

Received April 21, 2020, accepted May 4, 2020, date of publication May 11, 2020, date of current version May 19, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2993604

New High Capacity Reversible Data Hiding Using the Second-Order Difference Shifting

WEIQING WANG¹ AND WEIHUA WANG²

¹Business College, Southwest University, Chongqing 402460, China

²School of Artificial Intelligence, Chongqing University of Arts and Sciences, Yongchuan 402160, China

Corresponding author: Weiqing Wang (wwqlhy@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61304255, in part by the Natural Science Foundation of Chongqing under Grant cstc 2019 jcyj -msxmX034, and in part by the Southwest University under Grant SWU1909766.

ABSTRACT Reversible data hiding (*RDH*) algorithm is developing rapidly for many years. Usually, the histogram shifting (*HS*) based *RDH* algorithm has two main steps: Firstly, a steep difference histogram is generated with some effective methods for difference. Secondly, by extending and shifting some differences in the histogram, bits of information can be embedded into the cover image which can be restored reversibly. In this paper, we proposed the second-order difference to obtain a steeper difference histogram. Firstly, we slide a window with size 2×2 through the image. For each block, we can get the two first-order differences by calculating the absolute values of the two differences for its two columns. Thus, the second-order difference of each block, which is the absolute values of the difference of the two first-order differences, can be obtained. By extending and shifting the second-order difference, a bit may be embedded into the block finally. Experiments reveal that the proposed algorithm outperforms the previous state-of-the-art *RDH* methods in terms of the computational complexity, image distortion and the embedding performance.

INDEX TERMS Difference expanding, cover image, histogram shifting, reversible data hiding, the second-order difference, watermarking.

I. INTRODUCTION

Information security is becoming more and more important as the rapid development of information technology, so data hiding (*DH*) technology is becoming the best method to protect the information transmitted through public media [1], [2], where secret message is embedded into a cover digital medium such as image, video or text to produce a corresponding marked-medium. Usually, image is taken as the embedding cover medium to hide data. Recently, *DH* schemes are classified into two main types: irreversible data hiding (*IDH*) schemes [3], [4] and *RDH* schemes [5], [6]. For some *DH* applications, such as a communication, medical [7] or military application, their original image must be recovered without any distortion, so *RDH* methods are developed rapidly for the recipient cannot completely recover the cover image with the *IDH* schemes. In general, the existing *RDH* algorithms can be classified into three groups: difference expansion (*DE*), *HS* and encrypted *RDH* methods.

The associate editor coordinating the review of this manuscript and approving it for publication was Jiachen Yang.

The *DE* method is first proposed by Tian [8], where the difference between two adjacent pixels is expanded to embed a secret message. However, it will suffer from undesirable distortion for embedding a location map. Then, Alattar developed it for color image data hiding [9], where the differences of four pixels are employed to embed 3 secret message bits and improved the embedding capacity (*EC*). And to improve the distortion performance at low embedding capacities, Thodi *et al.* [10] embedded data with prediction error expansion (*PEE*) method instead of *DE* method, which makes use of the correlations among the adjacent. Later, Zhang *et al.* [11] computed the values of pairs of difference to generate a two-dimensional-difference histogram, and designed a specifically difference-pair-mapping (*DPM*) to hide data. In 2013, Ou *et al.* developed *PEE* to two dimensional-prediction-error expansion [12]. To obtain a sharp PE histogram and enhance the embedding performance Hu *et al.* [13] optimized histograms modification scheme to propose a minimum rate criterion *RDH* algorithm. To make better use of the pairs of difference with high frequencies, Xue *et al.* [14] proposed a difference pair mapping (*DPM*) *RDH* method, which employs a fine adjusting strategy

for the optimization of embedding. In [15], Wang *et al.* introduced a *DE* scheme to resist unintentional attacks. To enhance the quality of the marked image, Maniriho *et al.* [16] reduced *DE* to introduce a pixel-block based *RDH* method. Xiao *et al.* [17] optimize the path of expansion bins selection according to a specific division for two dimensional prediction error histogram, and with these optimal expansion bins, they can adaptively determine the histogram modification mapping. Recently, various *DE*-based *RDH* algorithms are developed rapidly [18]–[21].

Ni *et al.* [22] proposed the first *HS* based *RDH* algorithm, which employed the high frequencies in image histogram for embedding data. Nevertheless, it has low *EC* for the histogram of cover image. To fully make use of the correlations among the adjacent pixels, Hong *et al.* [23] improved the prediction accuracy with a set of basic pixels for embedding payload. Then, by changing the order of the Markov model, Wang *et al.* [24] designed a *HS* scheme to achieve an efficient tradeoff between the quality of marked image and *EC*. In [25], Li *et al.* designed a *HS* technique to construct some special shifting and embedding functions so that some conventional *HS* algorithms can be taken as specific cases. Unlike the previous methods, Zhang *et al.* [26] collected the pixels with a given complexity to generate a prediction error histogram (*PEH*), and adaptively selected the expansion bins to minimize the embedding distortion. In [27], Kim *et al.* employ the median edge detection to generate the predicted image of cover image so that the secret data can be embedded into the peak points. Later, Wang *et al.* [28] introduced a rate and distortion optimization model and employed genetic algorithm to search the nearly optimal zero and peak bins for the improvement of *EC* and the quality of marked image. To increase the quality of the marked image, Kumar *et al.* [29] construct the segments of size 2, 3, 4 and 5 elements based on the human visual and then secret data is embedded in them. Wang *et al.* [30] use fuzzy *C*-means (*FCM*) clustering method to classify the cover carriers into different clusters for building the multiple histograms. Recently, other *HS*-based *RDH* algorithms are developed [31]–[33].

By embedding the secret data into an encrypted image, the encrypted-image based *RDH* scheme was firstly proposed by Zhang *et al.* [34]. Then, by using public key cryptosystem, Chen *et al.* [35] presented an encrypted signal-based *RDH* scheme according to Paillier homomorphic encryption, which improved the payload and signal quality. In [36], the secret data were embedded into the encrypted image with additive modulo 256 for encryption, and the secret data were extracted and the cover image was restored with preserved mean value. Later, Zhou *et al.* [37] embedded secret data into an encrypted image with a public key modulation mechanism. And using compressive sensing and discrete fourier transform, Xin *et al.* [38] could embed secret data into an encrypted image, which employs both real and imaginary coefficients to recover the original image and provides flexible payload. In the recent years, many other *RDH* algorithms and effective encrypted *RDH* algorithms are developed [39]–[41].

For the encrypted technique, the encrypted image is first obtained from the original image with a special encryption method, which is taken as a cover image to embed secret data by a *DE* or *HS* embedding method. And for *HS*-based technique, it selects pairs of peak and zero bins from the histogram to be expanded for embedding and shifts other bins. Generally, *DE*-based technique selects the peak and zero bins at the center and tail of the histogram for embedding, so it can be taken as a special *HS*-based technique. As we know, the performance of *HS*-based *RDH* algorithm relies heavily on the steepness of histogram and the method for the decision of the pairs of peak and zero bins. To obtain a steeper histogram, a novel *HS*-based *RDH* based on the second-order difference expansion (*SODE*) is proposed.

Next, we'll organize this paper as: the related works will be described in Section 2. Section 3 will introduce the basic fundamentals of our new scheme. Then, we will discuss and analyze the parameters of our new scheme in Section 4. The comparative experiments and analysis will be carried out in Section 5. And Section 6 will summarize the conclusions.

II. RELATED WORKS

Two *DE*-based *RDH* algorithms will be discussed in this section.

A. OU *et al.*'s PAIRWISE ALGORITHM [12]

In a cover image, the prediction error of each pixel x_i is calculated by

$$e_i = x_i - \hat{x}_i \quad (1)$$

where \hat{x}_i is the prediction of x_i calculated with a particular strategy. So, the *PEH* can be generated with

$$h(k) = \#\{1 \leq i \leq N : e_i = k\} \quad (2)$$

where $\#$ is the frequency of prediction errors. Then the prediction error e_i can be expanded to embed one secret bit b or not by

$$e'_i = \begin{cases} 2e_i + b, & e_i \in [-T, T) \\ e_i + T, & e_i \in [T, +\infty) \\ e_i - T, & e_i \in [-\infty, -T) \end{cases} \quad (3)$$

where T is an integer parameter to control the embedding capacity. With the prediction error sequence (e_1, e_2, \dots, e_N) , Ou *et al.* got the pair of prediction error sequence $(e'_1, e'_2, \dots, e'_{N/2})$ by

$$e'_i = (e_{2i-1}, e_{2i}) \quad (4)$$

Then, they can generate the 2 dimensional prediction error histogram (*2D-PEH*) with

$$h(k_1, k_2) = \#\{1 \leq i \leq N/2 : e_{2i-1} = k_1, e_{2i} = k_2\} \quad (5)$$

By expanding and shifting the pairwise errors, Ou *et al.* embedded secret bits in the pairwise errors as Fig. 1 when $T = 1$ in (3).

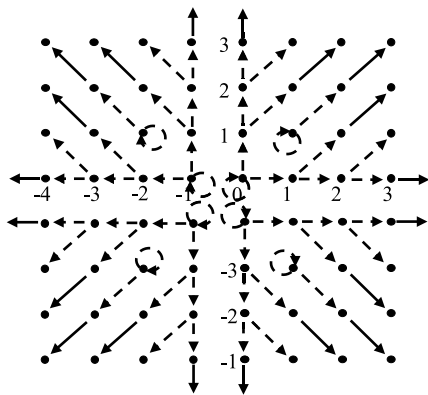


FIGURE 1. Modification mappings for Ou *et al.*'s PEE with $T = 1$. The dotted arrow denotes the expansion of the pairwise prediction errors for embedding, and the solid arrow indicates the shift of the pairwise prediction errors.

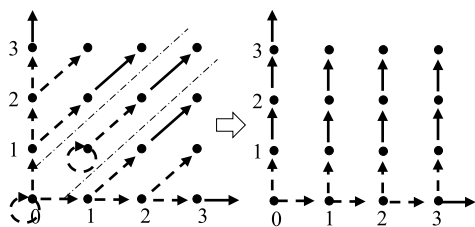


FIGURE 2. 2D-PEH of the first quadrant of Fig. 1 and the rectangle region of its lower triangle region.

B. XIAO *et al.*'s PAIRWISE ALGORITHM [17]

To improve the embedding performance and take full advantage of the redundancy of the cover image, Xiao *et al.* divided the 2D-PEH in Fig. 1 into three regions: the upper triangular region, the diagonal region and lower triangular region. Then, each of the triangle regions was converted into a rectangular region with coordinate transformation. Taking the first quadrant of the 2D-PEH in Fig. 1 as an example, the coordinate transformations of other quadrants were the same. The lower triangle region in the left of Fig. 2 transformed into the rectangle region in the right of Fig. 2 by

$$\begin{cases} e'_{2i-1} = e_{2i-1} - e_{2i} - 1 \\ e'_{2i} = e_{2i} \end{cases} \quad (6)$$

III. THE PROPOSED METHOD

To take advantage of the correlations within prediction errors, the 2D-PEH methods in the related works expand or shift bins in two directions, which provided lessons for our new algorithm. However, their PEE will not be changed, so their embedding performances are not significantly improved. Based on the idea of pairwise prediction error methods, we use two pairs of difference errors to form a second-order difference to obtain the steeper distributed histogram for the better embedding performance.

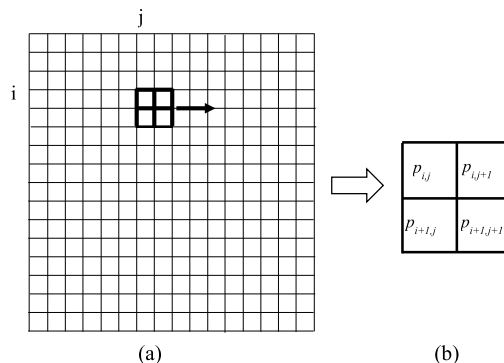


FIGURE 3. The sliding window sw_{ij} of the pixel p_{ij} . (a) is the sliding window sw_{ij} scanning at the location (i, j) of the image I , and (b) is the enlarged window $sw_{i,j}$.

A. SECOND-ORDER DIFFERENCE

For a cover image I with size $w \times h$, where w and h are the width and height of I respectively, a sliding window $sw_{i,j}$ with size 2×2 slides over the image I in raster scanning order as shown in Fig. 3. When it slides to the pixel $p_{i,j}$, which is the pixel at the location (i, j) of image I , the sliding window includes the four pixels $(p_{i,j}, p_{i,j+1}, p_{i+1,j}, p_{i+1,j+1})$. Then the first-order differences $(e1_{i,j}, e2_{i,j})$ can be obtained by

$$\begin{cases} e1_{i,j} = abs(p_{i,j} - p_{i+1,j}) \\ e2_{i,j} = abs(p_{i,j+1} - p_{i+1,j+1}) \end{cases} \quad (7)$$

where function $abs(x)$ returns the absolute value of the variable (x) . Then we can get the second-order difference with

$$d_{i,j} = abs(e1_{i,j} - e2_{i,j}) \quad (8)$$

Let $k = i * (h - 1) + j$, then, we can get the second-order difference sequence $(d_{1,1}, d_{1,2}, \dots, d_{h-1,w-1})$, which can be used to generate the second-order difference histogram (SODH) for embedding. So, the second-order difference occurrences are counted and the corresponding SODH is defined as:

$$hs(k) = \#\{1 \leq i \leq N : d_{i,j} = k\} \quad (9)$$

where $\#$ is the frequency of the second-order difference sequence.

As far as we know, there is a great relationship between the difference histogram distribution and the embedding performance. The steeper the SODH distribution is, the smaller the image distortion is namely the better the embedding performance of the algorithm is. Moreover, the steepness of the SODH for a cover image can be expressed by standard deviation. According to Eq. (1), the steepness of the related works can be calculated by

$$\sigma_{e1} = \sqrt{\frac{1}{(w-1) \times (h-1)} \sum_{i=1}^{h-1} \sum_{j=1}^{w-1} (e1_{i,j} - \bar{e1})^2} \quad (10)$$

$$\sigma_{e2} = \sqrt{\frac{1}{(w-1) \times (h-1)} \sum_{i=1}^{h-1} \sum_{j=1}^{w-1} (e2_{i,j} - \bar{e2})^2} \quad (11)$$

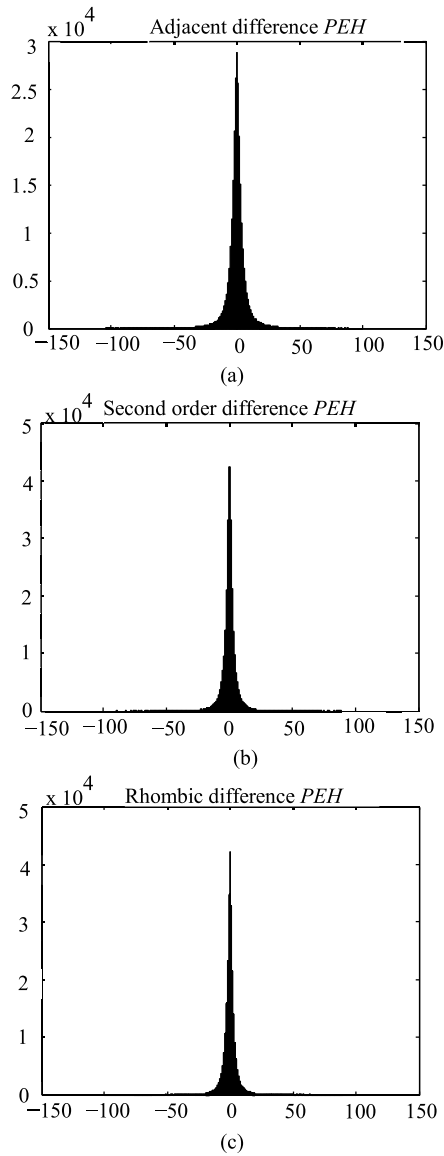


FIGURE 4. The PEHs and SODH of gray-scale image Lena. (a) Adjacent difference PEH, (b) SODH, (c) Rhombic difference PEH.

where $\bar{e}1$ and $\bar{e}2$ are means of the first-order difference sequences $(e1_{1,1}, e1_{1,2}, \dots, e1_{h-1,w-1})$ and $(e2_{1,1}, e2_{1,2}, \dots, e2_{h-1,w-1})$ respectively. And based on Eq. (8), the steepness of the proposed method can be calculated by

$$\sigma_d = \sqrt{\frac{1}{(w-1) \times (h-1)} \sum_{i=1}^{h-1} \sum_{j=1}^{w-1} (d_{i,j} - \bar{d})^2} \quad (12)$$

where \bar{d} is the mean of the second-order difference sequence $(d_{1,1}, d_{1,2}, \dots, d_{h-1,w-1})$.

Second-order difference makes use of the correlation among 4 adjacent pixels, while first-order difference only makes use of the correlation between 2 adjacent pixels, so the correlation of second-order difference is higher than the first-order difference. To further improve the correlation, some

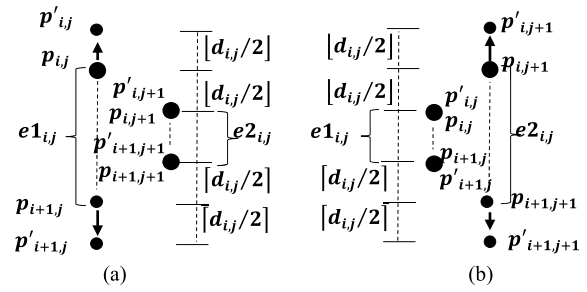


FIGURE 5. The second-order difference expansion. To turn the second-order difference into an even number for embedding secret bits, the pixel pair with the larger first-order difference was expanded by half of the second-order difference in both directions. (a) $e1_{i,j} \geq e2_{i,j}$ (b) $e1_{i,j} < e2_{i,j}$.

literatures make use of rhombus prediction error, but the pixel values are changed considerably when we expand the bins, which will increase the distortion of the cover image. For example, for the standard deviation of the standard gray-scale image Lena with size 512×512 , its standard deviations of first-order adjacent pixel prediction error, rhombic prediction error and the second-order difference are 71.9, 26.0 and 39.2 respectively. The PEHs and SODH of the image Lena are tested as Fig. 4. The standard deviation of the second-order difference is greater than that of the rhombic prediction error, and its histogram is closed to that of rhombic prediction error, rhombic prediction error can only be extended in one direction, that will change the pixels considerably, while second-order difference can use the bidirectional expansion embedding method as following, so it will result in low distortion.

B. BIDIRECTIONAL DIFFERENTIAL EXPANSION AND EMBEDDING

Based on SODH, we select some highest bins for expansion embedding while shift the other bins to create vacancies, so, for each second-order difference $d_{i,j}$, the marked second-order difference $d'_{i,j}$ is calculated by

$$d'_{i,j} = \begin{cases} 2d_{i,j} + b & d_{i,j} \in [0, T] \\ d_{i,j} + T + 1 & d_{i,j} > T \end{cases} \quad (13)$$

where $b \in \{0, 1\}$ is the secret bit to be embedded, and T is a non-negative integer parameter used to control the embedding capacity. To further reduce the distortion in the embedding process, we divide the expanded difference by halves and assign them to the pixel pair with a larger first-order difference, respectively, as shown in Fig. 5. So, the four marked pixels of the two pixel pairs used to calculate the second-order difference $d_{i,j}$ can be obtained with

$$(p'_{i,j}, p'_{i+1,j})$$

$$\begin{aligned}
 & \begin{cases} (p_{i,j}, p_{i+1,j}) & e1_{i,j} < e2_{i,j} \\ (p_{i,j} + \lfloor d_{i,j}/2 \rfloor + b, p_{i+1,j} - \lceil d_{i,j}/2 \rceil) \\ e1_{i,j} \geq e2_{i,j}, \quad p_{i,j} \geq p_{i+1,j}, \quad d_{i,j} \in [0, T] \\ (p_{i,j} - \lfloor d_{i,j}/2 \rfloor - b, p_{i+1,j} + \lceil d_{i,j}/2 \rceil) \\ e1_{i,j} \geq e2_{i,j}, \quad p_{i,j} < p_{i+1,j}, \quad d_{i,j} \in [0, T] \\ (p_{i,j} + \lceil T/2 \rceil + 1, p_{i+1,j} - \lceil T/2 \rceil) \\ e1_{i,j} \geq e2_{i,j}, \quad p_{i,j} \geq p_{i+1,j}, \quad d_{i,j} > T \\ (p_{i,j} - \lceil T/2 \rceil - 1, p_{i+1,j} + \lceil T/2 \rceil) \\ e1_{i,j} \geq e2_{i,j}, \quad p_{i,j} < p_{i+1,j}, \quad d_{i,j} > T \end{cases} \quad (14) \\
 & (p'_{i,j+1}, p'_{i+1,j+1}) \\
 & \begin{cases} (p_{i,j+1}, p_{i+1,j+1}) & e1_{i,j} \geq e2_{i,j} \\ (p_{i,j+1} + \lfloor d_{i,j}/2 \rfloor + b, \quad p_{i+1,j+1} - \lceil d_{i,j}/2 \rceil) \\ e1_{i,j} < e2_{i,j}, p_{i,j+1} \geq p_{i+1,j+1}, \quad d_{i,j} \in [0, T] \\ (p_{i,j+1} - \lfloor d_{i,j}/2 \rfloor - b, \quad p_{i+1,j+1} + \lceil d_{i,j}/2 \rceil) \\ e1_{i,j} < e2_{i,j+1}, \quad p_{i,j} < p_{i+1,j+1}, \quad d_{i,j} \in [0, T] \\ (p_{i,j+1} + \lceil T/2 \rceil + 1, p_{i+1,j+1} - \lceil T/2 \rceil) \\ e1_{i,j} < e2_{i,j+1}, \quad p_{i,j} \geq p_{i+1,j+1}, \quad d_{i,j} > T \\ (p_{i,j+1} - \lceil T/2 \rceil - 1, p_{i+1,j+1} + \lceil T/2 \rceil) \\ e1_{i,j} < e2_{i,j+1}, \quad p_{i,j} < p_{i+1,j+1}, \quad d_{i,j} > T \end{cases} \quad (15)
 \end{aligned}$$

Based on this embedding method, some marked pixels may be beyond the range [0,255] of gray-scale image pixel. That means they encounter the overflow/underflow problem. For brevity, to deal with the overflow/underflow problem, the same method as [12] was used in our proposed algorithm.

C. BIDIRECTIONAL DIFFERENTIAL COMPRESSION AND EXTRACTION

Similarly, for a marked image I' with size $w \times h$, where w and h are the width and height of I' respectively, a sliding window $sw_{i,j}$ with size 2×2 slides over image I' in the reverse order of Fig. 3. When it slides to the marked pixel $p'_{i,j}$, which is the pixel at the location (i, j) of the marked image I' , the sliding window includes the four pixels $(p'_{i,j}, p'_{i,j+1}, p'_{i+1,j}, p'_{i+1,j+1})$. So the marked pairwise difference $(e1'_{i,j}, e2'_{i,j})$ can be obtained by

$$\begin{cases} e1'_{i,j} = abs(p'_{i,j} - p'_{i+1,j}) \\ e2'_{i,j} = abs(p'_{i,j+1} - p'_{i+1,j+1}) \end{cases} \quad (16)$$

Then we can get the second-order difference with

$$d'_{i,j} = abs(e1'_{i,j} - e2'_{i,j}) \quad (17)$$

According to Fig. 5, an expanded difference $d'_{i,j}$ must be even before a secret bit is embedded in as the difference is doubled in both directions. Based on Eq. (14) and (15), if a secret bit 1 is embedded, it must be odd. So, if the marked second-order difference $d'_{i,j} \in [0, 2T + 1]$, we can extract the embedded bit and compress the marked second-order

difference $d'_{i,j}$ as

$$\begin{cases} b = mod(d'_{i,j}, 2) \\ d_{i,j} = \lfloor d'_{i,j}/2 \rfloor \end{cases} \quad (18)$$

The function $mod(x, y)$ returns the modulus after division of x by y . Then, the cover pixels can be restored as

$$\begin{aligned}
 & (p_{i,j}, p_{i+1,j}) \\
 & \begin{cases} (p'_{i,j}, p'_{i+1,j}) & e1'_{i,j} < e2'_{i,j} \\ (p'_{i,j} - \lfloor d_{i,j}/2 \rfloor - b, \quad p'_{i+1,j} + \lceil d_{i,j}/2 \rceil) \\ e1'_{i,j} \geq e2'_{i,j}, \quad p'_{i,j} \geq p'_{i+1,j}, \quad d'_{i,j} \in [0, 2T + 1] \\ (p'_{i,j} + \lfloor d_{i,j}/2 \rfloor + b, \quad p'_{i+1,j} - \lceil d_{i,j}/2 \rceil) \\ e1'_{i,j} \geq e2'_{i,j}, \quad p'_{i,j} < p'_{i+1,j}, \quad d'_{i,j} \in [0, 2T + 1] \\ (p'_{i,j} - \lceil T/2 \rceil - 1, p'_{i+1,j} + \lceil T/2 \rceil) \\ e1'_{i,j} \geq e2'_{i,j}, \quad p'_{i,j} \geq p'_{i+1,j}, \quad d'_{i,j} > 2T \\ (p'_{i,j} + \lceil T/2 \rceil + 1, p'_{i+1,j} - \lceil T/2 \rceil) \\ e1'_{i,j} \geq e2'_{i,j}, \quad p'_{i,j} < p'_{i+1,j}, \quad d'_{i,j} > 2T \end{cases} \quad (19) \\
 & (p_{i,j+1}, p_{i+1,j+1}) \\
 & \begin{cases} (p'_{i,j+1}, p'_{i+1,j+1}) & e1'_{i,j} \geq e2'_{i,j} \\ (p'_{i,j+1} - \lfloor d_{i,j}/2 \rfloor - b, \quad p'_{i+1,j+1} + \lceil d_{i,j}/2 \rceil) \\ e1'_{i,j} < e2'_{i,j}, \quad p'_{i,j+1} \geq p'_{i+1,j+1}, \quad d'_{i,j} \in [0, 2T + 1] \\ (p'_{i,j+1} + \lfloor d_{i,j}/2 \rfloor + b, \quad p'_{i+1,j+1} - \lceil d_{i,j}/2 \rceil) \\ e1'_{i,j} < e2'_{i,j}, p'_{i,j+1} < p'_{i+1,j+1}, \quad d'_{i,j} \in [0, 2T + 1] \\ (p'_{i,j+1} - \lceil T/2 \rceil - 1, p'_{i+1,j+1} + \lceil T/2 \rceil) \\ e1'_{i,j} < e2'_{i,j}, \quad p'_{i,j+1} \geq p'_{i+1,j+1}, \quad d'_{i,j} > 2T \\ (p'_{i,j+1} + \lceil T/2 \rceil + 1, p'_{i+1,j+1} - \lceil T/2 \rceil) \\ e1'_{i,j} < e2'_{i,j}, \quad p'_{i,j+1} < p'_{i+1,j+1}, \quad d'_{i,j} > 2T \end{cases} \quad (20)
 \end{aligned}$$

D. EMBEDDING CAPACITY AND DISTORTION

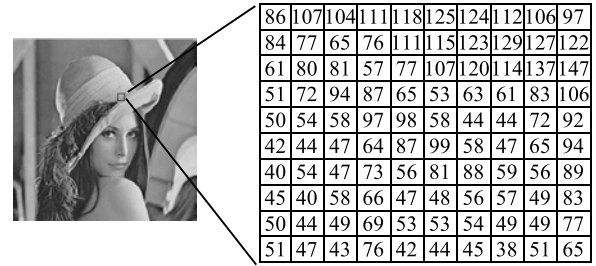
For the pixel $p_{i,j}$, after embedding 1 bit of information, $d_{i,j+1}$ may be changed, the probabilities of its value increasing or decreasing are 50 percent respectively. Based on the above embedding process as shown in Eq. (14) and Eq. (15), when $d_{i,j} = k \in [0, T]$, we can embed a bit into two pixels, thus, the embedding capacity (EC) can be approximated as

$$EC = \sum_{k=0}^T hs(k) \quad (21)$$

where $hs(k)$ is the second-order difference occurrences as shown in Eq. (9). In terms of probability, half of the embedding bits are '0's, and the other half of them are '1's. So, its embedding distortion (ED) can be approximated as

$$\begin{aligned}
 ED = & \sum_{k=0}^T \frac{1}{2} \left(\left(\left(\lfloor \frac{k}{2} \rfloor \right)^2 + \left(\lceil \frac{k}{2} \rceil \right)^2 \right) + \left(\left(\lfloor \frac{k}{2} \rfloor + 1 \right)^2 \right. \right. \\
 & \left. \left. + \left(\lceil \frac{k}{2} \rceil \right)^2 \right) \right) hs(k) + T^2 \sum_{k=T+1}^{+\infty} hs(k)
 \end{aligned}$$

$$= \sum_{k=0}^T \left(\frac{1}{2} \left(\left(\left\lfloor \frac{k}{2} \right\rfloor \right)^2 + \left(\left\lfloor \frac{k}{2} \right\rfloor + 1 \right)^2 \right) + \left(\left\lceil \frac{k}{2} \right\rceil \right)^2 \right) \times hs(k) + T^2 \sum_{k=T+1}^{+\infty} hs(k) \quad (22)$$



E. EMBEDDING ALGORITHM

Given gray-scale image I and a non-negative integer parameter T , the embedding algorithm as follows **Step 1** It scans image I in raster scanning order.

Step 2 For the current pixel $p_{i,j}$ and its corresponding pixel block $(p_{i,j}, p_{i+1,j}, p_{i,j+1}, p_{i+1,j+1})$, its first-order differences $(e_{1,i,j}, e_{2,i,j})$ are obtained by Eq. (7).

Step 3 For the current pixel $p_{i,j}$, its second-order difference $d_{i,j}$ is obtained by Eq. (8).

Step 4 It continues with steps 2-3 until all the pixels have been calculated.

Step 5 The *SODH* of cover image I is obtained by Eq. (9)

Step 6 According to this *SODH*, the threshold value T is determined with Eq. (21). (Note: as Eq. (21) is an approximate equation, the selected value for T in step 6 should be 1 or 2 bigger than that from Eq. (21).)

Step 7 It scans image I again in raster scanning order.

Step 8 For the current pixel $p_{i,j}$ and its second-order difference $d_{i,j}$, the corresponding marked pixels are calculated with Eq. (14) and (15).

Step 9 It continues to calculate all the pixels according to steps 7, and finally we get the marked image I' .

To understand the embedding process of our proposed scheme, we take an image block sized 10×10 from image Lena to embed the bits of information, as shown in Fig. 6. The image block from the hat of image Lena is shown as Fig. 6 (a), the first three specific examples, where the threshold $T = 5$, are given as follows.

Example 1: For pixel $p_{1,1}$, its first-order difference $e_{1,1,1}$ and $e_{2,1,1}$ are calculated as $e_{1,1,1} = abs(86 - 84) = 2$ and $e_{2,1,1} = abs(107 - 77) = 30$, so its second-order difference is calculated as $d_{1,1,1} = abs(2 - 30) = 28$. According to Eq. (14) and (15), its second-order difference is greater than threshold T , so the marked pixels are calculated as $p'_{1,1} = p_{1,1} = 86, p'_{2,1} = p_{2,1} = 84, p'_{1,2} = p_{1,2} + \lfloor T/2 \rfloor + 1 = 107 + 2 + 1 = 110, p'_{2,2} = p_{2,2} - \lceil T/2 \rceil = 77 - 3 = 74$. Here, no bit of information is embedded. Note, in Fig. 6 (d), the pixel p_{22} has been modified as 87, because it is calculated again when we scan the second line.

Example 2: According to example 1, pixels $p_{1,2} = 107$ and $p_{2,2} = 77$ have been modified as $p_{1,2} = 110$ and $p_{2,2} = 74$. For pixel $p_{1,2}$, its first-order differences $e_{1,1,2}$ and $e_{2,1,2}$ are calculated as $e_{1,1,2} = abs(110 - 74) = 36$ and $e_{2,1,2} = abs(104 - 65) = 39$, so its second-order difference is calculated as $d_{1,2} = abs(36 - 39) = 3$. According to Eq. (14) and (15), its second-order difference is less than threshold T , so the marked pixels are calculated as $p'_{1,2} = p_{1,2} = 110, p'_{2,2} = p_{2,2} = 74, p'_{1,3} = p_{1,3} + \lfloor d_{1,2}/2 \rfloor + b =$

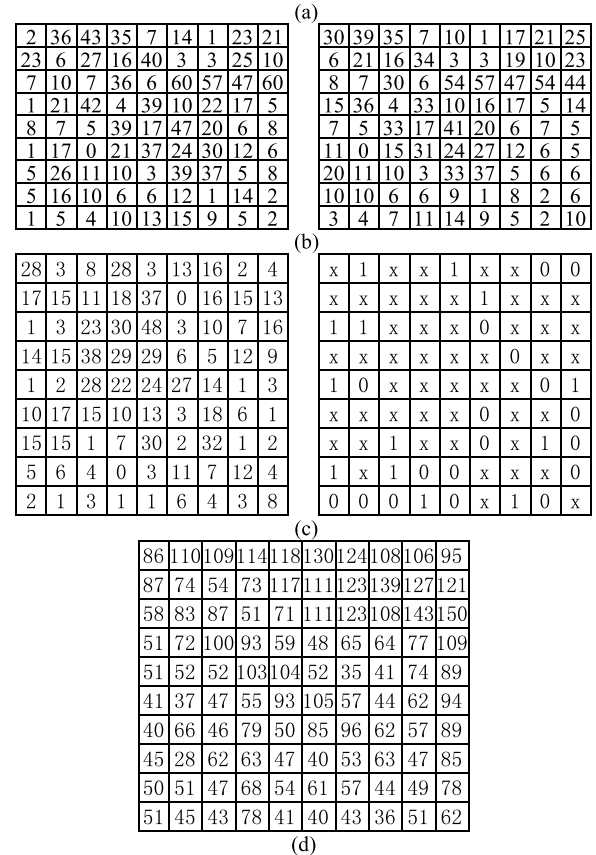


FIGURE 6. An example with $T = 5$. (a) an image block from image Lena, (b) pairwise difference $e_{1,i,j}$ and $e_{2,i,j}$ with Eq. (7), (c) the second-order difference $d_{i,j}$ with Eq. (8), and the secret bits b , where x denote no any bit of information, (d) the embedding results with Eq. (14) and (15).

$104 + 1 + 1 = 106, p'_{2,3} = p_{2,3} - \lceil d_{1,2}/2 \rceil = 65 - 2 = 63$. Here, one bit '1' of information is embedded.

Example 3: According to example 2, pixels $p_{1,3} = 104$ and $p_{2,3} = 65$ have been modified as $p_{1,3} = 106$ and $p_{2,3} = 63$. For pixel $p_{1,3}$, its first-order differences $e_{1,1,3}$ and $e_{2,1,3}$ are calculated as $e_{1,1,3} = abs(106 - 63) = 43$ and $e_{2,1,3} = abs(111 - 76) = 35$, so its second-order difference is calculated as $d_{1,3} = abs(43 - 35) = 8$. According to Eq. (14) and (15), its second-order difference is greater than threshold T , so the marked pixels are calculated as $p'_{1,3} = p_{1,3} + \lfloor T/2 \rfloor + 1 = 106 + 2 + 1 = 109, p'_{2,3} = p_{2,3} - \lceil T/2 \rceil = 63 - 3 = 60, p'_{1,4} = p_{1,4} = 111, p'_{2,4} = p_{2,4} = 76$. Here, no bit of information is embedded.

If we keep on doing that, the embedding results are obtained as Fig. 6. The first-order differences $e1_{i,j}$ and $e2_{i,j}$ obtained by Eq. (7) are shown as the blocks in Fig. 6 (b) respectively, so the second-order differences $d_{i,j}$ obtained by Eq. (8) is shown as the left block in Fig. 6 (c). If we use the randomly selected bits shown as the right block in Fig. 6 (c), then the embedded results with Eq. (14) and (15) is shown in Fig. 6 (d).

F. EXTRACTING ALGORITHM

After receiving the marked gray-scale image I' , we can extract the bits of information and restore the cover image I as following:

Step 1 It scans the marked image I' in the reverse order of Fig. 3.

Step 2 For the current marked pixel $p'_{i,j}$ and its corresponding pixel block $(p'_{i,j}, p'_{i+1,j}, p'_{i,j+1}, p'_{i+1,j+1})$, its first-order differences $(e1'_{i,j}, e2'_{i,j})$ are obtained by Eq. (16).

Step 3 For the current marked pixel $p'_{i,j}$, its second-order difference $d'_{i,j}$ is obtained by Eq. (17).

Step 4 The embedded bit b is extracted and the second-order difference $d_{i,j}$ is compressed respectively with Eq. (18).

Step 5 The cover pixels can be restored with Eq. (19) and (20).

Step 6 It continues with steps 1-4 until all the pixels have been calculated.

Step 7 The cover image I and embedded bits of information are obtained.

To understand the extraction process of our proposed scheme, we take the above pixel block in image Lena to extract the bits of information and recover the cover pixel block, as shown in Fig. 6. The image block from the hat of image Lena is shown as Fig. 6 (a), the last three specific examples, where threshold $T = 5$, are given as follows.

Example 4: According to the above embedding examples, the third pixel $p'_{1,3}$ from the end is the current marked pixel, and its corresponding pixel block $(p'_{1,3}, p'_{2,3}, p'_{1,4}, p'_{2,4})$ is (109, 60, 111, 76). So, its first-order differences $e1'_{1,3}$ and $e2'_{2,3}$ are calculated as $e1'_{1,3} = \text{abs}(109 - 60) = 49$ and $e2'_{1,3} = \text{abs}(111 - 76) = 35$ with Eq. (16). Based on the Eq. (17), its second-order difference is calculated as $d'_{1,3} = \text{abs}(49 - 35) = 14$, which is more than twice the threshold T . Based on Eq. (18), no bit of information is extracted. According to Eq. (19) and (20), the pixels can be restored as $p_{1,3} = p'_{1,3} - \lfloor T/2 \rfloor - 1 = 109 - 2 - 1 = 106$, $p_{2,3} = p'_{2,3} + \lceil T/2 \rceil = 60 + 3 = 63$, $p_{1,4} = p'_{1,4} = 111$ and $p_{2,4} = p'_{2,4} = 76$.

Example 5: According to example 4, the values of pixels $p'_{1,3}$ and $p'_{2,3}$ should be 106 and 63. So the second pixel $p'_{1,2}$ from the end is the current marked pixel, and its corresponding pixel block $(p'_{1,2}, p'_{2,2}, p'_{1,3}, p'_{2,3})$ is (110, 74, 106, 63). So, its first-order differences $e1'_{1,2}$ and $e2'_{1,2}$ are calculated as $e1'_{1,2} = \text{abs}(110 - 74) = 36$ and $e2'_{1,2} = \text{abs}(106 - 63) = 43$ with Eq. (16). Based on Eq. (17), its second-order difference is calculated as $d'_{1,2} =$

$\text{abs}(36 - 43) = 7$, which is less than twice the threshold T . Based on Eq. (18), a bit $b = \text{mod}(7, 2) = 1$ of information is extracted, and the second-order difference is calculated as $d_{1,2} = \lfloor 7/2 \rfloor = 3$. According to Eq. (19) and (20), the pixels can be restored as $p_{1,2} = p'_{1,2} = 110$, $p_{2,2} = p'_{2,3} = 74$, $p_{1,3} = p'_{1,3} - \lfloor d_{1,2}/2 \rfloor - b = 106 - 1 - 1 = 104$ and $p_{2,3} = p'_{2,3} + \lceil d_{1,2}/2 \rceil = 63 + 2 = 65$.

Example 6 According to example 5, the values of pixels $p'_{1,2}$ and $p'_{2,2}$ should be 104 and 65. So the last pixel $p'_{1,1}$ is the current marked pixel, and its corresponding pixel block $(p'_{1,1}, p'_{2,1}, p'_{1,2}, p'_{2,2})$ is (86, 84, 110, 74). So, its first-order differences $e1'_{1,1}$ and $e2'_{1,1}$ are calculated as $e1'_{1,1} = \text{abs}(86 - 84) = 2$ and $e2'_{1,1} = \text{abs}(110 - 74) = 36$ with Eq. (16). Based on the Eq. (17), its second-order difference is calculated as $d'_{1,1} = \text{abs}(2 - 36) = 34$, which is more than twice the threshold T . Based on Eq. (18), no bit of information is extracted. According to Eq. (19) and (20), the pixels can be restored as $p_{1,1} = p'_{1,1} = 86$, $p_{2,1} = p'_{2,1} = 84$, $p_{1,2} = p'_{1,2} - \lfloor T/2 \rfloor - 1 = 110 - 2 - 1 = 107$ and $p_{2,2} = p'_{2,2} + \lceil T/2 \rceil = 74 + 3 = 77$.

IV. DISCUSSIONS

In this paper, we use the peak signal-to-noise-ratio ($PSNR$) to assess the performance of the RDH schemes. The mean square error (MSE) can be calculated as

$$MSE = \frac{1}{w \times h} \sum_{i=0}^{w-1} \sum_{j=0}^{h-1} (p'_{i,j} - p_{i,j})^2 \quad (23)$$

So $PSNR$ can be calculated as

$$PSNR = 10 \log_{10} \left(\frac{(255)^2}{MSE} \right) = 20 \log_{10} \left(\frac{255}{\sqrt{MSE}} \right) \quad (24)$$

A. THRESHOLD T

As shown in Fig. 4 (b) as well as the Eq. (14) and (15), the EC and ED of our proposed algorithm depend on the parameter T . We select eight standard 512×512 gray-scale images as shown in Fig. 7 to test the relation among T , EC and ED , and Table 1 lists the experimental results.

Let us consider image Lena, when threshold T is 1, 5, 10, 15 and 20, respectively, the EC is 108236, 212024, 252279 and 256408 bit, respectively, and their $PSNR$ is 49.7, 43.7, 41.2, 40.4 and 39.7 dB, respectively. So, EC increases with T , and it is increasing more slowly than T does, while ED decreases with T , but it is decreasing more slowly than T does. The same results are obtained when we experiment on the other images shown in Fig. 7.

To further observe this relationship, four of the eight standard images were selected for further experiments under the condition of more T values. The experimental results are shown in Fig. 8. When $T \leq 3$, EC increases rapidly as T increases and ED decreases rapidly as T increases. When $T > 3$, with the increasing of T , EC increases slowly and ED decreases slowly too. And for most applications, $T \leq 3$ is sufficient.

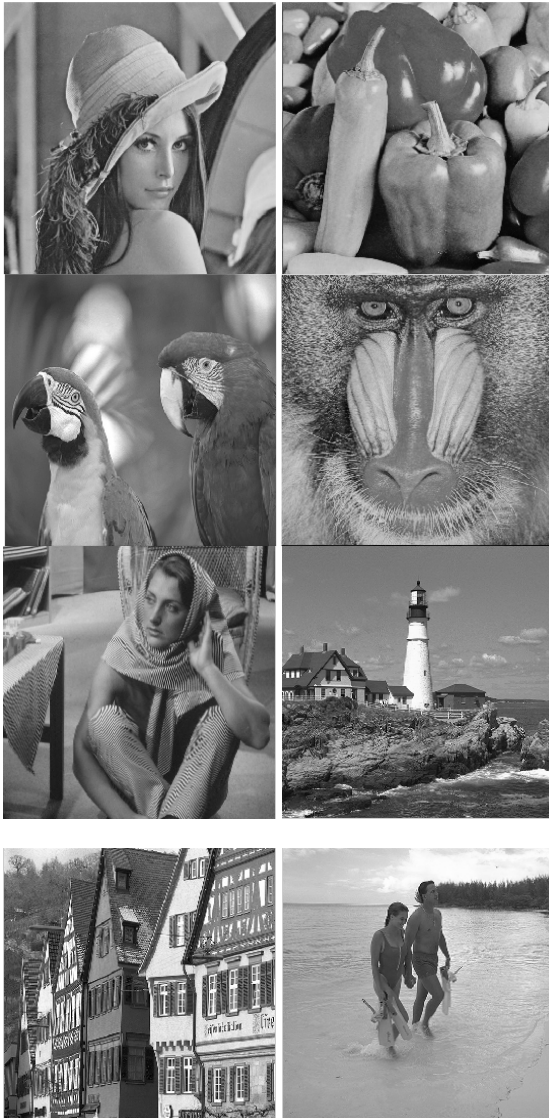


FIGURE 7. Eight standard 512 × 512 gray-scale images, which are Lena, Peppers, Bird, Baboon, Barbara, Chimney, House and Beach respectively.

B. PAYLOAD-LIMITED EMBEDDING

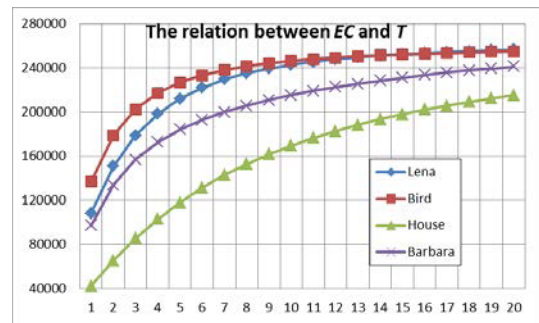
In an application, the payload EC is usually fixed and the corresponding ED is minimized. Based on Eq. (21) and (22), the proposed method can be considered as the following optimal problem

$$\begin{cases} \min ED = \sum_{k=0}^T \left(\frac{1}{2} \left(\left(\left\lfloor \frac{k}{2} \right\rfloor \right)^2 + \left(\left\lfloor \frac{k}{2} \right\rfloor + 1 \right)^2 \right) \right. \\ \quad \left. + \left(\left\lceil \frac{k}{2} \right\rceil \right)^2 \right) hs(k) + T^2 \sum_{k=T+1}^{+\infty} hs(k) \quad (25) \\ s.t. \sum_{k=0}^T hs(k) \geq EC \end{cases}$$

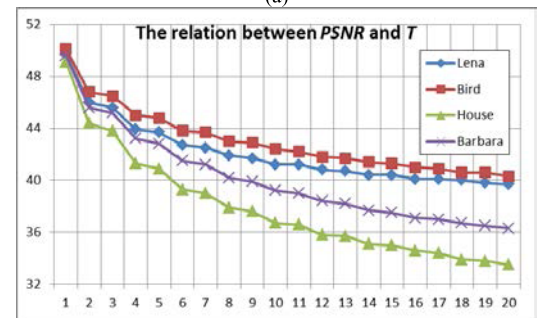
For a given payload EC while minimizing the ED , our optimal problem is to determine the minimal threshold T . Taking image Lena as an example shown in Table 1 and Fig. 8, the T 's

TABLE 1. The relation between threshold T and EC for different standard images with our proposed algorithm.

		T=1	T=5	T=10	T=15	T=20
Lena	$EC(\text{bit})$	108236	212024	242719	252279	256408
	$PSNR(\text{dB})$	49.7	43.7	41.2	40.4	39.7
Peppers	$EC(\text{bit})$	68111	178211	231607	249057	255235
	$PSNR(\text{dB})$	49.4	42.4	39.6	38.8	38.2
Bird	$EC(\text{bit})$	136779	226656	246210	252112	254941
	$PSNR(\text{dB})$	50.1	44.8	42.4	41.3	40.3
Baboon	$EC(\text{bit})$	30776	96457	149068	181589	203086
	$PSNR(\text{dB})$	48.9	40.4	36	34.1	32.5
Barbara	$EC(\text{bit})$	97157	184024	215188	230996	241174
	$PSNR(\text{dB})$	49.6	42.8	39.2	37.5	36.3
Chimney	$EC(\text{bit})$	69626	159073	197933	216397	228058
	$PSNR(\text{dB})$	49.3	41.9	38.0	36.3	34.8
House	$EC(\text{bit})$	42203	117978	169470	197940	215307
	$PSNR(\text{dB})$	49.1	40.9	36.7	35.0	33.5
Beach	$EC(\text{bit})$	96968	194272	232705	247125	253462
	$PSNR(\text{dB})$	49.5	42.9	39.9	38.9	38.1



(a)



(b)

FIGURE 8. The effect of T on the proposed algorithm for four of the standard images in Fig. 7. (a) The relation between EC and T , (b) The relation between $PSNR$ and T .

are 1, 5, and 15 respectively when EC 's are 95000, 110000 and 250000 bits respectively.

V. EXPERIMENTS AND ANALYSIS

To evaluate the performance of our proposed algorithm, we experiment it on the eight standard gray-scale mages with size 512 × 512 shown in Fig. 7, 100 gray-scale images with

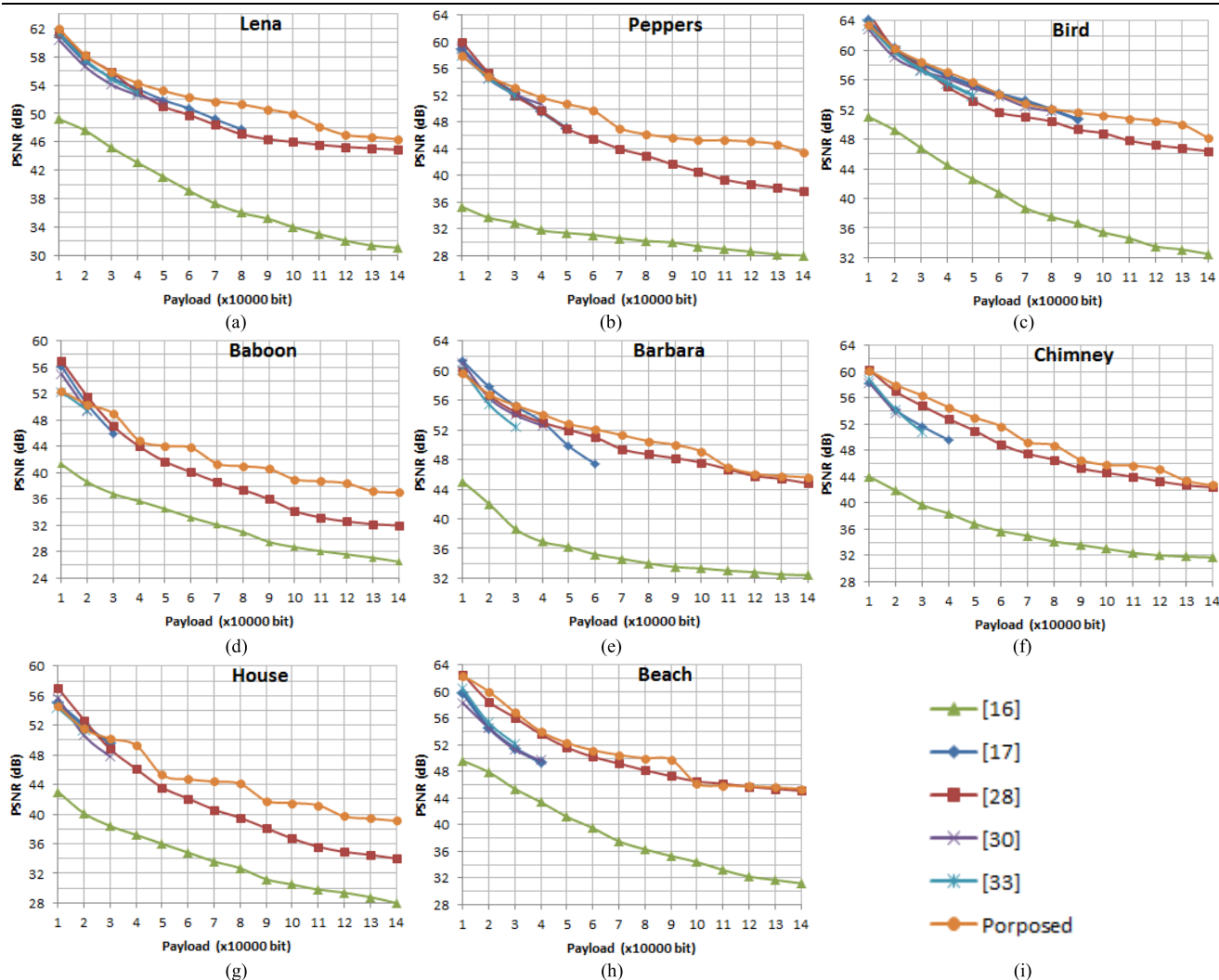


FIGURE 9. Performance comparison of our proposed algorithm with other related RDH algorithms. (a) Lena, (b) Peppers, (c) Bird, (d) Baboon, (e) Barbara, (f) Chimney, (g) House, (h) Beach, (i) legends.

size 816×616 randomly selected from Cambridge database. Before our experiments, we will convert all of the color images into gray-scale images. The software used in our experiments is MATLAB and the experimental hardware is a PC with i7 3.4 GHz CPU and 4.0 GB RAM. The secret bits in the experiments are generated with a pseudo random number generator. To demonstrate the superiority of the proposed algorithm, five state-of-the-art RDH methods [17], [28]–[33] are selected for the comparison.

A. COMPARISON OF PSNR WITH OTHER RELATED SCHEMES AT THE SAME EC

As PSNR can objectively reflect the quality of embedded images, we first compare the proposed algorithm with the state-of-the-art algorithms in terms of PSNR at the same EC for different images. The distortion performance of these algorithms is summarized as Table 2, where $EC = 30000$ and 60000 respectively. For each algorithm, different images

can be embedded with different EC, for example, when $EC = 30000$ bits, the PSNRs of [28] are 55.9, 52.0, 57.8, 47.1, 54.4, 54.8, 48.8 and 58.5 dB respectively, and PSNRs of the proposed algorithm are 55.9, 53.2, 58.5, 49.0, 55.3, 56.4, 50.2 and 60.0 dB respectively. And for each algorithm, the smoother the image, the better the embedding performance, for example, the PSNRs of the image Beach and Baboon are 39.5 and 33.2 dB respectively for [16] when $EC = 60000$, that is to say the image Beach has higher stego-image quality than the image Baboon, other algorithms have the same results. But for the same image, the proposed algorithm has the higher PSNRs than other algorithms at the same EC, for example, for image Peppers, when $EC = 60000$ bits, the PSNRs of [16], [28] and the proposed algorithm are 31.1, 45.7 and 49.7 dB respectively, and [17], [30] and [33] can't embed so much EC into image Peppers. To prove the superior performance of the proposed algorithm, we calculate the average embedding

TABLE 2. The PSNR(DB) on different images for $EC = 30000$ and 60000 bits respectively, 'N' denotes it can't embed so much EC.

Scheme	Lena	Peppers	Bird	Baboon	Barbara	Chimney	House	Beach	Average
[16]	45.2	32.9	46.9	36.8	38.6	39.7	38.4	47.9	40.80
	39.1	31.1	40.8	33.2	35.2	35.7	34.8	39.5	36.18
[17]	55.1	52.3	58.2	46.0	55.2	51.7	49.6	54.6	52.84
	50.7	N	54.1	N	47.4	N	N	N	50.73
[28]	55.9	52.0	57.8	47.1	54.4	54.8	48.8	58.5	53.66
	49.8	45.5	51.6	40.1	51.0	48.9	42.1	50.2	47.40
[30]	54.1	52.2	57.3	N	54.0	N	47.8	54.5	53.32
	N	N	53.0	N	N	N	N	N	N
[33]	54.9	52.1	57.4	N	52.4	50.8	N	55.3	53.82
	N	N	N	N	N	N	N	N	N
Proposed	55.9	53.2	58.5	49.0	55.3	56.4	50.2	60.0	54.81
	52.3	49.7	54.1	43.8	52.0	51.0	44.7	51.2	51.10

performance of these 8 standard images as shown in the last column of Table 2 when $EC = 30000$ bits, these average PSNRs are 40.80, 52.84, 53.66, 53.32, 53.82 and 54.81 respectively. It shows our proposed algorithm outperforms these state-of-the-art methods with about 34.3%, 4.7%, 2.1%, 2.8% and 1.8% increase in terms of PSNRs. Thus, the proposed algorithm is superior to other algorithms.

B. PSNR VALUES UNDER DIFFERENT PAYLOADS

Then, we continue to compare the performance of the proposed method with the five RDH methods as shown in Fig. 9. The results show that the curve of the proposed method is not very smooth and jumps at some points. For example, when $EC = 9 \times 10^4$ and $EC = 6 \times 10^4$ in image Lena and Peppers respectively, the slope of both sides of these points changed greatly. That's because, for some value of T for a certain image, EC is at its maximum at these points, and in order to increase EC , we have to increase the value of T . Although some algorithms occasionally perform better than the proposed algorithm on some images, for example, in the images Peppers and Baboon shown in Fig. 9 (b) and (d), algorithm [28] is superior to the proposed when $EC < 2 \times 10^4$, the proposed algorithm is superior to the algorithm [28] when $EC > 2 \times 10^4$, which can meet most application requirements. For most images, the curve of the proposed is located at the top of the curves of the other algorithms. It shows that the embedding performance of our method is better than other methods. To demonstrate this superior performance, we continued to experiment on 100 images randomly selected from the Cambridge image database, and averaged their performance as shown in Fig. 10, the same results happened.

Fig. 9 and Fig. 10 show that the ECs of algorithms [30] and [33] are less than 2×10^4 bits, and EC of algorithm [17] is less than 6×10^4 bits. At the same time, the curves of methods [30] and [33] are below the curve of our method. For [17],

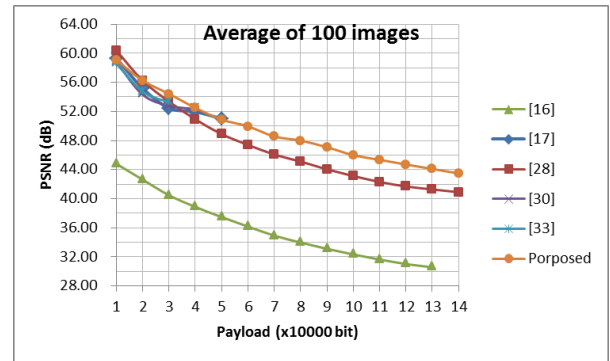


FIGURE 10. Average performance comparison with other related RDH algorithms on 100 images randomly selected from Cambridge image database.

it expands a specific 2 dimensional PEH for embedding, and PEH needs to satisfy two conditions to embed a bit, so its EC is low. And after embedding, each dimension has been modified, so it has large embedding distortion. In [30], it uses Fuzzy C-means (FCM) clustering method to classify the multiple histograms, only some parts of the clusters can be used for embedding, so it has low EC. While for [33], blocks are classified into highly-correlated and lowly-correlated smooth block, and the lowly-correlated block has a low utilization rate, so it also has low EC as [30]. Both [30] and [33] use the multilayer embedding method where some pixels may be modified more than once, so they will cause higher distortion. To search the nearly optimal pairs of zero and peak, a genetic algorithm was employed in [28]. So it has higher embedding performance when EC is low. But when large payload needs to be embedded, the multiple-shifting scheme is employed, which will cause higher distortion. So its performance is worse than the proposed algorithm for most applications and images. For [16], each pixel can be embedded with one bit, but each vector is modified by twice the difference, so it has large EC and distortion. While the proposed algorithm uses the second-order difference which has steeper histogram for each image to embed bits, so it has large EC, in each block, only half of the pixels can be reduced or increased half of the difference, so its distortion is small. Moreover, as the window slides, some pixels increase and then decrease, and some pixels decrease and then increase, so the overall distortion decreases further. So our proposed method is superior to the other methods.

When the payload is small, the steepness of the histogram has little influence on the embedding performance, so the advantage of the SODH is not obvious. Therefore, when the payload is small, some of the state-of-the-art algorithms may have better embedding performance than the proposed algorithm for some images, as shown in Fig. 9 and Fig. 10. As the performance of method [16] is far lower than that of other methods, we further test the embedding performance of the proposed algorithm and the other four state-of-the-art algorithms with low embedding payloads, as shown in Fig. 11

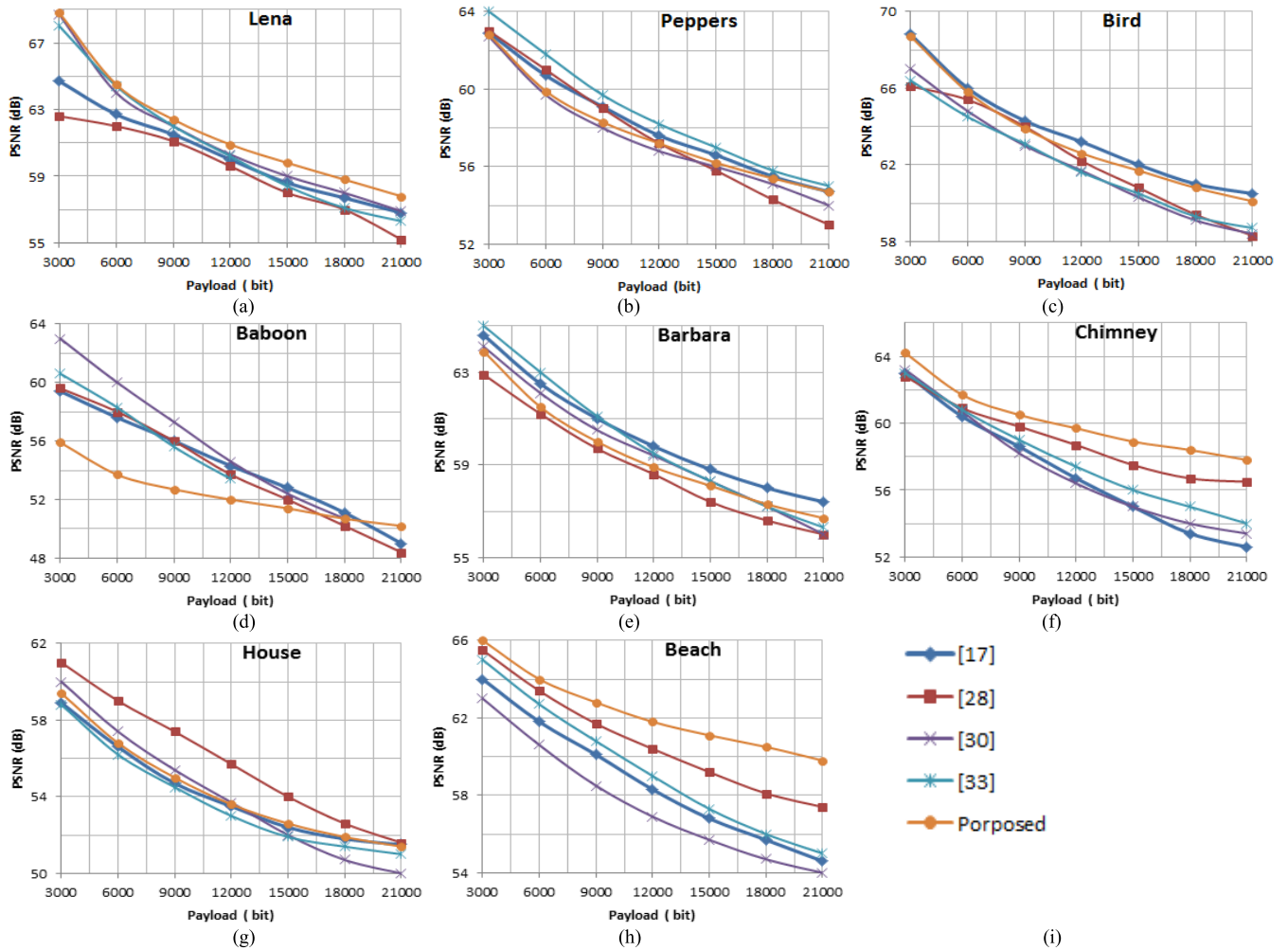


FIGURE 11. Performance comparison with low EC. (a) Lena, (b) Peppers, (c) Bird, (d) Baboon, (e) Barbara, (f) Chimney, (g) House, (h) Beach, (i) legends.

and Fig. 12. Fig. 11 shows, for some images, the embedding performance of our proposed method is worse than these of some other algorithms, and for others, the embedding performance of the proposed algorithm is better than these of other algorithms, for example, algorithm [17], [28], [33] are superior to the proposed for image Peppers, and algorithm [17] is superior to the proposed for image Bird, but the proposed is superior to other algorithms for images Chimney and Beach. Fig. 12 shows the average embedding performance of our proposed method is better than those of the other methods. So our proposed method is superior to other methods with low payload too.

Therefore, the proposed algorithm is superior to other algorithms, especially when the embedding payload is large, this advantage is more obvious, meanwhile, the proposed algorithm is more suitable for most data hiding applications.

C. PRACTICAL EVALUATION OF COMPUTATIONAL COMPLEXITY

Based on Fig. 9 and Fig. 10, for a gray-scale image with size 512×512 , the ECs of algorithms [17], [30] and [33]

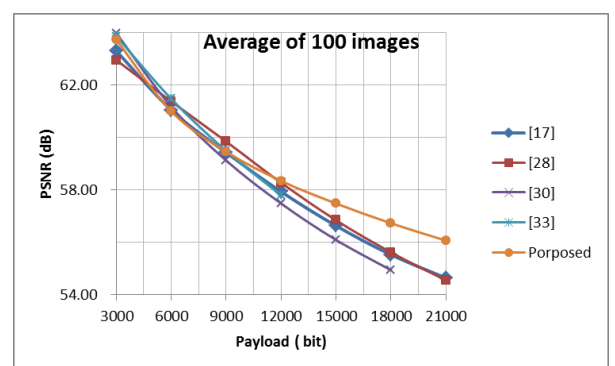


FIGURE 12. Average performance comparison with low EC on 100 images randomly selected from Cambridge image database.

are less than 6000 bits and 3000 bits respectively, and their embedded image quality are less than proposed algorithm. Although the EC of algorithm [16] is large, its embedded image quality is far lower than that of the proposed algorithm, so, its embedding performance is much lower than that of the proposed algorithm. When EC is less than 20000 bits,

TABLE 3. Average CT comparison with [28].

Test image size	Payload (bpp)	CT(second)	
		[28]	Proposed algorithm
512×512	0.2	2.214	0.274
	0.3	2.345	0.349
	0.6	2.617	0.599
1024×1024	0.2	3.188	1.001
	0.3	3.608	1.397
	0.6	4.676	2.595
2048×2048	0.2	5.917	3.308
	0.3	7.228	4.838
	0.6	12.054	9.686

the embedded image quality of [28] may be higher than that of the proposed algorithm for some gray-scale images with size 512×512 . To further evaluate the superior performance of the proposed algorithm, we continue to compare its computational cost with that of [28]. In the experiments, we test the computation time (CT) of the proposed and [28] with different payload on various image sizes. To ensure the effectiveness of the test, 100 gray-scale images were randomly selected from Cambridge image database, which were converted to the images with size 512×512 , 1024×1024 and 2048×2048 respectively. And the embedding payloads used in these experiments were 0.2, 0.3 and 0.6 bit per pixel (bpp) respectively. The results of the average CT on the 100 images are listed in Table 3. It shows that: (1) the CT of both algorithms increases with the increase of the embedding payload, for the EC increases as embedding payload increases, for example, when embedding payload are 0.2, 0.3 and 0.6 bpp respectively for the image with size 1024×1024 , the CT of proposed algorithm are 1.001, 1.397 and 2.595 seconds respectively, and the CT of [28] are 3.188, 3.608 and 4.676 seconds respectively. And (2) the CTs of the proposed algorithm are much smaller than those of [28], for example, for the image with size 1024×1024 , the CTs of the proposed algorithm are decreased by about 68.6%, 61.3% and 31.2% compared with those of [28]. Thus the computational performance of our proposed method is far better than that of [28].

VI. CONCLUSIONS

Based on the analysis of HS and DE technologies, a novel second-order HS algorithm is proposed in this paper. With the two first-order differences of two pairs of pixels in a pixel block, we can get the second-order difference of the four pixels. As the second-order difference histogram of the image is steeper than the first-order difference histogram, the proposed algorithm has good embedding performance. When the EC is small, the steepness of the histogram has little influence on the embedding performance of the image,

so the advantage of the second-order difference histogram is not obvious. Experimental results show that the proposed algorithm is superior to the state-of-the-art algorithms, and it is suitable for most data hiding applications.

ACKNOWLEDGMENTS

Here, we would like to thank the editors and reviewers for their hard work on our paper.

REFERENCES

- [1] C.-C. Chang, C.-Y. Lin, and Y.-H. Fan, "Lossless data hiding for color images based on block truncation coding," *Pattern Recognit.*, vol. 41, no. 7, pp. 2347–2357, Jul. 2008.
- [2] S. K. Ghosal, J. K. Mandal, and R. Sarkar, "High payload image steganography based on Laplacian of Gaussian (LoG) edge detector," *Multimedia Tools Appl.*, vol. 77, no. 23, pp. 30403–30418, Dec. 2018.
- [3] A. Malik, G. Sikka, and H. K. Verma, "A high payload data hiding scheme based on modified AMBTC technique," *Multimedia Tools Appl.*, vol. 76, no. 12, pp. 14151–14167, Jun. 2017.
- [4] I.-C. Lin, Y.-B. Lin, and C.-M. Wang, "Hiding data in spatial domain images with distortion tolerance," *Comput. Standards Interface*, vol. 31, no. 2, pp. 458–464, Feb. 2009.
- [5] P. Rahmani and G. Dastghaibiyfar, "A low distortion reversible data hiding scheme for search order coding of VQ indices," *Multimedia Tools Appl.*, vol. 74, no. 23, pp. 10713–10734, Dec. 2015.
- [6] H. Chen, J. Ni, W. Hong, and T.-S. Chen, "High-fidelity reversible data hiding using directionally enclosed prediction," *IEEE Signal Process. Lett.*, vol. 24, no. 5, pp. 574–578, May 2017.
- [7] G. Coatrieux, C. Le Guillou, J.-M. Cauvin, and C. Roux, "Reversible watermarking for knowledge digest embedding and reliability control in medical images," *IEEE Trans. Inf. Technol. Biomed.*, vol. 13, no. 2, pp. 158–165, Mar. 2009.
- [8] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 8, pp. 890–896, Aug. 2003.
- [9] A. M. Alattar, "Reversible watermark using difference expansion of quads," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, May 2004, p. 377.
- [10] D. M. Thodi and J. J. Rodriguez, "Expansion embedding techniques for reversible watermarking," *IEEE Trans. Image Process.*, vol. 16, no. 3, pp. 721–730, Mar. 2007.
- [11] X. Li, W. Zhang, X. Gui, and B. Yang, "A novel reversible data hiding scheme based on two-dimensional difference-histogram modification," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 7, pp. 1091–1100, Oct. 2013.
- [12] B. Ou, X. Li, Y. Zhao, R. Ni, and Y.-Q. Shi, "Pairwise prediction-error expansion for efficient reversible data hiding," *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 5010–5021, Dec. 2013.
- [13] X. Hu, W. Zhang, X. Li, and N. Yu, "Minimum rate prediction and optimized histograms modification for reversible data hiding," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 3, pp. 653–664, Apr. 2015.
- [14] B. Xue, X. Li, J. Wang, and Z. Guo, "Improved reversible data hiding based on two-dimensional difference-histogram modification," *Multimedia Tools Appl.*, vol. 76, no. 11, pp. 13473–13491, Jun. 2017.
- [15] W. Wang, J. Ye, T. Wang, and W. Wang, "Reversible data hiding scheme based on significant-bit-difference expansion," *IET Image Process.*, vol. 11, no. 11, pp. 1002–1014, Nov. 2017.
- [16] P. Manirihio and T. Ahmad, "Enhancing the capability of data hiding method based on reduced difference expansion," *Eng. Lett.*, vol. 26.1, no. 2018, pp. 45–55.
- [17] M. Xiao, X. Li, Y. Wang, Y. Zhao, and R. Ni, "Reversible data hiding based on pairwise embedding and optimal expansion path," *Signal Process.*, vol. 158, pp. 210–218, May 2019.
- [18] S.-H. Lee, E.-J. Lee, W.-J. Hwang, and K.-R. Kwon, "Reversible DNA data hiding using multiple difference expansions for DNA authentication and storage," *Multimedia Tools Appl.*, vol. 77, no. 15, pp. 19499–19526, Aug. 2018.
- [19] S. Das, R. Maity, and N. P. Maity, "VLSI-based pipeline architecture for reversible image watermarking by difference expansion with high-level synthesis approach," *Circuits, Syst., Signal Process.*, vol. 37, no. 4, pp. 1575–1593, Apr. 2018.

- [20] T. Luo, G. Jiang, M. Yu, C. Zhong, H. Xu, and Z. Pan, "Convolutional neural networks-based stereo image reversible data hiding method," *J. Vis. Commun. Image Represent.*, vol. 61, pp. 61–73, May 2019.
- [21] H.-H. Liu, Y.-C. Lin, and C.-M. Lee, "A digital data hiding scheme based on pixel-value differencing and side match method," *Multimedia Tools Appl.*, vol. 78, no. 9, pp. 12157–12181, May 2019.
- [22] Z. Ni, "Reversible data hiding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 3, pp. 354–362, May 2006.
- [23] W. Hong and T.-S. Chen, "A local variance-controlled reversible data hiding method using prediction and histogram-shifting," *J. Syst. Softw.*, vol. 83, no. 12, pp. 2653–2663, Dec. 2010.
- [24] C.-T. Wang and H.-F. Yu, "A Markov-based reversible data hiding method based on histogram shifting," *J. Vis. Commun. Image Represent.*, vol. 23, no. 5, pp. 798–811, Jul. 2012.
- [25] X. Ai, "A general framework to histogram-shifting-based reversible data hiding," *IEEE Trans. Image Process.* vol. 22, no. 6, pp. 2181–2191, Feb. 2013.
- [26] X. Li, W. Zhang, X. Gui, and B. Yang, "Efficient reversible data hiding based on multiple histograms modification," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 9, pp. 2016–2027, Apr. 2015.
- [27] P.-H. Kim, D.-S. Kim, and K.-Y. Yoo, "Improved Histogram-Shifting-Imitated reversible data hiding scheme," in *Proc. 12th Int. Conf. Inf. Technol. New Generat.*, Apr. 2015, pp. 668–673.
- [28] J. Wang, J. Ni, X. Zhang, and Y.-Q. Shi, "Rate and distortion optimization for reversible data hiding using multiple histogram shifting," *IEEE Trans. Cybern.*, vol. 47, no. 2, pp. 315–326, Feb. 2017.
- [29] R. Kumar, S. Chand, and S. Singh, "An improved Histogram-Shifting-Imitated reversible data hiding based on HVS characteristics," *Multimedia Tools Appl.*, vol. 77, no. 11, pp. 13445–13457, Jun. 2018.
- [30] J. Wang, N. Mao, X. Chen, J. Ni, C. Wang, and Y. Shi, "Multiple histograms based reversible data hiding by using FCM clustering," *Signal Process.*, vol. 159, pp. 193–203, Jun. 2019.
- [31] X. Liang and S. Xiang, "Robust reversible audio watermarking based on high-order difference statistics," *Signal Process.*, vol. 173, Aug. 2020, Art. no. 107584, doi: [10.1016/j.sigpro.2020.107584](https://doi.org/10.1016/j.sigpro.2020.107584).
- [32] S. Wenting, Q. Jing, Y. Jia, H. Rong, and H. Jiankun, "Enabling identity-based integrity auditing and data sharing with sensitive information hiding for secure cloud storage," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 2, pp. 331–346, Jun. 2011.
- [33] S. Weng, Y. Shi, W. Hong, and Y. Yao, "Dynamic improved pixel value ordering reversible data hiding," *Inf. Sci.*, vol. 489, pp. 136–154, Jul. 2019.
- [34] X. Zhang, "Reversible data hiding in encrypted image," *IEEE Signal Process. Lett.*, vol. 18, no. 4, pp. 255–258, Apr. 2011.
- [35] Y.-C. Chen, C.-W. Shiu, and G. Horng, "Encrypted signal-based reversible data hiding with public key cryptosystem," *J. Vis. Commun. Image Represent.*, vol. 25, no. 5, pp. 1164–1170, Jul. 2014.
- [36] S. Agrawal and M. Kumar, "Mean value based reversible data hiding in encrypted images," *Optik*, vol. 130, pp. 922–934, Feb. 2017.
- [37] J. Zhou, W. Sun, L. Dong, X. Liu, O. C. Au, and Y. Y. Tang, "Secure reversible image data hiding over encrypted domain via key modulation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 3, pp. 441–452, Mar. 2016.
- [38] X. Liao, K. Li, and J. Yin, "Separable data hiding in encrypted image based on compressive sensing and discrete Fourier transform," *Multimedia Tools Appl.*, vol. 76, no. 20, pp. 20739–20753, Oct. 2017.
- [39] R. Kumar and K.-H. Jung, "Robust reversible data hiding scheme based on two-layer embedding strategy," *Inf. Sci.*, vol. 512, pp. 96–107, Feb. 2020.
- [40] G.-D. Su, C.-C. Chang, and C.-C. Lin, "A high capacity reversible data hiding in encrypted AMBTC-compressed images," *IEEE Access*, vol. 8, pp. 26984–27000, 2020.
- [41] S. Xiang and X. Luo, "Reversible data hiding in homomorphic encrypted domain by mirroring ciphertext group," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 11, pp. 3099–3110, Nov. 2018.



WEIQING WANG was born in Rongchang, Chongqing, in 1969. He received the M.S. degree in computer software and theory from Southwest University, in June 2007, and the Ph.D. degree in instrument science and technology from Chongqing University, in December 2018. He is currently an Associate Professor with the Business College, Southwest University. His research interests include water marking, machine learning, pattern recognition, and artificial intelligence.



WEIHUA WANG received the M.S. degree in computer application technology from the Chongqing University of Posts and Telecommunications of China, in June 2005, and the Ph.D. degree in computer application technology from the University of Chinese Academy of Sciences, in June 2018. He is currently a Professor of computer science and technology with the Chongqing University of Arts and Sciences, China. His research interests include image processing, signal processing, and telecommunications.

• • •