Improved reversible data hiding based on PVO and adaptive pairwise embedding [*]

Haorui Wu[a,1,2]**, Xiaolong Li**[b,1,2]**, Yao Zhao**[c,1,2]**, Rongrong Ni**[d,1,2]

[1]Institute of Information Science, Beijing Jiaotong University, Beijing 100044, China
[2]Beijing Key Laboratory of Advanced Information Science and Network Technology, Beijing 100044, China

Abstract Pixel-value-ordering (PVO) is an efficient technique of reversible data hiding (RDH). By PVO, the cover image is first divided into non-overlapping blocks with equal size. Then, the pixel values in each block are sorted in ascending order. Next, take the second largest/samllest pixel value as a prediction of the largest/samllest pixel value to derive two prediction-errors. Finally, the data embedding is constructed by modifying the generated prediction-errors of each block. After data embedding, the PVO of each block is unchanged, which guarantees the reversibility. Our key observation is that, in each block, the modification for the two prediction-errors is independent without exploiting the correlation between them, although they are closely correlated to each other. Based on this consideration, an improved PVO-based RDH method is proposed in this work. The two prediction-errors of each block is considered as a pair, and the pairs are modified for data embedding based on adaptive two-dimensional histogram modification. The proposed method is experimentally verified better than the original PVO-based method and some of its improvements.

Keywords Reversible data hiding · Pixel-value-ordering · Two-dimensional histogram · Pairwise embedding · Adaptive embedding

1 Introduction

Reversible data hiding (RDH) is a special type of information hiding technique. By RDH, the decoder can perfectly recover the cover image after extracting the embedded secret data. The key question of RDH technology is how to minimize the embedding distortion for a given the embedded capacity.

So far, RDH has been widely studied and many effective reversible embedding methods have been proposed in the literature [1–24]. Among the present RDH methods, the ones based on prediction-error expansion (PEE) have shown an outstanding embedding performance by exploiting the correlations between adjacent pixels. The PEE technique is first proposed by Thodi and Rodriguez in [3]. In this method, a prediction-error histogram is first generated, and then this histogram is modified to embed data based on expansion and shifting the histogram bins. Later on, the PEE technique has been widely adopted and developed in RDH studies. These developments are mainly based on, for example, sharper histogram generation with an accurate predictor design [5, 15, 19], adaptive histogram modification [6, 14, 22], two-dimensional histogram modification [10, 11, 20], multiple histogram modification [17], etc.

The pixel-value-ordering (PVO) based RDH is first proposed by Li *et al.* in [25]. In this method, a new predictor is designed based on PVO for PEE. Specifically, the cover image is first divided into non-overlapping blocks with equal size, and, for each block, its pixel values are sorted in ascending order. Next, the second largest/samllest pixel value is taken as a prediction of the largest/samllest pixel value to derive two prediction-errors. Finally, the data embedding is conducted by modifying the generated prediction-errors of each block. After data embedding, the PVO of each block remains unchanged, which ensures the reversibility. The PVO-based technique can derive an accurate predictor, and its per-

formance is proved better than some other PEE-based methods, especially for low embedding capacities. Later on, the PVO-based method [25] has been improved in some works [26–32] in many aspects.

In this paper, we try to improve the embedding performance of PVO-based RDH. Our key observation is that, for the PVO-based methods such as [25] and [26], in each divided image block, the modification for the two prediction-errors is independent without exploiting the correlation between them, although they are closely correlated to each other. Based on this consideration, in this work, the two prediction-errors of each block is considered as a pair, and the pairs are modified for data embedding based on two-dimensional (2D) histogram modification. The 2D histogram modification manner is adaptively determined in our work such that the embedding performance is optimized. Moreover, inspired by the previous works [25, 26], a block selection strategy is employed as well, by which only the smooth image blocks are selected for data embedding while the rough ones are unmodified. The proposed method is experimentally verified better than the original PVO-based RDH method [25] and some of its improvements [26, 27].

The rest of the paper is arranged as follows. Section 2 provides a brief review for two typical PVO-based methods [25] and [26]. Section 3 introduces the proposed PVO-based RDH. Experimental results are reported in Section 4. Finally, Section 5 concludes this paper.

2 Related work

In this section, as a preparation, the original PVO-based RDH scheme [25] and its improvement [26] are reviewed.

2.1 PVO-based RDH [25]

The PVO-based RDH technique is first proposed by Li *et al.* in [25]. For this method, first, the cover image is divided into non-overlapping blocks sized $n_1 \times n_2$. Then, for a given block, sort its pixels $(p_1, ..., p_n)$ in ascending order according to the pixel values to obtain $(p_{\sigma(1)}, ..., p_{\sigma(n)})$, where $\sigma : \{1, ..., n\} \rightarrow \{1, ..., n\}$ is the unique one-to-one mapping satisfying $\sigma(i) < \sigma(j)$ if $p_{\sigma(i)} = p_{\sigma(j)}$ and $i < j$, and $n = n_1 \times n_2$. Next, take the second largest pixel value $p_{\sigma(n-1)}$ as the prediction of the largest pixel value $p_{\sigma(n)}$, and define the prediction-error as

$$d_{\max} = p_{\sigma(n)} - p_{\sigma(n-1)}. \tag{1}$$

Take the standard gray-scale Lena image with 2×2 sized blocks as an example, the histogram of prediction-error $d_{\max}$ is shown in Figure 1. It can be observed that the histogram is defined in the interval $[0, +\infty)$ since $d_{\max} \geq 0$, and it has a peak value at bin 1. Then, based on this peak property of
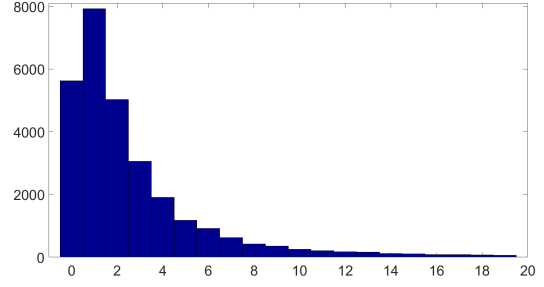

Fig. 1 Histogram of $d_{\max}$ defined in (1), for the standard 512×512 sized gray-scale image Lena with block size of 2×2.

the generated histogram, $d_{\max}$ is modified according to the following rule to derive the marked prediction-error $\widetilde{d}_{\max}$ as

$$\widetilde{d}_{\max} = \begin{cases} d_{\max}, & \text{if } d_{\max} = 0 \\ d_{\max} + b, & \text{if } d_{\max} = 1 \\ d_{\max} + 1, & \text{if } d_{\max} > 1 \end{cases} \tag{2}$$

where $b \in \{0, 1\}$ is a to-be-embedded secret data bit. Finally, the largest pixel value $p_{\sigma(n)}$ is modified as

$$\widetilde{p}_{\sigma(n)} = p_{\sigma(n-1)} + \widetilde{d}_{\max} \tag{3}$$

to derive the marked pixel value.

Notice that in the above procedure, for each block, only the largest pixel value with prediction-error larger than 0 maybe modified in which this largest value is either unchanged or increased by 1, while other pixels remain unchanged. As a result, the PVO of each block is unchanged as well, and this property guarantees the reversibility. Specifically, for the decoder, the same as the data embedding process, the marked image is also divided into non-overlapping blocks of size $n_1 \times n_2$. Then, for a given block, sort its pixel values in ascending order to obtain $(\widetilde{p}_{\sigma(1)}, ..., \widetilde{p}_{\sigma(n)})$. Notice that $\widetilde{p}_{\sigma(i)} = p_{\sigma(i)}$ holds for each $1 \leq i \leq n-1$. Next, compute the marked prediction-error

$$\widetilde{d}_{\max} = \widetilde{p}_{\sigma(n)} - \widetilde{p}_{\sigma(n-1)}. \tag{4}$$

Finally, recover the original pixel value $p_{\sigma(n)}$ as

$$p_{\sigma(n)} = \begin{cases} \widetilde{p}_{\sigma(n)}, & \text{if } \widetilde{d}_{\max} \in \{0, 1\} \\ \widetilde{p}_{\sigma(n)} - 1, & \text{if } \widetilde{d}_{\max} > 1 \end{cases} \tag{5}$$

and extract the embedded data as $b = \widetilde{d}_{\max} - 1$ in the case of $\widetilde{d}_{\max} \in \{1, 2\}$. Moreover, since only the largest pixel value of each block maybe modified in the data embedding procedure, for each $1 \leq i \leq n-1$, $p_{\sigma(i)}$ can be recovered as $\widetilde{p}_{\sigma(i)}$ itself.

Simultaneously, the smallest pixel value $p_{\sigma(1)}$ in each block can also be modified (either decreased by 1 or unchanged) to embed data. The similar data embedding and extraction procedures are omitted here.

2.2 Improved PVO-based RDH [26]

In the original PVO-based RDH scheme, the blocks with $d_{max} = 0$ are not utilized to carry data. However, these blocks are usually smooth ones and suitable for reversible embedding. Based on this consideration, in order to take the advantage of the blocks with $d_{max} = 0$, an improved PVO-based scheme is proposed by Peng *et al.* in [26].

For the data embedding of this method, first, for a given block with sorted values $(p_{\sigma(1)}, ..., p_{\sigma(n)})$, instead of computing the prediction-error d_{max} in (1) as the original PVO-based method does, it is redefined as follows considering the order of $\sigma(n-1)$ and $\sigma(n)$

$$d_{max} = \begin{cases} p_{\sigma(n)} - p_{\sigma(n-1)}, & \text{if } \sigma(n) > \sigma(n-1) \\ p_{\sigma(n-1)} - p_{\sigma(n)}, & \text{if } \sigma(n) < \sigma(n-1) \end{cases}. \quad (6)$$

Clearly, one can verify that the redefined prediction-error satisfies $d_{max} \geq 0$ if $\sigma(n) > \sigma(n-1)$, and $d_{max} < 0$ if $\sigma(n) < \sigma(n-1)$. That is to say, the prediction-error defined in this way is ranged from $-\infty$ to $+\infty$. For example, for the Lena image, the histogram of the redefined prediction-error d_{max} is shown in Figure 2. One can see that this histogram is a Laplacian-like distribution centered at 0 with two sides decay. Then, the bins 0 and -1 are expanded for data embedding. More specifically, d_{max} is modified to derive the marked prediction-error $\widetilde{d}_{max}$ in the following way

$$\widetilde{d}_{max} = \begin{cases} d_{max} + b, & \text{if } d_{max} = 0 \\ d_{max} - b, & \text{if } d_{max} = -1 \\ d_{max} + 1, & \text{if } d_{max} \geq 1 \\ d_{max} - 1, & \text{if } d_{max} \leq -2 \end{cases} \quad (7)$$

where $b \in \{0, 1\}$ is a to-be-embedded secret data bit. Accordingly, the largest pixel value $p_{\sigma(n)}$ is modified as

$$\widetilde{p}_{\sigma(n)} = p_{\sigma(n-1)} + |\widetilde{d}_{max}| \quad (8)$$

to derive the marked pixel value.

For this improved method, a key issue is that, unlike other expansion-shifting based RDH methods, the expansion bins can not be arbitrarily selected. To guarantee the reversibility, the sign of each prediction-error (i.e., "≥ 0" or "< 0") shouldn't be changed after data embedding.

The same as the original PVO-based method, in each block, only the largest pixel value $p_{\sigma(n)}$ is either increased by 1 or unchanged, while other pixel values remain unchanged. The PVO of each block is unchanged as well, and thus the recovery and extraction process can be conducted accordingly. Specifically, for the decoder, the marked prediction-error $\widetilde{d}_{max}$ is first computed for a marked block with sorted values $(\widetilde{p}_{\sigma(1)}, ..., \widetilde{p}_{\sigma(n)})$ as follows,

$$\widetilde{d}_{max} = \begin{cases} \widetilde{p}_{\sigma(n)} - \widetilde{p}_{\sigma(n-1)}, & \text{if } \sigma(n) > \sigma(n-1) \\ \widetilde{p}_{\sigma(n-1)} - \widetilde{p}_{\sigma(n)}, & \text{if } \sigma(n) < \sigma(n-1) \end{cases}. \quad (9)$$
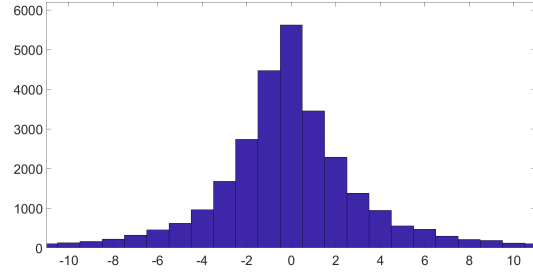

Fig. 2 Histogram of d_{max} defined in (6), for the standard 512×512 sized gray-scale image Lena with block size of 2×2.

Then, recover the original pixel value $p_{\sigma(n)}$ as

$$p_{\sigma(n)} = \begin{cases} \widetilde{p}_{\sigma(n)}, & \text{if } \widetilde{d}_{max} \in \{0, -1\} \\ \widetilde{p}_{\sigma(n)} - 1, & \text{otherwise} \end{cases}. \quad (10)$$

In addition, for each $1 \leq i \leq n-1$, $p_{\sigma(i)}$ is recovered as $\widetilde{p}_{\sigma(i)}$ itself. And, the embedded data bit is 0 if $\widetilde{d}_{max} \in \{0, -1\}$ or 1 if $\widetilde{d}_{max} \in \{1, -2\}$.

Besides, in this method, the smallest pixel value $p_{\sigma(1)}$ of each block is also modified for data embedding, by considering the prediction-error defined as

$$d_{min} = \begin{cases} p_{\sigma(2)} - p_{\sigma(1)}, & \text{if } \sigma(2) > \sigma(1) \\ p_{\sigma(1)} - p_{\sigma(2)}, & \text{if } \sigma(2) < \sigma(1) \end{cases}. \quad (11)$$

One can verify that $d_{min} \geq 0$ if $\sigma(2) > \sigma(1)$, and $d_{min} < 0$ if $\sigma(2) < \sigma(1)$. The histogram of d_{min} is also a Laplacian-like distribution centered at 0 with two sides decay. For brevity, the similar data embedding and extraction procedures by modfying the smallest pixel value are omitted here, and the details can be found in [26].

3 Proposed method

In this section, an improved RDH scheme based on PVO and adaptive pairwise embedding is proposed. Our idea is straightforward. Notice that in the original PVO-based method [25] and its improvement [26], the largest and smallest pixel values of each divided block are independently modified to embed data. For example, for the improved PVO-based method [26], this is conducted by modifying the prediction-errors d_{max} and d_{min} defined in (6) and (11), independently. However, inspired by the recently proposed RDH technique named pairwise PEE [10], we argue that this independent modification manner is unreasonable since d_{max} and d_{min} are related to each other. Then, for efficient reversible embedding, we propose to consider d_{max} and d_{min} as a pair, and then modify the prediction-error pairs based on 2D histogram modification strategies.

The notations utilized in this section are the same as Section 2.2, and we consider the prediction-errors d_{max} and d_{min} defined in (6) and (11) which take values in $(-\infty, \infty)$. Let
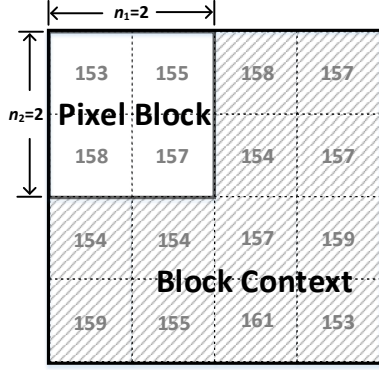
Fig. 4 Context (shadow pixels) of a given block for complexity computation.

us first see Figure 3, it shows the joint distribution of $d_{\max}$ and $d_{\min}$, for three standard 512×512 sized gray-scale images Lena, Baboon and Airplane. For each figure in Figure 3, we first divide the image into non-overlapping 2×2 sized blocks, and then compute the pair $(d_{\max}, d_{\min})$ for each divided block, and finally, derive the distribution of $(d_{\max}, d_{\min})$ based on all the divided blocks. Clearly, according to these figures, one can see that the distribution has a peak at origin $(0, 0)$ and it rapidly decreases toward the direction away from the origin, and the decrease trend is very significant for the smooth image Airplane. This observation confirms that $d_{\max}$ and $d_{\min}$ are correlated to each other. And, the smoother the image is, the stronger the correlation is. Then, to achieve better performance of PVO-based RDH, we propose to modify the prediction-error pair $(d_{\max}, d_{\min})$ for data embedding based on 2D histogram modification strategies. The idea of 2D histogram modification is motivated by the work of Ou *et al.* [10]. We describe the embedding details as below.

First, inspired by previous works [25–27], we adopt the block selection strategy to select smooth blocks for histogram generation. More specifically, in order to evaluate the complexity of a block, the same as [27], a block context is first defined as shown in Figure 4, and then the complexity of a block is computed as the sum of the vertical and the horizontal absolute difference of every two adjacent pixels in the context. For a given threshold T, only the blocks with complexities less than T are selected. The blocks with large complexities are considered as rough ones and will not be used in our data embedding. These blocks remain unchanged during the data embedding procedure. Then, the prediction-error pair $(d_{\max}, d_{\min})$ is computed for each selected blocks, and the 2D histogram counting the distribution of $(d_{\max}, d_{\min})$ is generated. After that, the data embedding will be conducted by modifying the generated 2D histogram.

Clearly, one can directly apply the pairwise embedding mechanism of Ou *et al.* [10] to modify the generated histogram. For illustration, the pairwise embedding mechanis-

m based on a 2D mapping is shown Figure 5(a). The 2D mapping here is defined as a function

$$f : \mathbb{Z}^2 \mapsto \mathscr{P}(\mathbb{Z}^2) \qquad (12)$$

where $\mathscr{P}(\mathbb{Z}^2)$ is the power set of $\mathbb{Z}^2$. For example, $f(0, 0) = \{(0, 0), (0, 1), (1, 0)\}$ meaning that in the data embedding process, the pair $(0, 0)$ will be modified as one element of the set $\{(0, 0), (0, 1), (1, 0)\}$ to embed $\log_2 3$ bits. Notice that, for a given embedding capacity, with pairwise embedding, the threshold T is determined as the smallest one such that the embedding capacity can be satisfied with the generated histogram. For example, for the Baboon image with 2×2 sized blocks and an embedding capacity of 10,000 bits, the threshold T is 235, and one can get a PSNR of 54.63 dB by applying the pairwise embedding to the corresponding histogram (this histogram is shown in Figure 5(c)). On the other hand, with the same block sizes and complexity computation method, the PSNR of Peng *et al.*'s method [26] is just 54.34 dB. That is to say, compared with [26], 0.29 dB of improvement is obtained by using pairwise embedding. Obviously, the improvement is due to the utilization of the correlation between $d_{\max}$ and $d_{\min}$.

Now, we have seen that based on pairwise embedding, the advantage of 2D histogram is verified. However, one can get better result since, with the pairwise embedding mechanism of Ou *et al.* [10], only a fixed modification manner is simply utilized for the 2D histogram. Actually, there are many ways for 2D histogram modification, and one can take the best one to optimize the embedding performance. Obviously, any 2D mapping defined in $\mathbb{Z}^2$ can derive a reversible embedding if it satisfies the following reversible condition: for any $x \in \mathbb{Z}^2$, there exists unique $y \in \mathbb{Z}^2$ such that $x \in f(y)$. For clarity, the 2D mappings considered in the following context are the ones satisfying the reversible condition. Based on this consideration, we then propose to use adaptive 2D mapping for performance enhancement.

We will exhaustively search all the 2D mappings and find the optimal one such that it can provide the required embedding capacity while the embedding distortion is minimized. Considering that the generated 2D histogram is usually symmetric for the four quadrants, we then suppose that the 2D mapping is symmetric as well. More specifically, we only consider the 2D mapping $f : \mathbb{Z}^2 \mapsto \mathscr{P}(\mathbb{Z}^2)$ which satisfies the following condition: for each $x \geq 0$ and $y \geq 0$, $f(x, y) = f(-x - 1, y) = f(x, -y - 1) = f(-x - 1, -y - 1)$. Moreover, as the generated 2D histogram is concentred on $(0, 0)$, we only modify the pairwise embedding in a small local region $[0, K] \times [0, K]$ to derive the 2D mappings. That is to say, besides the region $[0, K] \times [0, K]$, each tested 2D mapping coincides with the mapping of pairwise embedding in the first quadrant. Here, balancing the embedding performance and the running time cost, the parameter K is simply
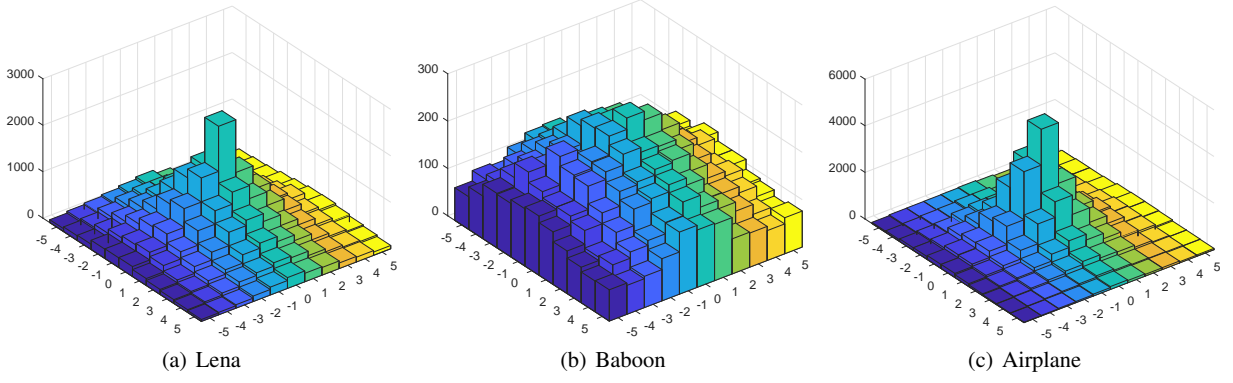
Fig. 3 Distribution of (d_{max}, d_{min}), for three standard 512×512 sized gray-scale images Lena, Baboon and Airplane, with block size of 2×2.
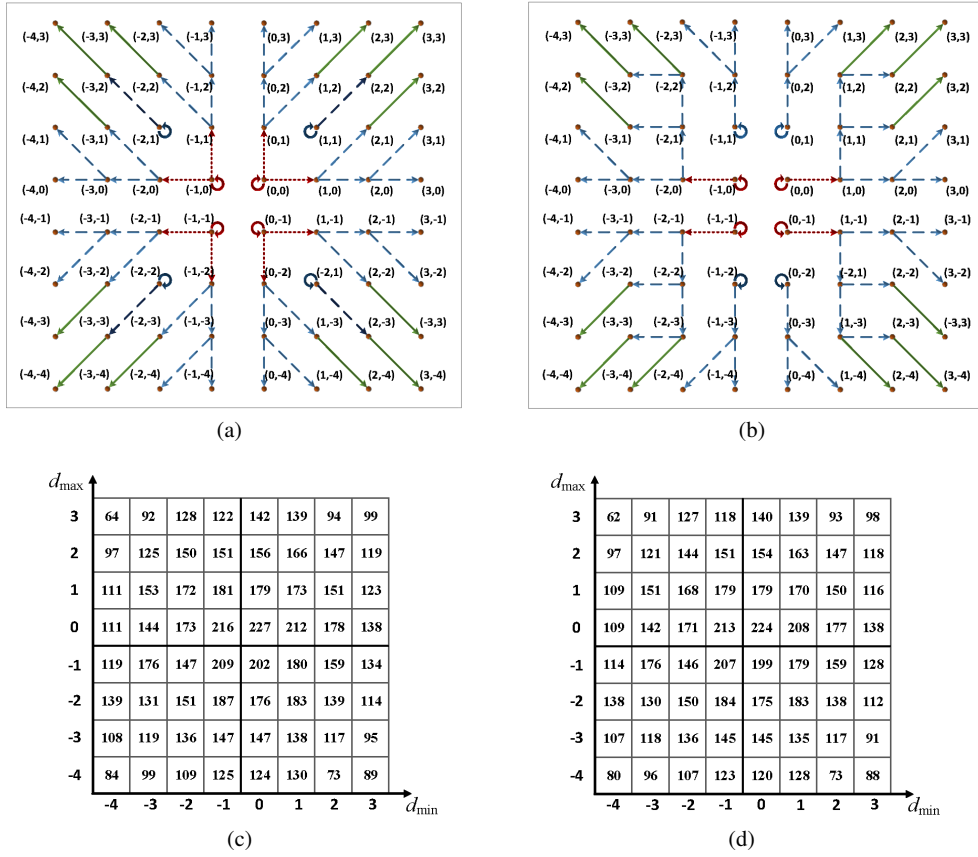

Fig. 5 (a) 2D mapping of the pairwise embedding [10], (b) Optimal 2D mapping obtained by the proposed method, (c) 2D histogram for the Baboon image with block size 2×2 and complexity threshold $T = 235$, (d) 2D histogram for the Baboon image with block size 2×2 and complexity threshold $T = 225$.

taken as 2. In this situation, we only need to test 1996 different 2D mappings in total. Then, for each 2D mapping, the same as the case of pairwise embedding, the complexity threshold T is determined as the smallest one such that the embedding capacity can be satisfied with the generated 2D histogram. After processing all the 2D mappings, one can finally get the optimal mapping with minimized distortion. For example, for the Baboon image with 2×2 sized

blocks and an embedding capacity of 10,000 bits, the optimal 2D mapping is shown in Figure 5(b), the corresponding complexity threshold $T = 225$, and the derived 2D histogram is shown in Figure 5(d). In this case, one can get a PSNR of 54.90 dB, and 0.27 dB of improvement is obtained compared with pairwise embedding. We see then, with optimal 2D mapping, the PVO-based method [26] is further improved.

Table 1 Optimal parameters of the proposed method for the embedding capacity of 10,000 bits.

Images	PSNR(dB)	n_1	n_2	T
Lena	61.09	4	2	67
Baboon	54.90	2	2	225
Airplane	63.87	3	2	24
Barbara	61.04	2	3	54
Lake	59.74	2	2	61
Boat	58.66	2	4	140
Peppers	59.63	3	3	107
Elaine	58.07	2	2	81

Table 2 Optimal parameters of the proposed method for the embedding capacity of 20,000 bits.

Images	PSNR(dB)	n_1	n_2	T
Lena	57.19	3	2	86
Airplane	59.98	2	2	25
Barbara	57.17	2	2	59
Lake	54.62	2	2	123
Boat	54.19	2	2	123
Peppers	55.40	3	2	143
Elaine	53.27	2	2	144

Finally, the same as the previous methods [25, 26], the above embedding procedure is implemented several times for different block size $n_1, n_2 \in \{2, 3, 4, 5\}$, and the best embedding result is taken as our final embedding result.

4 Experimental results

In this section, the proposed method is evaluated by comparing it with the original PVO-based method proposed by Li *et al.* [25], the improved PVO-based method proposed by Peng *et al.* [26], and a recently proposed PVO-based method of Ou *et al.* [27]. The proposed method is implemented on MatLab version 2018a with a personal laptop (ThinkPad X280), and the embedding process can be completed within five seconds for a cover image and a given embedding capacity.

Eight standard 512×512 sized gray-scale bitmap images including Lena, Baboon, Airplane, Barbara, Lake, Boat, Peppers and Elaine, are used in the experiment for evaluation. First, for a given embedding capacity of 10,000 bits, we shown in Table 1 and Table 2 the optimal parameters (including the block size (n_1, n_2) and the complexity threshold T) of the proposed method. We see that the parameters are rather different for different images. We may conclude that it is necessary to test different block size for PVO-based methods, and it is reasonable to take large sized blocks for low embedding capacity. Moreover, for a better illustration, the optimal 2D mappings for some test images with different embedding capacities, are shown in Figure 6, and the adaptivity can be observed from this figure. For brevity, only the 2D mapping in the first quadrant is plotted in Figure 6. Figure 7 shows the embedding performance presented by the capacity-distortion curves. It can be seen that the proposed method is superior to the compared methods with a higher PSNR whatever the cover image or the embedding capacity is.

For the original PVO-based method [25], as we have mentioned, the smooth blocks are not exploited for data embedding. As a result, this method is less efficient compared with the other tested methods. According to Table 3 and Table 4, for an embedding capacity of 10,000 and 20,000

Table 3 Comparison of PSNR (dB) among the proposed method, and the PVO-based methods of Li *et al.* [25], Peng *et al.* [26], Ou *et al.* [27]. The embedding capacity is 10,000 bits.

Images	Li *et al.*	Peng *et al.*	Ou *et al.*	Proposed
Lena	60.34	60.49	60.59	**61.09**
Baboon	53.46	53.58	54.48	**54.90**
Airplane	62.10	62.97	63.29	**63.87**
Barbara	60.39	60.48	60.59	**61.04**
Lake	58.15	58.81	59.36	**59.74**
Boat	58.05	58.26	58.23	**58.66**
Peppers	58.87	58.97	59.18	**59.63**
Elaine	56.84	57.37	57.37	**58.07**
Average	58.52	58.86	59.13	**59.63**

bits, compared with [25], our method can get better embedding result with an increase of PSNR of 1.11 dB and 1.18 dB in average, respectively. Peng *et al.*'s method [26] is an improvement of [25]. Although efficient, its embedding performance is still unsatisfactory with independent modification for $d_{\max}$ and $d_{\min}$. According to Table 3 and Table 4, compared with this improeved PVO-based method [26], our method can get respectively an increase of PSNR of 0.77 dB and 0.74 dB in average, for an embedding capacity of 10,000 and 20,000 bits. The advantage of the proposed method mainly lies in the utilization of 2D histogram and its adaptive modification.

The so-called PVO-k method [27] is another improvement of the original PVO-based method. For this method, if more than one pixels taking the largest value in a given block, all these pixels will be simultaneously modified in a same way for data embedding. This method is verified better than the original PVO-based method [25], however, the smooth blocks are still not utilized for data embedding (e.g., the block whose pixel values are identical). According to Table 3 and Table 4, for an embedding capacity of 10,000 and 20,000 bits, compared with [27], our method can get better embedding result with an increase of PSNR of 0.50 dB and 0.53 dB in average, respectively.
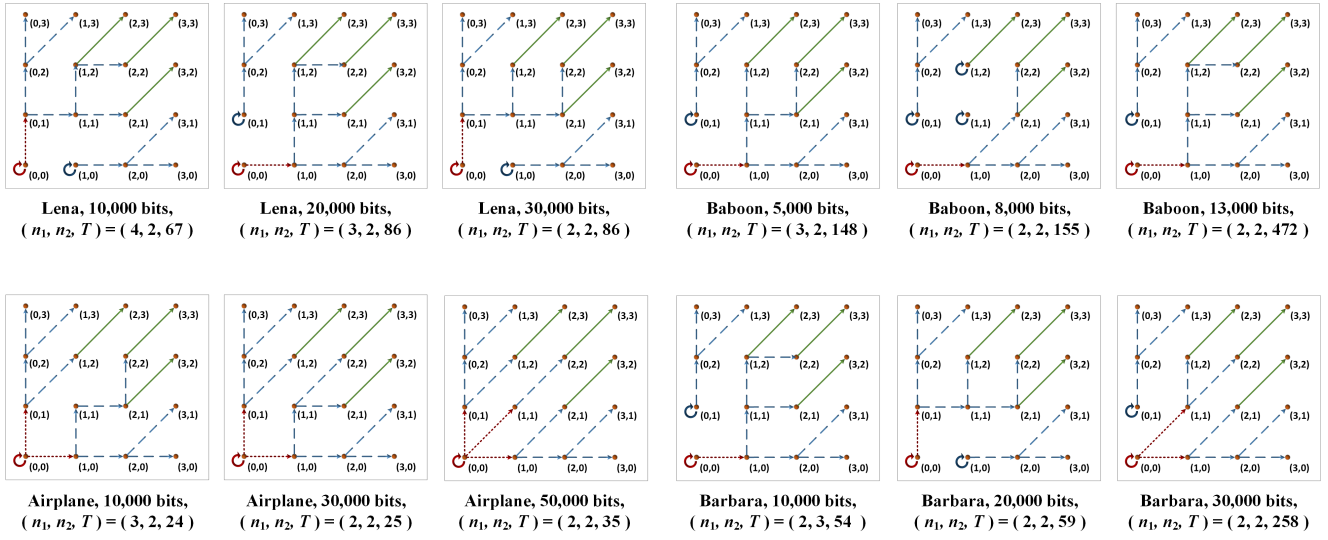
Fig. 6 Optimal 2D mappings for some test images with different embedding capacities.

Table 4 Comparison of PSNR (dB) among the proposed method, and the PVO-based methods of Li *et al.* [25], Peng *et al.* [26], Ou *et al.* [27]. The embedding capacity is 20,000 bits.

Images	Li *et al.*	Peng *et al.*	Ou *et al.*	Proposed
Lena	56.20	56.56	56.58	**57.19**
Airplane	58.20	59.07	59.33	**59.98**
Barbara	55.43	56.20	56.50	**57.17**
Lake	53.39	53.53	54.29	**54.62**
Boat	53.29	53.83	53.76	**54.19**
Peppers	54.66	54.77	54.93	**55.40**
Elaine	52.36	52.65	52.71	**53.27**
Average	54.79	55.23	55.44	**55.97**

5 Conclusion

Based on PVO and adaptive pairwise modification, an improved RDH scheme is proposed in this paper. After dividing the cover image into non-overlapping equal-sized blocks, unlike previous PVO-based embedding methods [25–27], the prediction-errors for the largest and smallest pixel values of each block are jointed as a pair. Then, the secret data is embedded into the cover image by adaptively modifying the 2D histogram which is generated based on the prediction-error pairs of smooth image blocks. The proposed method is experimentally verified better than the previous works [25–27]. One possible direction of PVO-based RDH approach is to take more pixels in each block for data embedding, to enhance the embedding capacity and to further improve the embedding performance.

References

1. Tian, J.: Reversible data embedding using a difference expansion. IEEE Trans. Circuits Syst. Video Technol. **13**(8), 890–896 (2003)
2. Ni, Z., Shi, Y.Q., Ansari, N., Su, W.: Reversible data hiding. IEEE Trans. Circuits Syst. Video Technol. **16**(3), 354–362 (2006)
3. Thodi, D.M., Rodriguez, J.J.: Expansion Embedding Techniques for Reversible Watermarking. IEEE Trans. Image Process. **16**(3), 721–730 (2007)
4. Hong, W., Chen, T.S., Shiu, C.W.: Reversible data hiding for high quality images using modification of prediction errors. J. Syst. Softw. **82**(11), 1833-1842 (2009)
5. Sachnev, V., Kim, H.J., Nam, J., Suresh, S., Shi, Y.Q.: Reversible Watermarking Algorithm Using Sorting and Prediction. IEEE Trans. Circuits Syst. Video Technol. **19**(7), 989–999 (2009)
6. Coatrieux, G., Le Guillou, C., Cauvin, J.M., Roux, C.: Reversible watermarking for knowledge digest embedding and reliability control in medical images. IEEE Trans. Inf. Technol. Biomed. **13**(2), 158–165 (2009)
7. Li, X., Yang, B., Zeng, T.: Efficient Reversible Watermarking Based on Adaptive Prediction-Error Expansion and Pixel Selection. IEEE Trans. Image Process. **20**(12), 3524–3533 (11)
8. Coltuc, D.: Improved Embedding for Prediction-Based Reversible Watermarking. IEEE Trans. Inf. Forens. Secur. **6**(3), 873–882 (2011)
9. Hong, W.: Adaptive reversible data hiding method based on error energy control and histogram shifting. Optics Communications. **285**(2), 101–108 (2011)
10. Ou, B., Li, X., Zhao, Y., Ni, R., Shi, Y.Q.: Pairwise prediction-error expansion for efficient reversible data hiding. IEEE Trans. Circuits Syst. Video Technol. **22**(12), 5010–5021 (2013)
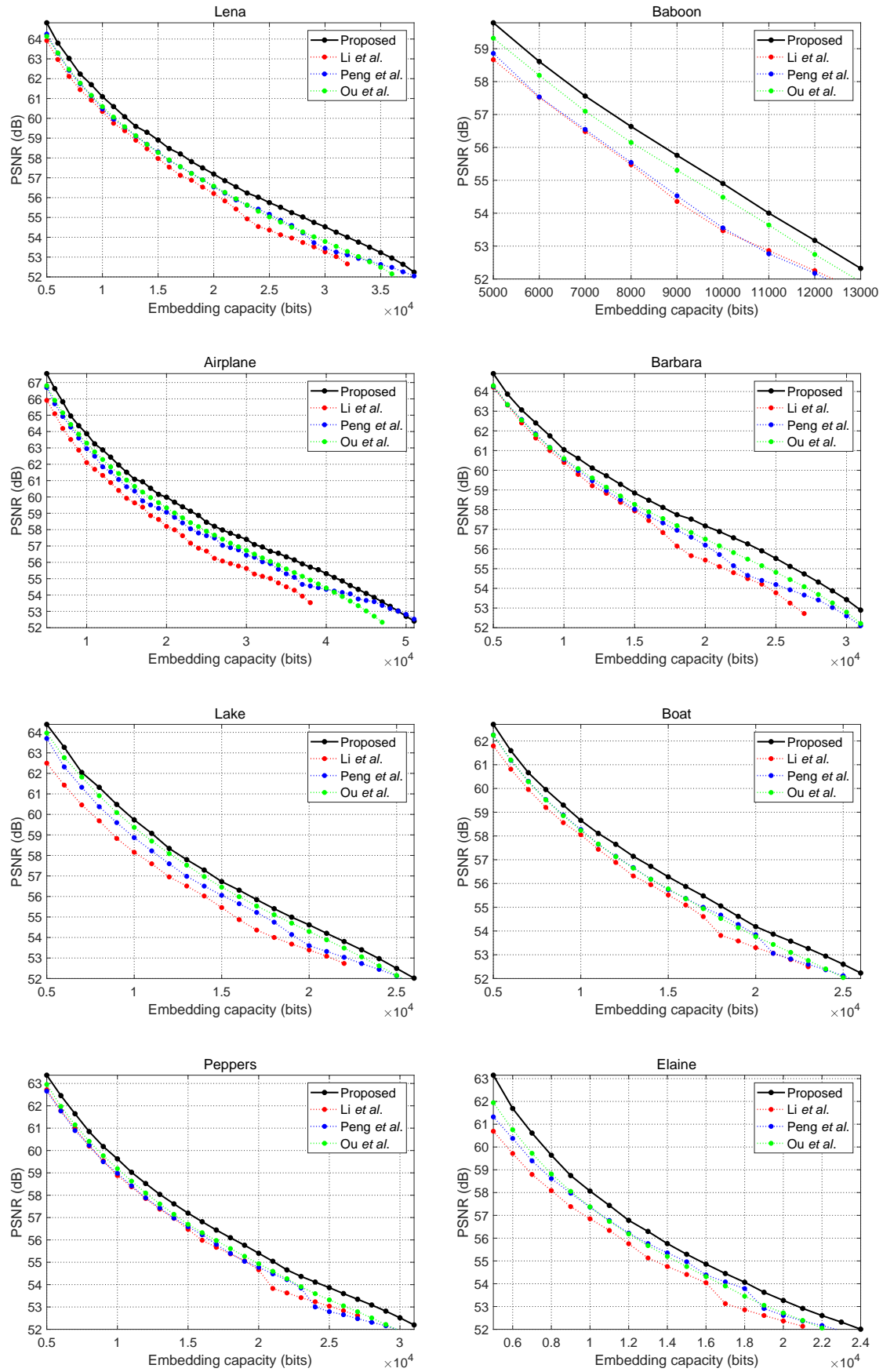11. Li, X., Zhang, W., Gui, X., Bin Y.: A Novel Reversible Data Hiding Scheme Based on Two-Dimensional

Fig. 7 Performance comparison between the proposed method and Li *et al.*'s method [25], Peng *et al.*'s method [26], and Ou *et al.*'s method [27].

Difference-Histogram Modification. IEEE Trans. Circuits Syst. Video Technol. **8**(7), 1091–1100 (2013)

12. Qin, C., Chang, C.C., Huang, Y.H., Liao, L.T.: An Inpainting-Assisted Reversible Steganographic Scheme Using a Histogram Shifting Mechanism. IEEE Trans. Circuits Syst. Video Technol. **23**(7), 1109–1118 (2013)

13. Li, X., Li, B., Yang, B., Zeng, T.: General Framework to Histogram-Shifting-Based Reversible Data Hiding. IEEE Trans. Image Process. **22**(6), 2181–2191 (2013)

14. Xuan, G., Tong, X., Teng, J., Zhang, X., Shi, Y.Q.: Optimal histogram-pair and prediction-error based image reversible data hiding. in Proc. IWDW, pp. 368–383. Springer(2013)

15. Dragoi, I.C., Coltuc, D.: Local-prediction-based difference expansion reversible watermarking. IEEE Trans. Image Process. **23**(4), 1779–1790 (2014)

16. Qian, Z., Zhang, X., Wang, S.: Reversible data hiding in encrypted JPEG bitstream. IEEE Trans. Multimedia **16**(5), 1486–1491 (2014)

17. Li, X., Zhang, W., Gui, X., Yang, B.: Efficient Reversible Data Hiding Based on Multiple Histograms Modification. IEEE Trans. Inf. Forens. Secur. **19**(9), 2016–2027 (2015)

18. Pan, Z., Hu, S., Ma, X., Wang, L.: Reversible data hiding based on local histogram shifting with multilayer embedding. J. Vis. Commun. Image Represent. **31**, 64–74 (2015)

19. Dragoi, I.C., Coltuc, D.: On local prediction based reversible watermarking. IEEE Trans. Image Process. **24**(4), 1244–1246 (2015)

20. Dragoi, I.C., Coltuc, D.: Adaptive pairing reversible watermarking. IEEE Trans. Image Process. **25**(5), 2420–2422 (2016)

21. Shi, Y.Q., Li, X., Zhang, X., Wu, H.T., Ma, B.: Reversible data hiding: advances in the past two decades. IEEE Access. **4**, 3210–3237 (2016)

22. Wang, J., Ni, J., Zhang, X., Shi, Y.Q.: Rate and distortion optimization for reversible data hiding using multiple histogram shifting. IEEE Trans. Cybern. **47**(2), 315–326 (2017)

23. Kim, D., Yoon, E., Kim, C., Yoo, K.: Reversible data hiding scheme with edge-direction predictor and modulo operation. J. Real Time Image Process. **14**(1), 137–145 (2018)

24. Jung, K.: High-capacity reversible data hiding method using block expansion in digital images. J. Real Time Image Process. **14**(1), 159–170 (2018)

25. Li, X., Li, J., Li, B., Yang, B.: High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion. Signal Process. **93**(1), 198–205 (2013)

26. Peng, F., Li, X., Yang, B.: Improved PVO-based reversible data hiding. Digit. Signal Process. **25**(2), 255–265 (2014)

27. Ou, B., Li, X., Zhao, Y., Ni, R.: Reversible data hiding using invariant pixel-value-ordering and prediction-error expansion. Signal Process.: Image Commun. **29**(7), 760–772 (2014)

28. Qu, X., Kim, H.J.: Pixel-based pixel value ordering predictor for high-fidelity reversible data hiding. Signal Process. **111**, 249–260 (2015)

29. Wang, X., Ding, J., Pei, Q.: A novel reversible image data hiding scheme based on pixel value ordering and dynamic pixel block partition. Information sciences. **310**, 16–35 (2015)

30. Ou, B., Li, X., Wang, J.: High-fidelity reversible data hiding based on pixel-value-ordering and pairwise prediction-error expansion. J. Vis. Commun. Image Represent. **39**, 12–23 (2016)

31. He, W., Xiong, G., Weng, S., Cai, Z., Wang, Y.: Reversible data hiding using multi-pass pixel-value-ordering and pairwise prediction-error expansion. Information Sciences. **49**, (2018)

32. Dragoi, I.C., Caciula I., Coltuc D.: Improved Pairwise Pixel-Value-Ordering for High-Fidelity Reversible Data Hiding. in Proc. IEEE ICIP, pp. 1668–1672 (2018)