scheme, two sets of stego images produced by the two schemes are provided in Figures 5 and 6, respectively. Compared to the cover images shown in Figure 4, it is difficult to distinguish the difference between the stego images and the original



p_{i+1} \ p_i	0	1	2	3	4	5	6	7	8	9
9	6	7	0	1	2	3	4	5	6	7
8	4	5	6	7	0	1	2	3	4	5
7	1	2	3	4	5	6	7	0	1	2
6	7	0	1	2	3	4	5	6	7	0
5	4	5	6	7	0	1	2	3	4	5
4	2	3	4	5	6	7	0	1	2	3
3	7	0	1	2	3	4	5	6	7	0
2	5	6	7	0	1	2	3	4	5	6
1	2	3	4	5	6	7	0	1	2	3
0	0	1	2	3	4	5	6	7	0	1

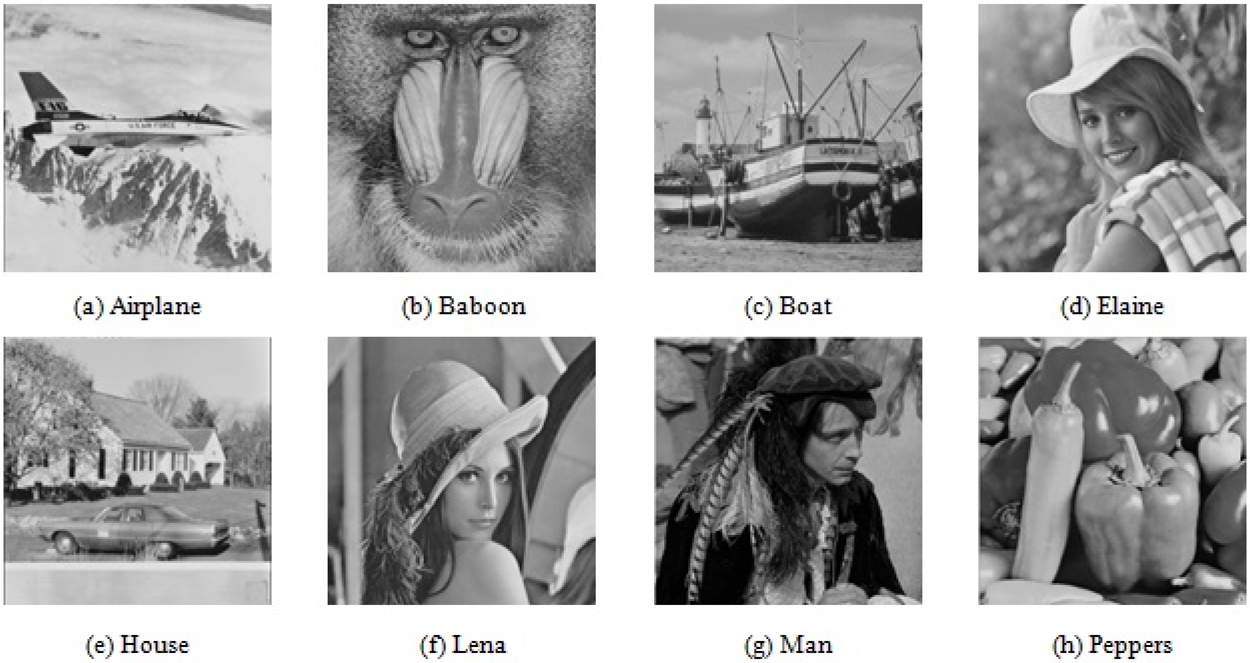
Figure 3: Examples of embedding process based on turtle shells



(a) Airplane (b) Baboon (c) Boat (d) Elaine

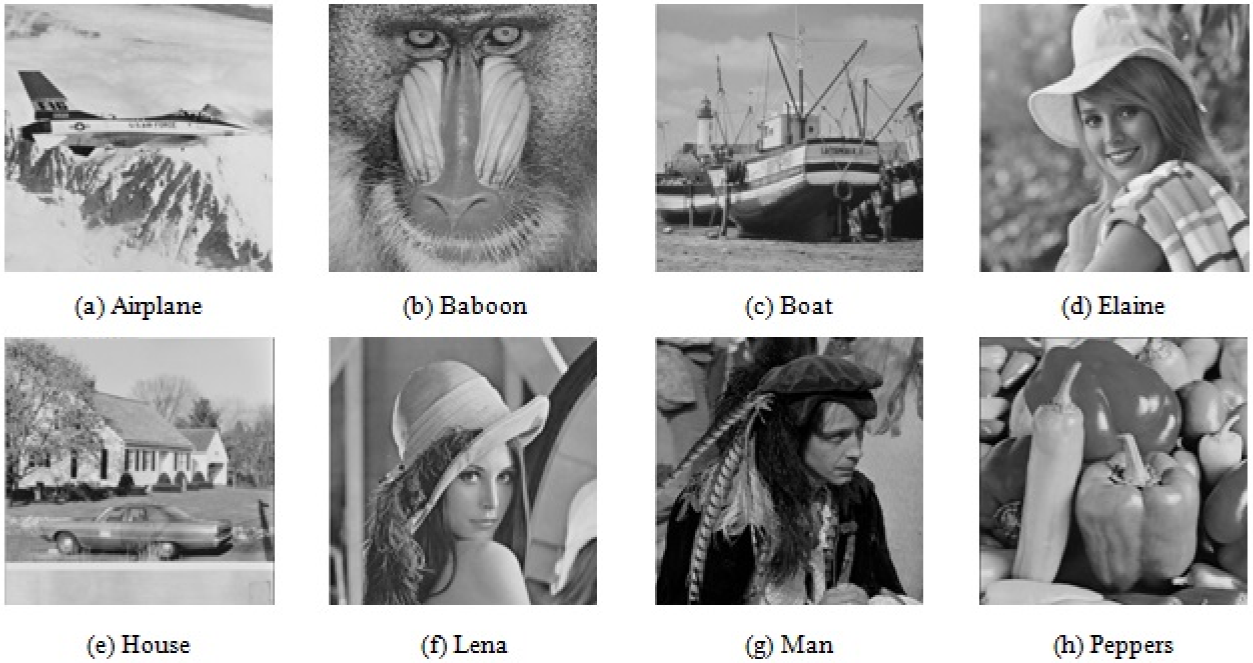
(e) House (f) Lena (g) Man (h) Peppers

Figure 4: Grayscale test images


(a) Airplane (b) Baboon (c) Boat (d) Elaine

(e) House (f) Lena (g) Man (h) Peppers

Figure 5: Stego images of Chang *et al.*'s scheme


(a) Airplane (b) Baboon (c) Boat (d) Elaine

(e) House (f) Lena (g) Man (h) Peppers

Figure 6: Stego images of the proposed scheme

Table 1: Central point bits of the binary block and its corresponding shadow blocks

Cover Images	Chang *et al.*'s scheme (TDH)			Proposed scheme		
	PSNR	*EC*	Running Time	*PSNR*	*EC*	Running Time
Airplane	49.75	1.5	0.64	50.17	1.5	0.60
Baboon	49.75	1.5	0.75	50.16	1.5	0.60
Boat	49.75	1.5	0.67	50.16	1.5	0.58
Elaine	49.75	1.5	0.67	50.17	1.5	0.58
House	49.76	1.5	0.69	50.16	1.5	0.58
Lena	49.75	1.5	0.63	50.17	1.5	0.59
Peppers	49.76	1.5	0.68	50.18	1.5	0.58
Man	49.48	1.5	0.68	49.93	1.5	0.56
Average	49.72	1.5	0.67	50.14	1.5	0.58

images. In other words, the visual qualities of the stego images of the two schemes are very good.

4.2 Analysis of the Experimental Results

In the data embedding process, the area for searching for the secret digit in our scheme is in a certain 3×3 block, while the search scope of Chang *et al.*'s scheme is in turtle shells or some 3×3 blocks. Due to the structural features of the turtle shell, there will be some cases in which the secret digit within the turtle is far from the element that corresponds to a cover pixel pair. For example, the distance between $R(3, 3)$ and the secret digit 7 inside the turtle shell reaches 2. And in our scheme, the secret digit 7 at (2, 2) is selected, and the distance is , which is less than 2. The cover pixel pair is replaced with the coordinate value of the nearest element to embed the secret digit, resulting in a small change in the cover pixel value. Thus, the *MSE* value computed by Eq. (2) is small, and the *PSNR* value is high. Therefore, the proposed scheme outperforms the TDH scheme proposed by Chang *et al.* in the quality of the stego images.

In terms of running time, the range of looking for the secret digit in our scheme is only a 3×3 block, while this search range in Chang *et al.*'s scheme is in a set that may involve more than one turtle shell or one 3×3 block, so the proposed scheme consumes less computational cost.

The above analysis shows that the proposed scheme outperforms the TDH scheme proposed by Chang *et al.* in terms of implementation efficiency and image quality. The experimental results validated this analysis.

5 Conclusions

An improved turtle shell-based data hiding scheme is proposed in this paper. The reference matrix used in the proposed scheme is the same as that in the TDH scheme. But, in the proposed scheme, a 3×3 block instead of a hexagon is used to perform the search for the element of the corresponding secret digit in the reference matrix. This improvement reduces the distortion of the embedded image and speeds up the search for elements for data embedding. The experimental results show that our scheme improves the visual quality and processing speed compared to the TDH scheme, in which the image quality is improved by an of 0.42 dB and the execution time is reduced by an average of 0.09 seconds.

References

[1] C. K. Chan and L. M. Cheng, "Hiding data in images by simple LSB substitution," *Pattern Recognition*, vol. 37, pp.469–474, 2003.

[2] C. S. Chan, C. C. Chang, and Y. C. Hu, "A color image hiding scheme using image differencing," *Optical Engineering*, vol. 44, no. 1, pp. 1–9, 2005.

[3] C. C. Chang, Y. C. Chou, and T. D. Kieu, "An information hiding scheme using sudoku," in *Proceedings of The Third International Conference on Innovative Computing Information and Control (ICICIC'08)*, pp. 17–21, June 2008.

[4] C. C. Chang, Y. J. Liu, and T. S. Nguyen, "A novel turtle shell based scheme for data hiding," in *Proceedings of The Tenth International Conference Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP'14)*, pp. 89–93, Aug. 2014.

[5] C. C. Chang, Y. H. Yu, and Y. C. Hu, "Hiding secret data in images via predictive coding," *Pattern Recognition*, vol. 38, no. 5, pp. 691–705, 2005.

[6] I. C. Chang, Y. C. Hu, W. L. Chen, and C. C. Lo, "High capacity reversible data hiding scheme based on residual histogram shifting for block truncation coding," *Signal Processing*, vol. 108, pp. 376–388, 2015.

[7] J. Chen, "A pvd-based data hiding method with histogram preserving using pixel pair matching," *Signal Processing: Image Communication*, vol. 29, no. 3, pp .375–384, 2014.

[8] W. Hong and T. S. Chen, "A novel data embedding method using adaptive pixel pair matching," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 1, pp. 176–184, 2012.

[9] W. Hong, T. S. Chen, and C. W. Shiu, "A minimal euclidean distance searching technique for sudoku steganography," in *Proceedings of International Symposium on Information Science and Engineering*, pp. 515–518, Dec. 2008.

[10] Y. C. Hu, "High capacity image hiding scheme based on vector quantization," *Pattern Recognition*, vol. 39, no. 9, pp. 1715–1724, 2006.

[11] T. D. Kieu and C. C. Chang, "A steganographic scheme by fully exploiting modification directions," *Expert Systems with Applications*, vol. 38, no. 8, pp. 10648–10657, 2011.

[12] H. J. Kim, C. Kim, Y. Choi, S. Wang, and X. Zhang, "Dual-image-based reversible data hiding method using center folding strategy," *Signal Processing*, vol. 60, no. 2, pp. 319–325, 2010.

[13] M. H. Lin, Y. C. Hu, and C. C. Chang, "Both color and gray scale secret image hiding in a color image," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 16, no. 6, pp. 697–713, 2002.

[14] Y. Liu, C. C. Chang, and T. S. Nguyen, "High capacity turtle shell-based data hiding," *IET Image Processing*, vol. 10, no. 2, 130–137, 2016.

[15] C. C. Lo, Y. C. Hu, W. L. Chen, and C. M. Wu, "Reversible data hiding scheme for btc-compressed images based on histogram shifting," *International Journal of Security and Its Applications*, vol. 8, no. 2, pp. 301–314, 2014.

[16] J. Mielikainen, "Lsb matching revisited," *IEEE Signal Process Letters*, vol. 13, no. 5, pp. 285–287, 2006.

[17] C. Qin, C. C. Chang, Y. H. Huang, and L. T. Liao, "An inpainting-assisted reversible steganographic scheme using a histogram shifting mechanism," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 7, pp. 1109–1118, 2013.

[18] C. Qin and Y. C. Hu, "Reversible data hiding in VQ index table with lossless coding and adaptive switching mechanism," *Signal Processing*, vol. 129, pp. 48–55, 2016.

[19] P. Y. Tsai, Y. C. Hu, and H. L. Yeh, "Reversible image hiding scheme using predictive coding and histogram shifting," *Signal Processing*, vol. 89, no. 6, pp. 1129–1143, 2009.

[20] Y. C. Tseng, Y. Y. Chen, and H. K. Pan, "A secure data hiding scheme for binary images," *IEEE Transactions on Communications*, vol. 50, pp. 1227–1231, 2002.

[21] A. Westfeld, "F5: A steganographic algorithm," *Lecture Notes in Computer Science*, vol. 2137, pp. 289–302, 2001.

[22] X. Z. Xie, C. C. Lin, and C. C. Chang, "Data hiding based on a two-layer turtle shell matrix," *Symmetry*, vol. 10, no. 2, 2018.

[23] C. H. Yang, C. Y. Weng, H. K. Tso, and S. J. Wang, "A data hiding scheme using the varieties of pixel-value differencing in multimedia images," *Journal of Systems and Software*, vol. 84, no. 4, pp. 669–678, 2011.

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

Biography

Yu Chen received the B.S. degree in Computer and Application from Hunan University, Hunan, China in 1993, and M.S. degree in Software Engineering from Fuzhou University, Fujian, China, in 2006. Currently, he is an associate professor in the School of Information Science and Engineering, Fujian University of Technology(FJUT), China. His current research interests include information retrieval, data mining, and digital image processing.

Jiang-Yi Lin received the B.S. and M.S. degrees in Computer science and Technology from FuZhou Uniersity, FuJian, China, in 2005 and 2008, repectively. He is currently pursuing the Ph.D degree with the Multimedia and Secure Networking Laboratory (MSN lab), the Department of Information Engineering and Computer Science of Feng Chia University, Taichung, Taiwan. His research interests include image processing, secret sharing and steganography.

Chin-Chen Chang received his Ph.D. degree in computer engineering from National Chiao Tung University. His current title is Chair Professor in Department of Information Engineering and Computer Science, Feng Chia University, from February 2005. He is currently a Fellow of IEEE and a Fellow of IEE, UK. And, since his early years of career development, he consecutively won Outstanding Talent in Information Sciences of the R. O. C., AceR Dragon Award of the Ten Most Outstanding Talents, Outstanding Scholar Award of the R. O. C., Outstanding Engineering Professor Award of the R. O. C., Distinguished Research Awards of National Science Council of the R. O. C., Top Fifteen Scholars in Systems and Software Engineering of the Journal of Systems and Software, and so on. On numerous occasions, he was invited to serve as Visiting Professor, Chair Professor, Honorary Professor, Honorary Director, Honorary Chairman, Distinguished Alumnus, Distinguished Researcher, Research Fellow by universities and research institutes. His current research interests include database design, computer cryptography, image compression, and data structures.

Yu-Chen Hu received his PhD. degree in computer science and information engineering from the Department of Computer Science and Information Engineering, National Chung Cheng University, Chiayi, Taiwan in 1999. Currently, Dr. Hu is a professor in the Department of Computer Science and Information Management, Providence University, Sha-Lu, Taiwan. He is a senior member of IEEE. He is also a member of Computer Vision, Graphics, and Image Processing (CVGIP), Chinese Cryptology and Information Security Association (CCISA), Computer Science and Information Management (CSIM) and

Phi Tau Phi Society of the Republic of China. He servers as the Editor-in-Chief of International Journal of Image Processing from June 2009 to May 2015. In addition, he is the managing editor of Journal of Information Assurance & Security since March 2009. He is the associated editor of Human-centric Computing and Information Sciences since Feb. 2011. He joints the editorial boards of several other journals. His research interests include digital forensics, information hiding, image and signal processing, data compression, information security, and data engineering.