

## Research Article

# An Improved Bidirectional Shift-Based Reversible Data Hiding Scheme Using Double-Way Prediction Strategy

Lin Li <sup>1,2</sup>, Chin-Chen Chang <sup>2,3</sup> and Hefeng Chen <sup>1</sup>

<sup>1</sup>Computer Engineering College, Ji Mei University, Xiamen 361021, China

<sup>2</sup>Department of Information Engineering and Computer Science, Feng Chia University, 100 Wenhwa Road, Seatwen, Taichung 40724, Taiwan

<sup>3</sup>School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China

Correspondence should be addressed to Chin-Chen Chang; alan3c@gmail.com

Received 16 April 2019; Revised 16 October 2019; Accepted 12 November 2019; Published 26 December 2019

Guest Editor: Ki-Hyun Jung

Copyright © 2019 Lin Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Reversible data hiding (RDH) is a method that allows a cover image to be completely recovered from its corresponding stego image without distortion after the embedded secret messages have been extracted. Prediction-error expansion (PEE), as a classic RDH scheme, has been studied extensively due to its high quality of stego images. Based on prediction errors, threshold values, and the relative distances between each bin and zero bin, we present a bidirectional shift and double-way prediction strategy to solve the multiple embedding problem. Compared with the original algorithm, this scheme only takes a little more time and reduces the PSNR slightly, but it improves the embedding capacity significantly and allows for reversible data hiding. When both threshold values of  $TH$  and  $TH^*$  are equal to 2, the average  $ER$  value of 108 test images is 1.2 bpp which is ideal for medium data payload. At the same time, the PSNR is above 30 dB, making embedded information visually imperceptible. These data, together with other experimental results, show that the method proposed in this paper has obvious advantages in image quality and embedding capacity.

## 1. Introduction

The concept of steganography was first recorded and used in 1499 by Johannes Trithemius in his *Steganographia*, which was placed on the Index Librorum Prohibitorum in 1609 [1] and removed in 1900 [2], but the practice of secret communication has existed since ancient times. There are stories from Roman times concerning how slaves could hide messages and send the messages to recipients even though they were closely watched. Currently, in the computer age, camouflaged information is still processed to make it unrecognizable to the human eye, but it can possibly be revealed by computer visual recognition. In recent years, with the increasing emphasis on information security, data hiding technology has been used extensively for copyright protection and the authentication of the content of digital multimedia [3–5]. The content of an original image is altered after some data have been embedded in the image, and the

changes are visually imperceptible. But there is an inherent shortcoming of the traditional data hiding scheme, which is that the cover image cannot be restored completely after the receiver extracts the secret information [6–8]. For some specific scenarios, such as military and medical information and the preservation of artwork, even slight distortions in the images are intolerable since it is always essential to recover the original raw data. Therefore, reversible data hiding techniques have been proposed and developed that enable the receiver to extract the secret information, as is done in the traditional scheme, as well as to reconstruct the entire original image without distortion. From the perspective of its application, RDH can be considered a covert communication channel for the transmission of secret information. Thus, it can be seen that its application is quite extensive, such as embedding patients' private information into corresponding medical images and providing lossless authentication watermark for satellite images.

The first reversible data hiding (RDH) scheme was proposed in 1997 by Barton in his patent [9]. Barton's scheme compresses some of the alternative superimposed bits, adds a stream of bits, and embeds them into data blocks. Since then, RDH has attracted the attention of many scholars, and related works have been published frequently. In 2001, an improved algorithm [10] focusing on lossless recovery was proposed by Honsinger et al. One year later, Fridrich et al. [11] developed a high-capacity scheme based on embedding message bits in the status of pixel groups. In 2002, Celik et al. presented a lossless image compression algorithm [12] by using a prediction-based conditional entropy coder. This method uses some feature sets  $S$  of the image to form saved space by lossless compression and then utilizes them to hide information. However, as  $EC$  (embedding capacity) increases, more bit planes must be compressed, resulting in a sharp increase in distortion, which prevents the method from achieving satisfactory performance.

Thus, as can be seen, it has become the focus of scholars to determine how to reduce distortion while considering as much secret storage capacity as possible. Subsequently, expansion-based RDH schemes have been proposed to achieve the desired effectiveness, which is defined by three basic classifications, i.e., histogram shifting (HS), difference expansion (DE), and prediction-error expansion (PEE). The first HS-based reversible data hiding was proposed by Ni et al. [13] in 2003. The scheme is based on the concept of histogram of an image, which normally refers to the histogram of grayscale values of all pixels in the image, which shows the number of pixels at different grayscale values found in that image. For an 8 bit grayscale image, there are 256 possible grayscale values. The histogram will display the distribution of the pixels among those grayscale values between 0 and 255, as shown in Figure 1. For instance, the red mark in Figure 1 shows image "Baboon" has 2759 pixels with a grayscale value of 121. The scheme is implemented by shifting a certain histogram bin, usually the *peak* bin, which has the maximum number of pixels in the image. But HS techniques do not work well when the histogram of an image is equal. Although multiple pairs of peaks and minimums can be used for embedding, the embedding capacity is still insufficient. In 2003, with *peak* and *zero* bins to be chosen from the middle and the edge part of the histogram, respectively, a special case of HS, called the DE-based scheme, was first proposed by Tian [14], and it was based on the difference expansions of pixel pairs. Later, several improved techniques for DE-based embedding [15] were proposed, and they included integer transformation, pixel difference [16, 17], and prediction-error expansion (PEE). PEE is the focus of this paper, which was first proposed by Thodi and Rodriguez [18], and it extended the pixels and predicted differences for embedding data. Compared with the methods based on DE and HS, prediction-error histogram distribution is sharper and more centralized, so PEE has better performance. In addition to this, unlike DE, which only considers the correlation of two neighbouring pixels, and also unlike HS, which only considers the current pixel itself, PEE uses more adjacent pixels to obtain better performance.

Existing PEE-based RDH methods are mainly based on the modification of the one- or two-dimensional prediction-error histogram (PEH). The approach we propose using the PEH to form a method with a bidirectional shift and double-way prediction strategy for RDH and the details associated with its use are covered in the next section.

In this paper, we propose an improved RDH method based on the study [19] with double-way prediction and bidirectional shifting. The difference between the current value and the prediction value is used to form a prediction-error histogram. This histogram will be used to hide secret messages by expanding and shifting, and the embeddable bins are called *peak* (peak point) bins. Through outward shifting, they will be used to insert secret bits into the vacant bins reserved by other unembeddable bins that are also known as outer region bins. This process will be accomplished through two stages of prediction and corresponding processing based on the predicted values. The process is repeated for all pixels of the entire image to produce the stego image. During the extraction procedure, to draw out all of the secret bits and recover the original image, the receiver end can utilize the auxiliary information, which includes the position information of *zero* bins, threshold values, and pixels in the last row and the last column that have been kept intact. First, we calculate the prediction values of pixels in neighbourhood in the last row and the last column in the stego image. And the difference, named the prediction error (PE), between the prediction value and the stego-pixel value can be worked out at the same time. According to the PE, the quantity of the pixel value changes and the hidden information can be determined. Thus, the original pixel value can be worked out. Then pixel by pixel, the prediction value of the currently processed pixel can be determined using the pixels in its neighbourhood that have been recovered before it. Finally, all the pixel values of the original image and secret message are recovered. Therefore, the algorithm proposed in this paper is an RDH scheme. In addition, the experimental results show that the proposed method can achieve higher embedding capacity and better image quality than the previous scheme.

The contributions of this paper are as follows:

- (1) We apply bidirectional histogram shifting to improve the hiding capacity of a secret message. During the shifting procedure, we make two shift directions opposite to each other to reduce distortion.
- (2) We utilize a proposed double-way prediction strategy to get more embedding capacity.
- (3) Through the experimental data of 108 images, it is found that when  $TH = TH^* = 2$ , the algorithm effectively and significantly improves the information hiding capacity by 65% but only reduces the PSNR value about 6%, and the PSNR value remains above 30 dB.
- (4) The method can completely recover the image without any error; therefore, it is an RDH scheme with high hiding capacity.

The rest of this paper is organized as follows: In Section 2, Wang et al.'s method is reviewed, and the proposed

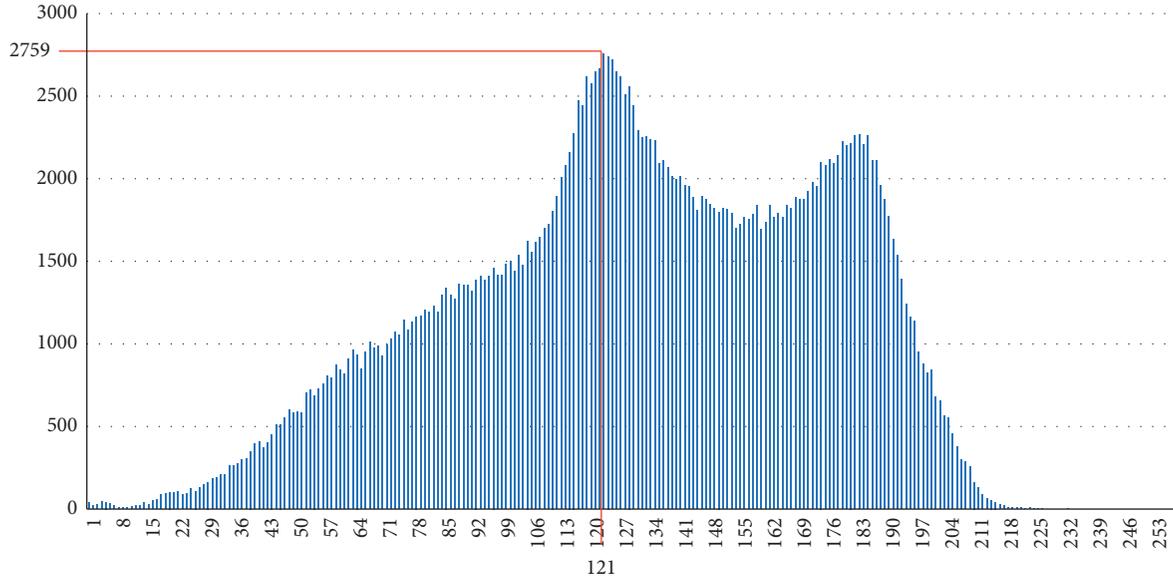


FIGURE 1: Histogram distribution of image “Baboon” with a peak point of 121.

method is described in Section 3. In Section 4, the experimental results are presented and discussed, and the conclusions are given in Section 5.

## 2. Review of Wang et al.’s Method

In this section, we introduce Wang et al.’s method [19], which is based on a double-way prediction error and right-left shift. To further decrease distortion of the stego image, the authors presented an extension of the traditional histogram shifting technique. The prediction error is produced by the difference between the value of the current pixel and the average value of the sum of its three adjacent pixels, i.e., the pixels in the right, bottom right, and bottom. Their technique is based on the principles that (1) the neighbouring pixels have a strong correlation and (2) the prediction-error distribution of adjacent pixels has a prominent maximum, that is, most of the prediction errors are expected to be very close to zero. The distribution of prediction errors has a zero mean value, similar to the Laplace distribution, which can be concluded from papers [20–23]. The authors utilize this characteristic of the distribution in combination with right-left shifting to improve the embedding capacity. To further introduce the algorithm, some basic concepts are provided in the next paragraph, and an example is given to illustrate the embedding and extraction procedures.

**2.1. Definition and Production of Prediction-Error Histogram.** To discuss the details of Wang et al.’s method, we use a grayscale image  $I$  of size  $H \times W$ , where  $H$  and  $W$  are the height and width of an image, respectively. Assume that the upper left coordinate position of  $I$  is  $(1, 1)$  and each pixel of  $I$  is assigned a coordinate. The coordinates of all pixels are successively increased from left to right and from top to bottom of the image. One of them is referred to as  $I(i, j)$ , as shown in Figure 2.

The scheme first predicts all pixels to form the prediction values  $P(i, j)$  of a pixel  $I(i, j)$  using the following equation:

$$P(i, j) = \text{fix}\left(\frac{I(i+1, j) + I(i, j+1) + I(i+1, j+1)}{3}\right), \quad (1)$$

where  $\text{fix}(\bullet)$  is a function, of which the input is a real number and the output is an integer. It rounds the input to the nearest integer toward zero. For example,  $\text{fix}(3.2) = 3$  and  $\text{fix}(8.6) = 8$ . Then, the PE (prediction-error) value  $P_e(i, j)$  of  $I(i, j)$  is obtained by the following equation:

$$P_e(i, j) = I(i, j) - P(i, j). \quad (2)$$

The pixels in the last row and the last column of an image are the reference pixels for recovery and thus cannot be predicted; the original values will be maintained. The preservation of these original data will play an important role in the extraction and recovery process.

Then,  $P_e(i, j)$  is produced by the formulation as formulated in equation (2). The prediction-error histogram (PEH) is generated by  $h(P_e)$  function, which counts the number of occurrences of a prediction-error (PE) value. It contains the *peak* bins, *zero* bins, and other bins, which are distributed symmetrically with values in the range of  $[-255, 255]$  for an 8 bit grayscale image. The PEH of “Barbara” is shown in Figure 3(a) as an example, where  $h(0) = 19,209$ ,  $h(81) = 337$ , and  $h(-68) = 100$ , which mean there are 19,209 pixels with a prediction-error value of “0,” 337 pixels with a prediction-error value of “81,” and 100 pixels with a prediction-error value of “-68.” Traditionally, the *peak* bin, which is also called the embeddable bin, is the bin that is located in the middle of the histogram. Here, the concept of peak bin is extended to all of the central bins within the threshold called  $TH$ , and the number of peak bins is defined as  $PK = 2 \times TH + 1$ , which consists of the number of traditional peak bins and the

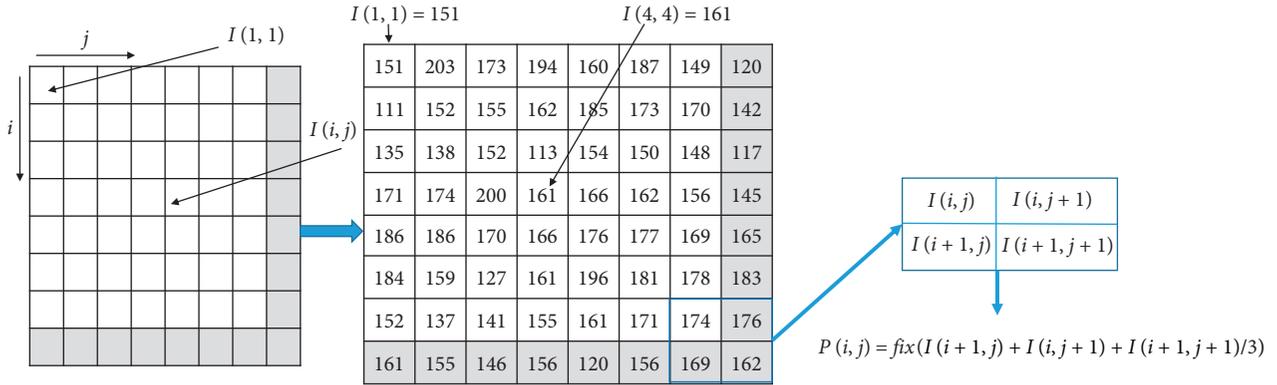


FIGURE 2:  $I(i, j)$  and  $P(i, j)$  of an image  $I$ .

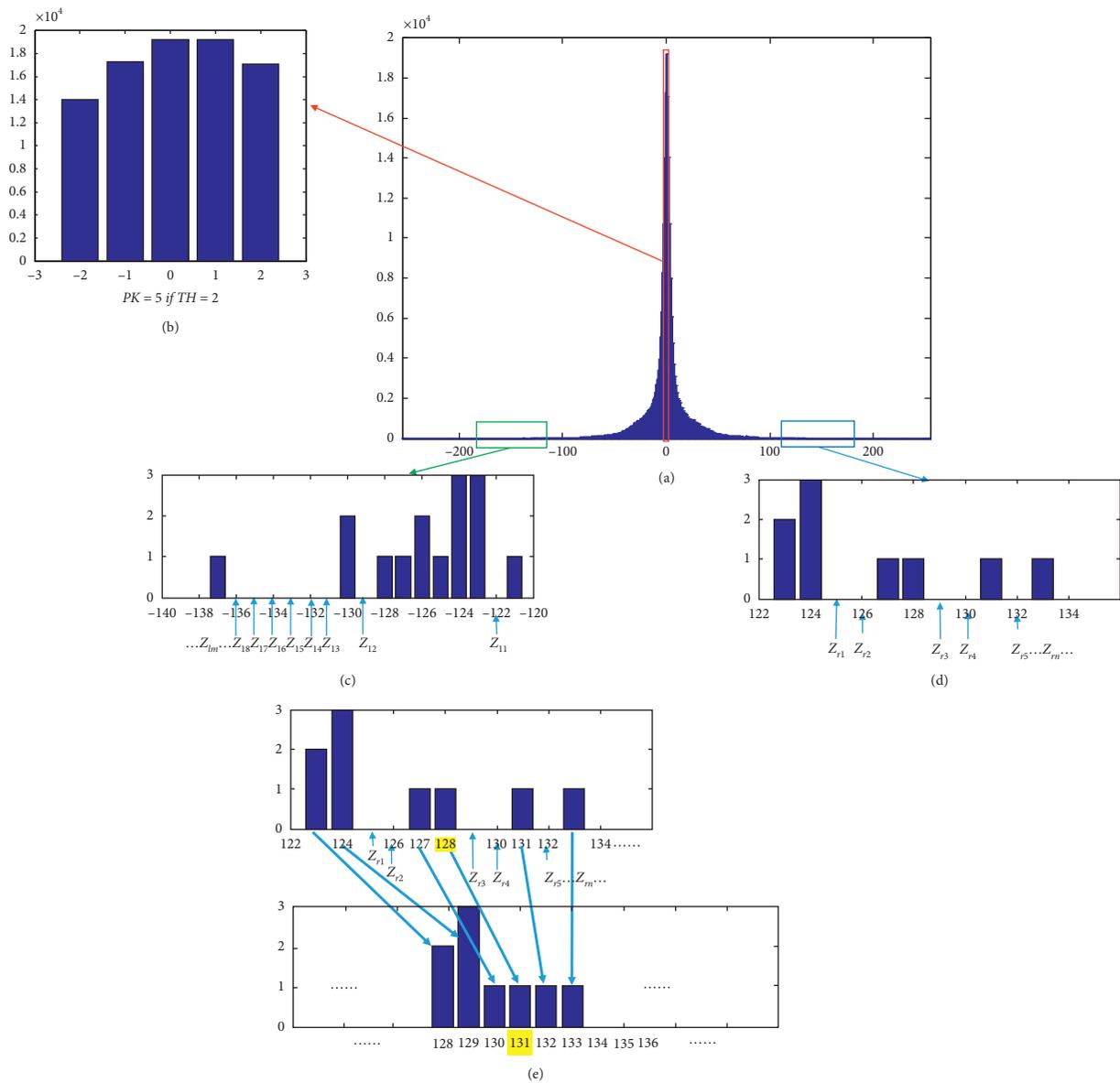


FIGURE 3: (a) Distribution of prediction errors of "Barbara." (b) Peak bins in the peak region. (c) Zero bins on the edge of the left outer region. (d) Zero bins on the edge of the right outer region. (e) Zero bins used for reducing shift and distortion.

bins on the left and right sides of it within the distance of  $TH$ , as shown in Figure 3(b).

During the embedding procedure, if the secret bit to be embedded is “1,” the corresponding *peak* bin will be expanded by 1 unit; otherwise, it remains unchanged. The outer bins will shift outward to make room for *peak* bins. *Zero* bins, which are located on two edges of the outer sides of the PEH, are denoted as  $Z_{rn}$  and  $Z_{lm}$ , respectively, acting as buffers for shifting. Here, zero means there are no PE values equal to that bin value. For example, in “Barbara,”  $Z_{r1} = 125$ , which means that there are no PE values equal to 125.  $Z_{r2} = 126$ ,  $Z_{r3} = 129$ ,  $Z_{r4} = 130$ ,  $Z_{r5} = 132$  and  $Z_{l1} = -122$ ,  $Z_{l2} = -129$ ,  $Z_{l3} = -131$  are the other *zero* bins, as shown in Figures 3(c) and 3(d), which are the enlarged version of Figure 3(a). In addition, subscripts  $n$  and  $m$  record the number of *zero* bins between the current bin and *peak* bins. For example, when the value of the current bin is 128, the largest  $Z_{rn}$  less than 128 is  $Z_{r2}$ , where  $n$  is 2, which means there are 2 *zero* bins between *peak* bins and 128. When right shifting is needed during the embedding procedure, these 2 *zero* bins can be occupied to reduce two moves and thus to reduce distortion. For instance, if we assume that all bins must shift to the right direction by  $\Delta bins = 5$ , then bin 128 will move to 131 rather than 133 because  $128 + \Delta bins - n = 128 + 5 - 2 = 131$ , and the occupancy of these 2 *zero* bins is achieved by subtracting 2, as shown in Figure 3(e). The same thing happens for a bin value of  $-130$ ; that is, the smallest  $Z_{lm}$  larger than  $-130$  is  $Z_{l2}$ , where  $m$  is 2, and when left shifting, these 2 *zero* bins can be occupied to reduce the quantity of moves by 2 and lessen distortion.

**2.2. Embedding Procedure.** The secret information to be hidden is denoted as a binary string  $S$ , with elements  $s_1, s_2, s_3, \dots, s_j, \dots, s_L$  of  $\{0, 1\}$ ,  $i, j \in [1, L]$ , where  $L$  is the length of  $S$ . We set  $TH = 2$  and take image “Barbara” as an example. The number of *peak* bins of “Barbara” is 5, as shown in Figure 3(a). According to the distribution of prediction errors of “Barbara,” Figure 4 is provided, and the details about the figure demonstrate right and left shifting of *peak* bins and the whole prediction-error histogram of “Barbara.” Because the image being discussed is a grayscale image, the histogram of prediction errors would be distributed in the interval  $[-255, 255]$ . Due to the similarity of the values of neighbouring pixels, the *peak* bins were concentrated on value 0 and its adjacent values. It is this feature that the authors used to hide information. There are two phases of movement of the PEH in the algorithm, which are first shifting to right and later shifting to left. The first line in Figure 4 is the original PEH of “Barbara,” the second line is the PEH after right shifting, and the third line is the PEH after left shifting.

When right shifting (RS), bin value  $-2$  remains as  $-2$  to hide the secret bit “0,” and it changes to  $-1$  to hide the secret bit “1.” In a similar way, bin value  $-1$  moves to 0 to embed the secret bit “0,” and it shifts to 1 to hide the secret bit “1.” One bin moved after another, bin value 2 will move to 6 and 7 to hide “0” and “1.” But the bins on the right and left sides of *peak* bins will not be used to hide secret bits. Due to the

movement of *peak* bins, the bins on the right side will have to shift outward to make room for *peak* bins. Thus, bin 3 will move to 8, bin 4 will move to 9, etc. To reduce the distortion, every *zero* bin can be used to buffer the amount of shifting by reducing the movement to the right side by one bin. Therefore, bin value 127 will move to 130 because there are 2 *zero* bins, i.e.,  $Z_{r1} = 125$  and  $Z_{r2} = 126$ , between bin value 127 and *peak* bins. The amount of moved units is  $PK - 2 = 5 - 2 = 3$ . Those bins with values less than  $-2$  will not be affected and maintain their original values.

When left shifting (LS),  $PK_l$  is 10, which is twice as large as the  $PK$  value. It is because after right shifting, bins of the *peak* region have already been expanded to twice the number of the original *peak* region. This time, all of the bin values that are larger than 8 remain unchanged, but those less than 8 will move. Concerning bin value 7, it remains unmoved to hide the secret bit, “0,” while it shifts to 6 to hide “1.” In a similar way, bin by bin, bin value  $-2$  moves to  $-11$  to embed the secret bit “0” and shifts to  $-12$  to hide the secret bit “1.” The bins on left sides of  $PK_l$  will not be used to hide secret data but must be moved to make room. For example, bin value  $-3$  will shift to  $-13$  and bin value  $-120$  will shift to  $-130$ . Instead of moving to the left side by  $PK_l = 10$ , bin value  $-123$  moves by  $10 - 1 = 9$  units because there is a *zero* bin  $Z_{l1} = -122$  between  $-123$  and *peak* bins, which is used to reduce one unit of movement.

To ensure better understanding, three examples are provided to explain how shifting and expanding work during the right-left shifting phase:

**Example 1.** The red values in Figure 4 are PE values of *peak* bins according to the definition of  $TH$ . In the cover image “Barbara,” the original pixel value  $I(2, 2)$  is 198, the corresponding prediction value  $P(2, 2)$  is 196 (calculated from equation (1)), and the prediction-error value is 2 (calculated from equation (2)), which is one of the *peak* bin values. In the RS (right shifting) stage, the amount of shifting is 5, which is the result of  $7 - 2$ , where the secret bit  $s_i$  is assumed to be “1” here. Thus, the value of the marked pixel is  $198 + 5 = 203$ . In the LS (left shifting) stage, the amount of shifting is 0, where the secret bit  $s_j$  is assumed as “0.” After the above procedure, the pixel value  $I(2, 2)$  of the cover image is changed from 198 to 203, which is the final value of its corresponding stego pixel, with embedding two secret bits.

**Example 2.** The green values in Figure 4 are bin values between 3 and 133. Number 3 is the smallest prediction error greater than  $TH$ , and number 133 is the greatest nonzero bin value. In the RS stage, these bin values will add a relative positive integer according to the subtraction calculation of  $PK - n$ , where  $n$  is the total quantity of *zero* bins less than the current bin value. In the LS stage, these bin values will remain unchanged. For example,  $I(371, 457) = 228$ ,  $P(371, 457) = 97$ , and  $P_e(371, 457) = 228 - 97 = 131$  from equation (2), and the amount of shifting is  $PK - n = 5 - 4 = 1$ , where  $n$  is 4 because the largest *zero* bin less than 131 is  $Z_{r4}$  (Figure 3(d)), which means there are 4 *zero* bins less than 131. Thus, bin 131 moves to 132, and the marked pixel value

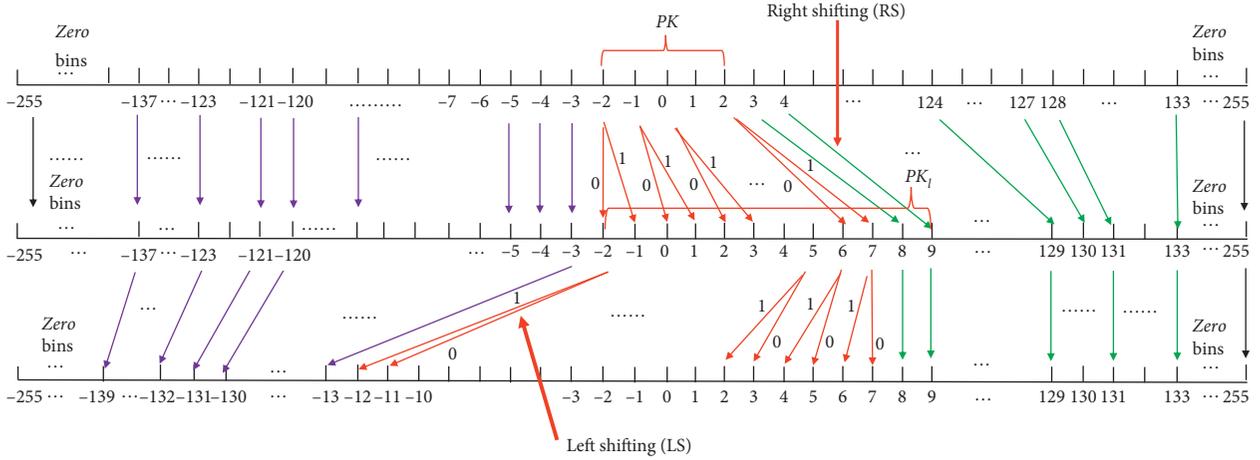


FIGURE 4: Right and left shifting of the whole prediction-error histogram (PEH) of “Barbara.”

is  $228 + 1 = 229$ . In the second-stage hiding procedure, the shifting of bin 132 is not needed, so the value of the final stego image pixel of  $I(371, 457)$  is 229.

*Example 3.* The purple values in Figure 4 are bin values between  $-3$  and  $-137$ . Number  $-3$  is the largest prediction error less than  $-TH$ , which is  $-2$ , and number  $-137$  is the smallest nonzero bin value. In the RS stage, these bin values will remain unchanged because they are less than  $-PK$ . In the LS stage, these bin values will subtract a relative positive integer according to the subtraction calculation of  $PK_l - m$ , where  $m$  is the total quantity of zero bins greater than the current discussing bin value. For example,  $I(378, 453) = 51$ ,  $P(378, 453) = 188$ , and  $P_e(378, 453) = 51 - 188 = -137$ . In the RS stage, bin value  $-137$  remained unchanged; thus, the amount of shifting of bin value  $-137$  is 0, which means pixel value 51 will remain unchanged. In the LS stage, the amount of shifting is  $PK_l - m = 10 - 8 = 2$ , where  $m$  is 8 because the smallest zero bin on the right side of bin  $-137$  is  $Z_{18}$  (Figure 3(c)). At last, the final stego-pixel value of  $I(378, 453)$  is  $51 - 2 = 49$ .

**2.3. Extracting Procedure.** After the embedding procedure,  $TH$ , the zero bins  $Z_{rn}$  and  $Z_{lm}$ , and the stego image, in which the last row and the last column are original and intact, will be sent to the receiver. Thus, the predicted pixel values can be calculated from the pixel values in the last row and the last column. To be specific, according to equation (1), for an image with a size of  $512 \times 512$ , the prediction value  $P(511, 511)$  is first calculated, and after that, the difference between the prediction value and the stego-pixel value of  $P(511, 511)$  is worked out. Then, based on the prediction error and how it relates to  $TH$ , the zero bins  $Z_{rn}$  and  $Z_{lm}$ , the original value of  $I(511, 511)$ , and the embedding bit value (if it had been hidden before) are solved. Now as a given message, the original value of  $I(511, 511)$  can be used to determine  $I(511, 510)$ . Thus, pixel by pixel, the whole secret information can be extracted and the original image can be recovered in the reverse processing order of the embedding procedure.

### 3. Proposed Scheme

To improve the quality of the stego image and increase the amount of secret information that can be hidden, we use bidirectional PEH shifting in double-way prediction based on Wang et al.’s method [19]. The process of generating the stego image is divided into two stages: embedding process 1 using the raster scan order and embedding process 2 using the inverse raster scan order. First of all, pixels in the last row and the last column are kept and used as reference pixels. The first prediction is made in a raster scan order, i.e., from the first pixel in the upper left corner of an image, top to bottom, and left to right, and the top-left pixel is predicted by the other three pixels in its square-shaped adjacent context. The calculation process is shown in equation (1), and the prediction error,  $P(i, j)$ , is calculated by equation (2); the second-stage prediction is processed in the reverse raster scan order from the pixel in the bottom right corner, right to left, and bottom to top, and the lower-right pixel is predicted by the other three pixels as follows, respectively:

$$P^*(i, j) = \text{fix} \left( \frac{I_1(i-1, j) + I_1(i, j-1) + I_1(i-1, j-1)}{3} \right), \quad (3)$$

$$P_e^*(i, j) = I_1(i, j) - P^*(i, j), \quad (4)$$

where  $I_1$  and  $I_1(i, j)$  are the stego image and its stego pixel after embedding process 1. In the process of information hiding in these two stages, PE (prediction-error) values are determined separately, and they are named  $P(i, j)$  and  $P^*(i, j)$ , respectively. In addition, during embedding process 1, we first expand or shift bins of the PEH to the right direction and afterward to the left direction. During embedding process 2, to avoid severe distortion which may occur, the movement is set to the opposite direction from the first embedding process. That is to say, during embedding process 2, we first expand or shift bins of the PEH to the left direction and then to right. The number of moving units depends on the interval in which the value of the bin falls. After the shifting of all the pixels has been finished, the

TABLE 1: Notations of the proposed method (all variables marked with a star are used in embedding process 2).

---

PEH: prediction-error histogram
PE: prediction error
S: secret information in binary bits with elements $s_1, s_2, s_i, \dots, s'_i, \dots, s_j, \dots, s'_j, \dots, s_L$ , where $L$ is the length of $S$
TH, $TH^*$ : $[-TH, TH]$ , $[-TH^*, TH^*]$ are the ranges of peak bins
PK, $PK^*$ : the quantity of peak bin values
$PK_b, PK_r^*$ : double the number of <i>peak</i> bins
$I, \bar{I}(i, j)$ : an 8 bit grayscale cover image with a size of $H \times W$ ( $H$ is the height and $W$ is the width), $I(i, j)$ is a pixel of $I$ in row number $i$ and column number $j$
$I_1, I_1(i, j)$ : the first-stage stego image and one of its pixels
$\bar{I}, \bar{I}(i, j)$ : the final stego image and one of its pixels
$P(i, j), P^*(i, j)$ : prediction value of $I(i, j), I_1(i, j)$
$P_e(i, j), P_e^*(i, j)$ : prediction error of pixel $I(i, j)$
$P_{el}(i, j), P_{el}^*(i, j)$ : prediction value of pixels $I_{1r}(i, j)$ and $I_1(i, j)$ after left shifting
$P_{er}(i, j), P_{er}^*(i, j)$ : prediction value of pixels $I_{2l}^*(i, j)$ and $\bar{I}(i, j)$ after right shifting
$h(p_e)$ : the frequency of the prediction-error value $p_e$ in the histogram, where $p_e \in [-255, 255]$
$I_{1r}(i, j)$ : the stego image after right shifting of PEH in embedding process 1
$I_{2l}^*(i, j)$ : the stego image after left shifting of PEH in embedding process 2
$\Delta bin(i, j), \Delta bin_l(i, j)$ : the quantity of units to move during expanding and shifting in right shift and left shift in embedding process 1
$\Delta bin^*(i, j), \Delta bin_r^*(i, j)$ : the quantity of units to move during expanding and shifting in left shift and right shift in embedding process 2
$n, n^*$ : the number of zero bins between the current positive bin value and the <i>peak</i> bins
$m, m^*$ : the number of zero bins between the current negative bin value and the <i>peak</i> bins

---

original prediction error (PE) is altered to the modified prediction error (MPE). The difference of the MPE and PE, called  $\Delta bins$ , is added to the original pixel to create the value of the stego pixel.

The notations of this scheme are presented in Table 1, and the subsequent discussion explains the algorithm process in detail, including the pseudocode of the hiding and extracting information procedure.

The details of embedding process 1 are explained by the pseudocode presented in Algorithm 1.

Figure 5 is an illustration of embedding process 1. We assume  $TH = 2$ , and there is a pixel  $P(i, j)$  with its PE value  $P_e(i, j) = 0$ , which is a value belonging to bin value 0 shown in Figure 5(a). During right shifting of the PEH, bin value 0 is shifted to bin value 2, with “0” embedded, as shown in Figure 5(c). That is to say,  $\Delta bin(i, j) = P_e(i, j) + TH + s_i = 0 + 2 + 0 = 2$ . Then,  $I(i, j)$  is changed to  $I_{1r}(i, j)$ , where  $I_{1r}(i, j) = I(i, j) + \Delta bin(i, j) = I(i, j) + 2$ . Then, left shifting is used to embed “1” in  $I_{1r}(i, j)$ , which changes bin value 2 to  $-4$ .  $P_{er}(i, j) = P_e(i, j) + \Delta bin(i, j) = P_e(i, j) + 2$ ;  $\Delta bin_l(i, j) = P_{er}(i, j) - TH - PK - s_j = P_e(i, j) + 2 - 2 - 5 - 1 = -6$ . Finally,  $I_1(i, j) = I_{1r}(i, j) + \Delta bin_l(i, j) = I(i, j) + 2 - 6 = I(i, j) - 4$ . After that, pixel  $I(i, j)$  has been embedded with “0” and “1” with pixel value adding  $-4$ . Thus, with varied quantity of movements of *peak* region bins in the PE histogram, secret data are hidden. Moving the outer region makes room for the *peak* region to move and hide secret information. Those pixel values whose prediction-error values are located in the right outer region are increased. Those pixel values whose prediction-error values are located in the left outer region are decreased. In addition, with help of *zero* bins, the amount of movement in the rear of the outer region can be reduced to some extent. Of all these, the maximum change is  $2 \times PK$ .

Next, the current stego image  $I_1$  will be predicted with equation (3), and the order of prediction processing is in the reverse order from the prediction in embedding process 1. Then, the PEH shifting direction is toward left and then right which is the opposite order of embedding process 1 also. Compared with Wang et al.’s scheme [19], the distortion will be a little bit larger due to the increase in the hiding capacity of secret information. The pseudocode of embedding process 2 is given in Algorithm 2.

The overflow/underflow problem occurs when the values of some pixels, called boundary pixels, are changed from 255 to greater than 255 or from 0 to negative values. For these pixels, according to reference [24], we use a different approach from Wang et al.’s. The values that will exceed the boundary after  $\Delta bins$  are added will not be allowed to participate in the addition; for them, the original values are kept unchanged and called pseudo-values. The pixel values, whose addition result values are equal to the corresponding pseudovalues, are called genuine values. A boundary array can be recorded to distinguish genuine values from pseudovalues to avoid the confusion. Each member of the array corresponds to a boundary pixel in the stego image, with genuine “0” and pseudo “1,” respectively.

**3.1. An Example of Embedding Process.** As shown in Figure 6, pixel  $I(31, 12)$  of “*Baboon*” is identified as  $I(31, 12)$ , and its pixel value is 119. Based on equation (1),  $P_e(31, 12) = -2$  satisfies the condition  $P_e(i, j) < TH$ , where  $TH = 0$ . Thus,  $\Delta bin$  value in right shifting of embedding process 1 of  $I(31, 12)$  is 0,  $I_{1r}(31, 12) = I(31, 12) = 119$ , and  $P_{er}(31, 12) = -2$ . In left shifting of embedding process 1, due to  $P_{er}(31, 12) < TH = 0$ ,  $\Delta bin_l(31, 12) = -PK_l = -2$ ; therefore,  $I_1(31, 12) =$

```

Input: grayscale cover image  $I$ , secret  $s_i, s'_i$  as one bit of the secret message  $S$ , and threshold  $TH$ 
Output: stego image  $I_1$ 
for each pixel  $I(i, j)$  (except for the pixels in the last column and the last row)
  Scan in the raster scan order;
  Calculate the prediction error of  $I(i, j)$  as  $P_e(i, j)$  with equations (1) and (2);
  Generate PEH of  $I, PK = 2 \times TH, PK_l = 2 \times PK$ ;
  Find out zero bins and denote them as  $Z_{rn}$  and  $Z_{lm}$ ;
  /*right expanding and shifting*/
  if  $-TH \leq P_e(i, j) \leq TH$ 
     $\Delta bin(i, j) = P_e(i, j) + TH + s_i$ ;
  elseif  $TH < P_e(i, j)$ 
     $\Delta bin(i, j) = PK - n$ , where  $Z_{rn} \leq P_e(i, j) < Z_{r(n+1)}, n \leq PK$ ;
  else
     $\Delta bin(i, j) = 0$ ;
  endif;
   $I_{1r}(i, j) = I(i, j) + \Delta bin(i, j)$ ;
end;
/*finish right and begin left expanding and shifting*/
for each pixel  $I_{1r}(i, j)$ 
   $P_{er}(i, j) = I_{1r}(i, j) - P(i, j) = I(i, j) + \Delta bin(i, j) - P(i, j) = P_e(i, j) + \Delta bin(i, j)$ ;
  if  $-TH \leq P_{er}(i, j) \leq TH + PK_l$ 
     $\Delta bin_l(i, j) = P_{er}(i, j) - TH - PK - s'_i$ ;
  elseif  $P_{er}(i, j) < -TH$ 
     $\Delta bin_l(i, j) = -PK_l + m$ , where  $z_{l(m+1)} \leq P_{er}(i, j) < z_{lm}, m \leq PK_l$ ;
  else
     $\Delta bin_l(i, j) = 0$ ;
  endif;
   $I_1(i, j) = I_{1r}(i, j) + \Delta bin_l(i, j)$ ;
   $P_{el}(i, j) = P_e(i, j) + \Delta bin(i, j) + \Delta bin_l(i, j)$ ;
end;
/*finish embedding process 1*/

```

ALGORITHM 1: Pseudocode of embedding process 1.

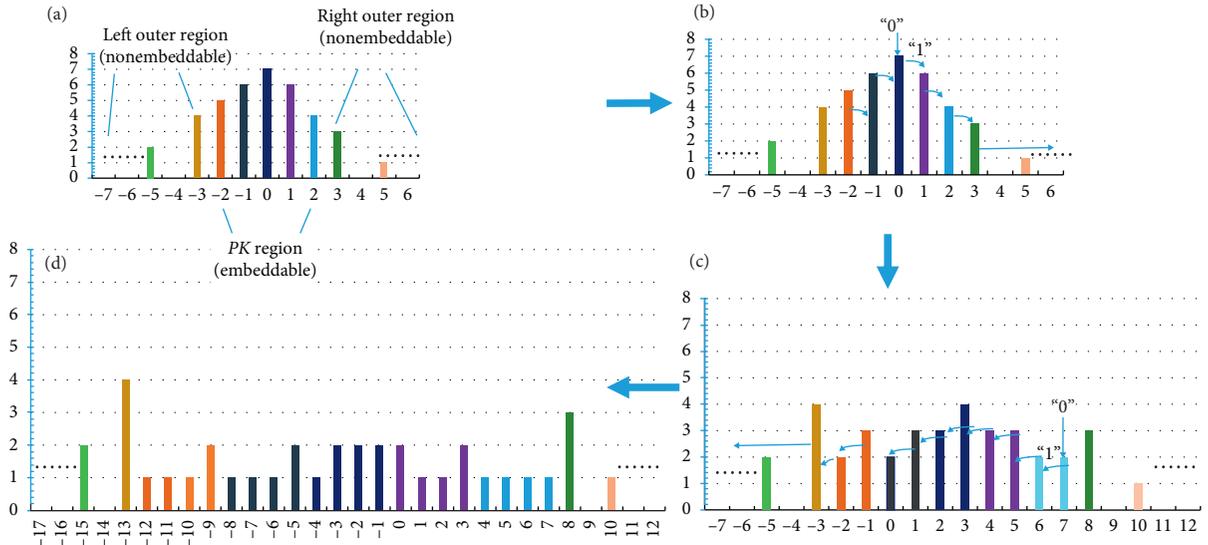


FIGURE 5: (a) Three regions of the histogram. (b) Right shift direction. (c) Left shift direction. (d) The final histogram.

$I_{1r}(31, 12) + \Delta bin_l(31, 12) = 119 + (-2) = 117$ , which is the result of embedding process 1. Next comes embedding process 2. The new prediction value is generated according

to equation (3), and the prediction error  $P_e^*(31, 12) = 0$  is obtained. Now comes left shifting of embedding process 2. According to the given threshold value, i.e.,  $TH^* = 0$ , we find

```

Input: first-stage stego image  $I_1$ , secret  $s_j$  and  $s'_j$  that are, respectively, one bit of the secret message  $S$ , and threshold  $TH^*$ 
Output: stego image  $\tilde{I}$ 
for each pixel,  $I_1(i, j)$ , except for the pixel in the first column and the first row of  $I_1$ 
  Scan in the reverse raster scan order from the pixel in the bottom right corner of  $I_1$ ;
  Calculate the prediction error of  $I_1(i, j)$  as  $P_e^*(i, j)$  with equations (3) and (4);
  Generate PEH* of  $I_1$  and  $PK^* = 2 \times TH^*$ ,  $PK_r^* = 2 \times PK^*$ ;
  Compute zero bins and denote them as  $Z_{rn}^*$  and  $Z_{lm}^*$ ;
  /*left expanding and shifting*/
  if  $-TH^* \leq P_e^*(i, j) \leq TH^*$ 
     $\Delta bin^*(i, j) = P_e^*(i, j) - TH^* - s_j$ ;
  elseif  $P_e^*(i, j) < -TH^*$ 
     $\Delta bin^*(i, j) = -PK^* + m^*$ , where  $Z_{l(m^*+1)}^* \leq P_e^*(i, j) < Z_{lm^*}^*$ ,  $m^* \leq PK^*$ ;
  else
     $\Delta bin^*(i, j) = 0$ ;
  endif;
   $I_{2l}^*(i, j) = I_1(i, j) + \Delta bin^*(i, j)$ ;
end;
/*finish left and begin right expanding and shifting*/
for each pixel  $I_{2l}^*(i, j)$ 
   $P_{el}^*(i, j) = I_{2l}^*(i, j) - P^*(i, j) = P_e^*(i, j) + \Delta bin^*(i, j)$ ;
  if  $-TH - PK^* \leq P_{el}^*(i, j) \leq TH^*$ 
     $\Delta bin_r^*(i, j) = P_{el}^*(i, j) + SH^* + PK^* + s'_j$ ;
  elseif  $P_{el}^*(i, j) > SH^*$ 
     $\Delta bin_r^*(i, j) = PK_r^* - n^*$ , where  $Z_{rn^*} \leq P_{el}^*(i, j) < Z_{r(n^*+1)}$ ,  $n \leq PK_r^*$ ;
  else
     $\Delta bin_r^*(i, j) = 0$ ;
  endif;
   $\tilde{I}(i, j) = I_{2l}^*(i, j) + \Delta bin_r^*(i, j)$ ;
   $P_{er}^*(i, j) = P_e^*(i, j) + \Delta bin^*(i, j) + \Delta bin_r^*(i, j)$ ;
end;
/*finish embedding process 2*/

```

ALGORITHM 2: Pseudocode of embedding process 2.

that  $P_e^*(31, 12) = TH^*$ . Assume that the current information to be hidden is  $s_j = 1$ . Thus,  $\Delta bin^*(31, 12) = -1$ , which expands bin value 0 to  $-1$  to hide the secret bit "1"; thus,  $I_{2l}^*(31, 12) = 116$ , and  $P_{el}^*(31, 12) = -1$ . Then, we proceed with right shifting of embedding process 2.  $P_{el}^*(31, 12)$  satisfies the condition  $-TH - PK^* \leq P_{el}^*(i, j) \leq TH^*$ , and we assume that the current information to be hidden is  $s'_j = 1$  so that  $\Delta bin_r^*(31, 12) = 1$  hides "1." Finally, the stego pixel is  $\tilde{I}(31, 12) = 117$ . At this point, the embedding processes are finished.

**3.2. An Example of Extraction Process.** Now, we assume that the receiver receives the stego image, threshold  $TH$ ,  $TH^*$ , zero bins  $Z_{rn}$ ,  $Z_{lm}$ ,  $Z_{rn}^*$ , and  $Z_{lm}^*$ , and a boundary array of over/underflow pixels trying to extract the secret and recover the original image. First, based on the first line and first column pixels that have been preserved, the receiver can determine the prediction value of pixel (2, 2) and, then, its  $I_1(2, 2)$  pixel value. Then pixel by pixel, in the raster scan order, all of the ones before  $I_1(31, 12)$  will be recovered. Then, calculating by equation (3) and using  $I_1(30, 11)$ ,  $I_1(30, 12)$ , and  $I_1(31, 12)$  that have been determined, the prediction value is determined, that is,  $P^*(30, 12) = 117$ . And the difference between  $P^*(30, 12)$  and stego pixel  $\tilde{I}(31, 12)$ , which is  $P_{er}^*(31, 12) = 0$ , can then be figured out.

Based on the difference value and  $TH^* = 0$ , we can determine that  $P_{er}^*(31, 12) = -1$  and  $s'_j = 1$ . Thus,  $I_{2l}^*(31, 12) = P_{er}^*(31, 12) + P^*(31, 12) = -1 + 117 = 116$ ,  $s_j = 1$ , and  $P_e^*(31, 12) = 0$ . Thus,  $I_1(31, 12) = 117$ . Through the same process, we can determine every  $I_1(i, j)$ , pixel by pixel. Subsequently, the first phase of the extraction process is completed. And, based on the values of  $I_1$  and all of the values in the last column and last row of the image that have maintained their original values from the cover image, we can determine, pixel by pixel, the prediction value and prediction errors of all the pixels. The difference between  $I_1(31, 12)$  and  $P(30, 12)$ , called  $P_{el}(31, 12)$ , is  $-4$ ; thus,  $P_{er}(31, 12) = -2$ ,  $\Delta bin_l^*(i, j) = -2$ , and  $I_{1r}(31, 12) = I_1(31, 12) - \Delta bin_l(31, 12) = 117 + 2 = 119$  because of the given threshold value  $TH = 0$ ; also,  $\Delta bin(31, 12) = 0$  and  $P_e(31, 12) = 0$ . Thus,  $I(31, 12) = I_{1r}(31, 12) - \Delta bin(31, 12) = 119$ , which is the original pixel value. The process is shown in Figures 6 and 7, in which the blue arrow points out the embedding process and the orange arrow indicates the extraction process.

## 4. Experimental Results and Analysis

The experimental platform we use consists of an Intel 3.41 GHz i7-6700 CPU, 32 GB RAM, and the Windows 10

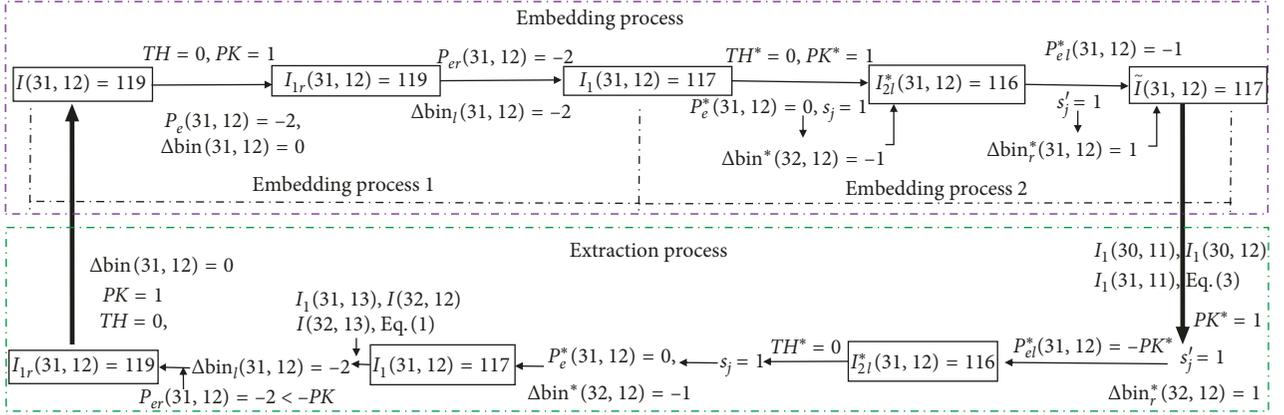
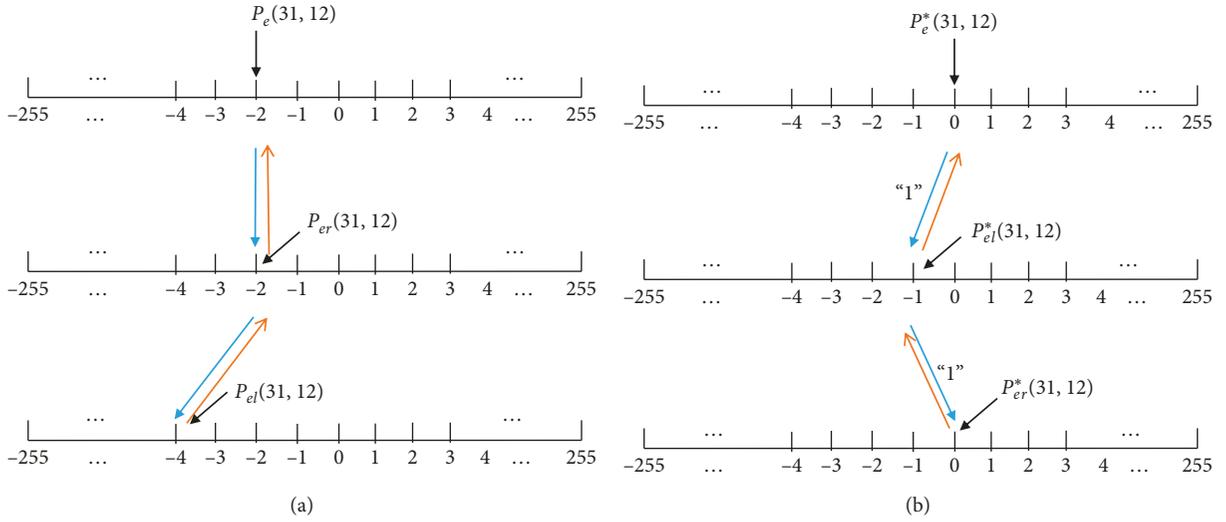


FIGURE 6: An example of processing the pixels of image "Baboon."

FIGURE 7: Histogram shifting of embedding process 1, embedding process 2, and the recovery process of  $I(31, 12)$ . (a) Right-left shifting ( $TH=0$ ). (b) Left-right shifting ( $TH^*=0$ ).

Professional 64 bit operating system. Our experiments are implemented by MATLAB. The performance of the proposed scheme is estimated by some criteria, i.e., embedding capacity (EC), peak signal-to-noise ratio (PSNR), and execution time (s: seconds). Tables 2–5 show some figures that are determined by using standard  $512 \times 512$  grayscale common, medical, texture, and aerial images, respectively.

EC can be controlled by quantity of *peak* bins, and it can be calculated by equation (6). The embedding rate (ER) represents the percentage of the embedded secret bits in the whole pixels of the cover image. ER is defined as in (7). Here,  $H \times W$  is the size of the cover image:

$$EC = h(p_e), \text{ where } -PK \leq p_e \leq PK, \quad (5)$$

$$ER = \frac{EC}{H \times W}, \quad (6)$$

where  $PK$  is the quantity of *peak* bin values we use to hide the secret messages, the value of which, in fact, will change in different embedding phases according to the  $TH$  value we use.

In general, the peak signal-to-noise ratio (PSNR) is used to measure the visual quality of the reconstructed image or the stego image. The PSNR is defined as

$$PSNR = 10 \times \log\left(\frac{255^2}{MSE}\right), \quad (7)$$

where MSE is the mean square error between the cover image and the stego image. For a grayscale cover image of size  $H \times W$ , the MSE is defined as

$$MSE = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W (p_{ij} - \bar{p}_{ij})^2, \quad (8)$$

where  $p_{ij}$  represents the pixel value in row  $i$  and column  $j$  of the original image and  $\bar{p}_{ij}$  is the corresponding stego-pixel value of  $p_{ij}$ .

As a common sense, the difference between the cover image and the stego image is invisible to naked eyes when the value of the PSNR exceeds 30 dB. Thus, to control the imperceptible distortion of a stego image, the embedding process will always be stopped when the PSNR value is less

TABLE 2: Increased embedding capacity ( $\Delta EC$ ) and its increasing rate (IR), the PSNR (dB) and its decreasing rate (DR), and the increased time ( $\Delta t$ ) of eight common images.

Value names		Lena	Barbara	Baboon	Lake	Boat	Bird	Peppers	Harbour
$TH=0$ $TH^*=0$	$\Delta EC$ (bits)	35188	27332	14824	25812	23924	20556	29530	25790
	IR	67.3%	71.1%	85.3%	66.9%	71.1%	63.1%	68.9%	62.2%
	PSNR (dB)	40.9	40.8	40.5	40.5	40.6	40.8	40.3	40.3
	DR	8.1%	8.2%	8.5%	8.9%	8.7%	8.0%	9.4%	9.2%
	$\Delta t$ (s)	1.74	1.35	1.34	6.05	4.24	1.44	1.38	1.32
$TH=1$ $TH^*=0$	$\Delta EC$ (bits)	16848	13533	8322	12697	11893	9979	13770	11795
	IR	11.2%	12.1%	8.21%	10.9%	7.8%	8.27%	8.6%	7.36%
	PSNR (dB)	34.44	34.60	34.12	34.39	34.45	34.9	34.33	34.37
	DR	3.0%	2.11%	2.40%	2.62%	2.48%	2.54%	3.05%	2.74%
	$\Delta t$ (s)	1.35	1.33	1.34	1.33	1.39	1.34	1.39	1.37
$TH=1$ $TH^*=1$	$\Delta EC$ (bits)	67764	54926	33240	50796	48708	43870	55856	47600
	IR	45.2%	49.3%	65.1%	46.1%	49.4%	60.7%	43.8%	40.1%
	PSNR (dB)	32.82	32.70	32.14	32.41	32.49	32.43	32.26	32.16
	DR	11.2%	11.2%	11.5%	12.1%	11.8%	11.0%	12.2%	12.3%
	$\Delta t$ (s)	1.35	1.32	1.35	1.39	1.40	1.43	1.37	1.34
$TH=2$ $TH^*=0$	$\Delta EC$ (bits)	9122	7428	4558	6847	6551	5425	7200	6536
	IR	3.9%	4.3%	5.4%	4.0%	4.2%	4.9%	3.6%	3.6%
	PSNR (dB)	31.15	30.85	30.17	30.69	30.71	30.33	30.78	30.72
	DR	2.0%	1.9%	2.2%	2.5%	2.3%	2.1%	2.2%	2.4%
	$\Delta t$ (s)	1.34	1.31	1.33	1.39	1.42	3.47	1.51	1.46
$TH=2$ $TH^*=1$	$\Delta EC$ (bits)	54966	44662	27666	41252	39150	33694	44082	38686
	IR	23.7%	25.7%	32.7%	24.1%	24.9%	30.2%	21.8%	21.3%
	PSNR (dB)	30.93	30.70	30.09	30.42	30.47	30.26	30.38	30.43
	DR	7.8%	7.6%	8.1%	8.6%	8.3%	7.7%	9.4%	8.6%
	$\Delta t$ (s)	1.41	1.39	1.42	1.68	1.38	1.35	1.51	1.36
$TH=2$ $TH^*=2$	$\Delta EC$ (bits)	90566	74417	46468	68132	65416	59622	73688	64756
	IR	39.1%	42.9%	55.1%	40.2%	41.6%	53.4%	36.4%	35.7%
	PSNR (dB)	30.07	30.12	30.08	30.15	30.18	30.25	30.19	30.17
	DR	13.4%	13.3%	13.7%	14.3%	14.0%	13.2%	15.2%	14.4%
	$\Delta t$ (s)	1.32	1.33	1.36	1.33	1.38	1.32	1.36	1.31

TABLE 3: Experimental results for eight medical images ( $TH=2$ ,  $TH^*=2$ ).

Images (512 $\times$ 512)	Payload (bits)	PSNR of the stego image (dB)		ER (bpp: bits per pixel)
	Increasing rate	Decreasing rate		
Mpic1	1021589	36.1		3.90
	95.6%	13.5%		
Mpic2	1017817	35.9		3.89
	94.9%	13.8%		
Mpic3	1020413	36.0		3.42
	95.4%	13.6%		
Mpic4	1021301	36.1		3.91
	95.6%	13.5%		
Mpic5	1021235	36.0		3.90
	95.6%	13.6%		
Mpic6	1021455	34.1		3.9
	95.6%	13.5%		
Mpic7	1021575	34.1		3.90
	95.7%	13.4%		
Mpic8	1010503	33.8		3.85
	93.5%	14.4%		

than 30. In our scheme, to avoid huge distortion, we only provide the experimental data with both of  $TH$  and  $TH^*$  values less than 2.

Compared with the original algorithm of Wang et al.'s [19] that made only one-way prediction, our algorithm uses

double-way predictions to generate two PEHs in order to hide more information. To minimize PSNR growth, during the second prediction, histogram shifting will be conducted from left to right, which is contrary to the shift order of the first prediction phase. Moreover, the original algorithm

TABLE 4: Experimental results for eight texture images ( $TH = 1$ ,  $TH^* = 1$ ).

Images (512 × 512)	PSNR of the stego image (dB)		ER (bpp: bits per pixel)
	Payload (bits) Increasing rate	Decreasing rate	
Texture1	88669	33.8	0.34
	73.2%	8.5%	
Texture2	144165	34.2	0.55
	60.5%	8.4%	
Texture3	33855	34.8	0.13
	70.5%	5.5%	
Texture4	71375	34.1	0.27
	77.1%	8.2%	
Texture5	860807	37.7	3.28
	70.4%	12.0%	
Texture6	61407	33.3	0.23
	82.8%	9.4%	
Texture7	91077	33.7	0.35
	71.8%	9.4%	
Texture8	31029	34.1	0.12
	86.4%	7.6%	

TABLE 5: Experimental results for eight aerial images ( $TH = 1$ ,  $TH^* = 1$ ).

Images (512 × 512)	PSNR of the stego image (dB)		ER (bpp: bits per pixel)
	Payload (bits) Increasing rate	Decreasing rate	
Aerial 1	94641	32.8	0.36
	66.5%	11.9%	
Aerial 2	155697	33.5	0.59
	53.0%	10.1%	
Aerial 3	235241	32.6	0.91
	37.3%	13.9%	
Aerial 4	87289	32.9	0.33
	70.0%	11.4%	
Aerial 5	151937	33.3	0.58
	54.8%	10.9%	
Aerial 6	104207	33.2	0.41
	65.2%	10.6%	
Aerial 7	81633	33.2	0.31
	62.7%	10.8%	
Aerial 8	122205	32.8	0.47
	43.0%	12.7%	

subtracted 255 when pixel values were greater than 255, or simply added 255 when pixel values were less than zero. The solution of the underflow/overflow problems they used was inefficient. In this paper, we solve the problems in an efficient way based on the study [24] by avoiding some boundary pixels from performing huge transition changes, such as from white to black or from black to white. Increased embedding capacity ( $\Delta EC$ ) in bits and the EC increasing rates (IR), as well as the PSNR and its decreasing rate (DR), for the eight images in Figure 8 are shown in Table 2. Comparing our proposed algorithm with the algorithm proposed by Wang et al., Table 2 shows that, by using our proposed algorithm, there are a huge increase in EC and a slight decrease in PSNR, and the decrease rate of PSNR is significantly smaller than the growth rate of EC.

Table 2 indicates that, by using the proposed double-way prediction, the embedding capacity is increased for all the

images we studied. However, the increment of each image is different, and the increase range of the same image under different  $TH$  and  $TH^*$  conditions is also different. Moreover, PSNR values of all of the images decreased, and the amounts of the decreases are different. It is worth noting that the increase rate of EC is far greater than the decrease rate of PSNR. And an example is used to illustrate this finding. For image “Lena,” when  $TH = 0$  and  $TH^* = 0$ ,  $\Delta EC$  is 35,188 bits, which means that compared with the original algorithm, the proposed algorithm can embed 35,188 more bits in the image, improving the embedding capacity by 67.3%; for the same conditions, the PSNR value maintains the ideal value of 40.9, with a decrease rate of only 8.1%; time cost increased by only 1.74 seconds. Thus, compared with the original algorithm, the proposed algorithm sacrifices a little bit of PSNR value and time cost in order to hide a lot more secret information. This phenomenon is even more pronounced for

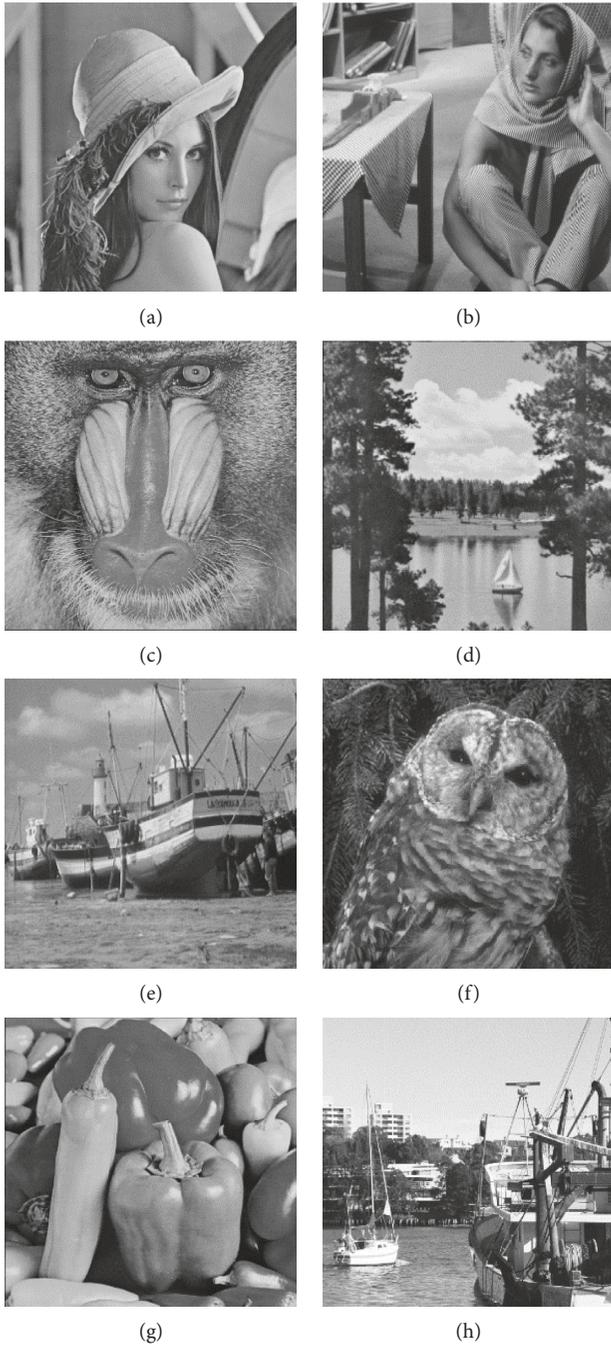


FIGURE 8: Eight common images with sizes of  $512 \times 512$  pixels. (a) Lena. (b) Barbara. (c) Baboon. (d) Lake. (e) Boat. (f) Bird. (g) Peppers. (h) Harbour.

image “*Baboon*.” The amount of hidden information is increased greatly by more than 85%, while the PSNR value decreased by slightly less than 9% when  $TH = 0$  and  $TH^* = 0$ . Similar results are obtained for the other images. Therefore, we come to the conclusion that, for common images, the proposed scheme provides a very large increase in information hiding capacity with only a small reduction in PSNR. Next, we use three groups of images, i.e., medical images, texture images, and aerial images, and each group has eight images to illustrate the relevant issues.

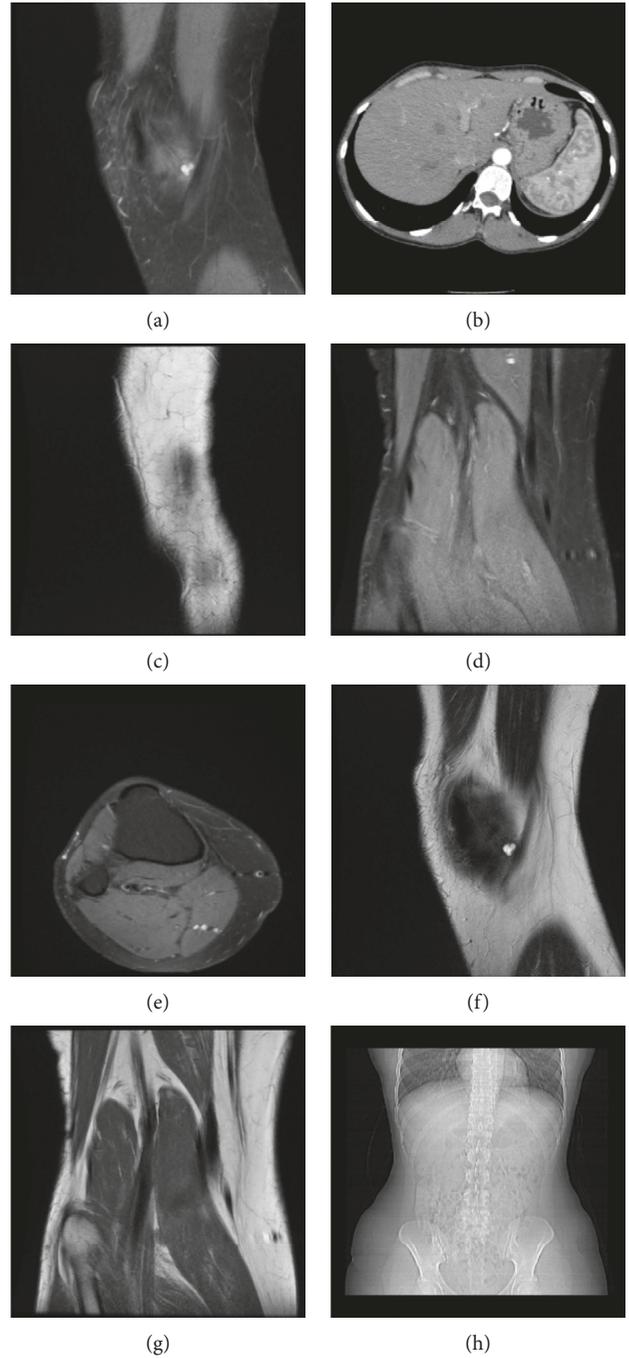


FIGURE 9: Eight medical images with sizes of  $512 \times 512$ .

Figure 9 shows eight medical images with sizes of  $512 \times 512$ , corresponding to these pictures; Table 3 shows us the payloads (bits), PSNR, and embedding rate (ER) with condition values of  $TH = 2$  and  $TH^* = 2$ . Table 3 indicates that the hiding capacities of all eight medical images are increased by about 95%, and their bpp (bits per pixel) values generally are greater than 3.8. In fact, most of the decrease rates in PSNR values are about 13%, which means that compared with the original algorithm [19], the proposed scheme provides significantly increased EC values, while PSNR values are reduced slightly. It is concluded that the

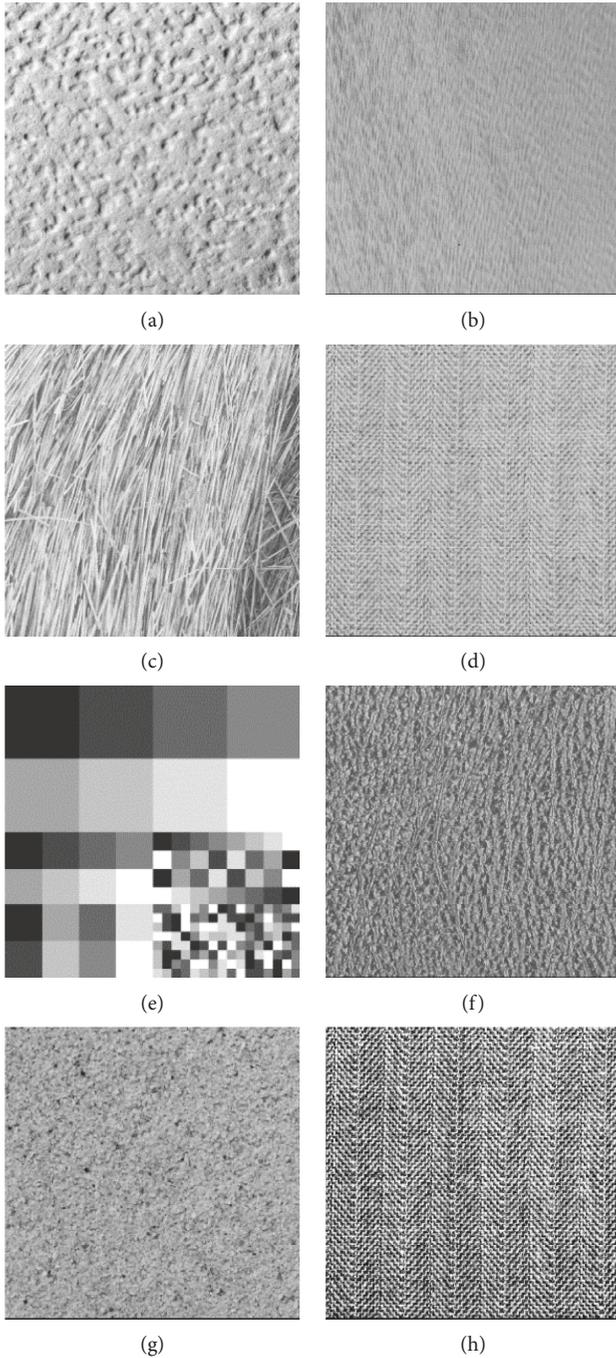


FIGURE 10: Eight texture images with sizes of  $512 \times 512$ .

proposed double-way prediction algorithm significantly increases storage capacity for secret messages and guarantees ideal ER and good PSNR values of stego medical images. Thus, the scheme is particularly suitable for medical images as cover images.

The texture images shown in Figure 10 illustrate the related information in Table 4. Since the structure of the texture of each image is different, the information hiding capacity of each image is also different. Some images, e.g., Figure 10(e), have large smooth blocks that make the histogram sharper and thus allow more information to be

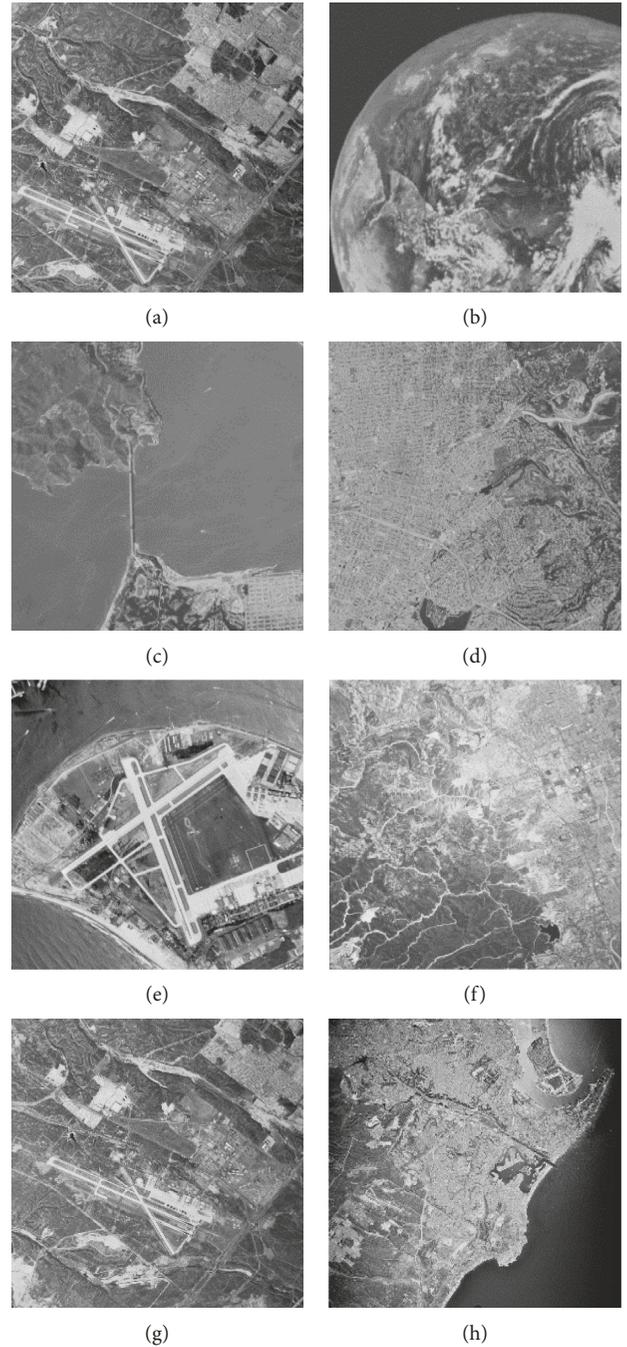


FIGURE 11: Eight aerial images with sizes of  $512 \times 512$ .

hidden, and its ER value is 2.95. However, Figure 10(h) has a complex and nonsmooth texture, and it produces a flatter histogram, thereby dramatically reducing the information that can be hidden, and its ER value is as low as 0.11. Thus, it is apparent that the proposed scheme will allow images with more smooth areas to hide much more information.

For the eight aerial images shown in Figure 11, the payload, PSNR, and ER of them are given in Table 5. From the experimental data, we can get to the conclusion that all of the increasing rates of payload are greater than 50% and all

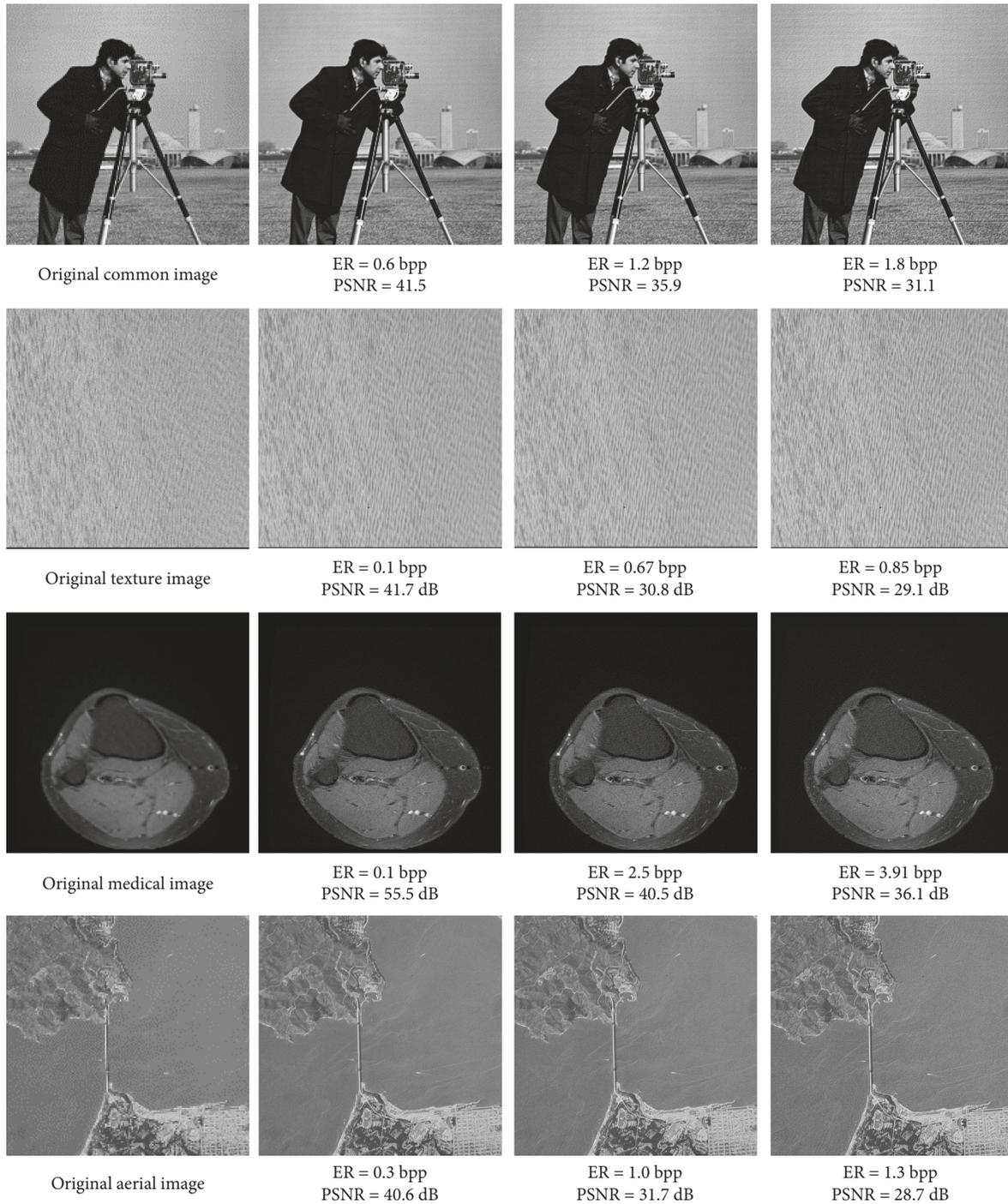


FIGURE 12: Four images of common, medical, texture, and aerial categories and their PSNR values with different ER.

of the decreasing rates of PSNR are about 10%. But the ER value here is a little bit low, being within 1.

From Figure 12, we can hardly see the visual differences of the stego images from common, medical, texture, and aerial categories with different ER and PSNR.

We use 108 images randomly selected from SIPI and Dicom image databases to get the experimental data, and

from Figure 13, we can see that ER is improved while the PSNR is decreased, but the increasing rate of ER is greater than the decreasing rate of PSNR obviously. Among them, Wang et al.'s algorithm has only one parameter,  $TH$ , and the proposed algorithm has two parameters,  $TH$  and  $TH^*$ . Therefore, ER and PSNR value curves of Wang et al.'s are not affected by the second parameter as the proposed

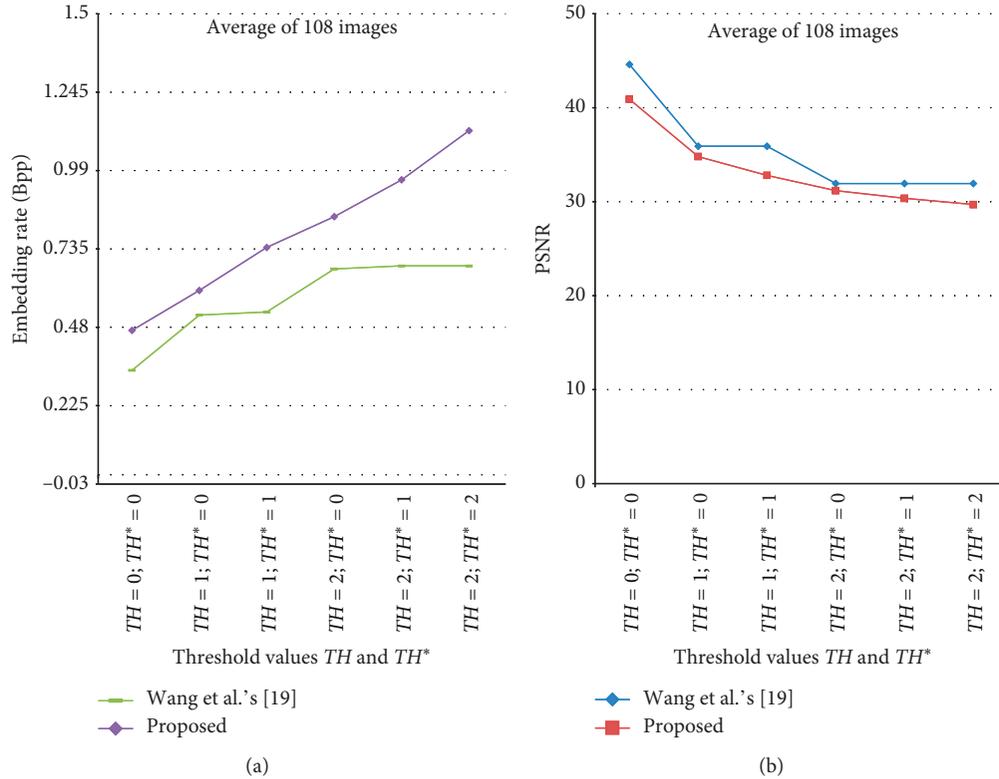


FIGURE 13: Average performance comparison for 108 test images randomly selected from SIPI and Dicom image databases at  $TH, TH^* = 0, 0, 1, 0, 1, 1, 2, 0, 2, 1, \text{ and } 2, 2$ , respectively.

scheme. Furthermore, the ER value of the proposed scheme is steadily increased, and when  $TH=2, TH^*=2$ , the average ER value of those 108 images is about 1.2 which is ideal for medium data payload. At the same time, the PSNR value is 29.7, making embedded information visually imperceptible.

## 5. Conclusions

In this paper, we proposed an improved, bidirectional shift-based reversible data hiding scheme using a double-way prediction strategy. The embedding process consists of two phases, i.e., (1) creation of the first histogram of prediction-error and right-left shifting with embedding and (2) creation of the second histogram of prediction-error and left-right shifting with embedding. Both of the extraction of the embedded message and the recovery of the original image can be realized exactly using the stego image and relevant information. After experiments on common, texture, medical, and aerial images, we found that the embedding rate of medical images increases the most. Whether it is medical images or other types of images, we can all come to the conclusion that the proposed scheme is significantly superior to the previous scheme in embedding capacity, with a little bit of time loss and PSNR decline, such that when  $TH = TH^* = 2$ , the algorithm effectively and significantly improves the information hiding rate by 65% but only reduces the PSNR value about 6%, and the PSNR value remains very close to 30 dB. In our future work, we will

further improve the visual quality with the increasing hiding rate.

## Data Availability

No external data were used to support this study. All derived data sets have been generated in our infrastructure.

## Disclosure

This research was performed as part of the employment of the authors.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported by the Soft Science Foundation of Fujian Province, China (Grant no. B19085).

## References

- [1] D. A. Perez, *Dubrull Indice de Libros Prohibidos*, Vatican, Europe, 1880.
- [2] Typis Vaticanis, *Index Librorum Prohibitorum*, Vatican, Europe, 1900.
- [3] M. Barni, F. Bartolini, I. J. Cox, J. Hernandez, and F. Perez-Gonzalez, "Digital watermarking for copyright protection: a

- communications perspective,” *IEEE Communications Magazine*, vol. 39, no. 8, pp. 90–91, 2001.
- [4] I. J. Cox, M. Miller, J. Bloom et al., *Digital Watermarking and Steganography*, 2nd edition, 2007, <https://www.elsevier.com/books/digitalwatermarkingnd-teganography/cox/978-12-1>.
- [5] C.-C. Lin, Y. Huang, and W.-L. Tai, “A novel hybrid image authentication scheme based on absolute moment block truncation coding,” *Multimedia Tools and Applications*, vol. 76, no. 1, pp. 463–488, 2017.
- [6] C. C. Chang, T. C. Lu, and Y. L. Liu, “A secret information hiding scheme based on switching tree coding,” in *Computer Security in the 21st Century*, D. T. Lee, S. P. Shieh, and J. D. Tygar, Eds., Springer, Boston, MA, USA, 2005.
- [7] C.-C. Chang, C.-Y. Lin, and Y.-Z. Wang, “VQ image steganographic method with high embedding capacity using multi-way search approach,” *Lecture Notes in Computer Science*, pp. 1058–1064, 2005.
- [8] M. Min Wu and B. Bede Liu, “Data hiding in image and video. I. Fundamental issues and solutions,” *IEEE Transactions on Image Processing*, vol. 12, no. 6, pp. 685–695, 2003.
- [9] J. Barton, “Method and apparatus for embedding authentication information within digital data,” US Patent 5,646,997, 1997.
- [10] C. W. Honsinger, P. W. Jones, M. Rabbani et al., “Lossless recovery of an original image containing embedded data,” Google Patents, 2001.
- [11] J. Fridrich, M. Goljan, and R. Du, “Lossless data embedding—new paradigm in digital watermarking,” *EURASIP Journal on Advances in Signal Processing*, vol. 2002, no. 2, 2002.
- [12] M. U. Celik, G. Sharma, A. M. Tekalp et al., “Reversible data hiding,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 8, pp. 157–160, 2002.
- [13] Z. Ni, Y. Q. Shi, N. Ansari et al., “Reversible data hiding,” in *Proceedings of the 2003 International Symposium on Circuits and Systems, ISCAS '03*, Bangkok, Thailand, May 2003.
- [14] J. Jun Tian, “Reversible data embedding using a difference expansion,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 8, pp. 890–896, 2003.
- [15] A. M. Alattar, “Reversible watermark using the difference expansion of a generalized integer transform,” *IEEE Transactions on Image Processing*, vol. 13, no. 8, pp. 1147–1156, 2004.
- [16] C.-C. Lin, W.-L. Tai, and C.-C. Chang, “Multilevel reversible data hiding based on histogram modification of difference images,” *Pattern Recognition*, vol. 41, no. 12, pp. 3582–3591, 2008.
- [17] W. Tai, C. Yeh, and C. Chang, “Reversible data hiding based on histogram modification of pixel differences,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 6, pp. 906–910, 2009.
- [18] D. M. Thodi, J. J. Rodríguez, and S. Member, “Expansion Embedding Techniques for Reversible Watermarking,” *IEEE Transactions on Image Processing*, vol. 16, no. 3, pp. 721–730, 2007.
- [19] W. Wang, J. Ye, and T. Wang, “A high capacity reversible data hiding scheme based on right-left shift,” *Signal Processing*, vol. 150, pp. 102–115, 2018.
- [20] R. M. Wang, “The double exponential distribution: using calculus to find a maximum likelihood estimator,” *The American Statistician*, vol. 38, no. 2, pp. 135–136, 1984.
- [21] D. Zou, Y. Q. Shi, and Z. Zhicheng Ni, “A semi-fragile lossless digital watermarking scheme based on integer wavelet transform,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 10, pp. 1294–1300, 2006.
- [22] Z. Wei Su, Y. Q. Shi, N. Ansari et al., “Robust lossless image data hiding designed for semi-fragile image authentication,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 4, pp. 497–509, 2008.
- [23] H. J. Kim, V. Sachnev, Y. Q. Shi et al., “A novel difference expansion transform for reversible data embedding,” *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 3, pp. 456–465, 2008.
- [24] L. Luo, Z. Chen, M. Chen et al., “Reversible image watermarking using interpolation technique,” *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 1, pp. 187–193, 2010.



**Hindawi**

Submit your manuscripts at  
[www.hindawi.com](http://www.hindawi.com)

