

# Reversible data hiding based on reducing invalid shifting of pixels in histogram shifting

Yujie Jia<sup>a</sup>, Zhaoxia Yin<sup>a,\*</sup>, Xinpeng Zhang<sup>b</sup>, Yonglong Luo<sup>c</sup>

<sup>a</sup>Key Laboratory of Intelligent Computing & Signal Processing, Ministry of Education, Anhui University, Hefei 230601, P.R.China

<sup>b</sup>School of Computer Science, Fudan University, Shanghai 201203, P.R.China

<sup>c</sup>Anhui Provincial Key Laboratory of Network and Information Security, Wuhu, Anhui 240002, P.R.China

## Abstract

In recent years, reversible data hiding (RDH), a new research hotspot in the field of information security, has been paid more and more attention by researchers. Most of the existing RDH schemes do not fully take it into account that natural image's texture has influence on embedding distortion. The image distortion caused by embedding data in the image's smooth region is much smaller than that in the unsmooth region, essentially, it is because embedding additional data in the smooth region corresponds to fewer invalid shifting pixels (ISPs) in histogram shifting. Thus, we propose a RDH scheme based on the images texture to reduce invalid shifting of pixels in histogram shifting. Specifically, first, a cover image is divided into two sub-images by the checkerboard pattern, and then each sub-image's fluctuation values are calculated. Finally, additional data can be embedded into the region of sub-images with smaller fluctuation value preferentially. The experimental results demonstrate that the proposed method has higher capacity and better stego-image quality than some existing RDH schemes.

**Keywords:** Histogram shifting, invalid shifting pixels, reversible data hiding

## 1. Introduction

As multimedia network technology develops rapidly, information security has attracted extensive attention in various fields. Data hiding refers to hiding data into the cover medium without being detected [1]. It is an effective technology to protect information. Data hiding methods mainly include steganography [2–5], digital watermarking [6–9], construction based data hiding [10–12] and so on.

Most data hiding methods can accurately extract the embedded data but will cause some damage to the cover. However, for some specific scenarios, the distortion of the cover image is not tolerated, as is the slight distortion. In order to satisfy these specific occasions, a reversible data hiding (RDH) scheme was first proposed in 1997 [13]. RDH means that not only the additional data can be extracted, but also the cover image can be reconstructed perfectly without any distortion. According to whether the cover image is encrypted, the RDH can be divided into two categories: RDH in encrypted domain and RDH in plaintext domain. Reversible data hiding in encrypted images (RDHEIs) [14–18] means that the cover image is encrypted, while the RDH in plaintext domain does not encrypt the cover image and embeds the additional data in the plaintext domain directly.

In the past few decades, RDH schemes had been widely studied based on different techniques in the plaintext domain [19],

such as lossless compression [20–24], difference expansion (DE) [25–27], histogram shifting (HS) [28–33], prediction error expansion (PEE) [34–39], etc.

The lossless compression algorithm [20] is the most common method in early RDH. The main idea is to save space for hiding additional data by performing lossless compression on the cover image. In 2016, Qin et al. [24] improved the search order coding based on VQ and proposed an adaptive switching method. Their approach has greatly improved embedding capacity compared to some methods. Obviously, this algorithm relies on the performance of the lossless compression algorithm, and its embedding capacity will be limited due to the weak correlation between bit planes. In addition, direct compression of the cover image will produce noise, which will degrade the image quality. Therefore, the performance of the RDH scheme based on bit plane compression is not satisfactory. Shortly afterwards, DE [25] was proposed to meet higher embedding capacity and better visual quality, and the basic idea of DE is to expand the difference between a pair of pixels to reversibly embed additional data.

In recent years, algorithms based on HS are most commonly used. HS [28] is a typical RDH method that embeds additional data into the peak point pixels of the image gray histogram and shifts others, and HS is put forward by Ni et al. in 2006. The main idea is to embed additional data by keeping the peak point pixels unchanged to hide '0' and shifting the peak point pixels by 1 to hide '1', and shift pixels which are between peak point and zero point by adding 1 or subtracting 1. As long as the peak point is known, the process of extraction and recovery can be easily accomplished. However, the embedding capacity of this method is relatively low. Later, in 2009, Tsai et al. [29] pro-

<sup>\*</sup>This work was supported in part by the National Natural Science Foundation of China under Grant 61872003, Grant U1636206, Grant 61672039 and Grant 61860206004.

\*Corresponding author

Email address: yinzhaoxia@ahu.edu.cn (Zhaoxia Yin)

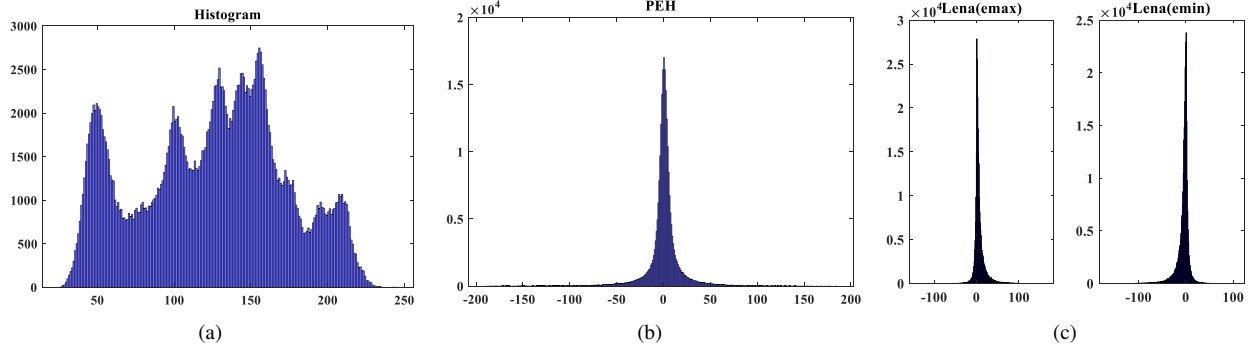


Figure 1: The histograms generated by the methods [28], [29], [31] on Lena. (a)Image histogram [28]. (b) PEH [29]. (c)Asymmetric PEH [31].

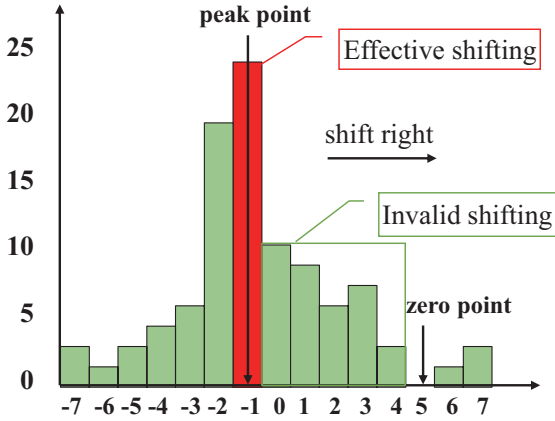


Figure 2: Example of pixels' valid shifting and invalid shifting.

posed a method to construct prediction error histogram(PEH) including the negative PEH and non-negative PEH, which effectively increases the embedding capacity. Since the distribution of the prediction error value is more compact than that of the image pixel value, the PEH is steeper than the image pixels histogram and the capacity can be effectively improved. Afterwards, a method of constructing asymmetric histograms is proposed to embed additional data, which is characterized by the compensation of secondary embedding. Chen et al. [31] used the maximum prediction error and the minimum prediction error of the image to form an asymmetric PEH in 2013. Firstly, embed additional data at the peak point of the maximum PEH, and the pixel value between the peak point and the left zero point subtracts 1. Then, the additional data is embedded at the peak point of the minimum PEH, and the pixel value between the peak point and the right zero point is added by 1. The distortion of the image is reduced by the complementary effect of pixels generated in the embedding process. Fig. 1 shows the histograms generated by the methods [28], [29], [31] on Lena, which are image histogram, prediction error histogram, and asymmetric histograms. It can be seen from Fig. 1(a)(b) that the prediction error histogram is steeper and the frequency of the peak point in the prediction error histogram is much higher than that in the image histogram. In addition, Fig. 1(c) constructs asymmetric histograms, which use secondary embedding at the

same position to achieve compensation, effectively improving the embedding capacity and reducing image distortion.

PEE [34], as another commonly used and effective method, embeds the additional data into prediction error employing DE, and PEE could be seen as a special HS [32]. Subsequently, combining with PEE, the method for achieving better embedding efficiency by pixel value ordering (PVO) was proposed by Li et al [36], which can achieve embedding two bits of additional data for every four pixels. In 2017, Jung K H [39] made improvements based on the method of Li et al. [36], including embedding capacity and image quality. Jung K H first divided the image into  $3 \times 1$  non-overlapping blocks, then predicted the maximum and minimum values within the block, and finally embedded the secret data and recovered the image by the method of PEE. Jung K H's algorithm achieves embedding two bits of additional data for every three pixels.

It can be found that most current methods, such as PEH, PEE or PVO, mostly embed additional data in a certain scanning order, without considering the texture information of the image. Fundamentally, the effect of the number of invalid shifting pixels (ISPs) in the HS on image distortion is not considered. Here, the ISPs means the pixel shifted but not selected for embedding. Fig. 2 gives an example of pixels' valid shifting and invalid shifting. For a series of additional data, the number of valid shifting pixels produced by histogram shifting is fixed during the embedding process, so that the distortion of the image is largely affected by the number of ISPs. That is, the more the number of ISPs, the greater the image distortion. Here, Fig. 3 shows the comparison of the numbers of ISPs in the unsmooth region and smooth region with the same size in Lena, and the same additional data is embedded in the PEHs of the two regions. The unsmooth region and the smooth region selected are the eye and shoulder, respectively, where the size of the block is  $20 \times 20$ , and the number of additional data is 15 bits. It has been experimentally found that the number of ISPs obtained in the unsmooth region is 39, and that in the smooth region is 6. In order to facilitate an intuitive comparison, we marked the number of ISPs in each area in the Fig. 3. Based on this, in this paper, we propose a RDH scheme based on reducing the invalid shifting of pixels in the histogram shifting, which is achieved by preferentially embedding additional data

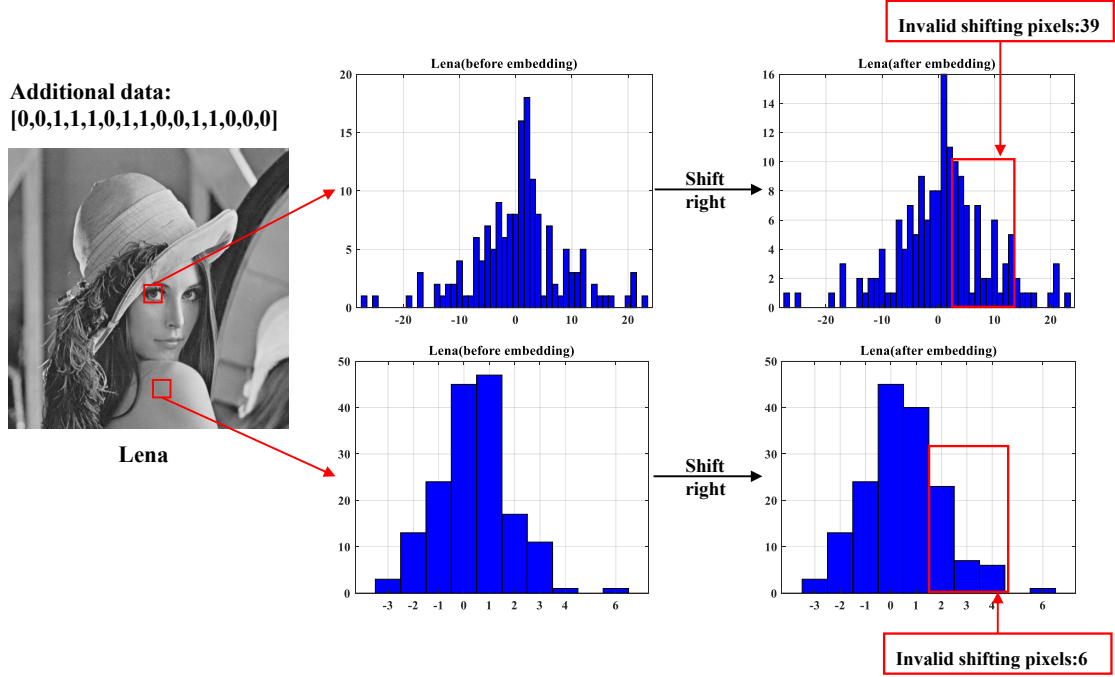


Figure 3: Comparison of the number of ISPs after embedding the same additional data in the PEHs of unsmooth region and smooth region with the same size in Lena.

in the smooth region of the image. Specifically, Firstly, we divide the pixel values in the image in a checkerboard manner. Secondly, we calculate the fluctuation value of each pixel and sort them. Finally, we combine the prediction error and the fluctuation to prioritize the embedding of additional data in small fluctuations. The experimental results of the proposed scheme show that not only the distortions effectively reduced but also the capacity is greatly improved.

The structure of the rest paper is as follows. The detailed process of the proposed scheme is introduced in Section 2. Next, Section 3 presents the results and analysis of the experiment. Finally, we summarize the paper in Section 4.

## 2. Proposed method

In this section, we will describe the proposed method in detail. Firstly, the detailed process of calculating the fluctuation value is introduced. Secondly, give the process of how to calculate the prediction error. Finally, the procedure of embedding and extracting is given.

### 2.1. Calculation of fluctuation value

As shown in Fig. 4, the image is divided into two sets, A and B, in a checkerboard pattern, regardless of the pixels on the boundary, set A is gray while set B is white. Since the calculation methods of the fluctuation values in A and B are the same, here, we will take A as an example to introduce the specific scheme of calculating the fluctuation value.

1. Calculate the local complexity of each pixel in A, the formula is as follows:

$$\Omega_p = |a-d| + |b-c| + |a+c-b-d| + |c+d-a-b|, \quad (1)$$

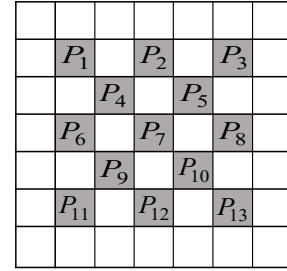


Figure 4: An example of allocating pixels with checkerboard, the gray and white belong to set A and set B respectively.

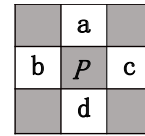


Figure 5: The adjacent pixels of a pixel.

where  $a$ ,  $b$ ,  $c$ , and  $d$  are the upper, left, right, and lower pixel values adjacent to the pixel point  $P$  as shown in Fig. 5. The four terms in the formula correspond to the changes between four nearest pixel neighbors of  $P$  along vertical, horizontal, positive diagonal and negative diagonal direction respectively.

2. In order to measure the smoothness of the pixel more accurately, we calculate the fluctuation value  $F$  of the current pixel value in combination with the local complexity of its adjacent pixel values. Taking the pixel value in Fig. 4 as an example, the computing of the fluctuation

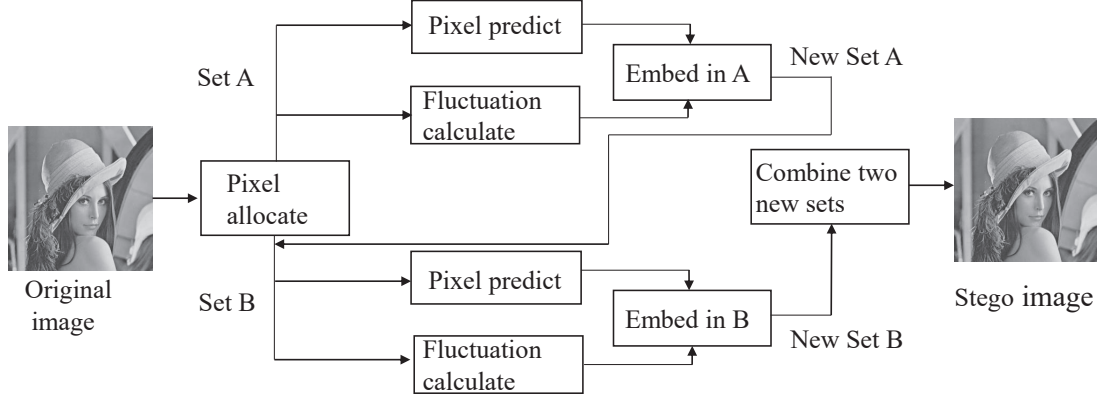


Figure 6: Framework of data embedding procedure.

value  $F$  of the pixel point  $P_1$  is following:

$$F_{P_1} = \Omega_{P_1} + \Omega_{P_4}. \quad (2)$$

Calculate the fluctuation value  $F$  of the pixel point  $P_2$  in Fig. 4 as:

$$F_{P_2} = \Omega_{P_2} + \left\lfloor \frac{\Omega_{P_4} + \Omega_{P_5}}{2} \right\rfloor. \quad (3)$$

Similarly, the calculation of the fluctuation of the pixel  $P_4$  in Fig. 4 should be:

$$F_{P_4} = \Omega_{P_4} + \left\lfloor \frac{\Omega_{P_1} + \Omega_{P_2} + \Omega_{P_6} + \Omega_{P_7}}{4} \right\rfloor. \quad (4)$$

The above can be concluded that when there is one, two, or four adjacent pixels of the pixel in A, their fluctuation values are gained by the equations (2), (3), or (4), respectively.

## 2.2. Calculation of prediction error

Since the processes of A and B layers are similar, we still cite A as an instance to illustrate how to calculate the prediction error. For each pixel, we use four nearest neighbors to predict it.

$$P''_{u,v} = \omega_1 \cdot P_{u-1,v} + \omega_2 \cdot P_{u+1,v} + \omega_3 \cdot P_{u,v-1} + \omega_4 \cdot P_{u,v+1}, \quad (5)$$

where  $P''$  represents the predicted value of the pixel value  $P_{u,v}$ ,  $u$  and  $v$  are the abscissa and ordinate of the pixel value  $P_{u,v}$ , respectively, and  $\omega_1, \omega_2, \omega_3, \omega_4$  are the weights of the corresponding positions, where  $\omega_1 + \omega_2 + \omega_3 + \omega_4 = 1$ .

The prediction error is calculated by

$$e_{u,v} = P_{u,v} - P''_{u,v}. \quad (6)$$

The solution to the four weights  $\omega_i$  in (5) is shown below:

1. Calculate the mean of the four nearest neighbors of each pixel value in A.

$$P'_{u,v} = \left\lfloor \frac{P_{u-1,v} + P_{u+1,v} + P_{u,v-1} + P_{u,v+1}}{4} \right\rfloor, \quad (7)$$

where  $\lfloor \cdot \rfloor$  represents the floor function.

2. The difference between  $P'_{u,v}$  and the nearest neighbors is obtained by subtracting the values of the upper, lower, left and right, respectively, and expressed as  $e_1, e_2, e_3$ , and  $e_4$ . The corresponding  $\omega_i$  is calculated as follows:

$$\omega'_i = \begin{cases} \frac{1}{4} & \text{if } \sum_{i=1}^4 |e_i| = 0 \\ \frac{\sum_{i=1}^4 |e_i|}{1 + |e_i|} & \text{otherwise,} \end{cases} \quad (8)$$

where  $i \in [1, 4]$ , normalize  $\omega'_1, \omega'_2, \omega'_3$ , and  $\omega'_4$  to get  $\omega_1, \omega_2, \omega_3$ , and  $\omega_4$  which are the four weights of upper, lower, left, and right, respectively.

## 2.3. Embedding procedure

Fig. 6 shows the framework of data embedding procedure. In order to avoid the underflow/overflow problem during embedding, regardless of boundary pixels, we modify the pixel with value of 0 or 255 which may occur underflow/overflow during embedding or shifting, the specific operation is that change 0 to 1 and change 255 to 254, and mark 1 in the location map. In addition, for other pixels, if they are modified to 0 or 255 due to embedding or shifting, we mark 0 in the location map. Then, location map is lossless compressed to a smaller size as part of the payload. The detailed embedding process is depicted as:

1. The fluctuation values and prediction errors of all pixels in A are acquired according to B, and then the fluctuation sequence and the prediction error sequence are obtained in the same scanning order. Sort the fluctuation sequence in an ascending order to gain the sequence  $\{F_{A_1}, F_{A_2}, \dots, F_{A_i}\}$  and the sequence  $\{e_{F_{A_1}}, e_{F_{A_2}}, \dots, e_{F_{A_i}}\}$  is got by sorting the prediction error sequence in ascending order of the fluctuation values.
2. Allocate half of the payload to A. We modify the prediction error in sequence  $\{e_{F_{A_1}}, e_{F_{A_2}}, \dots, e_{F_{A_i}}\}$  in turn to em-

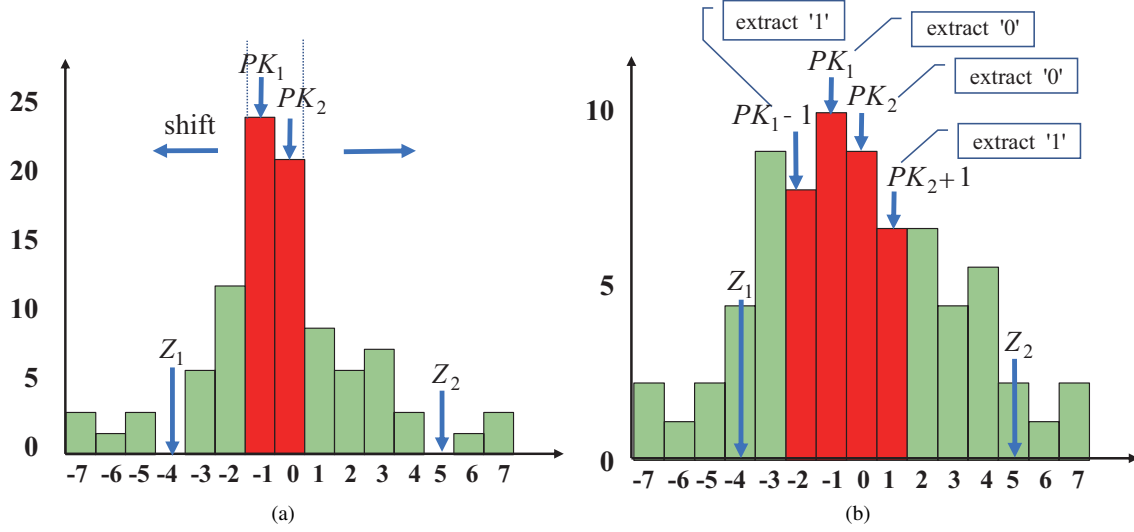


Figure 7: Examples of histogram shifting in embedding and extraction. (a) Embedding. (b) Extraction.

bed additional data. The regulation is:

$$e'_i = \begin{cases} e_i + b & \text{if } e_i = \max(PK_1, PK_2) \\ e_i - b & \text{if } e_i = \min(PK_1, PK_2) \\ e_i + 1 & \text{if } e_i > \max(PK_1, PK_2) \text{ and } e_i < \max(Z_1, Z_2) \\ e_i - 1 & \text{if } e_i < \min(PK_1, PK_2) \text{ and } e_i > \min(Z_1, Z_2) \\ e_i & \text{otherwise} \end{cases} \quad (9)$$

where  $b$  is additional data,  $b \in \{0, 1\}$ ,  $PK_1$  and  $PK_2$  are peak points when embedding, in addition,  $Z_1$  and  $Z_2$  are two zero points closest to the peak points.  $e'_i$  is the marked prediction error, and the corresponding original pixel value is modified by:

$$P'''_i = P''_i + e'_i, \quad (10)$$

where  $P''_i$  is the predicted value,  $e'_i$  is the corresponding marked prediction error, and  $P'''_i$  is the marked pixel value.

3. According to (10), the marked image of A is obtained, and on the basis of it, we implement the embedding of B in the same manner.

#### 2.4. Extraction and recovery procedure

Extraction and recovery are the reverse process, first extract the additional data from B and recover B, then do the same on A. The detailed process is described below:

1. Compute the fluctuation values and marked prediction errors of all pixels in B according to 2.1 and 2.2. It should be noted that because the fluctuation values of B are calculated using the pixels in A, the fluctuation value here is the same as that before B is embedded. The fluctuation value sequence and the marked prediction error sequence are obtained in the same scanning order as the embedding. The sequence of fluctuation values is arranged in ascending order to obtain  $\{F_{B_1}, F_{B_2}, \dots, F_{B_i}\}$ , sorting the

prediction error sequence according to the ascending order of the fluctuation value to obtain the marked prediction error sequence  $\{e'_{F_{B_1}}, e'_{F_{B_2}}, \dots, e'_{F_{B_i}}\}$ .

2. The additional data in sequence  $\{e'_{F_{B_1}}, e'_{F_{B_2}}, \dots, e'_{F_{B_i}}\}$  is sequentially extracted until half of the payload is reached. The way to extract additional data is as follows:

$$b = \begin{cases} 0 & \text{if } e'_i = PK_1 \text{ or } e'_i = PK_2 \\ 1 & \text{if } e'_i = \min\{PK_1, PK_2\} - 1, \text{ or } e'_i = \max\{PK_1, PK_2\} + 1 \end{cases} \quad (11)$$

where  $PK_1$ ,  $PK_2$ , and  $Z_1$ ,  $Z_2$  are the same as that in the procedure of embedding. For the extraction of data,  $b$  is extracted when  $e'_i \in \{\min(PK_1, PK_2) - 1, \max(PK_1, PK_2) + 1, PK_1, PK_2\}$ .

3. In the process of extracting additional data and recovering the original image, the prediction error is recovered as follows:

$$e_i = \begin{cases} e'_i - 1 & \text{if } e'_i = \max(PK_1, PK_2) + 1 \\ e'_i + 1 & \text{if } e'_i = \min(PK_1, PK_2) - 1 \\ e'_i & \text{if } e'_i = PK_1 \text{ or } e'_i = PK_2 \\ e'_i - 1 & \text{if } e'_i \leq \max(Z_1, Z_2) \text{ and } e'_i > \max(PK_1, PK_2) \\ e'_i + 1 & \text{if } e'_i \geq \min(Z_1, Z_2) \text{ and } e'_i < \min(PK_1, PK_2) \\ e'_i & \text{otherwise} \end{cases} \quad (12)$$

Correspondingly, the original pixel values are reconstructed as:

$$P_i = P''_i + e_i, \quad (13)$$

where  $P''_i$  is the predicted pixel value,  $e_i$  is the corresponding recovered prediction error, and  $P_i$  is the recovered pixel value.

4. After the extraction and recovery of B is done, the extraction and recovery of A is completed in the same way

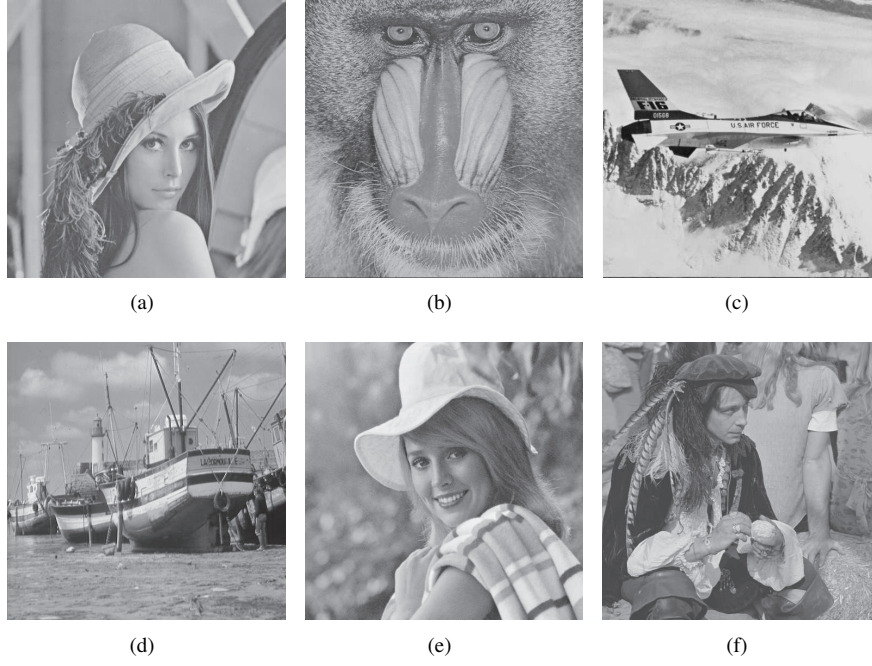


Figure 8: Test images: (a)Lena. (b)Baboon. (c)Airplane. (d)Boat. (e)Elaine. (f)Man.

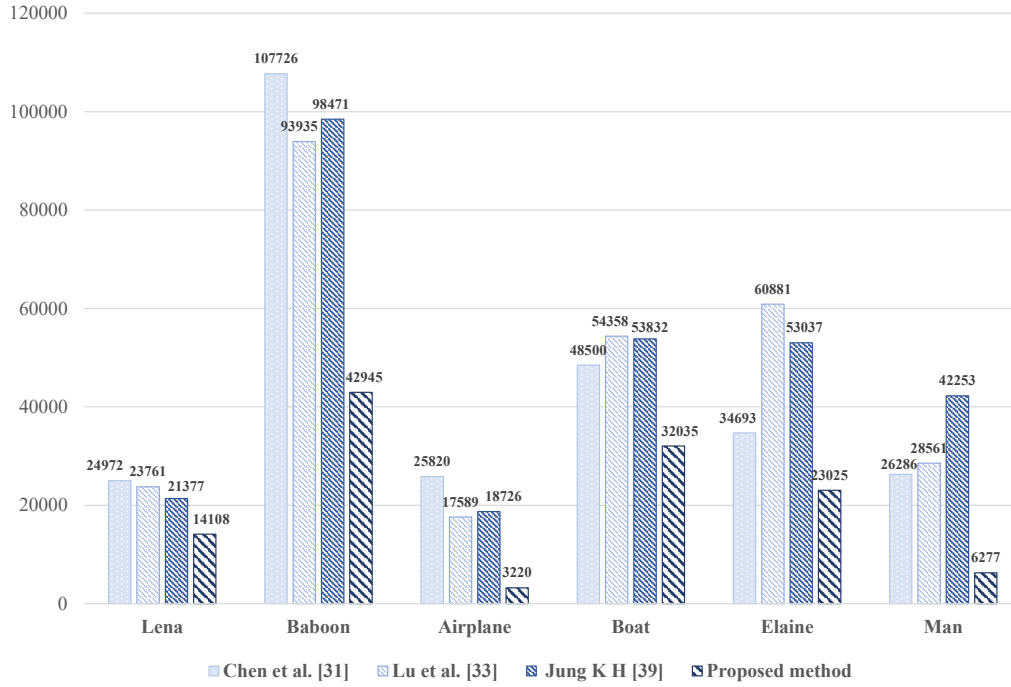


Figure 9: Comparison of the number of ISPs.

based on B. In addition, the pixel value is recovered based on the decompressed location map. Change 254 to 255 and 1 to 0 when the location map is marked 1. Finally, the entire image is completely recovered.

The examples of histogram shifting in embedding and extraction are shown in Fig. 7. It depicts the case of histogram shifting that embeds and extracts additional data when  $PK_2 >$

$$PK_1, Z_2 > Z_1.$$

### 3. Experimental results and analysis

In this section, we will evaluate the performance of the proposed method through several experiments and analyze the results of the experiment. Six standard  $512 \times 512$  sized test images, Lena, Baboon, Airplane, Boat, Elaine, Man, are used in



the experiments, as shown in Fig. 8. We compare with three state-of-the-art schemes of Chen et al. [31], Lu et al. [33], and Jung K H [39] to estimate the performance of the proposed method in terms of image visual quality and embedding capacity.

### 3.1. Comparison with other schemes

The schemes of Chen et al. [31], Lu et al. [33], and Jung K H [39] are chosen for comparing with the proposed scheme. For [31], Chen et al. used maximum prediction error and minimum prediction error to form an asymmetric prediction error histogram. Thus, the image quality is improved with pixel complementary methods. Lu et al. [33] improved on the basis of Chen et al.'s scheme and enhanced the accuracy of the prediction error combining the prediction methods of Feng et al.'s [30] and Lukac et al.'s [40], so there is a big improvement in image quality and embedding capacity. In [39], Jung K H sorted the pixels in the block after dividing the image into a number of  $3 \times 1$  non-overlapping blocks, and the maximum and minimum pixels were used to embed the additional data and the image was recovered by PEE. In our scheme, exploiting the fluctuation combined with PEH, the additional data is preferentially embedded in the position where the fluctuation value is small, hence, effectively reducing the number of ISPs and performance is greatly improved.

#### 3.1.1. Comparison of the number of ISPs

Peak signal-to-noise-ratio (PSNR) is often used as an objective measure of image quality, which reflects the degree of similarity between the stego image and the original image. For grayscale images, it is defined as follows:

$$\text{PSNR} = 10 \times \log_{10} \frac{255^2}{\text{MSE}}, \quad (14)$$

where MSE refers to the mean square error between the original image and the stego image. For a test image size of  $512 \times 512$ , MSE is defined as:

$$\text{MSE} = \frac{1}{512 \times 512} \sum_{i=1}^{512} \sum_{j=1}^{512} (o_{ij} - s_{ij})^2, \quad (15)$$

where  $o_{ij}$  is the original pixel value and  $s_{ij}$  is the marked pixel value. Furthermore, the MSE between the original image and the stego image can be regarded as the sum of the  $\text{MSE}_1$  generated by the valid shifting pixel and the  $\text{MSE}_2$  generated by the ISPs, expressed as:

$$\text{MSE} = \text{MSE}_1 + \text{MSE}_2. \quad (16)$$

Given a string of additional data, the number of valid shifting pixels is fixed, and the corresponding  $\text{MSE}_1$  is also fixed. Therefore, the more the number of ISPs, the larger the corresponding  $\text{MSE}_2$ , resulting in larger MSE. Correspondingly, the smaller the PSNR, the larger the image distortion. In this paper, we reduce the invalid shifting of the pixels by preferentially embedding additional data into smooth pixels. In addition, in order

to illustrate that the proposed method can effectively reduce the invalid shifting of pixels, Fig. 9 gives the comparison of the number of ISPs between the other three methods and our proposed method on six text images when additional data is 10000 bits. It can be seen intuitively from the Fig. 9 that the number of ISPs of the proposed scheme is much less than that of the other three methods on the six test images. For the smooth image, taking the Airplane as an example, the ISPs of the proposed method is 3,220, which is 22600, 14369, and 15506 less than the three methods, respectively. For the unsmooth image, Baboon is taken as an example. The method has fewer ISPs than half of the three methods.

#### 3.1.2. Comparison of PSNR

Fig. 10 is a comparison of the performance of our scheme with other schemes. Here, combined with the image characteristics, we set the payload step size to 0.01(bpp) or 0.02(bpp) to compare the performance. In addition, since all schemes are embedded at one time, for the sake of fairness, Lu et al.'s [33] scheme is embedded only once. It can be seen from Fig. 10 that the proposed scheme has better performance under different payloads than Chen et al. [31], Lu et al. [33] and Jung K H [39]. Taking the texture image Baboon and the smooth image Airplane as example, the PSNR of the proposed scheme is greater than 58 dB when the embedding capacity of baboon is 0.02 (bpp), but the PSNR of the schemes of Chen et al., Lu et al. and Jung KH cannot reach 56 dB. For Airplane, when the embedding capacity is 0.03 (bpp), the proposed schemes, Chen et al., Lu et al. and Jung K H, have PSNR of 64.45, 58.97, 59.64, and 59.49 dB, respectively. Our scheme outperforms [31], [33] and [39] with an increase of PSNR by 5.48, 4.81 and 4.96 dB.

Further, Table 1 demonstrates the average gains of PSNR when the embedding capacity is 0.01~0.05 (bpp) compared with the above three schemes. As seen in the table that compared with the schemes of Chen et al. [31], Lu et al. [33] and Jung K H [39], the proposed scheme provides a consistent gain in different test images. In particular, compared to [39], the gain of the test image Man reaches 6.76. Combined with the given Fig. 10 and Table 1, we can find that the proposed method can reduce the distortion to some extent compared with the other three methods, whether it is smooth image or unsmooth image.

#### 3.1.3. Comparison of embedding capacity

The comparison of embedding capacity between the proposed scheme and Chen et al.'s [31], Lu et al.'s [33], Jung K H's [39] schemes in six test images is displayed in Fig. 11. It can be obtained intuitively from the table that the proposed scheme effectively improves the embedding capacity. For the texture image Baboon, the embedding capacity is 18371 bits, 24997 bits, 16271 bits respectively in [31], [33], [39], while the proposed scheme is 25038 bits. Compared with the three schemes, the proposed scheme increased by 36.29%, 0.16% and 53.88%, differently. For the smooth image Lena, the proposed scheme is 38.83%, 0.33%, and 96.75% higher than the other three schemes. This is because the proposed scheme selects double peak, which effectively increases the embeddable

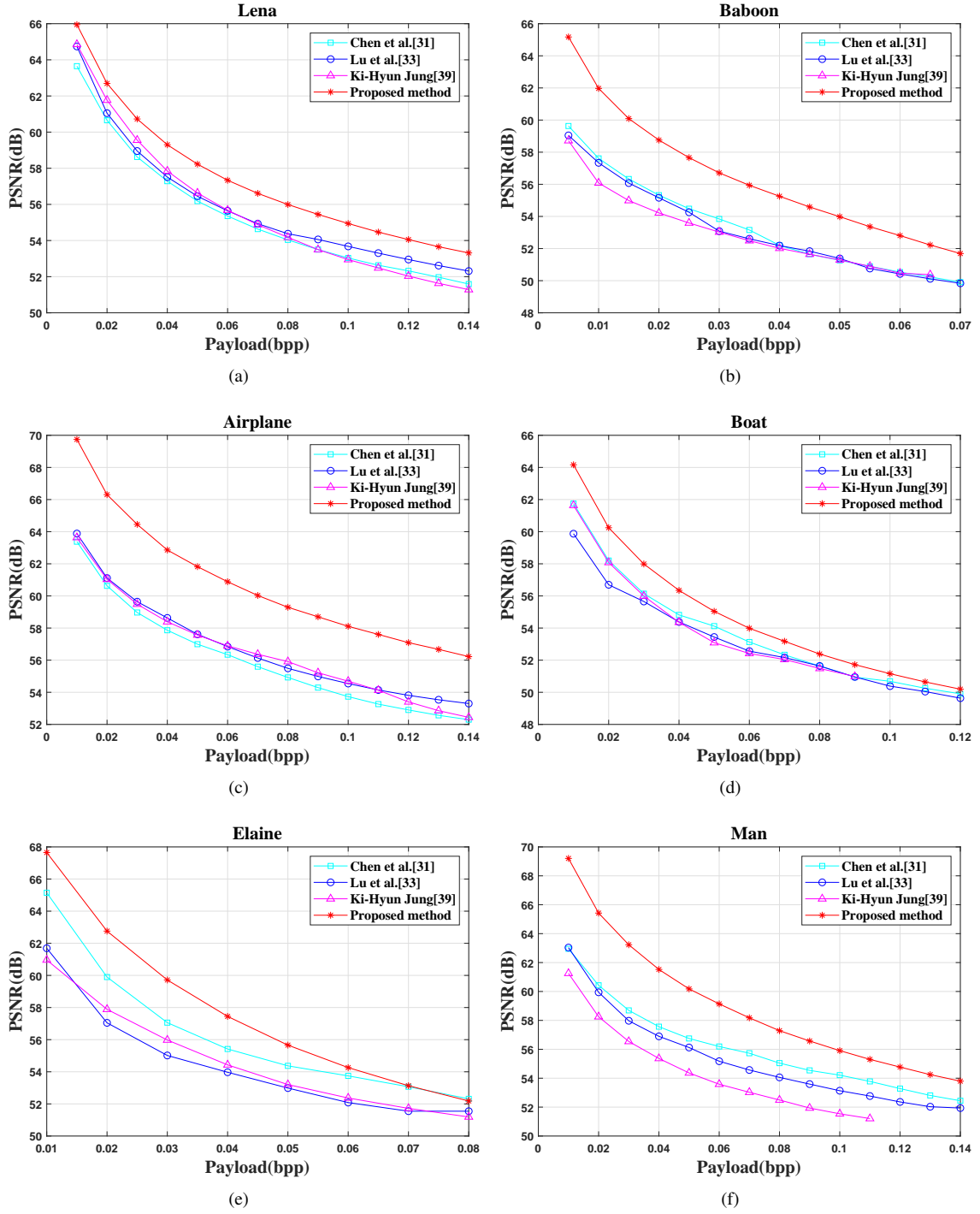


Figure 10: Comparison of the performance with other schemes.

Table 1: The average PSNR gains (in dB) for the payload 0.01~0.05 (bpp) by comparing with three schemes.

Image	Gain vs Chen et al. [31]	Gain vs Lu et al. [33]	Gain vs Jung K H [39]
Lena	2.10	1.64	1.25
Baboon	3.29	3.48	4.00
Airplane	5.47	4.86	5.01
Boat	1.75	2.74	2.12
Elaine	2.27	4.50	4.16
Man	4.63	5.12	6.76



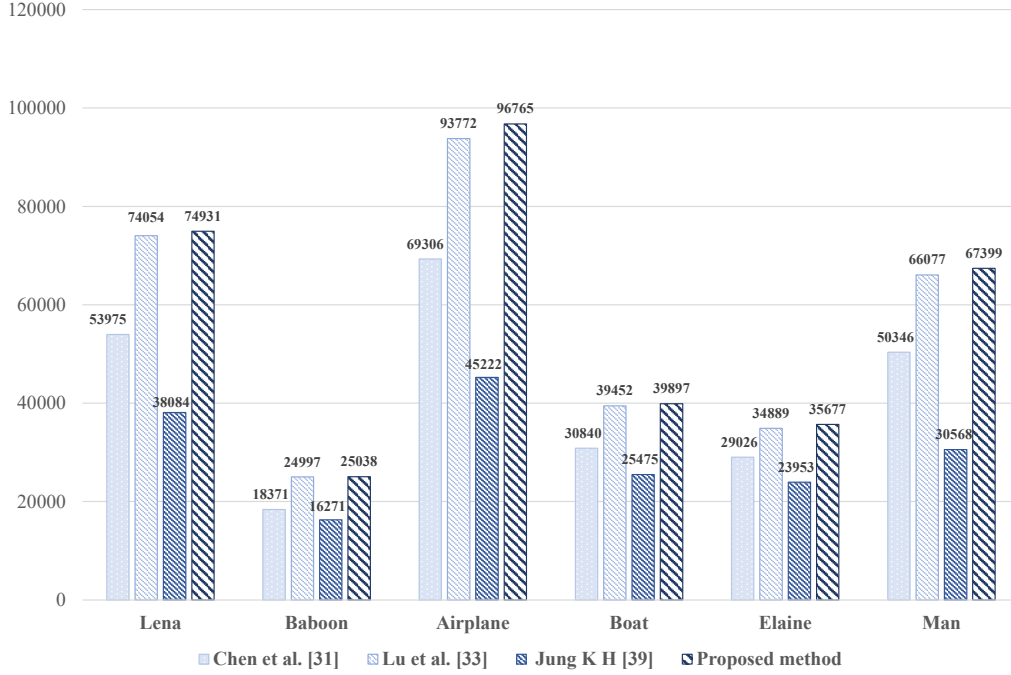


Figure 11: Comparison of capacity for each image by using four schemes.

position in the image. Here, it should be noted that the embedding capacity of Chen et al. [31], Lu et al. [33] and the proposed method may vary slightly due to the embedded additional data, but the variation is very small. This is because these three methods rely on the value of the stego pixel during the embedding process. In addition, it can be seen that the method in [33] has a small gap between the embedded capacity and the proposed method because the image is divided into two parts in [33], and each part is embedded twice.

#### 4. Conclusions

In this paper, we propose a RDH scheme to reduce the number of ISPs combined with image texture. The image is divided into two parts by the checkerboard pattern, the fluctuation value of each pixel in each part is calculated to determine the degree of smoothness, and the fluctuation value is sorted in ascending order. Furthermore, combined with the prediction error, the additional data is preferentially embedded in the position of the smooth pixel, thereby effectively reducing the invalid shifting of pixels. It is the advantage of using checkerboard method that makes the auxiliary information less. The improvement of experimental performance is divided into two parts. First of all, compared with three state-of-the-art schemes [31], [33], [39], due to the preference for smooth pixel point while embedding, our PSNR has been significantly improved, and good image visual quality is acquired. Secondly, our embedding capacity has been greatly raised by selecting double peak. In future research, we will further study the performance differences between smooth images and texture images, explore the characteristics of smooth image and texture image, and design more

effective methods to reduce the number of ISPs for smooth image and non-smooth image respectively.

#### References

- [1] W. Zeng, Digital watermarking and data hiding: technologies and applications, in: Proc. Int. Conf. Inf. Syst., Anal. Synth., Vol. 3, 1998, pp. 223–229.
- [2] W. Lu, L. He, Y. Yeung, Y. Xue, H. Liu, B. Feng, Secure binary image steganography based on fused distortion measurement, *IEEE Transactions on Circuits and Systems for Video Technology* (2018) 1–1.
- [3] Z. Wang, Z. Qian, X. Zhang, M. Yang, D. Ye, On improving distortion functions for jpeg steganography, *IEEE Access* 6 (2018) 74917–74930.
- [4] Z. Wang, X. Zhang, Z. Yin, Joint cover-selection and payload-allocation by steganographic distortion optimization, *IEEE Signal Processing Letters* 25 (10) (2018) 1530–1534.
- [5] J. Zhang, W. Lu, X. Yin, W. Liu, Y. Yeung, Binary image steganography based on joint distortion measurement, *Journal of Visual Communication and Image Representation* 58 (2019) 600–605.
- [6] X. Zhang, S. Wang, G. Feng, Fragile watermarking scheme with extensive content restoration capability, in: *International Workshop on Digital Watermarking*, Springer, 2009, pp. 268–278.
- [7] C. Qin, P. Ji, X. Zhang, J. Dong, J. Wang, Fragile image watermarking with pixel-wise recovery based on overlapping embedding strategy, *Signal Processing* 138 (2017) 280–293.
- [8] X.-C. Yuan, M. Li, Local multi-watermarking method based on robust and adaptive feature extraction, *Signal Processing* 149 (2018) 103–117.
- [9] Y. Peng, X. Niu, L. Fu, Z. Yin, Image authentication scheme based on reversible fragile watermarking with two images, *Journal of information security and applications* 40 (2018) 236–246.
- [10] H. Otori, S. Kuriyama, Texture synthesis for mobile data communications, *IEEE Computer graphics and applications* (6) (2009) 74–81.
- [11] Z. Qian, H. Zhou, W. Zhang, X. Zhang, Robust steganography using texture synthesis, in: *Advances in Intelligent Information Hiding and Multimedia Signal Processing*, Springer, 2017, pp. 25–33.
- [12] S. Li, X. Zhang, Toward construction-based data hiding: From secrets to fingerprint images, *IEEE Transactions on Image Processing* 28 (3) (2019) 1482–1497.

- [13] J. M. Barton, Method and apparatus for embedding authentication information within digital data, uS Patent 5,646,997 (Jul. 8 1997).
- [14] W. Zhang, K. Ma, N. Yu, Reversibility improved data hiding in encrypted images, *Signal Processing* 94 (2014) 118–127.
- [15] S. Yi, Y. Zhou, Binary-block embedding for reversible data hiding in encrypted images, *Signal Processing* 133 (2017) 40–51.
- [16] Z. Yin, A. Abel, J. Tang, X. Zhang, B. Luo, Reversible data hiding in encrypted images based on multi-level encryption and block histogram modification, *Multimedia Tools and Applications* 76 (3) (2017) 3899–3920.
- [17] S. Yi, Y. Zhou, Parametric reversible data hiding in encrypted images using adaptive bit-level data embedding and checkerboard based prediction, *Signal Processing* 150 (2018) 171–182.
- [18] Z. Yin, X. Niu, X. Zhang, J. Tang, B. Luo, Reversible data hiding in encrypted ambtc images, *Multimedia Tools and Applications* (2018) 1–17.
- [19] Y.-Q. Shi, X. Li, X. Zhang, H.-T. Wu, B. Ma, Reversible data hiding: advances in the past two decades, *IEEE Access* 4 (2016) 3210–3237.
- [20] J. Fridrich, M. Goljan, R. Du, Invertible authentication, in: *Security and Watermarking of Multimedia contents III*, Vol. 4314, International Society for Optics and Photonics, 2001, pp. 197–209.
- [21] M. U. Celik, G. Sharma, A. M. Tekalp, E. Saber, Lossless generalized-lsb data embedding, *IEEE transactions on image processing* 14 (2) (2005) 253–266.
- [22] M. U. Celik, G. Sharma, A. M. Tekalp, Lossless watermarking for image authentication: a new framework and an implementation, *IEEE Transactions on Image Processing* 15 (4) (2006) 1042–1049.
- [23] W. Zhang, X. Hu, X. Li, N. Yu, Recursive histogram modification: establishing equivalency between reversible data hiding and lossless data compression, *IEEE transactions on image processing* 22 (7) (2013) 2775–2785.
- [24] C. Qin, Y.-C. Hu, Reversible data hiding in vq index table with lossless coding and adaptive switching mechanism, *Signal Processing* 129 (2016) 48–55.
- [25] J. Tian, Reversible data embedding using a difference expansion, *IEEE transactions on circuits and systems for video technology* 13 (8) (2003) 890–896.
- [26] Y. Hu, H.-K. Lee, J. Li, De-based reversible data hiding with improved overflow location map, *IEEE Transactions on Circuits and Systems for Video Technology* 19 (2) (2009) 250–260.
- [27] X. Li, W. Zhang, X. Gui, B. Yang, A novel reversible data hiding scheme based on two-dimensional difference-histogram modification, *IEEE Transactions on Information Forensics and Security* 8 (7) (2013) 1091–1100.
- [28] Z. Ni, Y.-Q. Shi, N. Ansari, W. Su, Reversible data hiding, *IEEE Transactions on circuits and systems for video technology* 16 (3) (2006) 354–362.
- [29] P. Tsai, Y.-C. Hu, H.-L. Yeh, Reversible image hiding scheme using predictive coding and histogram shifting, *Signal processing* 89 (6) (2009) 1129–1143.
- [30] G. Feng, L. Fan, Reversible data hiding of high payload using local edge sensing prediction, *Journal of Systems and Software* 85 (2) (2012) 392–399.
- [31] X. Chen, X. Sun, H. Sun, Z. Zhou, J. Zhang, Reversible watermarking method based on asymmetric-histogram shifting of prediction errors, *Journal of Systems and Software* 86 (10) (2013) 2620–2626.
- [32] X. Li, B. Li, B. Yang, T. Zeng, General framework to histogram-shifting-based reversible data hiding, *IEEE Transactions on Image Processing* 22 (6) (2013) 2181–2191.
- [33] T.-C. Lu, C.-Y. Tseng, J.-H. Wu, Asymmetric-histogram based reversible information hiding scheme using edge sensitivity detection, *Journal of Systems and Software* 116 (2016) 2–21.
- [34] D. M. Thodi, J. J. Rodríguez, Expansion embedding techniques for reversible watermarking, *IEEE transactions on image processing* 16 (3) (2007) 721–730.
- [35] X. Li, B. Yang, T. Zeng, Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection, *IEEE Transactions on Image Processing* 20 (12) (2011) 3524–3533.
- [36] X. Li, J. Li, B. Li, B. Yang, High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion, *Signal processing* 93 (1) (2013) 198–205.
- [37] B. Ou, X. Li, Y. Zhao, R. Ni, Y.-Q. Shi, Pairwise prediction-error expansion for efficient reversible data hiding, *IEEE Transactions on Image Processing* 22 (12) (2013) 5010–5021.
- [38] X. Qu, H. J. Kim, Pixel-based pixel value ordering predictor for high-fidelity reversible data hiding, *Signal Processing* 111 (2015) 249–260.
- [39] K.-H. Jung, A high-capacity reversible data hiding scheme based on sorting and prediction in digital images, *Multimedia Tools and Applications* 76 (11) (2017) 13127–13137.
- [40] R. Lukac, K. Martin, K. Platanoitis, Digital camera zooming based on unified cfa image processing steps, *IEEE Transactions on Consumer Electronics* 50 (1) (2004) 15–24.



**Yujie Jia** received her bachelor degree in electronics and information engineering in 2016 and now is a master student in the School of Computer Science and Technology, Anhui University. Her current research interests include reversible data hiding.



**Zhaoxia Yin** received her B.Sc., M.E. & Ph.D. from Anhui University in 2005, 2010 and 2014 respectively. She is an IEEE/ACM/CCF member, CSIG senior member and the Associate Chair of the academic committee of CCF YOCSEF Hefei 2016–2017. She is currently working as an Associate Professor and a Doctoral Tutor in School of Computer Science and Technology at Anhui University. She is also the Principal Investigator of two NSFC Projects. Her primary research focus including Information Hiding, Multimedia Security and she has published many SCI/EI indexed papers in journals, edited books and refereed conferences.



**Xinpeng Zhang** received the B.S. degree in computational mathematics from Jilin University, China, in 1995, and the M.E. and Ph.D. degrees in communication and information system from Shanghai University, China, in 2001 and 2004, respectively. Since 2004, he had been with the faculty of the School of Communication and Information Engineering, Shanghai University, and he is currently a Professor of School of Computer Science, Fudan University. He was with the State University of New York at Binghamton as a visiting scholar from January 2010 to January 2011, and Konstanz University as an experienced researcher sponsored by the Alexander von Humboldt Foundation from March 2011 to May 2012. He served *IEEE Transactions on Information Forensics and Security* as an Associate Editor from 2014 to 2017. His research interests include multimedia security, image processing, and digital forensics. He has published more than 200 papers in these areas.



**Yonglong Luo** received his Ph.D. degree from the School of Computer Science and Technology, University of Science and Technology of China in 2005. Since 2007, he has been a professor and a Ph.D supervisor at Anhui Normal University (AHNU). Now he is the Director of Anhui Provincial Key Laboratory of Network and Information Security and the Dean of the School of Computer and Information, AHNU. His main research interests include information security and spatial data processing.