*Journal of Network Intelligence*
**Taiwan Ubiquitous Information**
 ISSN 2414-8105 (Online)
Volume **3**, Number **2**, May **2018**

# Data Hiding Scheme Based on A Flower-Shaped Reference Matrix

Chin-Feng Lee

Department of Information Management,
Chaoyang University of Technology Taichung 41349, Taiwan

Corresponding Author: lcf@cyut.edu.tw

Ya- Chen Li

Collage of Information,
Liaoning University Shenyang 110000, China

liyachen19940622@163.com

Shu-Chuan Chu, John F. Roddick

College of Science and Engineering,
Flinders University Sturt Rd,

Bedford Park SA 5042, South Australia
jan.chu@flinders.edu.au, john.roddick@flinders.edu.au

Received Jan. 2018 Revised Feb. 2018

Abstract. *Data hiding is a widely used technique to embed secrets in multimedia area so to achieve lower distortion and higher embedding capacity. Up to now, many methods which focus on solving these two problems have proposed constantly. Previous methods use various shaped shells to carry secret data in an image, but their schemes have high distortion of images and low capacity of carrying secrets due to their simple geometry layouts. We find out adjacent pixels have similar values, which means we can utilize this large area, and, therefore, we manipulate the data embedding and extracting on a difference-coordinate plan instead of the traditional pixel-coordinate plan. In this paper we propose a method aiming at solving these two problems mentioned above. This scheme embeds our secrets by using 3 pixels every time with the guidance of the flower-shaped reference matrix under a different coordinate system. The flower-shaped reference matrix combines three parts including petal matrix, calyx matrix and stamen matrix for secret embedding, which brings outstanding payload with good visual quality.*
**Keywords:** Steganography; Data hiding; Data embedding and extracting.

1. **Introduction.** When people want to deliver secret messages to others and prevent these messages from malicious attacks, one of the accessible ways is data hiding technique. Data hiding technique, a significant subject of information security, is widely used to transfer secret messages to others safely on public channels instead of highly costly and conspicuous private channels.

Data hiding focuses on finding a secure way to embed secrets in multimedia. Pictures, known as a common multimedia, can be a perfect means to carry secret messages. At the current stage in steganography, gray-scale images are common and convenient carriers as we know, owing to the value of every gray-scale image pixel ranging from 0 to 255, so a pixel can be easily represented by 8 bits in a binary system. Performance of each

data hiding method can be estimated by two aspects: 1. We want lower distortions of images after embedding secrets; 2. We need a higher capacity of carrying secret messages. Generally, a higher embedding capacity will result in higher distortions of stego-images, and vice versa. Thus, how to find a feasible way to make a trade-off is a big problem that the data hiding technique faces nowadays.

In 1989, Turner [1] presented a method for steganography, which describes a concept utilizing replacement of the least-significant bits (LSB) of each cover pixels value from a host image by a secret message to carry secrets. LSB for data hiding is the simplest and achievable method with a satisfactory capacity of carrying secret digits; although it can escape from human eyes, it is vulnerable under the malicious attacks based on the statistical analysis. Fortunately, decreasing the number of secret messages that we hide in the stego image is a feasible way to solve this problem. In 2001, Wang [2] proposed a scheme to reduce the distortion of stego images. And then in 2002, Tseng et al. [3] introduced a method which can hide $[\log_2(n+1)]$ bits of a secret string by changing at most 2 bits from a $n$ sized binary image block.

In 2006, Mielikainen [4] presented a modified LSB method called LSB to match a revisited approach which devotes to control the distortion of host images in a lower level with the same payload. In this way we can generate stego images under the guidance by both of the corresponding two original images pixels and two secret digits. Compared with a traditional LSB method, Mielikainens method performs better in visual imperceptibility apparently. Subsequently, in the same year Zhang and Wang [5] pointed out that LSB matching revisited method exploits incompletely, and introduces a new stenographic called exploiting modification direction (EMD), in which each unit composed of n pixels of a host image can carry one secret digit in (2n+1)-ary notational system during embedding processes every time, and only one pixel of the unit is modified by 1 every time. Therefore, it shows a larger payload and better quality of stego images. Since then, many methods emerged contain [6, 7, 8, 9, 10]. In 2009, Chao et al. [9] proposed a novel scheme called diamond encoding. They use 2 pixels every time to hide/extract a secret digit after calculating the diamond characteristic value, so that it can conceal a $(2k^2+2k+1)$ ary digit each time during embedding processes ($k$ is a embedding parameter). Chang et al. [10] proposed a method which utilizes the Sudoku solutions to hide secrets in host images. Further, in order to improve the ability of payload, Chang et al. proposed a turtle-shell based data hiding method [11]. This method provides an easy way to establish a layout looks like a turtle shell, so every secret digit ranging from 0 to 8 can be embedded by 2 pixels each time. During the process of extraction, we can use the same layout conveniently. In 2016, Liu et al. [12] introduced a scheme that can make a flexible trade-off to balance payload and visual quality by changing the size of turtle shells. Research for improving payload never stops. In 2017 Jin et al. [13] introduced a minimized turtle-shell scheme devoting to decreasing the distortion of stego images. In 2018, Leng [14] proposed a scheme similar with the scheme based on a turtle-shell based scheme. Lengs method arranges a layout based on a regular octagon-shape to improve the capacity of carrying secret digits. The scheme we proposed in this paper places octagon shells as a part of geometric layout combining octagon-shaped shells in order to hide more secret digits into a cover image with a high visual quality. Of course, the methods for hiding information are booming now, and more and more scholars like [15, 16, 17, 18, 19] are paying more attentions to reversible data hiding (RDH) because RDH can restore to the original carrier after out messages have been extracted.

After studying previous methods mentioned above, we refer to the geometric layout in [14] as a kind of inspiration to propose a steganographic method based on a flower-shaped reference matrix extended on a pixel-differencing plan to hide secret completely.

The flower-shaped reference matrix combines three parts: petal matrix, calyx matrix and stamen matrix) for secret embedding, bringing an outstanding payload with a good visual quality.

This paper is organized as follows. Section 2 introduces the related works. Section 3 introduces our proposed method focusing on the petal, the calyx, and stamen embedding. The experimental results show the advantages of our method in Section 4. Section 5 presents our conclusions.

2. **Related Work.** In this section, we will introduce the Zhang and Wang's EMD scheme, Chang et al.'s turtle-shell based scheme and Leng's octagon-shaped shells based scheme chronologically. Before introducing these three methods, we show the definition of symbols used in this paper. Definition of symbols: H: height of an image. W: weight of an image. Cover/host image: represents the original gray-scale image. Stego image: represents the gray-scale image after embedding a secret message. i: represents the index of pixels. $(p_{(i-1)}, p_i, p_{(i+1)})$: a triple of consecutive cover pixels. $d_1, d_2$: represents the difference-values of pixel pairs $(p_{(i-1)}, p_i)$ and $(p_{(i+1)}, p_i)$ from a cover image, respectively. $M(d_1, d_2)$: represents the value with the guidance of flower-shaped reference matrix mod: represents the modulo operation. $p_i$': represents the stego-pixle of $p_i$ after carrying secret by Least Significant Bit Substitution method. $d_1', d_2'$ : represents the difference-values of pixel pairs $(p_{(i-1)}, p_i')$ and $(p_{(i+1)} - p_i')$, respectively. $p'_{(i-1)}, p'_{(i+1)}$ : represents the stego-pixel values of $p_{(i-1)}$ and $p_{(i+1)}$ after embedding secret, respectively. $l_s$ : represents the length of secret that we are going to hide $M(d_1', d_2')$ : represents the value of the secret we are going to embed from the reference matrix $\sigma$ : represents standard deviation. num: represents the number of statistics in the histogram.

2.1. **Zhang and Wangs EMD scheme.** Zhang and Wangs EMD scheme can embed a $(2n+1) - ary$ notational secret data under a group of $n$ cover pixels from the host image every time. They showed the EMD scheme could achieve embedding efficiency and secrecy with low distortions. We will briefly introduce EMD's embedding procedure First, divide a cover image into a series of non-overlapping groups, and each group is composed of $n$ pixels which are $G = (p_1, p_2, , p_n)$. Then, convert a binary secret message into a sequence of secret digits in (2n +1)-ary notational system. Every secret digit can be shown as $s_j (j = 1, 2, , l)$, where l depends on $n$. Apply the EMD method on the group G by Eq. (1) where "$mod$" represents a modulo operation. Equation (2) calculates how to carry a $(2n+1) - ary$ secret digit $s_j$.

$$\rho = \ f\,(p_1 \ , \ p_2 \ , \ldots , \ p_n) = \left[ \sum_{i=1}^{n} (p_i i) \right] \ mod \ \ (2n+1) \tag{1}$$

$$\mathrm{D} = (\mathrm{s}_j - y)\, mod \ \ (2n+1) \tag{2}$$

Equation (3) can be used to evaluate how to change a certain $p_i$ value of $G$ by at most adding or substrating one.

$$p_i^{'} = \begin{cases} p_i, \ if \ s_j = \\ p_D + 1, \ if \ s_j \neq, and \ D \leq n \\ p_{(2n+1)-D} - 1, \ if \ s_j \neq, and \ D \leq n \end{cases} \tag{3}$$

Now, we demonstrate when $n = 2$, and illustrate with Figure1.

If the cover pixel pair is $(p_1, p_2) = (1, 2)$, $\sigma$ is 0 according to Eq. (1). When the to-be-embedded secret digit $s_j = 2$, the stego-pixel pair will be $(p_1', p_2') = (1, 3)$ according to Eqs. (2) and (3). When the receiver wants to extract the secret, they can also utilize

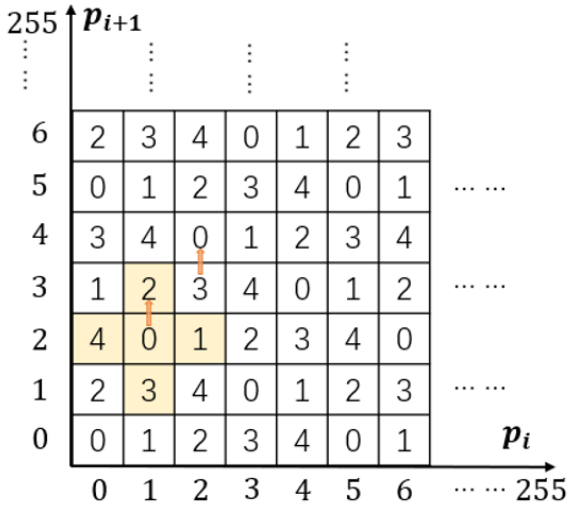


FIGURE 1. : A reference matrix based on EMD scheme, when $n = 2$

the function shown in Eq. (1). That is, the secret digit 2 can be extracted. This method ensures a high data payload (about 1.16bpp when $n = 2$) and a good image quality (about 52dB measured by Peak signal-to-noise ratio, often abbreviated PSNR).

2.2. **Chang et al.s turtle shell-based scheme.** In 2014, Chang et al. propose an EMD scheme based on a turtle-shell reference matrix. In this scheme every two pixels can carry a secret data range from $(000)_2$ to $(111)_2$ each time. We will generally introduce this scheme as follows: First of all, Chang et al. [11] build a 256×256 reference matrix containing as many turtle shells as possible to hide the secret data and each turtle shell is a hexagon that contains 8 different distinct numbers ranging from 0 to 7, including 6 edge digits and 2 back digits. In the reference matrix, the turtle shell is arranged one by one without overlapping. The turtle-shell reference matrix M is shown in Figure 2. Figure 2 shows that the value difference between the bottom row and the upper row is 2, and the next value difference is 3; then it is 2 again. By applying this rule, alternately add 2 and 3 to every row to complete the entire matrix. Therefore, the value difference between two adjacent numbers in the same row of the reference matrix is set to "1", and the value difference between two adjacent numbers in the same column is set alternately to "2" and "3". They keep going to continuously write down 0 to 7 in every row. Every turtle shell contains 8 numbers ranging from $(000)_2$ to $(111)_2$, so that each cover pixel pair is expressed as $(p_i, p_{(i+1)})$ , can carry a 3-bit digit $s_j$. Assume that the grayscale cover image I with sized of $H \times W$ is composed by $I = \{p_i \| i = 1, 2, \ldots (H \times W)$. To embed secret digits, the location of each pixel pair $(p_i, p_{(i+1)})$ will be determined as $M(p_i, p_{(i+1)})$ in the reference matrix $M$, where $p_i$ and $p_{(i+1)}$ are the column value and row value, respectively.

All elements of turtle-shell reference matrix M are categorized into 3 groups: digits inside, digits on the edge, and digits off the turtle. When $M(p_i, p_{(i+1)}) = s_j$, the pixel pair remains unchanged, which causes the stego-pixel pair is same as the cover pixel. On the contrary, when $M(p_i, p_{(i+1)}) \neq s_j$, there are three possible concealment strategies can be explored. The three situations are $M(p_i, p_{(i+1)})$ inside, on the edge, and off a turtle shell, respectively. Firstly, when $M(p_i, p_{(i+1)})$ is inside a turtle shell, find a digit $M(p'_i, p'_{(i+1)})$ within a turtle shell where $M(p_i, p_{(i+1)})$ belongs to such that $M(p'_i, p'_{(i+1)})$ is equal to $s_j$. Then the cover pixel pair is changed into the corresponding stego-pixel pair $(p'_i, p'_{(i+1)})$. Secondly, when $M(p_i, p_{(i+1)}) \neq s_j$, and $M(p_i, p_{(i+1)})$ is on the edge of a turtle shell, then a new pixel pair $(p'_i, p'_{(i+1)})$ is the stego-pixel pair such that $M(p'_i, p'_{(i+1)})$

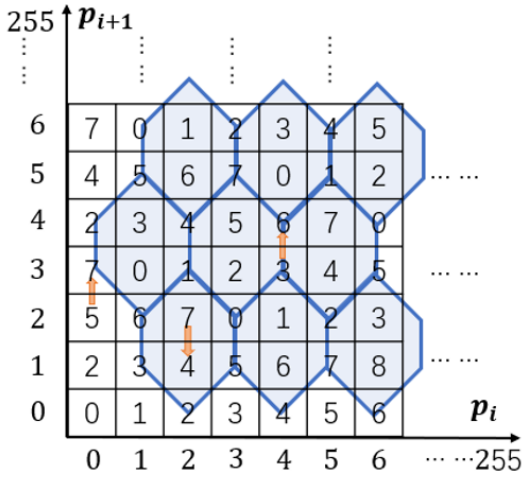


FIGURE 2. A turtle-shell reference matrix

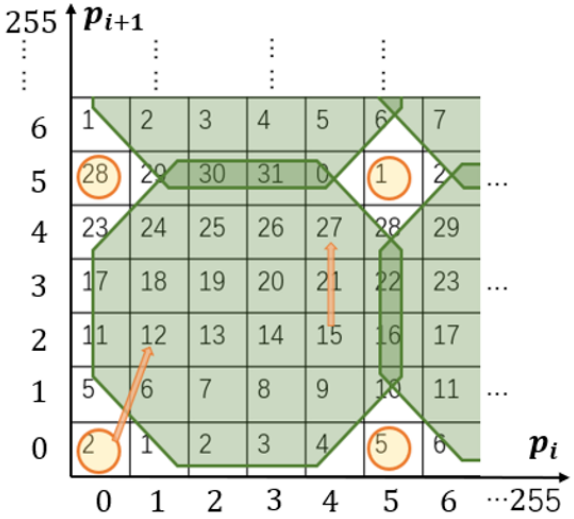


FIGURE 3. A reference matrix based on regular octagon-shaped shells

is equal to $s_j$ which is located from one of 3 turtle shells where $M(p_i, p_{(i+1)})$ belongs to, and the distance between $(p_i, p_{(i+1)})$ and $(p'_i, p'_{(i+1)})$ is the shortest for sure. Thirdly, when $M(p_i, p_{(i+1)}) \neq s_j$, and $M(p_i, p_{(i+1)})$ is off the turtle: We need to find a new pixel pair $(p'_i, p'_{(i+1)})$ such that $M(p'_i, p'_{(i+1)})$ is equal to $s_j$ which is located at 3×3 sub-block where $M(P_i, P_{(i+1)})$ belongs to, and the distance between $(p_i, p_{(i+1)})$ and $(p'_i, p'_{(i+1)})$ needs to be the shortest one for sure. Finally, we got a stego-mage that is composed of new pixels $(p'_i, p'_{(i+1)})$ after all secret data are embedded. Example 2. Assume that we want to embed three secret digits "467" into the image as shown in Figure 2. According to the embedding rules, tree cover pixel pairs (2,2), (4,3) , and (0,2), are changed into the pixel pairs which are (2,1),(4,4),and(0,3), relatively.

2.3. **Lengs octagon-shaped shells.** Leng's method is similar with Chang's turtle-shell based scheme, but he improves the payload from 1.5bpp to 2.5 bpp. In this method every pixel pair $(p_i, p_{(i+1)})$ from a cover image can carry 5 bits of secret data. We will introduce this scheme briefly as follows: First of all, Figure 3 shows a reference matrix based on regular octagon-shaped shells. For every column of the matrix, they set the difference as

1, and the range is from 0 to 31. For every row of the matrix, they set the differences as 5, 6, 6, 6, and 6 in turns, and the range is also from 0 to 31. Finally, the matrix, expressed as M, is composed of many octagons. Every octagon contains 32 digits expressed from $(00000)_2$ to $(11111)_2$, so that each pixel pair $(p_i, p_{(i+1)})$, can carry 5 bits of the secret data. The embedding procedure and extraction procedure are similar with that of the turtle-shell based scheme.

Example 3. Assume that two secret digits 12 and 27 will be embedded into the image as shown in Figure 3. According to the embedding rule developed by Leng, the pixel pairs as (0,0) and (4,2) will be changed to another two pixel pairs (1,2) and (4,4), respectively.

3. **Proposed Method.** In this section we will introduce how flower-shaped shell schemes work. Our scheme embeds a secret message by using 3 pixels every time with the guidance of the flower-shaped reference matrix under a difference-coordinate system. The flower-shaped reference matrix combines three parts: petal matrix, calyx matrix and stamen matrix for secret embedding, which brings a great payload with a good visual quality.

3.1. **Matrix Construction Procedure.** A coordinate system $(d_1, d_2)$, where$d_1$ and $d_2$ range from -255 to 255, represents the difference-value of pixel pairs $(p_{(i-1)} - p_i)$ and $(p_{(i+1)} - p_i)$ , respectively. There is a large number of difference-values are close to 0s, due to the feature of images that adjacent pixels have nearly similar values. Therefore, when$d_1$ and $d_2$ range from -1 to 1, we arrange a 33 rectangle-shaped matrix called stamen matrix which is marked in orange in Fig.4. The stamen matrix can be expressed by Eq. (4). Every pair of $(d_1, d_2)$ in the stamen matrix can carry a secret digit ranging from $(000)_2$ to $(111)_2$.

$$M(\ d_1\ ,\ d_2\ ) = (\ d_1\ + 3d_2 + \ 4)\ \bmod\ 8,\ \text{if}\ \ d_1\ , d_2 \in \{-1,\ 0,\ 1\} \tag{4}$$

Then settle the second part of the big matrix, when either $d_1$ or $d_2$ is equal to 0. The calyx matrix is marked in blue as shown in Figure 4. The Eq. (5) describes the calyx matrix.

$$M(d_1,\ d_2) = \begin{cases} d_1\ mod\ 4,\ if\quad d_1\,\{-1, 0,\ 1\}, d_2 = 0 \\ d_2\ mod\ 4,\ if\quad d_1 = 0,\ d_2\,\{-1, 0,\ 1\} \end{cases} \tag{5}$$

The positive axis of d1 ranges from 2 to 255, and the negative axis of d1 ranges from -2 to -255; meanwhile, d2 is kept as 0. We can obtain the other 2 calyxes on the positive axis and on the negative axis of d2 by transposing calyxes, respectively. Every element $M(d_1, d_2)$ in these calyxes can carry secret digits from $(00)_2$ to $(11)_2$. In the end we arrange the third part of the big matrix called petal that is marked in green as shown in Figure 4. And it can be describe as follow: For every column of the matrix, we set difference value as 1, and the range is from 0 to 31. For every row of the matrix, we set difference value as 5, 6, 6, 6, 6, in turns, and the range is from 0 to 31, like that proposed by Leng in 2017. Then we can get the whole matrix called flower-shaped reference matrix composed of petal matrix, calyx matrix and stamen matrix as shown in Figure 4.

3.2. **Payload Calculation.** When we need to conduct information hiding in an unreliable environment, we expect that the total volume of secret messages will be as much as possible during one transmission. Each cover image's payload depends on the resolution of the host image. Figure 5 illustrates the procedure flowchart of calculating the length of the secret message.

**Step 1:** Extract a triple of consecutive cover pixels $(p_(i-1), p_i, p_(i+1))$, where $i = 2, 5, \ldots,$(W×H -($W \times H$mod 3)-1). Convert a message S to a bitstream. First, extract 3

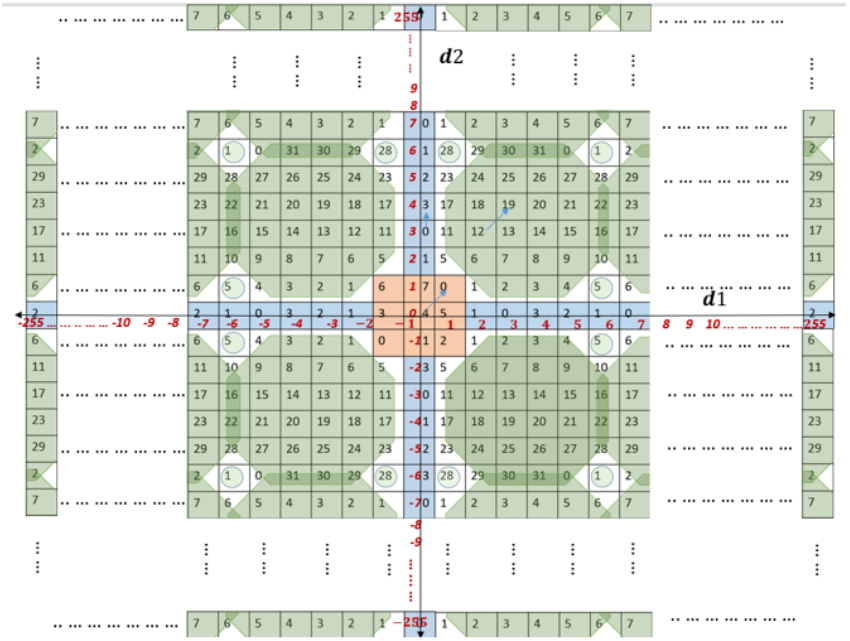


FIGURE 4. A reference matrix from the proposed scheme

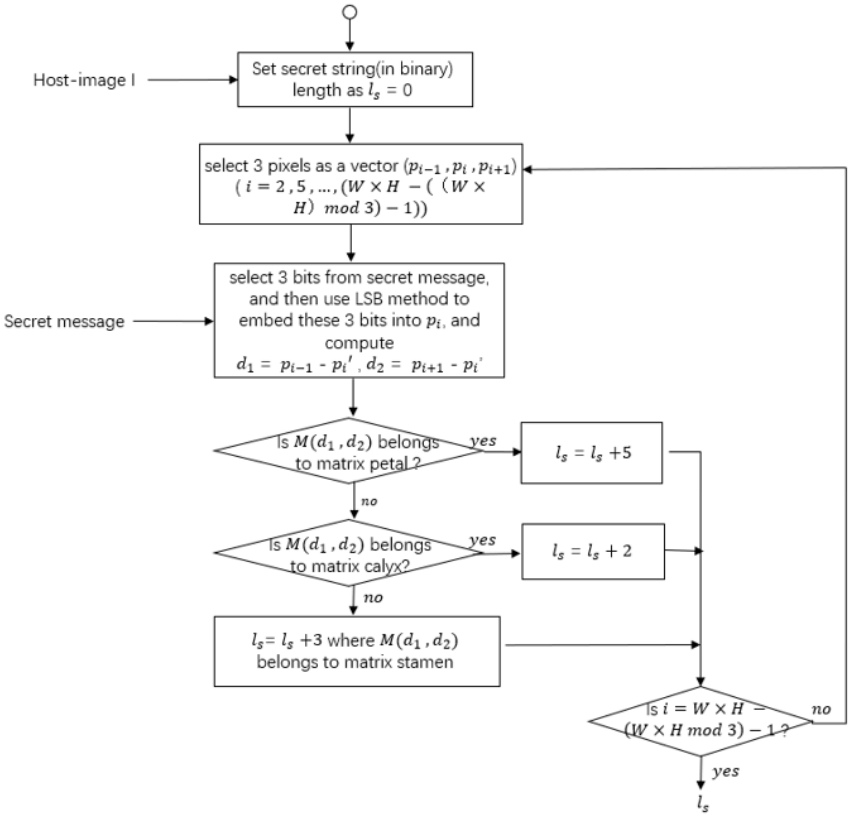


FIGURE 5. The procedure flowchart of calculating the length of the secret message

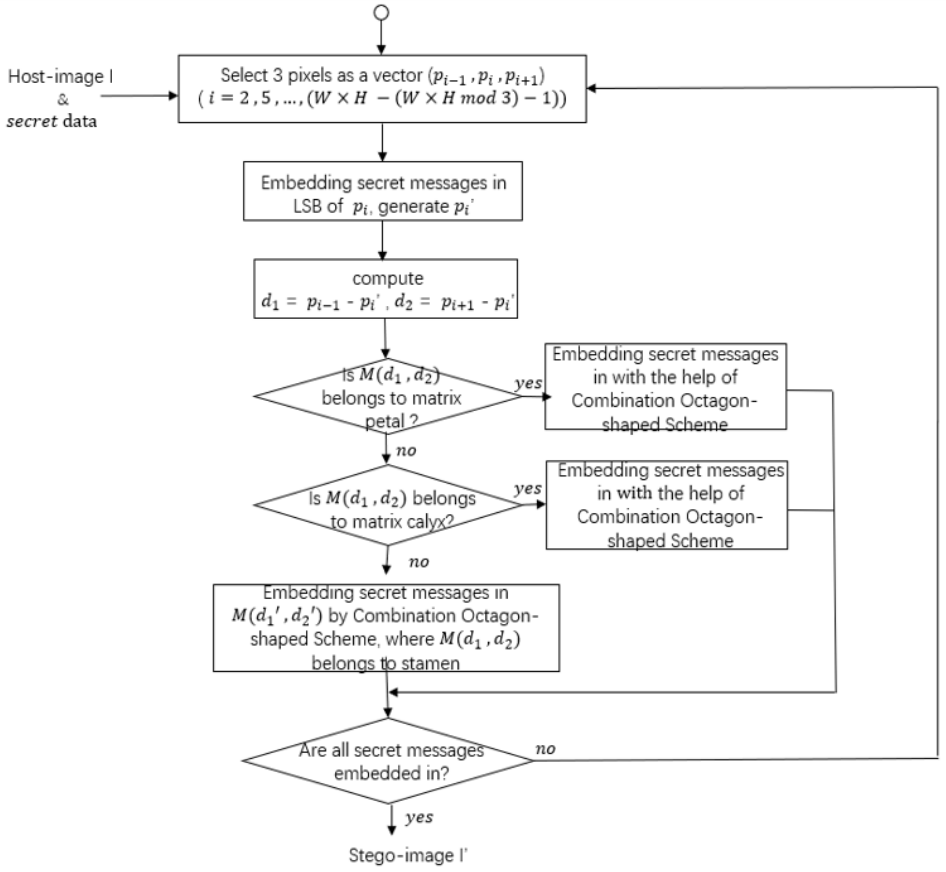


FIGURE 6. The procedure flowchart of embedding secrets based on the proposed scheme

bits from the secret string and embed the segment into the host image using $p_i$ by LSB substitution method, and update $l_s = l_s + 3$, where $l_s$ represents the length of the secret string which is going to be embedded into a cover image. Relative to $p_i$ is a cover pixel, and then $p_i'$ is a camouflaging pixel.

**Step 2:** Calculate the difference values $d_1 = p_{(i-1)} - p_i'$ and $d_2 = p_(i+1) - p_i'$, respectively.

**Step 3:** Recognize $M(d_1, d_2)$ belonging to which part of flower matrix: If it belongs to the calyx area, then $l_s = l_s + 2$; if it belongs to the stamen part, $l_s = l_s + 3$; otherwise, $l_s = l_s + 5$, which means it belongs to the petal matrix.

**Step 4:** Repeat from Steps 1 to 3 until all pixels in the cover image are completely processed. Return the payload length$l_s$. Our scheme embeds a 3-bit sub-secret string to the LSB pixel of $p_i$, and we also embed a $l_s$-bit sub-secret string to the pair of difference values $(d_1, d_2)$. The binary value s is converted to its corresponding decimal value $s_d$. The length $l_s$ of to-be-embedded secret data $s$ depends on where the pair $(d_1, d_2)$ locates on the flow-shaped reference matrix.

3.3. **Embedding Procedure.** We can embed a secret message which is in binary system into a host image as shown in Figure 6. During the procedure of embedding a secret, our scheme is efficient due to the embedding time less than 25 seconds with more than 2.6 bit per pixel (bpp). In Figure 6, it shows the embedding procedure.

**Step1:** Extract a triple of consecutive cover pixels $(p_{(i-1)}, p_i, p_{(i-1)})$, where $i = 2, 5, \ldots,$ (W×H -(W×H mod 3)-1).

**Step2:** Embed 3 bits of secret message into the LSB of $p_i$ to generate a stego image pixel $p'_i$. Then, compute $d_1 = p_{(i-1)} - p'_i, d_2 = p_{(i-1)} - p'_i$.

**Step3:** If the decimal secret $s_d$ is equal to $M(d_1, d_2)$, then keep $d_1, d_2$ unchanged; otherwise, embed $s_d$ as the following rules:

**Case 1** : $M(d_1, d_2)$ belongs to the petal matrix, that means this pair of $(d_1, d_2)$ can carry 5 digits of secret message ranging from $(00000)_2$ to $(11111)_2$.While the sub-secret $s_d$ is unequal to $M(d_1, d_2)$, find the pair $(d'_1, d'_2)$ which has the shortest distance with $(d_1, d_2)$ and is equal to sub-secret $s_d$ with the guidance of matrix flower. Change $(d_1, d_2)to(d'_1, d'_2)$ later, according to $d'_1 = p'_{(i-1)} - p'_i, d'_2 = p'_{(i-1)} - p'_i$ , to generate the stego-pixels: $p'_{(i-1)}$ and $p'_{(i-1)}$.

**Case 2:** $M(d_1, d_2)$ belongs to the stamen matrix, that means this pair of $(d_1, d_2)$ can carry 3 digits of secret message ranging from $(000)_2$to $(111)_2$. While the sub-secret $s_d$ is unequal to $M(d_1, d_2)$, find the pair $(d'_1, d'_2)$ that is equal to sub-secret $s_d$ with the guidance of matrix flower. Change $(d_1, d_2)to(d'_1, d'_2)$ later, according to $d'_1 = p'_{(i-1)} - p'_i, d'_2 = p'_{(i-1)} - p'_i$ , to generate the stego-pixels: $p'_{(i-1)}$ and $p'_{(i-1)}$.

**Case 3:** $M(d_1, d_2)$ belongs to the calyx matrix, that means this pair of $(d_1, d_2)$ can carry 2 digits of secret message ranging from $(00)_2$ to $(11)_2$. While sub-secret $s_d$ is unequal to $M(d_1, d_2)$, find the pair $(d'_1, d'_2)$ that is equal to sub-secret $s_d$ with the guidance of matrix flower. Change $(d_1, d_2)to(d'_1, d'_2)$ later, according to $d'_1 = p'_{(i-1)} - p'_i, d'_2 = p'_{(i-1)} - p'_i$, to generate the stego-pixels: $p'_{(i-1)}$ and $p'_{(i-1)}$. So far, the triple of consecutive stego-pixels $(p'_{(i-1)}, p'_i, p'_{(i-1)})$ are generated.

**Step 4:** Repeat Steps 1 to 4 until all secret messages are embedded. We obtain the stego image finally.

Example 4. When we want embed a secret digit "3" by the flower-shaped reference matrix as shown in the above mentioned Figure 4, and a secret digit "7" by LSB substitution under the triple cover pixels $(p_{(i-1)}, p_i, p_{(i-1)}) = (79, 74, 82)$, the LSB substitution procedure changes the central pixel from 74 $(1001010)_2$ to 79 $(1001111)_2$ such that $(p_{(i-1)}, p'_i, p_{(i-1)}) = (79, 79, 82)$. Then compute $(d_1, d_2) = (0, 3)$, and find $M(0, 4)is3$, so that $(d'_1, d'_2) = (0, 4)$, and finally change the stego vector $(p'_{(i-1)}, p'_i, p'_{(i-1)}) = (79, 79, 83)$. When we want embed a secret digit "2" by the flower-shaped reference matrix as shown in 5: The procedure flowchart of calculating the length of the secret message 4, and a secret digit "5" by LSB substitution under the triple cover pixels $(p_{(i-1)}, p_i, p_{(i-1)}) = (77, 74, 77)$, the embedding procedure first has $(p_{(i-1)}, p'_i, p_{(i-1)}) = (77, 77, 77)$, next compute $(d_1, d_2) = (0, 0)$, and last $(d'_1, d'_2) = (-1, 1)$. Therefore, the triple stego-pixels are $(p'_{(i-1)}, p'_i, p'_{(i-1)}) = (78, 77, 76)$. When we want embed a secret digit "14" the flower-shaped reference matrix as shown in Figure 4, and a secret digit "7" by LSB substitution under the triple cover pixels $(p_{(i-1)}, p_i, p_{(i-1)}) = (49, 45, 50)$, apply LSB procedure to change $(p_{(i-1)}, p'_i, p_{(i-1)}) = (49, 47, 50)$ and compute $(d_1, d_2) = (2, 3)$ , so that $(d'_1, d'_2) = (3, 4)$. At last, the triple stego-pixels are $(p'_{(i-1)}, p'_i, p'_{(i-1)}) = (50, 47, 51)$

3.4. **Secret Extraction Procedure.** First, select the triple of consecutive stego-pixels $(p'_{(i-1)}, p'_i, p'_{(i+1)})$, from a stego-image. The secret data can be obtained from the 3 least significant bits of $p'_i$. Then compute $d'_1 = p'_{(i-1)} - p'_i and d'_2 = p'_{(i+1)} - p'_i$. Based on the location indication of the two values $(d'_1, d'_2)$ on the flower-shaped reference matrix, $M(d'_1, d'_2)$ is the secret data. Repeat to process the secret extraction procedure, and then we can get the whole secret message. Example 5. Assume $(p'_{(i-1)}, p'_i, p'_{(i+1)})$ from a stego-image is (83, 79, 79). According the extraction procedure, we can extract secret 7 (111)2 from $p'_i$ and 3 (11)2 from $(d'_1, d'_2)$, respectively. What about the triples $(p'_{(i-1)}, p'_i, p'_{(i+1)}) = (78, 77, 76)$? Secret data 5 (101)2 and 2 (101)2 can be extracted from $p'_i and (d'_1, d'_2)$, respectively. Let's

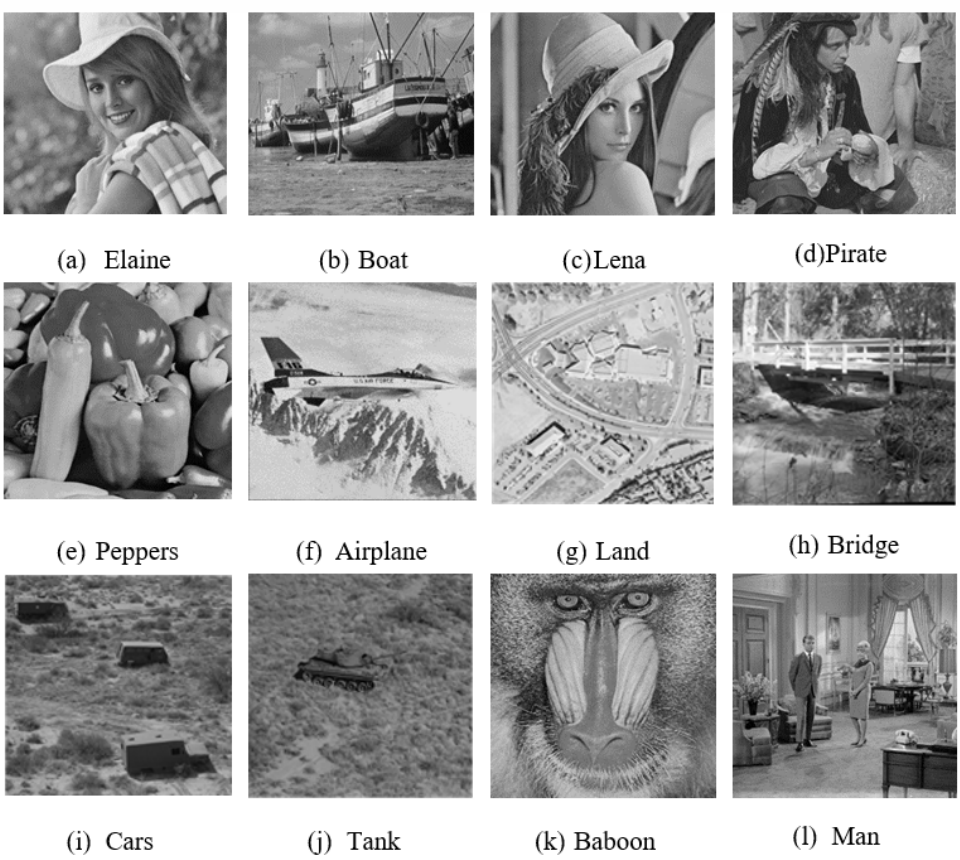


FIGURE 7. Twelve text images

look at what the secret message will be extracted from the $(p'_{(i-1)}, p'_i, p'_{(i+1)}) = (51, 47, 50)$. That are 7 $(111)_2$ from $p'_i = 47$ and 14 from $(d'_1, d'_2) = (4, 3)$.

4. **Experimental Result.** In this section, we performed some experiments to demonstrate our performance in terms of embedding payload and image quality. Figure 7 shows this experiment uses seven 512×512 grayscale test images: (a)Elaine (b)Boat (c)Lena (d)Pirate (e)Peppers (f)Airplane (g)Land (h)Bridge (i)Cars (j)Tank (k)Baboon (l)Man, respectively. The secret message is generated. All the experimental results are obtained based on the MATLAB R2010a platform. The system uses a random function supported by MATLAB to generate a sequence of bit streams as secret data. In order to quantify the performance of the proposed method, the embedding capacity (EC), peak signal-to-noise ratio (PSNR) and other parameters are measured in the experiment. Among them, EC refers to the number of secret data embedded in a test image; while PSNR is a kind of objective criteria for the evaluation of the image, the greater the PSNR, the better the quality of the image. PSNR can be calculated as Eqs. (6)-(7), where in, and represents the height and width of the cover image, respectively, $p_{(i,j)}$ represents the original cover pixels, and $p'_{(i,j)}$ represents the camouflage image pixels, respectively.

$$\text{PSNR} = 10 \ \log_{10} \left( \frac{255^2}{MSE} \right) \tag{6}$$

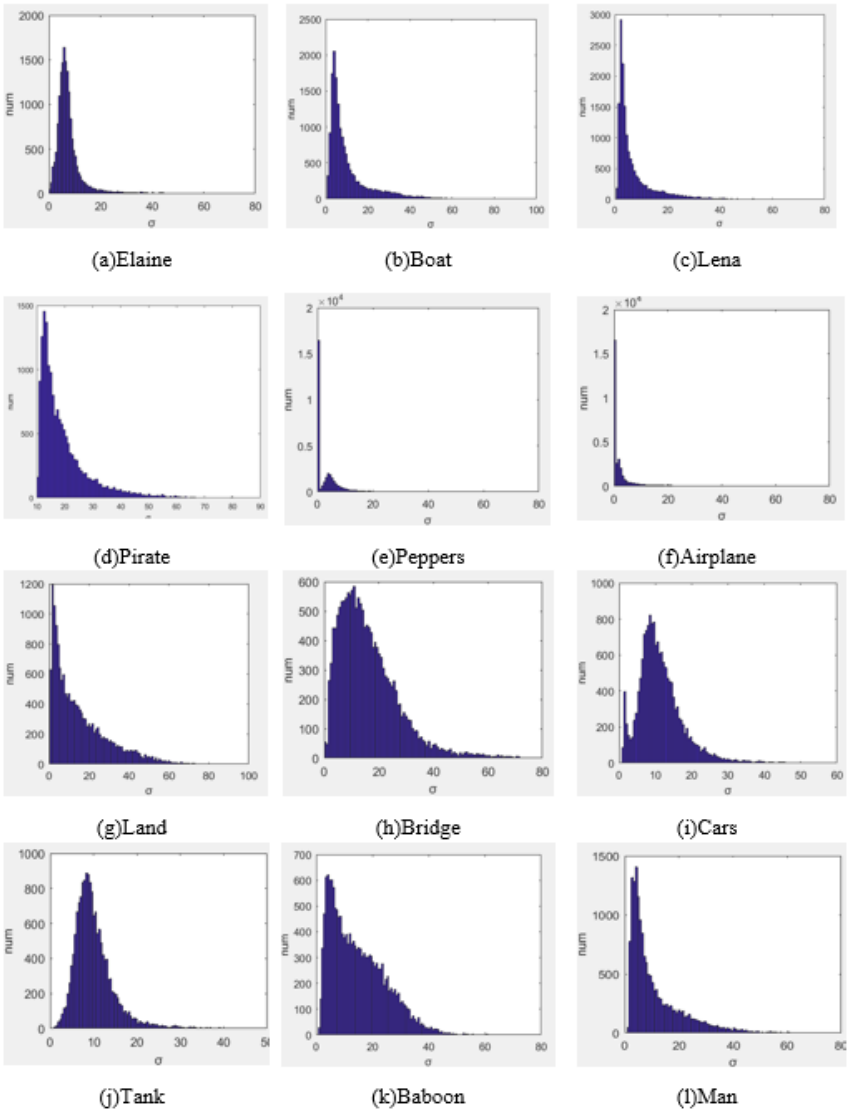


FIGURE 8. Twelve histograms for identifying image types

Where MSE is defined as Eq. (7) shows:

$$\text{MSE} = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} ( \ p_{i,j} \ - \ p_{i,j}{}' ) \tag{7}$$

As we know, a cover image can be categorized into 2 groups: smooth images and complex images. We firstly divide each test image into 4×4 non-overlapping blocks and then calculate the block standard deviations. Next we use the histogram to present the relationship between the standard deviation and the number of blocks as shown in Figure 8. From Figure 8, we classify images (a)-(f) are six smooth images whose block standard deviations are mostly around 0s. Images (g)-(l) are another six complex images of which variances are hardly around over 0.

After compared with embedding by smooth images and complex images, we can clearly demonstrate that our scheme performs better in smooth images with a high visual quality which is more than 40 dB and outstanding payload which is over 2.6 bpp. Concrete experimental results are shown in Table 1 and Table 2. The smooth regions are more suitable for embedding secret information due to the smaller difference between the pixel

(a)Original image (b)Stego image

FIGURE 9. Original image and stego-image based on baboon

TABLE 1. Smooth images' PSNR & Payload

| | (a)Elaine | (b)Boat | (c)Lena | (d)Pirate | (e)Peppers | (f)Airplane |
|---|---|---|---|---|---|---|
| **PSNR (dB)** | 40.4181 | 40.3879 | 40.3699 | 40.3774 | 40.3431 | 40.4213 |
| **EC (bpp)** | 2.6379 | 2.6037 | 2.6134 | 2.6022 | 2.6211 | 2.6135 |

TABLE 2. Complex images' PSNR & Payload

| | (f)Land | (g)Bridge | (h)Cars | (i)Tank | (j)Baboon | (l)Man |
|---|---|---|---|---|---|---|
| **PSNR (dB)** | 38.7091 | 38.7694 | 39.1367 | 39.1366 | 38.1356 | 38.3421 |
| **EC (bpp)** | 2.5934 | 2.6172 | 2.6481 | 2.6539 | 2.6402 | 2.6287 |

values. We compare the original Baboon image, and its camouflage one has low PSNR of 38.1356(dB). Though Babbon is classified as a complex image, we can't tell the difference between the original image and the stego-image with human beings eyes as shown in Figure 9.

4.1. **Image quality evaluation.** In the following experiment, six text images are used: (a)Elaine, (b)Boat, (c)Lena, (d)Pirate, (e)Peppers, and (f)Baboon as shown in Figure 10. The results of experiments are shown as Table 3. Compared with the schemes proposed by Turtle-shaped based scheme and regular-octagon based scheme, ours performs outstandingly in terms of payload with more than 2.6 bpp, and the satisfied visual quality of 40 dB on average.

4.2. **Result and Comparisons.**

5. **Conclusion.** Transferring more secrets messages at once is safer under most situations because delivering stego-image frequently will cause suspects. The results demonstrate our scheme has a very high payload so that our method can carry the largest secret

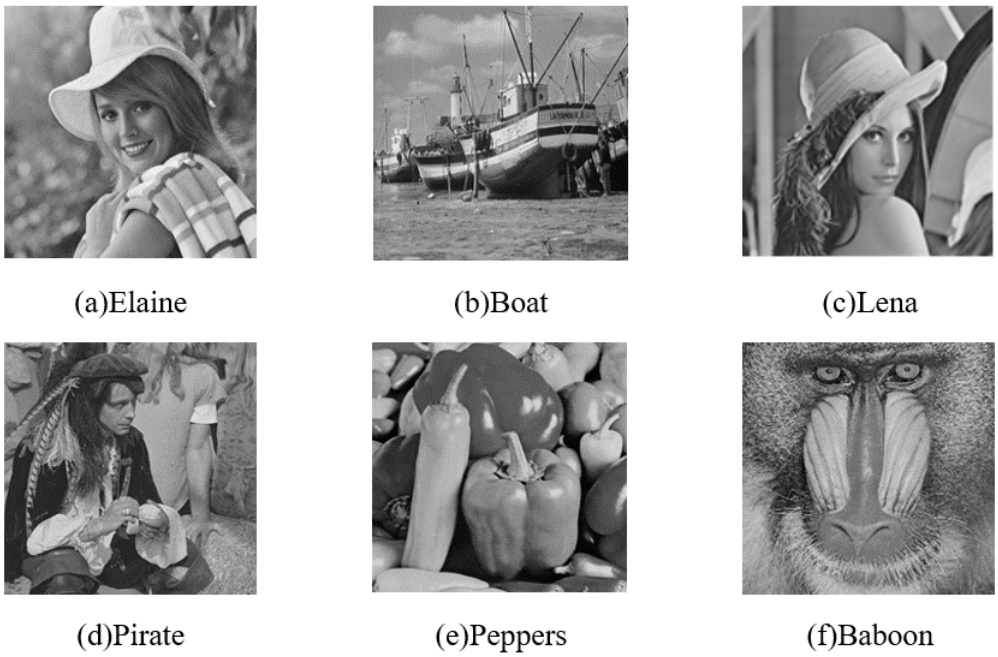


FIGURE 10. Six text images used in experiment

TABLE 3. Complex images' PSNR & Payload

| | **Proposed Scheme** | | **Turtle-shaped Scheme** | | **Regular-Octagon Scheme** | |
|---|---|---|---|---|---|---|
| | PSNR (dB) | EC (bpp) | PSNR (dB) | EC (bpp) | PSNR (dB) | EC (bpp) |
| **Elaine** | 40.4181 | 2.6379 | 49.41 | 1.5 | NA | NA |
| **Lena** | 40.3699 | 2.6134 | 49.42 | 1.5 | 43.0017 | 2.5 |
| **Pirate** | 40.3774 | 2.6022 | 49.42 | 1.5 | NA | NA |
| **Boat** | 40.3879 | 2.6037 | 49.40 | 1.5 | 43.0069 | 2.5 |
| **Peppers** | 40.3431 | 2.6211 | 49.40 | 1.5 | 42.9873 | 2.5 |
| **Baboon** | 38.1356 | 2.6402 | 49.39 | 1.5 | NA | NA |

messages with satisfied visual quality. Both EMD and turtle shell based schemes show a low payload because of their plain and single layouts. Our method introduced an effective strategy to make a trade-off to achieve a large payload with more than 2.6(bpp) and satisfied visual quality with more than 40.36(dB) due to elaborating layout and absorbing previous research features.

**Acknowledgment.** This research was partially supported by the Ministry of Science and Technology of the Republic of China under the Grant MOST106-2221-E-324-006-MY2.

## REFERENCES


[1] L.F. Turner and L.M. Cheng, Hiding data in images by simple LSB Substitution, *Pattern Recognition,* vol. 37, no. 3, pp. 469-474, 2004.

[2] R.Z. Wang, C.F. Lin, and J.C. Lin, Image hiding by optimal LSB substitution and genetic algorithm, *Pattern Recognition*, vol. 34, no. 3, pp. 671- 683, 2001.

[3] Y.C. Tseng, Y.Y. Chen, and H.K. Pan, A secure data hiding scheme for binary images, *IEEE Transactions on Communications,* vol. 50, no. 8, pp. 1227-1231, 2002.

[4] J. Mielikainen, LSB matching revisited, *IEEE Signal Processing Lett,* vol. 13, no. 5, pp. 285-287, 2006.

[5] X. Zhang and S. Wang, Efficient steganographic embedding by exploiting modification direction, *IEEE Communication Letters*, vol. 10, no. 11, pp.781-783, 2006.

[6] H. Hajizadeh, A. Ayatollahi and S. Mirzakuchaki, A new high capacity and EMD-based image steganography scheme in spatial domain, *Proceedings of the International Conference on Electrical Engineering (ICEE)*, 2013.

[7] T.D. Kieu, C.C. Chang, A steganographic scheme by fully exploiting modification directions, *Expert Syst Appl*, vol. 38, no. 8, pp. 10648-10657, 2011.

[8] H.J. Kim, C. Kim, Y. Choi, S. Wang and X. Zhang, Improved modification direction methods, Comput Math Appl , vol. 60, no. 2, pp. 319-325, 2010.

[9] R.M. Chao, H.C. Wu, C.C. Lee and Y.P. Chu, A novel image data hiding scheme with diamond encoding, *EURASIP Journal on Information Security,* pp. 1-9, doi:10.1155/2009/658047, May 2009.

[10] C.C. Chang, Y.C. Chou, and T.D. Kieu, An information hiding scheme using Sudoku, *Proceedings of the 3rd International Conference on Innovative Computing, Information and Control (ICICIC), Homburg, Germany*, pp. 18-20, 2008.

[11] C.C. Chang, Y.J. Liu and T.S. Nguyen, A novel turtle shell based scheme for data hiding, *Proceedings of 2014 Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pp. 89-93, Dec. 2014.

[12] L. Liu, C.C. Chang and A. Wang, Data hiding based on extended turtle shell matrix construction method, Multimedia Tools and Applications, doi:10.1007/s11042-016-3624-7, 2016.

[13] Q. Jin, Z. Li, C.C. Chang, A. Wang and L. Liu, Minimizing turtle-shell matrix based stego image distortion using particle swarm optimization, *International Journal of Network Security, vol.* 19, no. 1, pp. 154-162, 2017.

[14] H.S. Leng, Data hiding scheme based on regular octagon-shaped shells, *International Conference on Intelligent Information Hiding and Multimedia Signal Processing,* vol. 81, pp. 29-35, 2018.

[15] C. F. Lee, C. Y. Weng, and K. C. Chen, An Efficient Reversible data hiding with reduplicated exploiting modification direction using image interpolation and edge detection, *Multimedia Tools and Applications,* vol.76, Issue 7, pp. 9993-10016, April 2017.

[16] Y. H. Huang and Ching-Chun Chang, A multiple image based reversible data hiding scheme with high embedding capacity, *Journal of Information Hiding and Multimedia Signal Processing,* vol. 8, no. 4, pp. 880-887, July 2017.

[17] J. J. Li, Yun-He Wu, Chin-Feng Lee, and Chin-Chen Chang, Generalized PVO-K embedding technique for reversible data hiding, *International Journal of Network Security,* vol.20, no.1, pp.65-77, January 2018.

[18] Qiu-Yu Zhang, Qi-Yan Dou, Yan Yan and Wen-Jin Hu, High capacity reversible data hiding algorithm for color image based on bicubic interpolation extension, *Journal of Information Hiding and Multimedia Signal Processing,* vol. 9, no. 1, pp. 61-74, January 2018.

[19] C. C. Chen, Ching-Yen Lee, Gwoboa Horng, Lee-Jang Yang and Ying-Hsuan Huang, Prediction-based reversible data hiding using energy deviation strategy, *Journal of Information Hiding and Multimedia Signal Processing,* vol. 9, no. 2, pp. 293-302, March 2018.