



*Research article*

## **An IPVO-based reversible data hiding scheme using floating predictors**

**Rong Li<sup>1</sup>, Xiangyang Li<sup>2,\*</sup>, Yan Xiong<sup>2</sup>, An Jiang<sup>1</sup> and David Lee<sup>3</sup>**

<sup>1</sup> School of Computer Science, Guangdong Agriculture Industry Business Polytechnic, Guangzhou, 510507, China

<sup>2</sup> Institute of Computer Application, Guangdong Agriculture Industry Business Polytechnic, Guangzhou, 510507, China

<sup>3</sup> School of Information Technology and Engineering, University of Ottawa, Ottawa, Canada

\* **Correspondence:** Email: [xyli@gdaib.edu.cn](mailto:xyli@gdaib.edu.cn); Tel: + (86)1-371-087-5483.

**Abstract:** This work optimizes an improved high-fidelity reversible data hiding scheme of Peng et al. which is based on improved pixel-value-ordering (IPVO) and prediction-error expansion. In Peng et al.'s method, the difference between the maximum and second largest value (or, the minimum and second smallest value) of a block is defined considering the pixel locations of maximum and second the largest value (or, the pixel locations of minimum and second the smallest value). When the difference between the maximum and second largest value (or, the minimum and second smallest value) of a block is equal to 0 or 1, the block can be exploited to embed data. Otherwise, the block should be shifted or remain unchanged. However, different prediction-error used to embed information can lead to different histogram modification and different pixel shift rate, to further reduces the change in the carrier image. In this work, we list all the different prediction-error, which are used as the selection object for the embedded error when hiding information. As a prerequisite of meeting the demand of the embedding capacity, some appropriate prediction-errors are selected for embedding to reduce the number of the pixel shifted in the marked image as small as possible. An IPVO-based reversible data hiding scheme with floating predictor is also extended. Experimental results show that the proposed scheme yields a superior performance than the state-of-the-art works, under the condition of same embedding capacity, especially for relatively rough images.

**Keywords:** Reversible data hiding; optimization; floating prediction-error; improved pixel-value-ordering; PSNR

---

## 1. Introduction

Reversible data hiding (RDH) is referred to as a process to hide secret data into a cover image without any permanent distortion, and the original image can be perfectly recovered after the secret data are extracted from the marked image. In recent year, many RDH schemes have been exploited. The earlier versions of them concentrate on increasing payload capacity of the embedded data, and the latter ones gradually focus on the visual quality of the marked image, or the trade-off of payload capacity and visual quality.

Reversible data hiding algorithm has been extensively studied and developed by now. The early RDH algorithms are mainly based on lossless compression [1,2], and provide a lower embedding capacity ( $EC$ ) and bring severe degrade in marked image. Later, difference expansion (DE) [3,4] has received more extensive attention, which perform a spatial domain transform on pixel pairs, and embed the secret data in a reversible manner by expanding pixel difference, providing a variable embedding capacity by adjusting a threshold value. Histogram shifting (HS) [5,6] is another method that focuses on the trade-off of large data embedding capacity and high quality of the marked image, in which the peak points of image histogram are modified to embed secret data. However, when the image histogram distribution is flat, the embedding efficiency of HS is greatly decreased, so other HS techniques [7,8] are further exploited to enlarge  $EC$ . Prediction error expansion (PEE) [9–13] is a promising reversible data hiding method through introducing the prediction mechanism. PEE fully utilizes the correlation of the neighboring pixels in a nature image, and obtains significant increase in  $EC$ . Afterwards, PEE is developed in various ways such as constructing a payload dependent location map [14], adaptive embedding based on pixel selection strategy [15] and context embedding [16,17].

The high-level image fidelity RDH algorithms are usually implemented by modifying a certain histogram. Among these algorithms, in 2013, Li et al. [18] proposed a famous RDH scheme based on pixel-value-ordering (PVO). Li et al.'s work acquired immense success for moderate payload size. PVO method divides the cover image into some non-overlapped pixel blocks. For each pixel block, the pixels are sorted in an ascending order, then the maximum/minimum pixel value is predicted by the second largest/smallest pixel value to obtain the prediction error. If the prediction error is equal to 1, the pixel is used to carry a bit of secret data; and if the prediction error is greater than 1, then the pixel is shifted to create vacancy; otherwise, the pixel is discarded for data embedding. Obviously, for each pixel block, only two pixels at most may be changed by 1 to carry secret bits or shift, the visual quality is well guaranteed. Take the *Lena* image as an example, when the embedding amount is 10000 bits, the  $PSNR$  value is 59.2049 dB, which is about 5% higher than the previous PEE algorithm.

Later, many other methods [19–22] based on PVO have been proposed. Ou et al. proposed a new prediction strategy named PVO-k [19] and a pixel-based PVO (PPVO) method [20], these methods take full advantage of the pixels in the smooth area that are completely ignored by the PVO method, to get bigger the embedding capacity than PVO method. Peng et al. [21] extended an improved reversible data hiding scheme based on pixel value ordering (IPVO), in which a new prediction error has been computed and new histogram modification strategy has been utilized, the blocks which the largest pixel value equals to the second largest pixel value can be exploited to embed data, and these blocks are not utilized in Li et al.'s [18] method, resulting that this method exploits more image redundancy space and outperforms better performance than Li et al.'s work. When the embedding amount is 10000 bits, the  $PSNR$  value of the *Lena* image is 59.5834 dB, which is higher than PVO method. Subitha P et al. [22] proposed novel reversible pixel-value-ordering

technique, which contains two improvement strategies: Novel Difference Computation (DC) and Novel Histogram Shifting (HS), this technique significantly improves the *EC* along with *PSNR*. Jung et al. [23] proposed an RDH method based on sorting and prediction for three pixels in pixel blocks, which provides high embedding capacity and good image quality. He et al. [24] proposed a new reversible data hiding scheme based on multi-pass PVO and pairwise PEE, which can achieve efficient capacity-distortion trade-off. Yang et al. [25] proposed a lossless and high payload data hiding scheme for JPEG images by sorting the histogram of VLCs in descending order and modifying the histogram, which can lead less file size expansion in identical payload and higher embedding efficiency. Coverless image information steganographic scheme based on generative model [26] and the privacy-preserving outsourcing scheme of reversible data hiding over encrypted image data in cloud computing [27] were studied to enhance the security and reliability of the practical application process.

To our best knowledge, PVO-based predictors [18–22] have been confirmed to be more suitable for high-level fidelity reversible data hiding than median-edge-detector [28], gradient-adjusted-prediction [29] and mean-value-predictor [30]. However, there is still room for improvement. For example, in Peng et al.'s [21] method, two histogram bins namely 0 and 1 are fixedly utilized to embed secret data and other bins are shifted. From the perspective of reducing the *PSNR*, however, it is not always the best choice that using bins 0 and 1 fixedly to embed secret data for all situations. It may bring higher shift rate and lower *PSNR*. In fact, the performance of IPVO is not good for a rough picture. Actually, for a given image, we can use different combinations of difference histogram bins to embed secret data for a certain embedding capacity. Among these combinations, we can finally find out a best alternative to acquire the lowest shift rates. Let's take the *baboon* image as an example, when the embedding payload is 5000 bits, the image shift rate will be 0.9070 if two bins 1 and 0 are used according to Peng et al.'s art. But there are other alternatives: the two histogram bins namely bin 2 and  $-1$  can be utilized to embed secret data and the shift rate is 0.9045; similarly, the two bins 5 and  $-3$  can be also used and the shift rate is 0.8993. This demonstrates that there are some other pairs of difference bins than the pair of bins 0 and 1 can be exploited to embed the secret data, and can acquire smaller shift rates.

Based on this consideration, a new floating predictor is defined to search for the proper difference histogram bins pair adaptively to get higher *PSNR* of the marked image. And an improved data hiding scheme based on IPVO with floating predictor is proposed in this paper. To determine which pair of bins is the most suitable for embedding a certain amount of secret data, we calculate all the possible combinations of the bins and select the most suitable pair from these possibilities.

The rest of this paper is organized as follows. In Section 2, some related works are briefly reviewed. Our starting point for proposed floating predictor is also introduced in the section. In section 3, an improved IPVO-based scheme with floating predictor is proposed by developing an embedded difference optimization mechanism. The data embedding procedure and extraction procedure of our algorithm are also introduced in detail at the end of this section. The experimental results comparing our work with the state-of-the-art works are shown in Section 4. Finally, Section 5 concludes this paper.

## 2. Related works

### 2.1. IPVO-based RDH scheme

Li et al. [18] proposed a data hiding scheme based on pixel-value-ordering and predication-error expansion. Peng et al. [21] extended an improved reversible data hiding scheme (IPVO), which expands the prediction error 0 to be used for embedding and has larger embedding capacity than PVO scheme. The main process of IPVO can be summarized as follows: Dividing an image into non-overlapping blocks with a certain number of pixels, such as  $n$  pixels  $x_1, x_2, \dots, x_n$ . Pixels in a block are sorted into an ascending order and are denoted as  $x_{\sigma(1)}, x_{\sigma(2)}, \dots, x_{\sigma(n-1)}, x_{\sigma(n)}$ , and  $x_{\sigma(1)} \leq x_{\sigma(2)} \leq \dots \leq x_{\sigma(n-1)} \leq x_{\sigma(n)}$ ,  $\sigma(1), \sigma(2), \dots, \sigma(n-1), \sigma(n)$  is the position order in the original block are considered when calculating prediction errors.

Calculate the prediction error of the largest pixel  $e_{max}$  using the following formula (2.1).

$$e_{max} = x_u - x_v \quad (2.1)$$

where  $u = \min(\sigma(n-1), \sigma(n))$  and  $v = \max(\sigma(n-1), \sigma(n))$ .  $u$  and  $v$  give the position identifiers of maximum pixel and second maximum pixel, respectively. IPVO scheme uses prediction-errors 0 and 1 to embed the secret data as shown in formula (2.2).

$$e'_{max} = \begin{cases} e_{max} + s & \text{if } e_{max} = 1 \\ e_{max} + 1 & \text{if } e_{max} > 1 \\ e_{max} - s & \text{if } e_{max} = 0 \\ e_{max} - 1 & \text{if } e_{max} < 0 \end{cases} \quad (2.2)$$

The maximum pixel value is changed as formula (2.3).

$$x'_{\sigma(n)} = \begin{cases} x_{\sigma(n)} + s & \text{if } e_{max} = 1 \\ x_{\sigma(n)} + 1 & \text{if } e_{max} > 1 \\ x_{\sigma(n)} - s & \text{if } e_{max} = 0 \\ x_{\sigma(n)} - 1 & \text{if } e_{max} < 0 \end{cases} \quad (2.3)$$

As the same reason, calculate the prediction error of the smallest pixel  $e_{min}$  using the following formula (2.4).

$$e_{min} = x_m - x_n \quad (2.4)$$

where  $m = \min(\sigma(1), \sigma(2))$  and  $n = \max(\sigma(1), \sigma(2))$ .  $m$  and  $n$  give the position identifiers of minimum pixel and second minimum pixel, respectively. Using prediction-errors 0 and 1 to embed secret data as shown in formula (2.5).

$$e'_{min} = \begin{cases} e_{min} + s & \text{if } e_{min} = 1 \\ e_{min} + 1 & \text{if } e_{min} > 1 \\ e_{min} - s & \text{if } e_{min} = 0 \\ e_{min} - 1 & \text{if } e_{min} < 0 \end{cases} \quad (2.5)$$

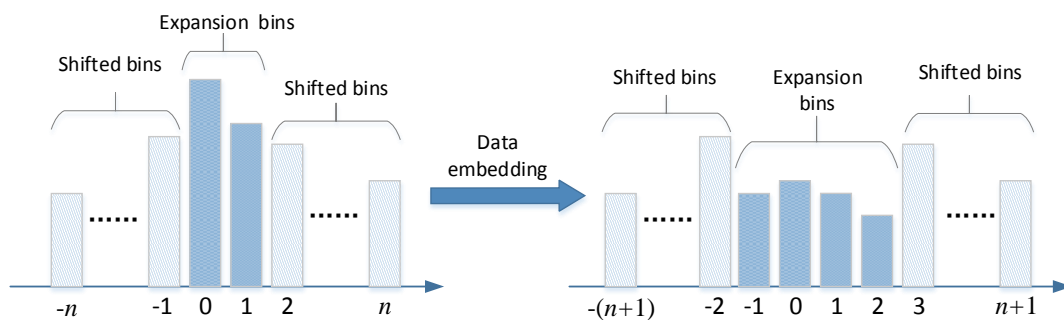
The minimum pixel value is changed as formula (2.6).

$$x'_{\sigma(1)} = \begin{cases} x_{\sigma(1)} - s & \text{if } e_{min} = 1 \\ x_{\sigma(1)} - 1 & \text{if } e_{min} > 1 \\ x_{\sigma(1)} + s & \text{if } e_{min} = 0 \\ x_{\sigma(1)} + 1 & \text{if } e_{min} < 0 \end{cases} \quad (2.6)$$

Obviously, the locations of the largest pixel  $x_{\sigma(n)}$  and the smallest pixel  $x_{\sigma(1)}$  have no changes in the block after the information is embedded, we can extract secret information as shown in [21].

## 2.2. Performance analysis

IPVO is based on the concept that pixel value in a block is similar for a natural image. The information is embedded when the difference is 1 and 0, the image histogram changes as shown in Figure 1.



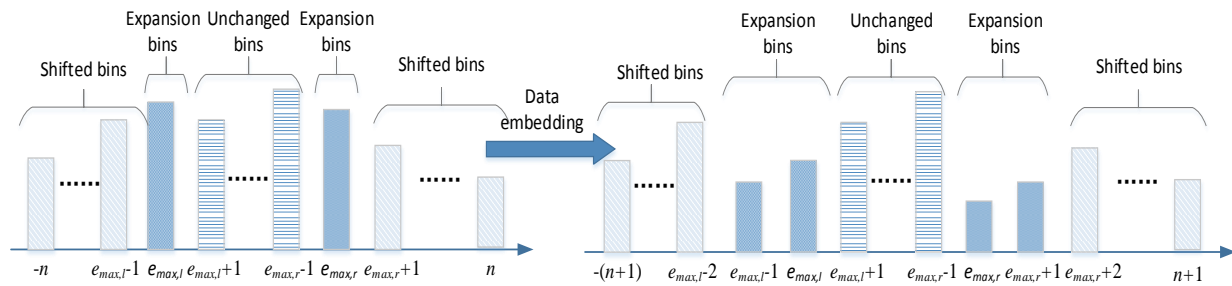
**Figure 1.** Histogram modification in IPVO.

From Figure 1 we can see that the maximum pixel is shifted right by 1 in value when its prediction error is greater than 1 and is shifted left by 1 when its prediction error is less than 0, the minimum prediction error selected for embedding information is the same as the maximum prediction error. In fact, there is a difference between the distributions of the maximum prediction error and the minimum prediction error, and the distributions of the prediction-errors for different images are also different.

Suppose that the difference that satisfies the pixel right shift is named the right-shift prediction error and the difference that satisfies the pixel left shift is named the left-shift prediction error. Then four kinds of different prediction-errors can be extended on the basis of the original IPVO algorithm, which are marked as the maximum right-shift prediction error  $e_{max,r}$ , the maximum left-shift prediction error  $e_{max,l}$ , the minimum right-shift prediction error  $e_{min,r}$  and the minimum left-shift prediction error  $e_{min,l}$ . If we use floating prediction errors,  $e_{max,r}$  or  $e_{max,l}$  may take a value other than 1 and  $e_{min,r}$  or  $e_{min,l}$  may take a value other than 0, image histogram modification will have new change, see Figure 2. In meeting the needs of embedding capacity, for different images, selecting the more appropriate prediction-error (not fixed to 0 and 1) for RDH can change the prediction-error histogram modification progress and further reduce the changes in the marked image.

Usually, assume that  $N_s$  is the number of shifted pixels and  $N_e$  is the number of embedded pixels. The shift rate  $SR$  is defined as formula (2.7), which is a metric to reflect the performance of data hiding algorithm. The smaller is  $SR$ , the better is the performance of the algorithm.

$$SR = \frac{N_s}{N_s + N_e} \quad (2.7)$$



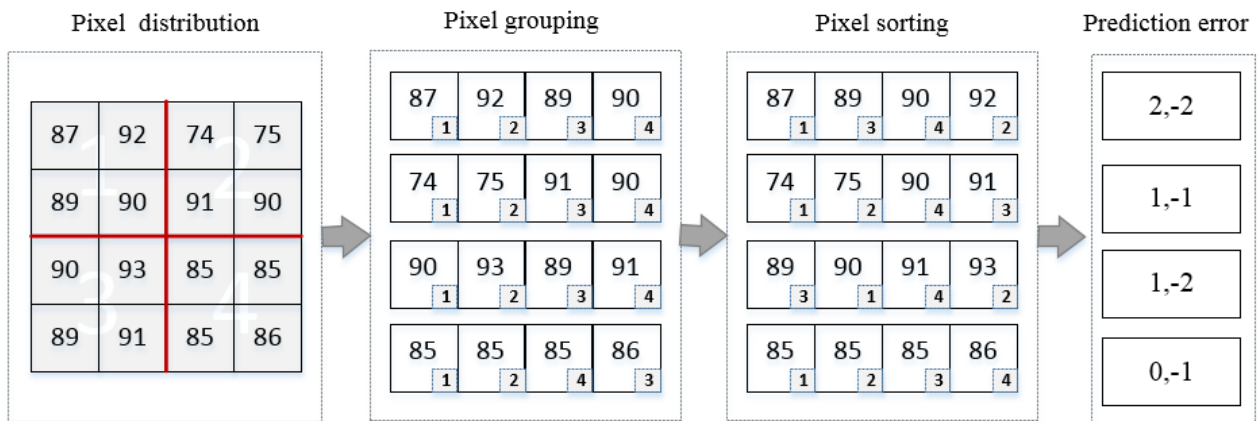
**Figure 2.** Histogram modification using floating prediction errors.

Utilization of floating prediction-errors  $e_{max,r}$ ,  $e_{max,l}$ ,  $e_{min,r}$  and  $e_{min,l}$  for embedding can cause the shift rate  $SR$  to be significantly changed, which gives the IPVO method a possible chance to improve its performance by selecting different combinations of prediction-errors for embedding. Take the *Baboon* image for example, when embedding capacity is 5000 bits, the shift rates are shown in Table 1 as the prediction-errors combination changes.

**Table 1.** Shift Rate ( $SR$ ) on floating predictor (*Baboon*).

$e_{max,r}$	$e_{max,l}$	$e_{max,l}$	$e_{min,l}$	$SR$
1	-1	1	-1	0.9064
2	-1	2	-1	0.9045
3	0	3	0	0.9065
3	-1	3	-1	0.9023
4	-2	4	-2	0.9019
4	-3	4	-3	0.9002
5	-1	5	-1	0.9012
5	-3	5	-3	0.8993
5	-5	5	-5	0.8982
6	-6	6	-6	0.8997
7	-5	7	-5	0.8992
7	-6	7	-6	0.8987
8	-4	8	-4	0.8983
8	-7	8	-7	0.8995
9	-4	9	-4	0.8983
10	-4	10	-4	0.8961

As can be easily seen from Table 1, the different combinations of prediction-errors show different  $SR$  values, especially more obvious for rough images. We can choose a combination of prediction-errors with a smaller shift rate to complete the information hiding process. On the other hand, different image has different pixel distribution, as can be seen from Figure 3, for a given pixel distribution and embedding capacity requirement, the selection of floating prediction-errors will bring different shift rate. If the requirement of the embedding capacity is changed, the combination of prediction errors will be reselected according to the principle that the distortion of the marked image is as small as possible.



- If secret message is 2 bits, when embedding prediction errors are 1 and 0, the shift rate is  $3/5$ , when embedding prediction errors are 1 and -1, the shift rate is  $1/2$ , when embedding prediction errors are 2 and -2, the shift rate is 0;
- If secret message is 3 bits, when embedding prediction errors are 1 and 0, the shift rate is  $4/7$ , when embedding prediction errors are 1 and -1, the shift rate is  $2/5$ , when embedding prediction errors are 2 and -2, the shift rate is 0;

**Figure 3.** Effect of pixel distribution on the selection of floating prediction-error.

### 3. Proposed method

In this section, we improve the scheme based on the IPVO and develop an embedded difference selection mechanism using float predictor, and introduce the data embedding procedure and extraction procedure.

#### 3.1. Floating prediction-error selection mechanism

In fact, the selected embedded difference combination can not only reduce the embedding distortion, but also definitely improve embedding performance. Suppose we choose four prediction errors to perform the embedding operations, which are the maximum right-shift prediction error  $e_{max,r}$ , the maximum left-shift prediction error  $e_{max,l}$ , the minimum right-shift prediction error  $e_{min,r}$  and the minimum left-shift prediction error  $e_{min,l}$  respectively. The goal of selection is to find the most suitable combination of  $e_{max,r}^*, e_{max,l}^*, e_{min,r}^*, e_{min,l}^*$  to get the minimal shift rate  $SR$  of the overall image. Under meeting the premise of a certain embedding capacity, we build selection model as formula (3.1).

$$\min SR(e_{max,r}^*, e_{max,l}^*, e_{min,r}^*, e_{min,l}^*) = \frac{\sum_{i=max,min;j=r,l} N_s(e_{i,j})}{\sum_{i=max,min;j=r,l} N_s(e_{i,j}) + \sum_{i=max,min;j=r,l} N_e(e_{i,j})}$$

$$s. t. \begin{cases} \sum_{i=max,min;j=r,l} N_e(e_{i,j}^*) \geq CR \\ e_{max,r}^*, e_{min,r}^* > 0 \\ e_{max,l}^*, e_{min,l}^* \leq 0 \\ |e_{max,l}^*| + |e_{max,r}^*| \leq TS \\ |e_{min,l}^*| + |e_{min,r}^*| \leq TS \end{cases} \quad (3.1)$$

Amongst them,  $CR$  is the capacity requirement of the embedded secret message. The number of embedding differences must be no less than  $CR$ . Hence, the secret message can be restored, right shift prediction error should be greater than left shift prediction error, so  $e_{max,l}^* < e_{max,r}^*$  and  $e_{min,l}^* < e_{min,r}^*$ .  $TS$  is a threshold parameter, which is used to control the complexity of the algorithm and the distortion degree of carrier image. From the perspective of algorithm performance, the time complexity of model (3.1) is  $O(n^2)$  and the range of  $e_{i,j}$  is  $[-255, 255]$ , ideal results are obtained in finite time, so we can get results by enumeration method.

### 3.2. Embedding procedure

We embed the secret messages into a cover image and obtain a marked image. The embedding process is presented as follows.

(1) For a cover image  $I$  with  $M \times N$  pixels, the cover image pixels are divided into non-overlapped blocks using the original PVO algorithm [18] (e.g. block size is  $2 \times 2$ ). Suppose the number of blocks is  $u$ , then  $u = (M \times N) / (2 \times 2)$ .

(2) Each block contains four neighboring pixels that are denoted by  $x_i, x_{i+1}, x_{i+2}, x_{i+3}$ , respectively, in block  $i$  ( $i = 1, 2, \dots, u$ ), here  $0 \leq x_i, x_{i+1}, x_{i+2}, x_{i+3} \leq 255$ .

(3) For the block  $i$  ( $i = 1, 2, \dots, u$ ), the pixels are sorted into an ascending order to get  $x_{\sigma(i)}, x_{\sigma(i+1)}, x_{\sigma(i+2)}, x_{\sigma(i+3)}$ , where  $x_{\sigma(i+3)}$  is the largest pixel value and  $x_{\sigma(i)}$  the smallest pixel value.  $\sigma: \{i, i+1, i+2, i+3\} \rightarrow \{i, i+1, i+2, i+3\}$  is the unique one-to-one mapping such that:  $x_{\sigma(i)} \leq x_{\sigma(i+1)} \leq x_{\sigma(i+2)} \leq x_{\sigma(i+3)}$ ,  $\sigma(m) < \sigma(n)$  if  $x_{\sigma(m)} = x_{\sigma(n)}$  and  $m < n$ ,  $m, n = i, i+1, i+2, i+3$ .

(4) For the block  $i$  ( $i = 1, 2, \dots, u$ ), calculate the difference between the largest pixel value  $x_{\sigma(i+3)}$  and the second largest pixel value  $x_{\sigma(i+2)}$  and obtain the maximum prediction-error. According to the previous analysis, there are two cases when information is embedded into the cover image: maximum prediction-errors are right shifted in and maximum prediction-errors are left shifted in. The corresponding maximum prediction errors are

$$\begin{cases} e_{max,r}^i = x_{\sigma(i+3)} - x_{\sigma(i+2)}, & \text{if } \sigma(i+3) < \sigma(i+2) \\ e_{max,l}^i = x_{\sigma(i+2)} - x_{\sigma(i+3)}, & \text{if } \sigma(i+3) \geq \sigma(i+2) \end{cases} \quad (3.2)$$

(5) In the similar way, for the block  $i$  ( $i = 1, 2, \dots, u$ ), calculate the difference between the smallest pixel value  $x_{\sigma(i)}$  and second largest pixel value  $x_{\sigma(i+1)}$  and obtain the minimum prediction-error. According to the previous analysis, there are two cases when information is embedded to cover image, minimum prediction-errors are right shifted in and minimum prediction-errors are left shifted in, so the corresponding minimum prediction errors are

$$\begin{cases} e_{min,r}^i = x_{\sigma(i+1)} - x_{\sigma(i)}, & \text{if } \sigma(i+1) < \sigma(i) \\ e_{min,l}^i = x_{\sigma(i)} - x_{\sigma(i+1)}, & \text{if } \sigma(i+1) \geq \sigma(i) \end{cases} \quad (3.3)$$

(6) Suppose bit string  $M = m_0 m_1 \dots m_{l-1}$  is the secret message with  $l$  bits, which will be embedded into the cover image, where  $m_j \in \{0, 1\}$  and  $0 \leq j \leq l-1$ .

(7) To meet the requirement of embedding capacity, search for the best combination of various differences by using the floating prediction-error selection mechanism which is created by



formula (3.1), we can obtain the best prediction-errors that are noted as  $e_{max,r}^*$ ,  $e_{max,l}^*$ ,  $e_{min,r}^*$ ,  $e_{min,l}^*$  respectively.

(8) Next, by using  $e_{max,r}^*$ ,  $e_{max,l}^*$ , the embedding rules for (2.3) can be improved, instead of taking bins 1 and 0 in the prediction-error histograms for expansion embedding, we utilize here the bins  $e_{max,r}^*$  and  $e_{max,l}^*$  to complete the information embedded.  $e_{max}^{i'}$  is modified as show in formula (3.4).

$$e_{max}^{i'} = \begin{cases} e_{max,r}^i + m_j & \text{if } e_{max,r}^i = e_{max,r}^* \\ e_{max,r}^i + 1 & \text{if } e_{max,r}^i > e_{max,r}^* \\ e_{max,l}^i - m_j & \text{if } e_{max,l}^i = e_{max,l}^* \\ e_{max,l}^i - 1 & \text{if } e_{max,l}^i < e_{max,l}^* \end{cases} \quad (i = 1, 2, \dots, u; 0 \leq j \leq l - 1) \quad (3.4)$$

Then, the marked pixel value of maximum  $x_{\sigma(i+3)}$  is changed as formula (3.5)

$$x'_{\sigma(i+3)} = \begin{cases} x_{\sigma(i+3)} + m_j & \text{if } e_{max,r}^i = e_{max,r}^* \\ x_{\sigma(i+3)} + 1 & \text{if } e_{max,r}^i > e_{max,r}^* \\ x_{\sigma(i+3)} + m_j & \text{if } e_{max,l}^i = e_{max,l}^* \\ x_{\sigma(i+3)} + 1 & \text{if } e_{max,l}^i < e_{max,l}^* \end{cases} \quad (i = 1, 2, \dots, u; 0 \leq j \leq l - 1) \quad (3.5)$$

(9) The same reason, by using  $e_{min,r}^*$ ,  $e_{min,l}^*$ , the embedding rules for (2.6) can be improved, instead of taking bins 1 and 0 in the prediction-error histograms for expansion embedding, we utilize here the bins  $e_{min,r}^*$  and  $e_{min,l}^*$  to complete the information embedded.  $e_{min}^{i'}$  is modified as show in formula (3.6).

$$e_{min}^{i'} = \begin{cases} e_{min,r}^i + m_j & \text{if } e_{min,r}^i = e_{min,r}^* \\ e_{min,r}^i + 1 & \text{if } e_{min,r}^i > e_{min,r}^* \\ e_{min,l}^i - m_j & \text{if } e_{min,l}^i = e_{min,l}^* \\ e_{min,l}^i - 1 & \text{if } e_{min,l}^i < e_{min,l}^* \end{cases} \quad (i = 1, 2, \dots, u; 0 \leq j \leq l - 1) \quad (3.6)$$

Then, the marked pixel value of minimum  $x_{\sigma(i)}$  is changed as formula (3.7)

$$x'_{\sigma(i)} = \begin{cases} x_{\sigma(i)} - m_j & \text{if } e_{min,r}^i = e_{min,r}^* \\ x_{\sigma(i)} - 1 & \text{if } e_{min,r}^i > e_{min,r}^* \\ x_{\sigma(i)} - m_j & \text{if } e_{min,l}^i = e_{min,l}^* \\ x_{\sigma(i)} - 1 & \text{if } e_{min,l}^i < e_{min,l}^* \end{cases} \quad (i = 1, 2, \dots, u; 0 \leq j \leq l - 1) \quad (3.7)$$

Here  $m_j \in \{0,1\}$  is a secret bit, the prediction error  $e_{max,r}^*$ ,  $e_{max,l}^*$ ,  $e_{min,r}^*$ ,  $e_{min,l}^*$  are used to embedded information, so the capacity is the sum of  $e_{max,r}^*$ ,  $e_{max,l}^*$ ,  $e_{min,r}^*$  and  $e_{min,l}^*$ .

In this improved method, the mapping  $\sigma$  also keeps unchanged; therefore, the decoder can achieve data extraction and image lossless recovery.

(9) The complexity of block  $i$  is measured by threshold  $TS$ , a block is taken as a flat one and is chosen to embed secret data when its complexity is less than  $TS$ . The reason is that, in a natural image, most image blocks are smooth, so we should obtain more space for embedding message than without prediction. Using formula (3.5) and (3.7) to embed all the information, we can obtain the marked image  $I'$ .

(10) For a block  $i$  ( $i = 1, 2, \dots, u$ ), the changes of pixel value may cause overflow or underflow, when  $x_{\sigma(i+3)} = 255$  and  $e_{max,r}^i \geq e_{max,r}^*$  or  $e_{max,l}^i \leq e_{max,l}^*$ , the pixel change will overflow, in this case, we set  $h1_i = 1$ , else set  $h1_i = 0$ , after marked all blocks as bitmap  $H_1 = h1_1 h1_2 \dots h1_u$ ; another when  $x_{\sigma(i)} = 0$  and  $e_{min,r}^i \geq e_{min,r}^*$  or  $e_{min,l}^i \leq e_{min,l}^*$ , the pixel change will underflow, in this case, we set  $h2_i = 1$ , else set  $h2_i = 0$ . After having marked all blocks as  $H_2 = h2_1 h2_2 \dots h2_u$ ,  $H_1$  and  $H_2$  are the required overhead information on extracting the embedded messages. The lengths of  $H_1$  and  $H_2$  are both  $u$ ,  $h1_j \in \{0, 1\}$  and  $h2_j \in \{0, 1\}, 1 \leq j \leq u$ . Because only a few 1 in the overhead information, we can use lossless compression to significantly reduce the size of  $H_1$  and  $H_2$  and denote compressed location map as  $H1_c$  and  $H2_c$ , and their length are  $l1_c$  and  $l2_c$ .

(11) Finally, embed supplementary information  $SI$  and  $H1_c$  and  $H2_c$  to the least significant bits ( $LSB$ ) of first  $l_s + l1_c + l2_c$  image pixels to get a binary sequence  $S_{LSB}$ . Symbol  $l_s$  is the length of supplementary information  $SI$  which include capacity  $CR$  (16 bits), threshold  $TS$  (8 bits),  $e_{max,r}^*$  (8 bits),  $e_{max,l}^*$  (8 bits),  $e_{min,r}^*$  (8 bits) and  $e_{min,l}^*$  (8 bits),  $l1_c$  ( $\lceil \log_2 M \times N \rceil$  bits),  $l2_c$  ( $\lceil \log_2 M \times N \rceil$  bits), where  $M \times N$  is the total number of image pixels and  $\lceil \cdot \rceil$  is the cell function. Next, embed the sequence  $S_{LSB}$  into the rest part (i.e. the blocks  $\{end+1, \dots, u\}$ ) of the cover image using the same method in formula (3.5) and (3.7).

### 3.3. Extraction procedure

The corresponding data extraction process is detailed as follows.

(1) Read  $LSB$  of the first  $56 + 2\lceil \log_2 M \times N \rceil$  pixels of the marked image to get the auxiliary information, which include  $CR$ ,  $TS$ ,  $e_{max,r}^*$ ,  $e_{max,l}^*$ ,  $e_{min,r}^*$ ,  $e_{min,l}^*$ ,  $l1_c$  and  $l2_c$ . Then read the  $LSB$  of next  $l1_c$  and  $l2_c$  pixels to get the compressed location map  $H1_c$  and  $H2_c$ , determine the location map  $H_1$  and  $H_2$  by decompressing  $H1_c$  and  $H2_c$ .

(2) Divide the marked image  $I'$  into some blocks that include four pixels as it was divided by the encoder during the embedding process.

(3) For a block  $i$  ( $i = 1, 2, \dots, u$ ), sort the pixel values  $y_i, y_{i+1}, y_{i+2}, y_{i+3}$  in an ascending order and marked as  $y_{\sigma(i)}, y_{\sigma(i+1)}, y_{\sigma(i+2)}, y_{\sigma(i+3)}$ .

(4) For a block  $i$  ( $i = 1, 2, \dots, u$ ), when  $h1_i = 0$  and  $h2_i = 0$ , calculate the maximum difference as following:

$$\begin{cases} e_{max,r}^i = y_{\sigma(i+3)} - y_{\sigma(i+2)}, & \text{if } \sigma(i+3) < \sigma(i+2) \\ e_{max,l}^i = y_{\sigma(i+2)} - y_{\sigma(i+3)}, & \text{if } \sigma(i+3) \geq \sigma(i+2) \end{cases} \quad (3.8)$$

At the same time, calculate the minimum difference as following:

$$\begin{cases} e_{min,r}^i = y_{\sigma(i+1)} - y_{\sigma(i)}, & \text{if } \sigma(i+1) < \sigma(i) \\ e_{min,l}^i = y_{\sigma(i)} - y_{\sigma(i+1)}, & \text{if } \sigma(i+1) \geq \sigma(i) \end{cases} \quad (3.9)$$

(5) For a block  $i$  ( $i = 1, 2, \dots, u$ ), the secret bit  $m_j$  ( $0 \leq j \leq l-1$ ) is released from the marked image as follows:

$$m_j = \begin{cases} 1 & \text{if } e_{max,r}^i = e_{max,r}^* + 1 \text{ or } e_{max,l}^i = e_{max,l}^* - 1 \\ 0 & \text{if } e_{max,r}^i = e_{max,r}^* \text{ or } e_{max,l}^i = e_{max,l}^* \end{cases} \quad (3.10)$$

$$m_j = \begin{cases} 1 & \text{if } e'_{min,r}{}^i = e_{min,r}^* + 1 \text{ or } e'_{min,l}{}^i = e_{min,l}^* - 1 \\ 0 & \text{if } e'_{min,r}{}^i = e_{min,r}^* \text{ or } e'_{min,l}{}^i = e_{min,l}^* \end{cases} \quad (3.11)$$

(6) For the largest pixel  $y_{\sigma(i+3)} (i = 1, 2, \dots, u)$ , we can restore pixel as formula (3.12) and marked as  $y'_{\sigma(i+3)}$ ; the smallest pixel  $y_{\sigma(i)} (i = 1, 2, \dots, u)$  can be restored as formula (3.13) and marked as  $y'_{\sigma(i)}$ .

$$y'_{\sigma(i+3)} = \begin{cases} y_{\sigma(i+3)} - 1 & \text{if } e'_{max,r}{}^i > e_{max,r}^* + 1 \\ y_{\sigma(i+3)} - m_j & \text{if } e'_{max,r}{}^i = e_{max,r}^* \text{ or } e'_{max,r}{}^i = e_{max,r}^* + 1 \\ y_{\sigma(i+3)} - m_j & \text{if } e'_{max,l}{}^i = e_{max,l}^* \text{ or } e'_{max,l}{}^i = e_{max,l}^* - 1 \\ y_{\sigma(i+3)} - 1 & \text{if } e'_{max,l}{}^i < e_{max,l}^* - 1 \end{cases} \quad (3.12)$$

$$y'_{\sigma(i)} = \begin{cases} y_{\sigma(i)} + 1 & \text{if } e'_{min,r}{}^i > e_{min,r}^* + 1 \\ y_{\sigma(i)} + m_j & \text{if } e'_{min,r}{}^i = e_{min,r}^* \text{ or } e'_{min,r}{}^i = e_{min,r}^* + 1 \\ y_{\sigma(i)} + m_j & \text{if } e'_{min,l}{}^i = e_{min,l}^* \text{ or } e'_{min,l}{}^i = e_{min,l}^* - 1 \\ y_{\sigma(i)} + 1 & \text{if } e'_{min,l}{}^i < e_{min,l}^* - 1 \end{cases} \quad (3.13)$$

(7) According to the block index, rearrange the message bits  $m_j (0 \leq j \leq l - 1)$  extracted from blocks. For the blocks  $\{end+1, \dots, u\}$ , we will extract the sequence  $S_{LSB}$  defined in data embedding progress and meanwhile realize the restoration, replace the least significant bits (LSB) of  $l_s + ll_c + l2_c$  image pixels by the sequence  $S_{LSB}$ . Therefore, we obtain the secret information  $M$  and recover the original cover image  $I$  lossless.

### 3.4. Example

An example of embedding progress and extracting progress is shown as Figure 4.

Given a cover image with  $4 \times 4$  pixels, which is divided into four non-overlapped blocks. For each block, sort its pixels in ascending order. The maximum prediction errors  $e_{max,r}^1 = 2$ ,  $e_{max,r}^2 = 1$ ,  $e_{max,r}^3 = 2$ ,  $e_{max,l}^4 = -1$  and the minimum prediction errors  $e_{min,l}^1 = -2$ ,  $e_{min,l}^2 = -1$ ,  $e_{min,r}^3 = 1$ ,  $e_{min,r}^4 = 0$  are calculated according to formula (3.2) and (3.3). Given secret message  $M = \{101\}$  with 3 bits, according to capacity requirement, prediction errors  $e_{max,r}^* = 2$ ,  $e_{max,l}^* = -1$ ,  $e_{min,r}^* = 1$ ,  $e_{min,l}^* = -2$  are selected by using scheme (3.1) to embedded information. Use the method in (3.5) and (3.7), in first block, the largest pixel ( $e_{max,r}^1 = e_{max,r}^* = 2$ ) and the smallest pixel ( $e_{min,l}^1 = e_{min,l}^* = -2$ ) are both expanded to carry hidden data; in third block, the largest pixel ( $e_{max,r}^3 = e_{max,r}^* = 2$ ) are expanded to carry hidden data, all other largest and smallest pixels are unchanged or shifted, so far, we have got the marked image.

Extraction operation is a reverse process. Divide the marked image into non-overlapping blocks and sort the pixel values according to the same rules as the embedding process, calculate the max prediction errors  $e'_{max,r}{}^1 = 3$ ,  $e'_{max,r}{}^2 = 1$ ,  $e'_{max,r}{}^3 = 3$ ,  $e'_{max,l}{}^4 = -1$  and min prediction errors  $e'_{min,l}{}^1 = -2$ ,  $e'_{min,l}{}^2 = -1$ ,  $e'_{min,r}{}^3 = 1$ ,  $e'_{min,r}{}^4 = 0$  as formula (3.8) and (3.9). In first block, the largest pixel ( $e'_{max,r}{}^1 = e_{max,r}^* + 1$ ) and the smallest pixel ( $e'_{min,l}{}^1 = e_{min,l}^*$ ) are both extracted secret bit; in third

block, the largest pixel( $e'_{max,r}^3 = e_{max,r}^* + 1$ ) is extracted secret bit, all other largest and smallest pixels contain no secret bits. Finally, we can extract the secret data and recover the cover image from the stego-image without extra information.

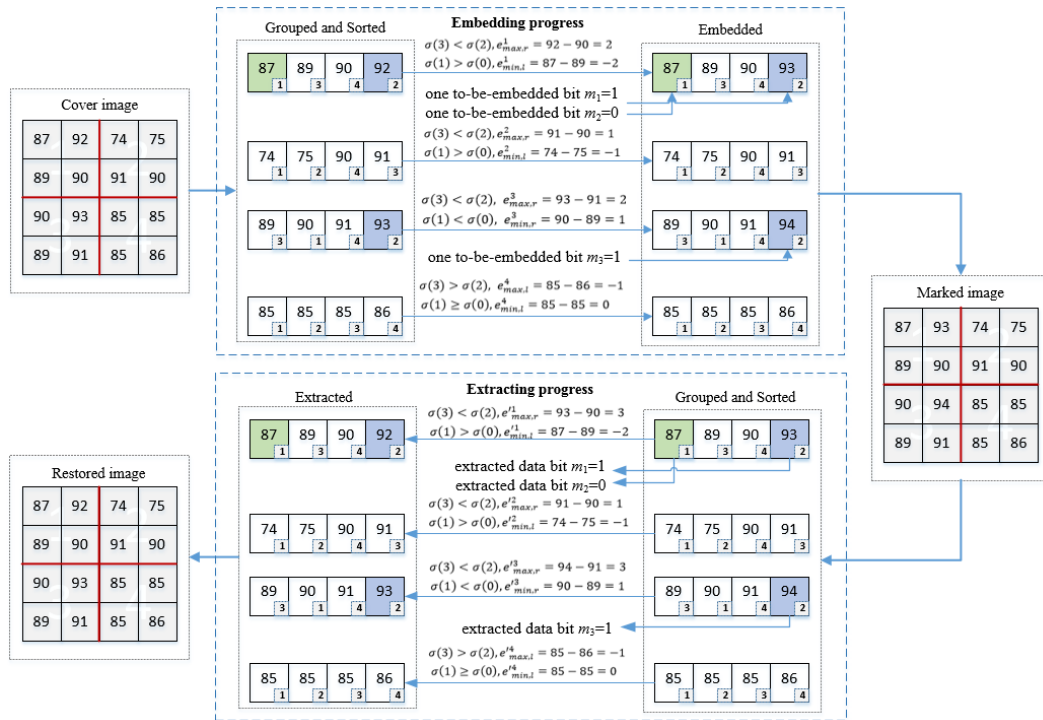


Figure 4. Illustration of data embedding and data extraction.

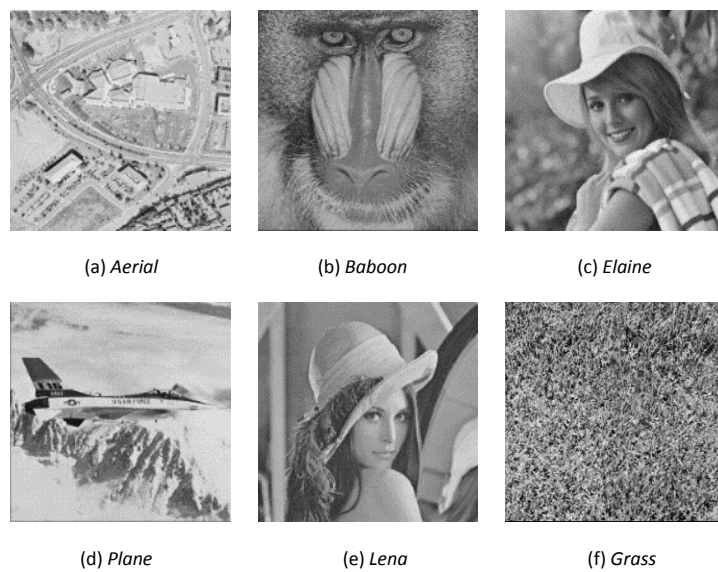
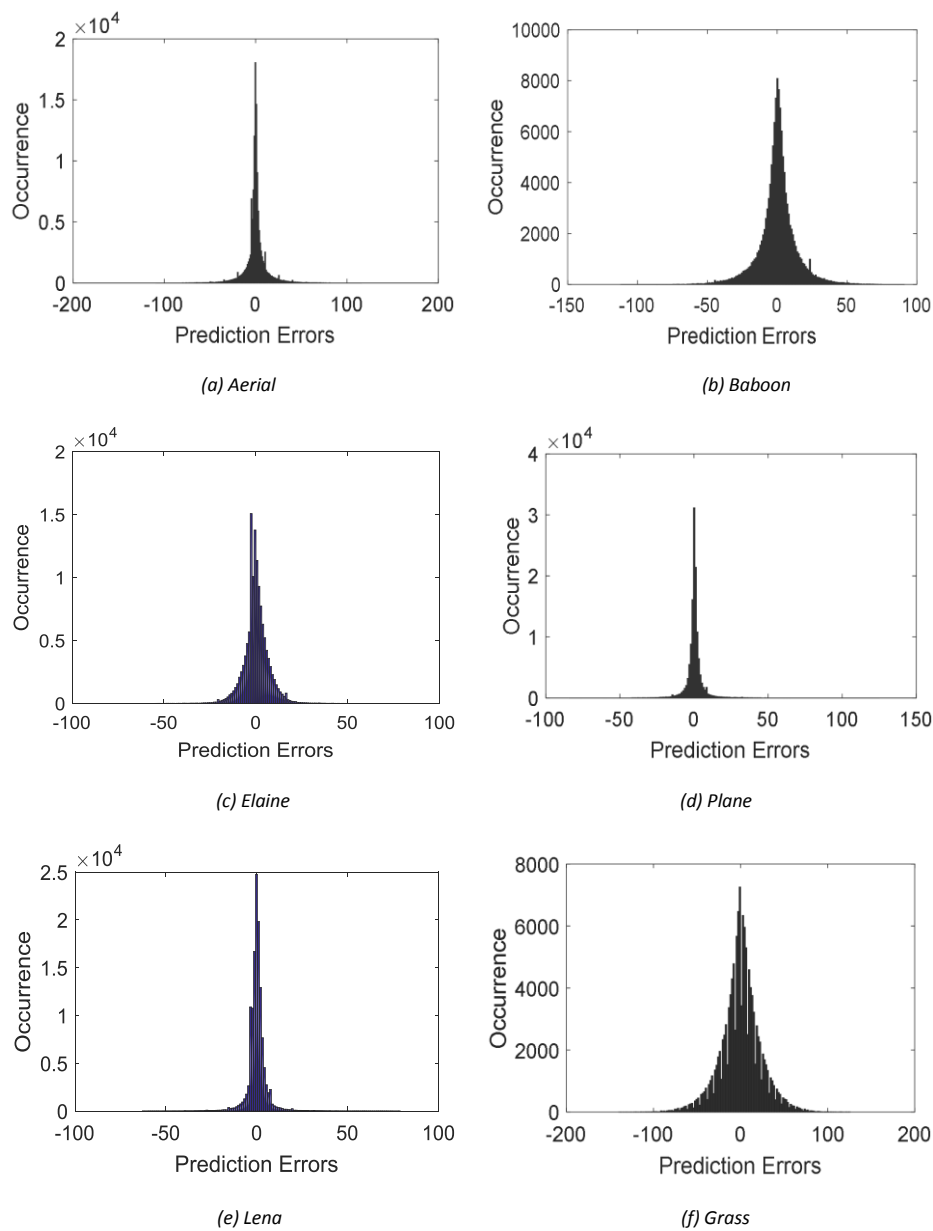


Figure 5. Test images for reversible data hiding.

#### 4. Experimental results

In this section, the performance of the proposed scheme was evaluated, and validity of the improved IPVO algorithm was verified by comparing the proposed algorithm with other recently proposed algorithm using some test images. The test images are  $512 \times 512$  sized standard gray-scale images of pixel value between 0 and 255, taken from the standard database and include *Aerial*, *Baboon*, *Elaine*, *Plane*, *Lena*, and *Grass*, as shown in Figure 5.

As described in the previous algorithm, the test images are divided into  $512 \times 512/4 = 65536$  non-overlapping blocks. For each image, after calculating the maximum prediction error and minimum prediction error of all blocks by formula (3.2) and (3.3), and histogram of prediction errors was drawn as shown in Figure 6.



**Figure 6.** Histogram of Prediction Errors Distribution.

For each test image, floating range of the prediction errors was shown in Table 2, at the same time, number of pixels that may occur overflow ( $x_{\sigma(i+3)} = 255$ ) or underflow ( $x_{\sigma(i)} = 0$ ) in these blocks was also shown in Table 2, these pixels may need special processing in the process of embedding.

**Table 2.** Relevant statistical information of Test image.

Test image	Range of $e_{max,r}$	Range of $e_{min,r}$	Overflow Pixels	Underflow Pixels
	or $e_{max,l}$	or $e_{min,l}$	Number	Number
<i>Aerial</i>	[-112,91 ]	[-95 , 86 ]	256	0
<i>Baboon</i>	[-109,105]	[-98 , 116]	0	0
<i>Elaine</i>	[-127,126]	[-140,125]	0	1
<i>Plane</i>	[-66 , 72 ]	[-76 , 72 ]	0	3
<i>Lena</i>	[-55 , 79 ]	[-63 , 74 ]	0	0
<i>Grass</i>	[-81 , 125]	[-84 , 79 ]	0	288

We change  $EC$  with a step size, the maximum prediction errors  $e_{max,r}^*, e_{max,l}^*$  and minimum prediction errors  $e_{min,r}^*, e_{min,l}^*$  selected are constantly changing, the specific situations are shown in Table 3 ~ Table 8.

**Table 3.** Different selection of prediction-errors with various capacity (*Aerial*).

$EC$ (bit)	$e_{max,r}^*$	$e_{max,l}^*$	$e_{min,r}^*$	$e_{min,l}^*$	$SR$		Pixels number	
					proposed scheme	IPVO[21]	overflow	underflow
5000	1	0	10	0	0.7468	0.7543	256	0
10000	1	0	1	-9	0.7445	0.7449	256	0
15000	1	0	1	0	0.7299	0.7299	256	0
20000	1	0	1	0	0.7396	0.7396	256	0
25000	1	0	1	0	0.7429	0.7429	256	0
30000	1	0	1	0	0.7435	0.7435	256	0

**Table 4.** Different selection of prediction-errors with various capacity (*Baboon*).

$EC$ (bit)	$e_{max,r}^*$	$e_{max,l}^*$	$e_{min,r}^*$	$e_{min,l}^*$	$SR$		Pixels number	
					proposed scheme	IPVO[21]	overflow	underflow
4000	10	-10	3	-10	0.8995	0.9165	0	0
6000	10	-3	3	-8	0.8907	0.9036	0	0
8000	2	-10	4	-4	0.8844	0.8946	0	0
10000	3	-3	3	-8	0.8808	0.8900	0	0
12000	2	-2	3	-4	0.8776	0.8848	0	0
14000	2	-1	2	-1	0.8774	0.8822	0	0

**Table 5.** Different selection of prediction-errors with various capacity (*Elaine*).

$EC$ (bit)	$e_{max,r}^*$	$e_{max,l}^*$	$e_{min,r}^*$	$e_{min,l}^*$	$SR$		Pixels number	
					proposed scheme	IPVO[21]	overflow	underflow
3000	9	0	3	0	0.7164	0.7170	0	1
6000	1	0	3	0	0.7566	0.7577	0	1
9000	1	0	1	0	0.7825	0.7825	0	1
12000	8	-6	7	0	0.8006	0.8038	0	1
15000	9	0	10	0	0.8040	0.8141	0	1
18000	4	0	9	0	0.8030	0.8140	0	1
21000	10	0	8	0	0.8041	0.8139	0	1
24000	1	0	2	0	0.8085	0.8113	0	1

**Table 6.** Different selection of prediction-errors with various capacity (*Plane*).

$EC$ (bit)	$e_{max,r}^*$	$e_{max,l}^*$	$e_{min,r}^*$	$e_{min,l}^*$	$SR$		Pixels number	
					proposed scheme	IPVO[21]	overflow	underflow
5000	3	-3	3	-3	0.5173	0.6472	0	1
10000	1	-2	3	-3	0.5336	0.6143	0	1
15000	1	-1	1	-3	0.5457	0.5920	0	3
20000	1	0	1	-1	0.5559	0.5726	0	3
25000	1	0	1	0	0.5756	0.5756	0	3
30000	1	0	1	0	0.5824	0.5824	0	3
35000	1	0	1	0	0.5953	0.5953	0	3
40000	1	0	10	0	0.6016	0.6016	0	0

**Table 7.** Different selection of prediction-errors with various capacity (*Lena*).

$EC$ (bit)	$e_{max,r}^*$	$e_{max,l}^*$	$e_{min,r}^*$	$e_{min,l}^*$	$SR$		Pixels number	
					proposed scheme	IPVO[21]	overflow	underflow
5000	2	0	2	0	0.5007	0.5078	0	0
10000	1	0	1	0	0.5784	0.5784	0	0
15000	1	0	1	0	0.6150	0.6150	0	0
20000	1	0	1	0	0.6353	0.6353	0	0
25000	1	0	1	0	0.6481	0.6481	0	0
30000	1	0	1	-1	0.6531	0.6535	0	0
35000	1	-1	1	-1	0.6559	0.6588	0	0
40000	1	-1	1	0	0.6598	0.6618	0	0

**Table 8.** Different selection of prediction-errors with various capacity (*Grass*).

EC (bit)	$e_{max,r}^*$	$e_{max,l}^*$	$e_{min,r}^*$	$e_{min,l}^*$	SR		Pixels number	
					proposed scheme	IPVO[21]	overflow	underflow
1000	10	-5	1	-9	0.9462	0.9531	0	7
2000	7	-3	1	-5	0.9458	0.9507	0	10
3000	8	-3	5	-5	0.9449	0.9500	0	6
4000	10	-10	9	-9	0.9435	0.9496	0	3
5000	10	-6	5	-9	0.9420	0.9495	0	3
6000	9	-3	8	0	0.9419	0.9485	0	284
7000	4	0	5	0	0.9416	0.9464	0	284

From the above results, it can be seen that the shift rate of the marked image is further reduced and the *PSNR* of the marked image is further improved due to the optimization embedding difference in different embedding requirements. For example, the *SR* of the proposed scheme for *Plane* is 0.5559 when embedding capacity is 20000 bits, that is a decrease of 0.0167 over the IPVO method, the rate of decline is 2.91%; the *PSNR* of the proposed scheme for *Plane* is 56.8626 when embedding capacity is 20000 bits, that is an increase of 0.225 over the IPVO method, the upgrade rate is 0.4%, the best four embedding prediction errors are 1, 0, 1, -1. Especially for the case of small capacity and high-fidelity, the advantage of this algorithm is more obvious. For example, the *SR* of the proposed scheme for *Plane* is 0.5173 when embedding capacity is 5000 bits, that is a decrease of 0.1299 over the IPVO method, the rate of decline is 20.07%; the *PSNR* of the proposed scheme for *Plane* is 63.3333 when embedding capacity is 5000 bits, that is an increase of 1.7082 over the IPVO method, the upgrade rate is 2.78%, the maximum prediction error 3, -3 and minimum prediction error 3, -3 selected for embedding.

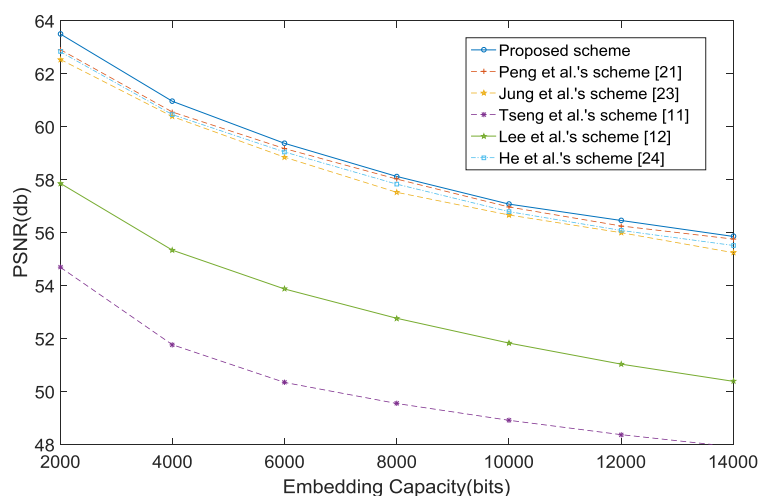
Table 2 lists the pixels number of overflow or underflow in each test image. From the results, we found that for most of the image it is easy to deal with the problem of overflow or underflow, the number of overflow or underflow is small, and the additional information that need to be recorded is relatively small, which has little impact on the performance of the algorithm. However, there are still some unsatisfactory situations, such as *Aerial* image, the pixels number of overflow are 256, at this point the length of bitmap information we need to record is  $16 \times 256 = 4096$  bits, the additional information length is the sum of the auxiliary information length and bitmap information length, and the result is  $58 + 32 + 4096 = 4186$  bits. When the length of secret information is 5000 bits, the actual embedding amount is  $5000 + 4186 = 9186$  bits, in this case, the performance of the algorithm has no obvious advantages.

In this algorithm, data hiding is implemented by predictive difference of non-fixed 0 and 1. In the actual operation, not all pixels with 255 or 0 will perform embedding or shifting operations, only the pixels satisfying the conditions for embedding or shifting operations may overflow. Therefore, compared with IPVO method, we deliberately add judgment conditions in dealing with overflow information. Such process enables us to find that different prediction errors are used because of different embedding quantities, which lead to the change of overflow situation. For image *plane*, when the embedding amount is 5000 or 10000 bits, the number of overflow pixel is 1; when the embedding amount is 40000 bits, the number of overflow pixel is 0; For other test embedding capacity, the number of overflow pixel is 3. For image *grass*, although the number of overflow pixels may be 288, the number of actual overflow is very small due to the change of selected prediction

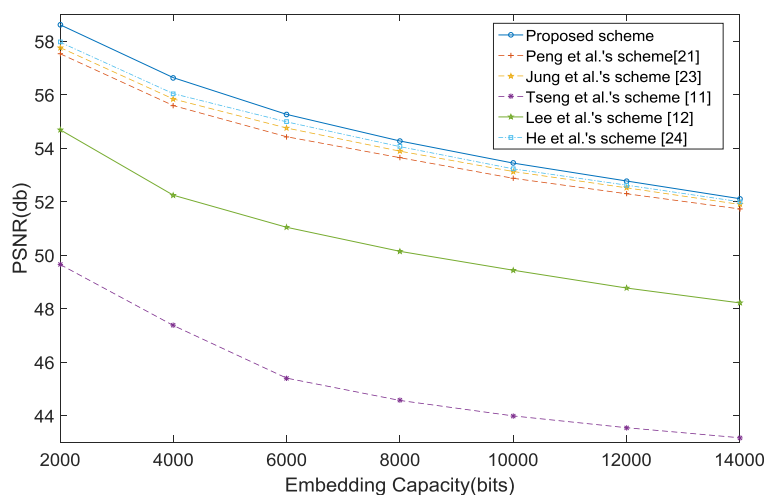


differences in the process of experiment. When the embedding amount is 1000 or 2000 etc. bits, the actual overflow amount is only 3 and the bitmap size is small; However, when the embedding amount is 6000 or 7000 bits, the actual amount of overflow increases to 284, and the performance of the algorithm decrease significantly, even because of the overflow information is too long to exceed the embedding capacity of image, the embedding process cannot be completed properly. These result fully reflect that for certain image the performance of RDH method based on pixel value ordering is closely related to the image's own characteristics.

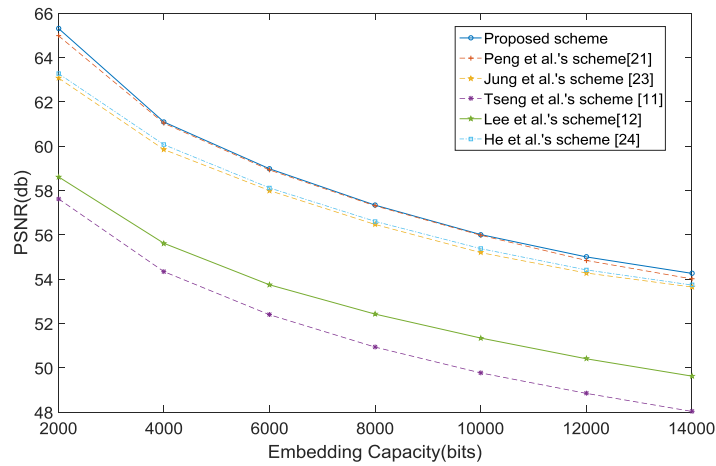
Next, comparing the performances between the proposed method and other four methods that are Peng et al.'s scheme (IPVO)[21], Jung et al.'s scheme [23], Lee et al.'s scheme (PEE)[12], Tseng et al.'s scheme [13] and He et al.'s scheme [24], the results are shown from Figure 7 to Figure 12.



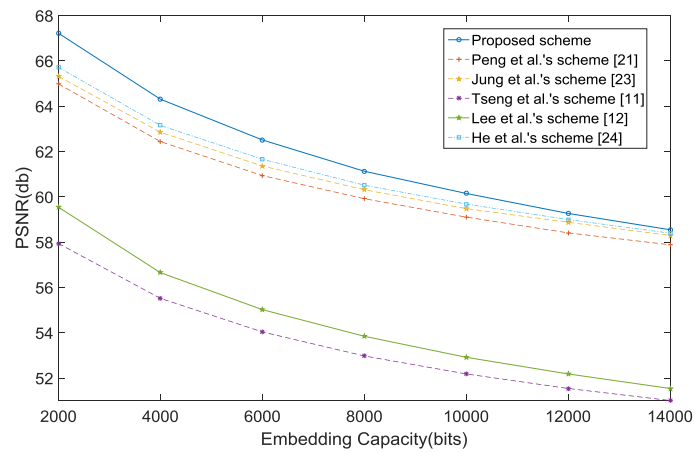
**Figure 7.** Performance comparison between the proposed scheme and other schemes (*Aerial*).



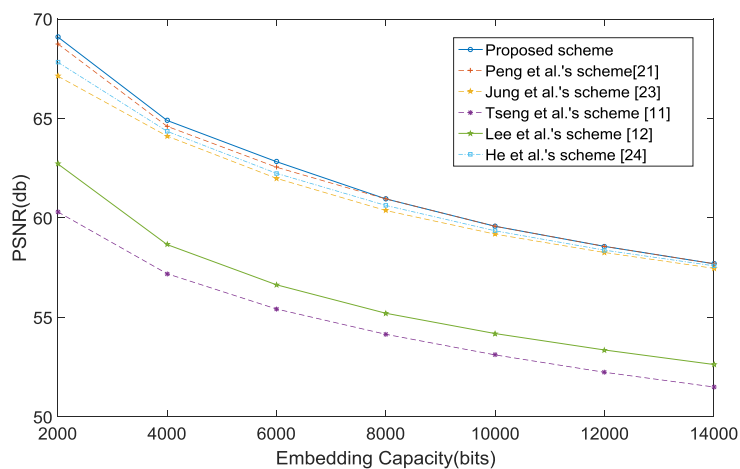
**Figure 8.** Performance comparison between the proposed scheme and other schemes (*Baboon*).



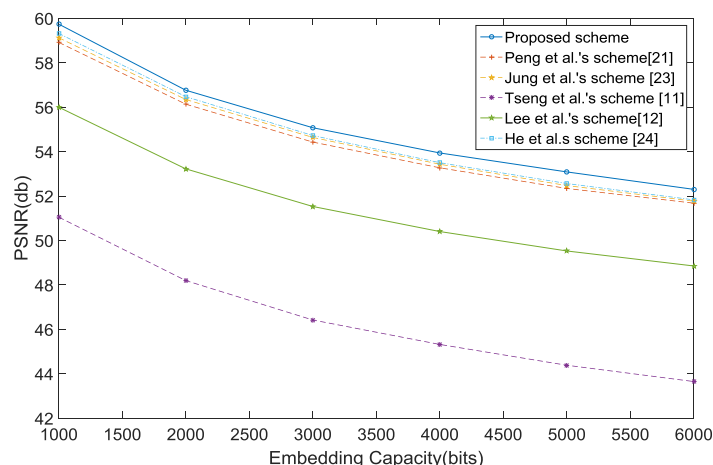
**Figure 9.** Performance comparison between the proposed scheme and other schemes (*Elaine*).



**Figure 10.** Performance comparison between the proposed scheme and other schemes (*Plane*).



**Figure 11.** Performance comparison between the proposed scheme and other schemes (*Lena*).



**Figure 12.** Performance comparison between the proposed scheme and other schemes (*Grass*).

It can be obtained from the experiments in Figures 7 to 12 that the marked images created by our method have less distortions and a higher *PSNR* values for the majority of *EC*. Take the *Grass* picture as an example, when embedding capacity increases from 1000 bits to 8000 bits with a step size of 1000 bits, the *PSNR* values of the proposed method are always higher than that of other methods, because the most appropriate combinations of difference errors are selected to embed information and the number of shift pixel is the smallest. For other pictures, the *PSNR* values of ours are also not lower than that of other methods.

The second fact that we can observe from the experiments in Figures 7 to 12 is that the *PSNR* values of the proposed algorithm are significantly superior to that of other methods when embedding capacity is small; and as the *EC* is getting large, the difference of the *PSNR* value between the proposed method and other methods gets smaller and smaller. When embedding capacity increases to a certain degree, the proposed algorithm may use the same difference errors as the traditional IPVO method to embed information, and the proposed algorithm degrades to a traditional IPVO method. Actually, the performance of the proposed algorithm is more superior especially for small capacity. Therefore, in the case of high fidelity and low payload, the ideas presented in this paper can achieved very impressive embedding performance and bring better visual experience.

The third fact that we can observe from the experiments in Figures 7 to 12 is that for rough images, such as *Plane* and *Baboon*, the proposed method is significantly better than the other methods. However, for flat images, for example, *Lena* and *Elaine*, only slightly better than the other methods. There are many prediction-error values of 0 and 1 in flat images, and other values are less. The traditional IPVO method is to use prediction-error values of 0 and 1 to embed secret information. Since the number of prediction-error values other than 0 and 1 is too small to satisfy the requirement of embedding capacity, the proposed method is also downgraded to a traditional IPVO method to use prediction-error values of 0 and 1. On the other hand, in rough images, the prediction error value is likely distributed evenly. The numbers of some prediction-error values other than 0 and 1 are big enough to satisfy the requirement of embedding capacity. The proposed method can use a combination of prediction-error values of other than 0 and 1 to get lower shift rate; however, the traditional IPVO method can only use prediction-error values of 0 and 1 fixedly to embed secret

information. Therefore, in the rough image, the proposed method can obtain a lower shift rate than the conventional IPVO method.

In order to further illustrate the improvement effect of the algorithm, we test the influence of different  $TS$  values on embedding process and algorithm performance. Take *grass* image as an example, result as shown in Table 9.

**Table 9.** Effect of different  $TS$  on algorithm performance.

$TS$	$e_{max,r}^*$	$e_{max,l}^*$	$e_{min,r}^*$	$e_{min,l}^*$	$SR$	$PSNR$
2	1	-1	1	-1	0.9484	52.5622
4	2	-2	1	-2	0.9475	52.6633
6	3	-3	3	-3	0.9459	52.7774
8	4	-3	4	-3	0.9449	52.8598
10	5	-3	5	-3	0.9440	52.9255
12	6	-6	5	-3	0.9432	52.9934
14	7	-6	5	-3	0.9431	52.9988
16	8	-6	8	-8	0.9428	53.0420
18	8	-6	8	-9	0.9422	53.0678
20	10	-6	5	-9	0.9420	53.0845
22	10	-6	5	-9	0.9420	53.0845

Assuming the embedding amount was 5000 bits,  $TS$  was gradually increased from 2 to 20 in step 2 to complete RDH respectively, we found that the prediction errors are different for each embedding. With the increase of  $TS$ , the range of prediction errors became larger, while  $SR$  decreased and  $PSNR$  increased gradually. Of course, this did not mean that the bigger  $TS$  is, the bigger  $PSNR$  is. When the value of  $TS$  increased to 22 or greater, selection of the embedding difference did not change again. So we found that, the number of pixels distributed at both ends of the error histogram (Fig. 6) is small, which made it difficult to meet the requirement of embedding capacity. Increasing  $TS$  not only didn't improve the algorithm performance continuously, but also increased the algorithm complexity. Conversely, if the  $TS$  is too small, the algorithm performance cannot be brought into play, especially for rough images. Therefore, how to choose right value of  $TS$  is a topic worthy discussing,  $TS$  should be properly selected according to the actual situation.

## 5. Conclusion

In this paper, an improved IPVO-based RDH scheme using floating predictor was proposed. The maximum prediction-errors are divided into two cases: right shift and left shift, and the minimum prediction-errors are also divided into right shift and left shift. Thus, we get four prediction-errors for information embedding. The prediction error combination includes a maximum right shift prediction error, a maximum left shift prediction error, a minimum right shift prediction error, and a minimum left shift prediction error to embed information. A model is utilized to get the best combination of four prediction errors. The selection principle is to minimize image distortion for a given embedding capacity. When the prediction error of each pixel block is equal to the selected embedding error, the maximum pixel value and the minimum pixel value is expanded respectively, otherwise, when the prediction error of each pixel block is greater or smaller than the selected embedding error, the shift

operation is performed respectively. Experimental results showed that the proposed method can achieve a higher *PSNR* than current IPVO scheme, under the condition of same embedding capacity, especially for relatively rough images. In the case of low embedding capacity, the proposed method is a high-fidelity reversible data hiding scheme with low distortion and satisfied visual quality.

## Acknowledgments

This research work is supported by National Natural Science Research Project of China (No. 17BQNS01007).

## Conflict of interest

The authors declare that there is no Conflict of interest.

## References

1. J. Fridrich, M. Goljan and R. Du, Lossless data embedding-new paradigm in digital watermarking, *EURASIP J. Adv. Signal Process.*, **2** (2002), 185–196.
2. M. U. Celik, G. Sharma and A. M. Tekalp, Lossless generalized-LSB data embedding, *IEEE Transact. Image Process.*, **14** (2005), 253–266.
3. J. Tian, Reversible data embedding using a difference expansion, *IEEE Transact. Circuit. Syst. Video Technol.*, **13** (2003), 890–896.
4. A. M. Alattar, Reversible watermark using the difference expansion of a generalized integer transform, *IEEE Transact. Image Process.*, **13** (2004), 1147–1156.
5. Z. Ni, Y.Q. Shi and N. Snsari, Reversible data hiding, *IEEE Transact. Circuit. Syst. Video Technol.*, **16** (2006), 354–362.
6. C. Kim, Content-based image copy detection, *Signal Process. Image Commun.*, **18** (2003), 169–184.
7. S. K. Lee, Y. H. Suh and Y. S. Ho, Reversible image authentication based on water-marking, *Process IEEE ICME*, (2006), 1321–1324.
8. X. Li, B. Li and B. Yang, General framework to histogram-shifting-based reversible data hiding, *IEEE Transact. Image Process.*, **23** (2013), 2181–2191.
9. D. M. Thodi and J. J. Rodrigues, Expansion embedding techniques for reversible watermarking, *IEEE Transact. Image Process.*, **16** (2007), 721–730.
10. D. M. Thodi and J. J. Rodrigues, Reversible watermarking by prediction-error expansion, *IEEE Southwest Symposium on Image Analysis & Interpretation IEEE*, (2004), 21–25.
11. H. W. Tseng and C. P. Hsieh, Prediction-based reversible data hiding, *Inform. Sci.*, **179** (2009), 2460–2469.
12. C. F. Lee, H. L. Chen and H. K. Tso, Embedding capacity raising in reversible data hiding based on prediction of difference expansion, *J. Syst. Software*, **83**(2010), 1864–1872.
13. Q. Shen, G. Liu and W. Liu, Adaptive image steganography based on pixel selection, *2015 IEEE International Conference on Progress in Informatics & Computing*, 2016.
14. Y. Hu, H. K. Lee and J. Li, De-based reversible data hiding with improved overflow location map, *IEEE Transact. Circuits Syst. Video Technol.*, **19** (2009), 250–260.

15. X. Li, B. Yang and T. Zeng, Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection, *IEEE Transact. Image Process.*, **20** (2011), 3524–3533.
16. D. Coltuc, Improved embedding for prediction-based reversible watermarking, *IEEE Transact. Inform. Forens. Secur.*, **6** (2011), 873–882.
17. D. Coltuc, Low distortion transform for reversible watermarking, *IEEE Transact. Image Process.*, **21** (2012), 412–417.
18. X. Li, J. Li and B. Li, High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion, *Signal Process.*, **93**(2013), 198–205.
19. B. Ou, X. Li and Y. Zhao, Reversible data hiding using invariant pixel value ordering and prediction-error expansion, *Signal Process.*, **29** (2013), 760–772.
20. X. Qu and H. J. Kim, Pixel-based pixel value ordering predictor for high-fidelity reversible data hiding, *Signal Process.*, **111**(2015), 249–260.
21. F. Peng, X. Li and B. Yang, Improved PVO-based reversible data hiding, *Digital Signal Process.*, **25** (2014), 255–265.
22. P. Subitha and V. Vaithyanathan, Novel reversible pixel-value-ordering technique for secret concealment, *Indian J. Sci. Technol.*, **8** (2015), 628–636.
23. K. H. Jung, A high-capacity reversible data hiding scheme based on sorting and prediction in digital images, *Mult. Tools Appl.*, **76**(2017), 13127–13137.
24. W. He, G. Xiong and S. Weng, Reversible data hiding using multi-pass pixel value ordering and prediction-error expansion, *Inform. Sci.*, **467**(2018), 784–799.
25. Y. Du, Z. X. Yin and X. P. Zhang, Improved lossless data hiding for jpeg images based on histogram modification, *Comput. Mater. Cont.*, **55**(2018), 495–507.
26. X. T. Duan, H. X. Song and C. Qin, Coverless steganography for digital images based on a generative model, *Comput. Mater. Cont.*, **55**(2018), 483–493.
27. L. Z. Xiong and Y. Q. Shi, On the privacy-preserving outsourcing scheme of reversible data hiding over encrypted image data in cloud computing, *Comput. Mater. Cont.*, **55**(2018), 523–539.
28. M. J. Weinberger, G. Seroussi and G. Sapiro, The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS, *IEEE Transact. Image Process.*, **9** (2000), 1309–1324.
29. X. Wu and N. Memon, Context-based, adaptive, lossless image coding, *IEEE Transact. Commun.*, **45** (1997), 437–444.
30. V. Sachnev, H. J. Kim and J. Nam, Reversible watermarking algorithm using sorting and prediction, *IEEE Transact. Circuit. Syst. Video Technol.*, **19** (2009), 989–999.



AIMS Press

©2019 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)