Improved PPVO-based high-fidelity reversible data hiding

Haorui Wu, Xiaolong Li, Yao Zhao*, Rongrong Ni

[a] *Institute of Information Science, Beijing Jiaotong University, Beijing 100044, China*
[b] *Beijing Key Laboratory of Advanced Information Science and Network Technology, Beijing 100044, China*

Abstract

Most pixel-value-ordering (PVO) based reversible data hiding (RDH) methods conduct pixel prediction on image blocks, while a recently proposed pixel-based PVO (PPVO) method changes the blockwise prediction to pixelwise manner and achieves better embedding performance. However, in PPVO, the prediction result is not accurate since some pixels close to the to-be-predicted one are not utilized for prediction. Moreover, pixels with different complexity are not fully exploited and the embedding performance of PPVO is not optimized. Then, to better determine pixel context for prediction as well as make full use of image local correlation, an improved PPVO-based RDH method is proposed in this paper. First, to improve the prediction accuracy, a new prediction strategy is proposed in which the pixel context is properly selected. Then, a new embedding strategy is proposed for performance optimization based on multiple histograms generation and modification with multi-sized prediction contexts. Experimental results verify that the proposed method is superior to PPVO and some other PVO-based RDH methods.

Keywords: Reversible data hiding, pixel-value-ordering, pixelwise prediction, multiple histograms generation and modification

1. Introduction

Information hiding has received extensive attention nowadays [1, 2]. Among existing information hiding approaches, reversible data hiding (RDH) can exactly extract the embedded data and losslessly recover the cover image as well, which is useful for the information hiding of sensitive images [3].

Up to now, many RDH methods have been proposed, such as difference expansion (DE) [4–8], histogram shifting (HS) [9–12], prediction-error expansion (PEE) [5, 13–23] and so on. The first DE-based method is proposed by Tian [4]. By DE, the pixel difference is expanded for reversible embedding. For HS-based RDH, it is first proposed by Ni *et al.* [24], and the data embedding is conducted by modifying the image intensity histogram. Later on, the technique prediction-error expansion (PEE) is proposed by Thodi and Rodriguez [6], in which the image pixels are first predicted and then the resulting prediction-errors are expanded or shifted for reversible embedding. Compared with DE and HS, PEE is the most effective RDH approach nowadays due to its fine balance between the embedding capacity and distortion. After that, many improvements for PEE focusing on pixel prediction and prediction-error histogram (PEH) modification

*Corresponding author. Tel./Fax: +86 10 51688667.
Email addresses: hrwu@bjtu.edu.cn (Haorui Wu), lixl@bjtu.edu.cn (Xiaolong Li), yzhao@bjtu.edu.cn (Yao Zhao), rrni@bjtu.edu.cn (Rongrong Ni)

are proposed, e.g., accurate predictor design [6, 7, 9, 13, 25–27], adaptive histogram modification [], high dimensional histogram modification [8, 19, 22], and multiple histograms modification [21, 28, 29], etc.

Recently, a novel improvement of PEE taking pixel-value-ordering (PVO) to provide accurate prediction has been proposed [28–37]. The original PVO-based method is proposed by Li *et al.* [30]. In this method, the cover image is first divided into equal-sized non-overlapping blocks and the secret data is embedded into the largest/smallest pixel of each block via modifying its prediction-errors. After that, Peng *et al.* proposed an improved PVO-based method in [31] by utilizing more smooth blocks for reversible embedding. In [32], Ou *et al.* proposed another PVO-based method called PVO-k which modifies the first k largest/smallest pixel values in each block for performance enhancement. Later on, Ou *et al.* [29] and He *et al.* [36] proposed a way to better explore the pixel relations in each block for PVO. In these methods, two prediction-errors in a block are paired together to generate a two-dimensional PEH and proper histogram modification manners are designed for reversible embedding. Besides, Wang *et al.* [28] and Weng *et al.* [34] proposed to dynamically determine the block size to better use pixels in smooth regions. Moreover, unlike the blockwise prediction used in previous PVO-based methods, Qu *et al.* proposed the so-called pixel-based PVO (PPVO) in [33] by utilizing pixelwise prediction. In this method, each pixel is predicted by the largest/smallest pixel value of its context, and high embedding capacity is achieved. PPVO is an effective improvement of PVO, and it is experimentally verified better than some PVO-based RDH methods with both higher embedding capacity and better marked image quality. However, for the prediction in PPVO, some pixels close to the to-be-predicted one are not involved so that the prediction is not accurate. This drawback limits its embedding performance. On the other hand, since all pixels are predicted with identical context without considering the different local complexity, the embedding performance of PPVO is not optimized.

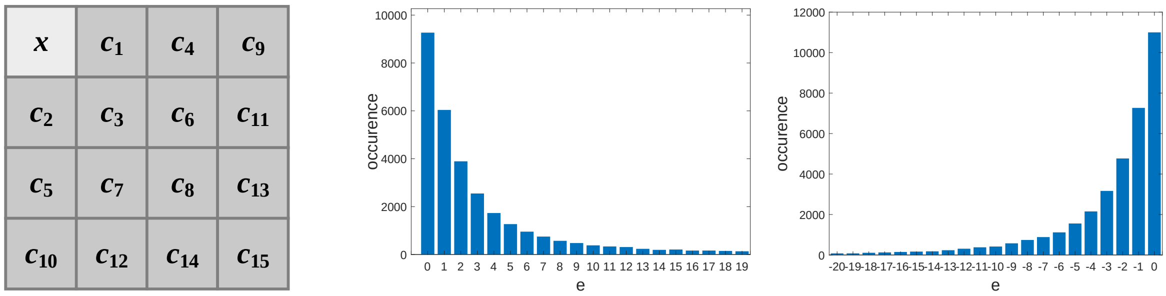
In this paper, based on the above considerations, a novel RDH method is proposed to improve PPVO [33]. Unlike PPVO, some pixels close to the to-be-predicted one are taken as a part of context for prediction, contributing to the prediction accuracy enhancement. In this way, shaper PEH can be generated which is more suitable for reversible embedding. Moreover, to better exploit image redundancy and for performance optimization of PPVO, a new embedding strategy is proposed by conducting multiple histograms generation and modification with multi-sized prediction contexts. By this strategy, pixels with different local complexity are predicted by different sized context. Then, multiple PEHs are generated according to the employed different pixel context and then modified for data embedding. Experimental results verify that the proposed method outperforms PPVO [33], and it is superior to some other PVO-based RDH methods [31, 32] as well.

The rest of the paper is organized as follows. Section 2 presents a brief review of PPVO [33]. The details of the proposed method improving the original PPVO is introduced in Section 3. Then, the experimental results are given in Section 4. Finally, Section 5 concludes the paper.

2. Related Work

In this section, as a prior knowledge, the PPVO method of Qu *et al.* [33] is briefly reviewed. Unlike the previous PVO-based methods such as [30] and [31], PPVO conducts prediction in a pixelwise manner. Specific data embedding and extraction procedures are presented as follows.

First of all, to avoid the overflow and underflow problem, pixels valued 0 are modified to 1 and pixels valued 255 are modified to 254. Then, for a target pixel x, its prediction context is defined by its neighboring pixels in the down right region, as shown in Fig. 1(a). More specifically, a prediction context $C = \{c_1, ..., c_n\}$



(a) Context of x with $n = 15$. (b) PEH for $x \in S_1 \cup S_4$. (c) PEH for $x \in S_2 \cup S_3$.

Figure 1: Prediction context and the PEHs for the image Lena with pixels in $S_1 \cup S_4$ and $S_2 \cup S_3$, for PPVO [33].

is formed, where n is the number of context pixels. Next, sort the context pixels, and four cases may occur for the prediction of x. That is to say, cover pixels used for data embedding are grouped into four sets including

$$\begin{aligned} S_1 &= \{x | c_{\max} \neq c_{\min}, x \geq c_{\max}\} \\ S_2 &= \{x | c_{\max} \neq c_{\min}, x \leq c_{\min}\} \\ S_3 &= \{x | c_{\max} = c_{\min}, x \leq c_{\min}, c_{\min} \neq 254\} \\ S_4 &= \{x | c_{\max} = c_{\min}, x \geq c_{\max}, c_{\max} = 254\} \end{aligned} \tag{1}$$

where $c_{\max}$ and $c_{\min}$ represent the largest and smallest pixel values in C, respectively, i.e., $c_{\max} = \max\{c_1, ..., c_n\}$ and $c_{\min} = \min\{c_1, ..., c_n\}$. Notice that, pixels outside the four sets are skipped and not utilized for embedding. In this way, x is predicted as $\widehat{x}$ by

$$\widehat{x} = \begin{cases} c_{\max}, & \text{if } x \in S_1 \cup S_4 \\ c_{\min}, & \text{if } x \in S_2 \cup S_3 \end{cases}. \tag{2}$$

Then, the prediction-error is defined as $e = x - \widehat{x}$. Accordingly, $e \geq 0$ if $x \in S_1 \cup S_4$, and $e \leq 0$ if $x \in S_2 \cup S_3$. Here, the standard 512×512 sized gray-scale image Lena is taken for example, and the corresponding PEHs counting the frequency of e are shown in Fig. 1(b) and (c), for pixels in $S_1 \cup S_4$ and $S_2 \cup S_3$, respectively. The highest frequency of both two PEHs locates at bin 0. Therefore, for each of the two PEHs, the bin 0 is expanded to embed data, and other bins are shifted to create vacancies. That is to say, the prediction-error e is modified to $\widetilde{e}$ as follows for data embedding

$$\widetilde{e} = \begin{cases} e + b, & \text{if } x \in S_1 \cup S_4 \text{ and } e = 0 \\ e + 1, & \text{if } x \in S_1 \cup S_4 \text{ and } e > 0 \\ e - b, & \text{if } x \in S_2 \cup S_3 \text{ and } e = 0 \\ e - 1, & \text{if } x \in S_2 \cup S_3 \text{ and } e < 0 \end{cases} \tag{3}$$

where $b \in \{0, 1\}$ is a data bit to be embedded. Finally, the marked pixel is determined as $\widetilde{x} = \widehat{x} + \widetilde{e}$.

The embedding process of PPVO for a cover image is performed in the raster-scan order and the extraction process is implemented in reverse order. Thus, the prediction context of each cover pixel and its corresponding marked pixel are the same. Moreover, in the embedding process, for $x \in S_1 \cup S_4$, it is either increased by 1 or unchanged, and for $x \in S_2 \cup S_3$, it is either decreased by 1 or unchanged, the marked pixel $\widetilde{x}$ can be classified into the same group as its cover x according to (1). Then, the extraction and recovery process can be conducted as follows. First, a marked pixel $\widetilde{x}$ is predicted as

$$\widehat{x} = \begin{cases} c_{\max}, & \text{if } \widetilde{x} \in S_1 \cup S_4 \\ c_{\min}, & \text{if } \widetilde{x} \in S_2 \cup S_3 \end{cases}. \tag{4}$$

Then, the marked prediction-error is computed by $\widetilde{e} = \widetilde{x} - \widehat{x}$. Next, the original prediction-error can be determined as

$$e = \begin{cases} \widetilde{e}, & \text{if } e = 0 \\ \widetilde{e} - 1, & \text{if } e > 0 \\ \widetilde{e} + 1, & \text{if } e < 0 \end{cases}. \tag{5}$$

Meanwhile, the embedded data bit can be extracted as $b = 0$ if $\widetilde{e} = 0$ and $b = 1$ if $\widetilde{e} \in \{-1, 1\}$. Finally, the original pixel is recovered by $x = \widehat{x} + e$.

Besides, for a given embedding capacity, in the embedding process, the proper number of context pixels is exhaustively determined by testing n from 1 to 15, and the one generating marked image with the highest PSNR will be selected. In addition, to better exploit the image redundancy, a pixel selection strategy with a threshold T is employed to improve the embedding performance. Pixels whose local complexity is smaller than T, indicating that it is easier to be accurately predicted, will be used for data embedding. In detail, the complexity of each pixel is defined as the difference of the maximum and the minimum of its 15 context pixels, as illustrated in Fig. 1(a). Experimental results show that PPVO provides larger embedding capacity than the previous PVO-based methods [30, 31] due to the pixelwise prediction. At the same time, sharper PEHs which contribute to better embedding performance are obtained by this method.

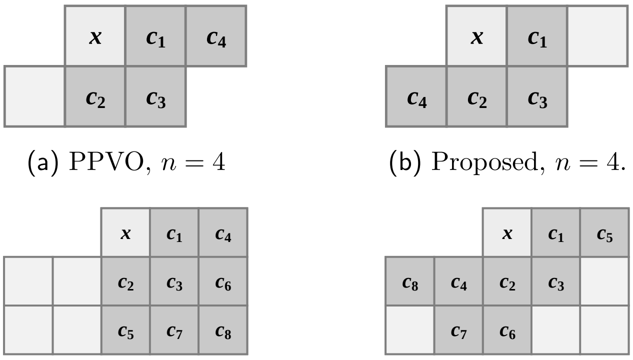
However, although efficient, we argue that PPVO can be further improved from two aspects. On one hand, since the prediction context of a given pixel is chosen only from its down right region (see Fig. 1(a)), pixels in the down left region which are closer to the target pixel are not exploited. Actually, we will see later that the pixels in the down left region are useful for accurate prediction. On the other hand, all the pixels with different local complexity are predicted utilizing identical context without taking their difference into consideration. Therefore, to better choose the prediction context as well as fully exploit image redundancy considering different local complexity, we propose an improved PPVO-based RDH method, and the details is given later in the next section.

3. Proposed Method

In this section, an improved PPVO-based RDH method is presented. First, a new predictor improving the one utilized in PPVO [33] is introduced. Then, a new embedding strategy is proposed for performance optimization of PPVO based on multiple histograms generation and modification utilizing multi-sized prediction contexts. Finally, the embedding and extraction procedures are summarized.

3.1. Extended predictor

The prediction in PPVO is conducted in a pixelwise manner. Specifically, for a given n, each cover pixel is predicted by its n context pixels in the down right direction. Take $n = 4$ and $n = 8$ for example, the context pixels of x used in the prediction of PPVO are shown in Fig. 2(a) and (c), respectively. Obviously, some pixels closing to x which are more suitable for accurate prediction are not utilized. For example, compared to the pixel c_4 in Fig. 2(a), pixel c_4 in Fig. 2(b) is closer to x and it is actually not utilized for



(a) PPVO, $n = 4$

(b) Proposed, $n = 4$.

(c) PPVO, $n = 8$

(d) Proposed, $n = 8$.

Figure 2: Prediction context of x, for PPVO and the proposed predictor with different number n of context pixels.

the prediction of x in PPVO. Similarly, in the case of $n = 8$, compared with the prediction context used in PPVO shown in Fig. 2(c), a better way is taking the prediction context as the one shown in Fig. 2(d) to full utilize the neighboring pixels. Then, to achieve accurate prediction, an extended predictor using refined prediction context is proposed.

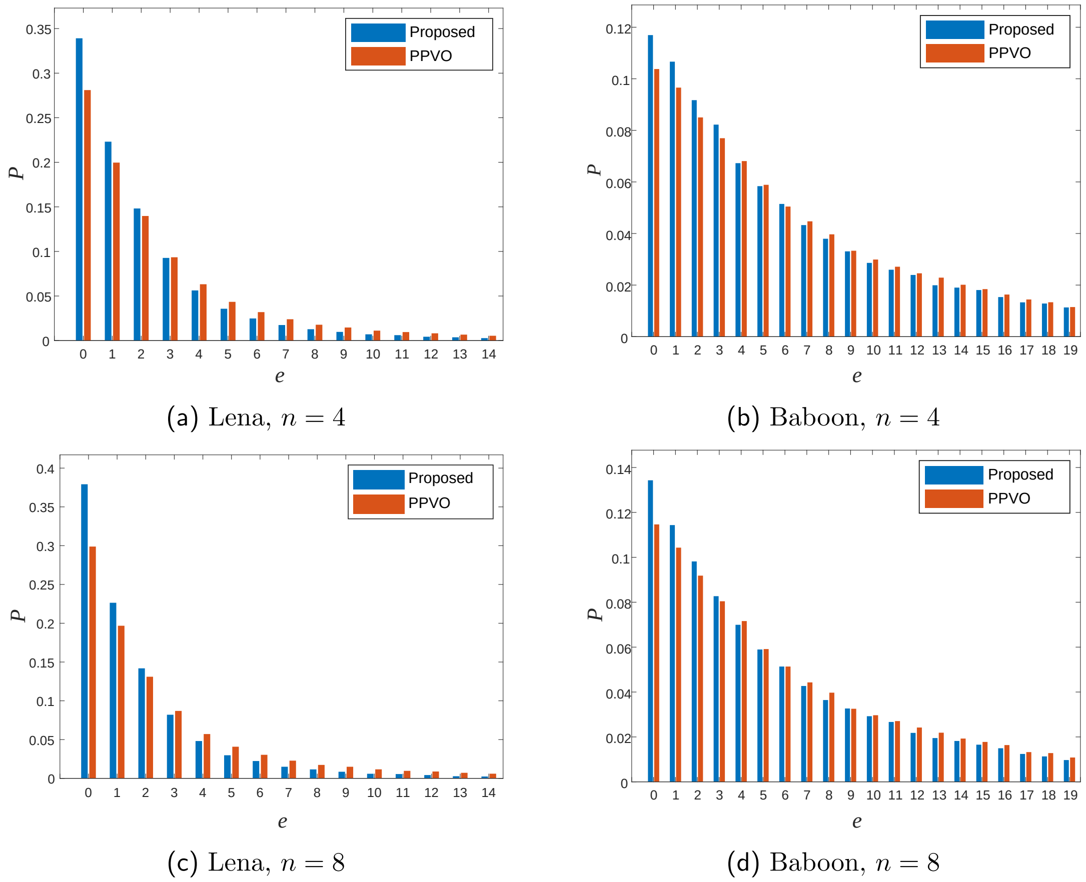
An experiment is conducted to verify the effectiveness of the proposed predictor. The standard 512×512 sized gray-scale images Lena and Baboon are used here. Fig. 3 shows the comparison between the normalized PEHs derived from PPVO and the proposed predictor. Two different prediction contexts with $n = 4$ and $n = 8$ are compared. Specifically, for PPVO, the prediction contexts are the ones shown in Fig. 2(a) and (c). For the proposed predictor, we use the contexts shown in Fig. 2(b) and (d) for prediction. Notice that, only the PEHs based on the pixels in $S_1 \cup S_4$ are shown in Fig. 3 for simplicity. According to these figures, compared with the PEH derived from PPVO, the one based on the proposed predictor with blue color has higher peak at bin 0 in each case, and the proposed PEHs are sharper. Moreover, the proportion of shifted pixels defined in [30] is used to measure the statistical characteristics of PEH in a quantitative way, and it is calculated by

$$P = \frac{\#\{\text{shifted pixels}\}}{\#\{\text{expanded or shifted pixels}\}} = \frac{\sum_{i>0} h(i)}{h(0) + \sum_{i>0} h(i)} \tag{6}$$

where h represents the generated PEH. Obviously, the smaller the value of P, the sharper the PEH and hence the better the embedding performance. In Fig. 3(a) and (c), for image Lena, when $n = 4$, the values of P corresponding to PPVO and the proposed predictor are 0.72 and 0.66, respectively. When $n = 8$, P equals 0.7 and 0.62 with these two predictors. It is obvious that, the values of P are decreased based on the proposed predictor, indicating that the embedding distortion can be decreased as well. Besides, similar results also can be observed on the PEHs of image Baboon. In summary, the proposed predictor provides more accurate prediction results, and hence the embedding performance can be improved.

3.2. Multi-size based Embedding Method for MHM

As mentioned above, in PPVO [33], the number of context pixels used for the prediction of each pixel is fixed, and the value of n is experimentally chosen from 1 to 15. However, different number of context pixels can lead to different embedding performance. In order to analyze the impact of different n on the embedding performance, we set $n \leq 24$ and choose four different context vectors for experiments. Specifically,



(a) Lena, $n = 4$

(b) Baboon, $n = 4$

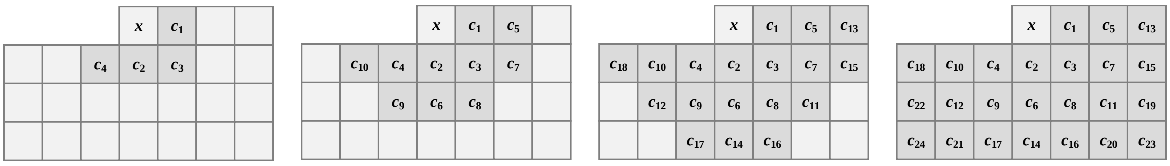
(c) Lena, $n = 8$

(d) Baboon, $n = 8$

Figure 3: Comparison of normalized PEHs derived from PPVO [33] and the proposed predictor, for images Lena and Baboon with different prediction contexts shown in Fig. 2.

the context vectors are $C_1 = \{c_1, ..., c_4\}$, $C_2 = \{c_1, ..., c_{10}\}$, $C_3 = \{c_1, ..., c_{18}\}$ and $C_4 = \{c_1, ..., c_{24}\}$, as shown in Fig. 4. Specific analysis is conducted based on the threshold-capacity curves and the curves about the threshold with the proportion of shifted pixels. Fig. 5 shows the comparison results between the context vectors C_1 and C_2. Here, the threshold means the local complexity threshold used for pixel selection, and only pixels whose local complexity is smaller than the threshold are used for embedding. Note that, for a target pixel, the local complexity, denoted as NL, is calculated by the sum of the absolute values of the differences between each pair of adjacent context pixels in the horizontal and vertical directions. In this experiment, the context pixel region used for NL calculation is shown in Fig. 4(b). Besides, the proportion of shifted pixels is exactly the value of P defined in (6). And, the embedding capacity is equal to the number of prediction-errors valued 0. According to the comparison results shown in Fig. 5, several observations are given as follows.

1. In Fig. 5(a), for a fixed threshold, the maximum embedding capacity derived based on C_1 is much larger than that based on C_2. The main reason is that, when n is small, almost all the cover pixels are used for embedding while part of the pixels are skipped with large value of n based on C_2.
2. In Fig. 5(a), the embedding capacity is dramatically increased with the increase of the threshold at the beginning, especially for the curve corresponding to C_1, indicating that pixels belong to smooth



(a) C_1 of $n = 4$ (b) C_2 of $n = 10$ (c) C_3 of $n = 18$ (d) C_4 of $n = 24$

Figure 4: Context pixels of x by Extended PPVO.

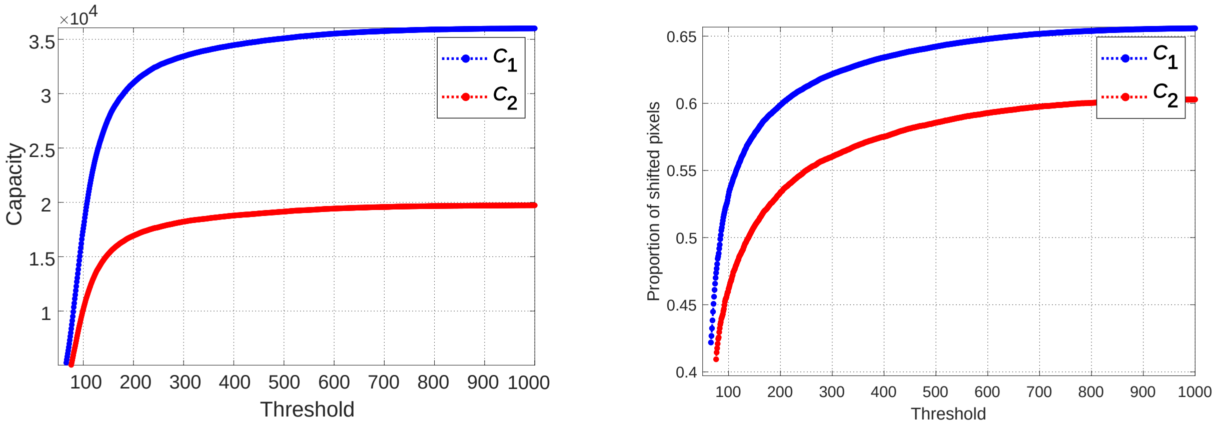
areas are accurately predicted and they are more suitable for data embedding.

3. According to Fig. 5(a) and (b), for a fixed threshold, although the embedding capacity corresponding to C_2 is smaller than C_1, the proportion of shifted pixels corresponding to C_2 is much smaller than that of C_1. In other words, the skipped pixels in the case of C_2 are difficult to be accurately predicted. Thus, they make little contribution on the improvement of the embedding performance.

In a word, large embedding capacity is obtained by using small number of context pixels, but corresponding distortion is also increased due to the inaccurate prediction. While, when a large number of context pixels are used, lots of cover pixels are skipped and the distortion caused by shifting is decreased. However, the embedding capacity is decreased at the same time. Therefore, to make use of the advantages of both sizes of context pixels, a multi-size context pixels based embedding strategy is developed in the proposed method.

Now, the proposed strategy is introduced in detail. First, cover pixels are divided into three classes through two thresholds $0 \leq t_1 \leq t_2$. For simple description, the first set consists of pixels whose local complexity is smaller than t_1, and it is denoted as X_1. The second set consists of pixels whose local complexity is bigger or equal to t_1 and smaller than t_2, and it is denoted as X_2. Then, since pixels belong to X_1 are smoother, each of them is predicted based on four context pixels according to C_1. And pixels belong to X_2 are predicted by context pixels according to C_2. As for pixels belong to the third class, they are skipped and not used for embedding. This strategy is reasonable for the following reason. As shown in Fig. 5(b), the gap between these two curves are growing with the increase of the threshold. That is to say, in complex areas, pixels that are difficult to accurately predicted are skipped by using context pixels according to C_2. Therefore, we tend to take context pixels based on C_2 for the prediction of pixels belong to X_2 to decrease the embedding distortion. At the same time, we take context pixels based on C_1 to predict pixels belong to X_1 to increase the embedding capacity.

A simple example to illustrate the embedding process for pixels belong to three different classes is shown in Fig. 7. The thresholds used here are $(t_1, t_2) = (15, 25)$. As shown in Fig. 7(a), the cover pixel is $x = 150$, and its local complexity is $\text{NL} = 10$ which is smaller than $t_1 = 15$. Thus, it is classified into the first class, and predicted by the context pixels $C = \{148, 150, 150, 148\}$. According to the prediction method described in Section 2, the prediction-error of x is $e = 0$. Then, without loss of generality, a data bit $b = 1$ can be embedded, and the marked pixel equals 151. Note that, if the context pixels are chosen based on C_2 in this case, this pixel will be skipped due to $C_{\min} < x < C_{\max}$. For the second case in Fig. 7(b), the cover pixel is $x = 135$, and its local complexity is NL= 24. Since $(t_1, t_2) = (15, 25)$, it is classified to the second class. Thus, the context pixels $C = \{137, 136, 139, 138, 140, 135, 142, 137, 136, 139\}$ are used for prediction, and the prediction-error is $e = 0$. Then, a data bit $b = 0$ is embedded into this pixel, and the marked pixel remains 135. In this case, if context pixels is chosen based on C_1, the prediction-error will be $e = 1$, and this pixel



(a)

(b)

Figure 5: Eval of Lena.

will be shifted. For the third case in Fig. 7(c), since its local complexity is $\mathrm{NL} = 34 > t_2$, it is skipped.

Next, a minimization problem is formulated to find the optimal thresholds (t_1^*, t_2^*) for a cover image as

$$\begin{array}{ll} \underset{0\le t_1\le t_2}{\arg\min} & \frac{\frac{1}{2}\mathrm{H}_{t_1,t_2}(0)+\sum_{i\ge 1}\mathrm{H}_{t_1,t_2}(i)}{\mathrm{H}_{t_1,t_2}(0)} \\ s.t. & \mathrm{H}_{t_1,t_2}(0)\ge EC \end{array} \tag{7}$$

where H_{t_1} represents the PEH derived from pixels in X_1 based on the context pixels chosen as C_1, and H_{t_2} represents the PEH derived from pixels in X_2 based on the context pixels chosen as C_2. And, aaa denotes the size of the required payload. Fig. 8 shows some optimal thresholds (t_1^*, t_2^*) obtained by (7) for some specific capacities on image Lena. It is note that the relationship between these two optimal thresholds are difficult to analyze, so that the optimal thresholds are determined by exhaustive search in practice. Besides, Fig. 6 shows three curves about embedding capacity and the proportion of shifted pixels. The blue and the red curves are corresponding to the cases that only context pixels based on C_1 or C_2 are used for prediction, while the pink curve corresponds to the case that both C_1 and C_2 are used for prediction in an optimal way. Obviously, the proposed multi-size context pixels based embedding strategy effectively reduced the distortion caused by pixel shifting and the performance is improved.

In addition, more sizes of context pixels can be used for further improvement of the embedding performance. Based on Fig. 4, four cases with different sizes of context pixels are tested here, and corresponding experimental results on images of Lena and Baboon are shown in Fig. 9. Specifically, the first case only takes C_1 into consideration, and it is shown as C_1 in Fig. 9. The second case employs two sizes of context pixels with C_1 and C_2 for prediction, and the corresponding result is shown in Fig. 9 as C_{12}. Similarly, C_{123} represent the case that C_1, C_2 and C_3 are used, and C_{1234} represent the case that C_1, C_2, C_3 and C_4 are used for prediction. Obviously, by employing more different sizes of context pixels for prediction, the embedding performance is improved. And the best embedding performance is obtained in the case of C_{1234}. However, since the embedding results of C_{123} is very close to the case of C_{1234}, while it is more time-saving, we finally take three different sizes of context pixels as C_1, C_2 and C_3 for the prediction of the proposed method. Accordingly, three thresholds $\{t_1, t_2, t_3\}$ are needed for the implementation of the proposed method, and specific procedures of the embedding and the extraction process are given in the next subsection.

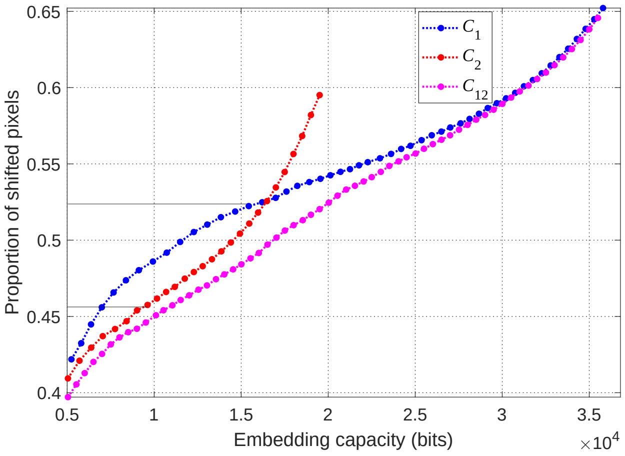


Figure 6: Eval of Lena.

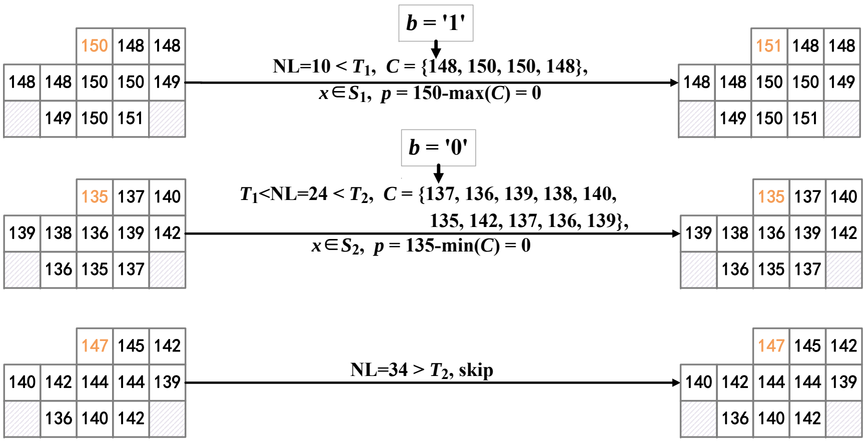


Figure 7: Example.

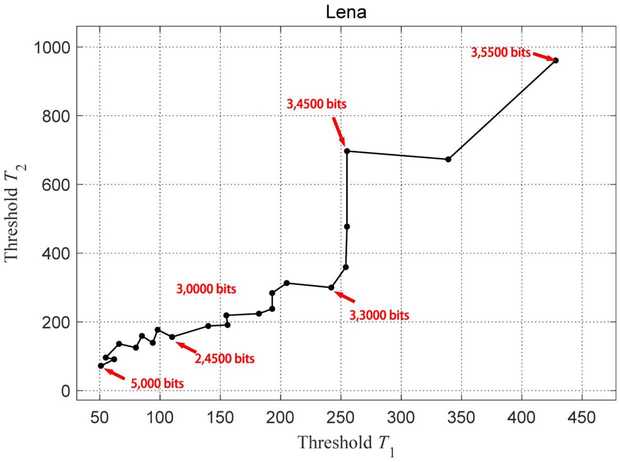


Figure 8: Thresholds.

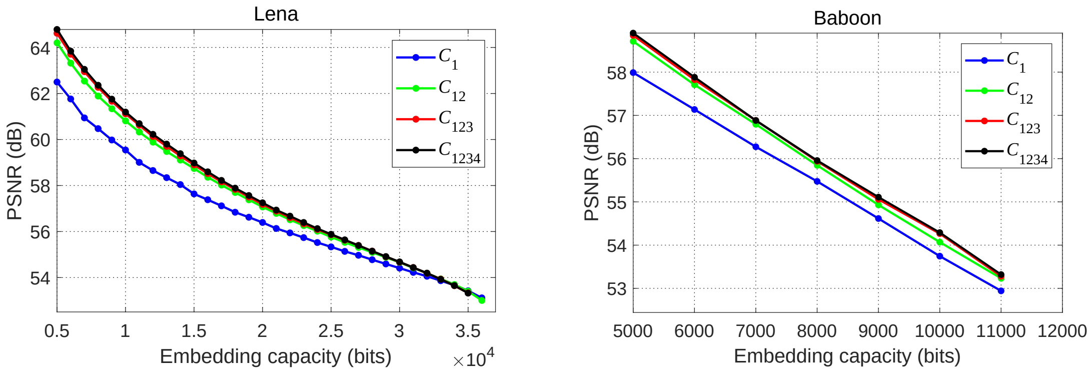


Figure 9: size.

3.3. Data Embedding and Extraction Procedures

For the data embedding process, it is conducted by following steps. First, a location map LM is constructed to avoid the overflow and underflow problem. For the i-th cover pixel x_i, if x_i equals 0 or 255, we set $\mathrm{LM}(i) = 1$. Otherwise, $\mathrm{LM}(i) = 0$. And it is further compressed by arithmetic coding to reduce its length. Then, for the cover image, pixels valued 255 are modified to 254 and pixels valued 0 are changed to 1. Next, the local complexity of all the pixels are calculated by the context region based on C_3. In addition, for a certain value of the three thresholds (t_1, t_2, t_3), the cover pixels are classified into four classes based on their local complexity, and pixels in different classes are predicted based on different number of context pixels as described in Section 3.2. Note that, the embedding procedures presented in Section 2 are used here except for the prediction of cover pixels. Accordingly, the data embedding process is performed in a pixelwise way until all the secret data is embedded into the cover image, and the position of the last pixel used for embedding is recorded as k_{end}. Moreover, for blind extraction, some auxiliary information also should be embedded, including

- the local complexity thresholds t_1, t_2 and t_3 ($11 \times 3 = 33$ bits),
- the position of the last pixel used for embedding k_{end} (18 bits),
- the length of the compressed location map L (18 bits),
- the compressed location map.

The auxiliary information is embedded into the least significant bits (LSB) of the first $69 + L$ pixels in the first row. And the original LSBs of these pixels are embedded into the cover image together with the payload. Note that, for a fixed payload, the optimal thresholds used for embedding are determined by exhaustively search.

The extraction process is conducted based on reverse procedures of the embedding process, and specific procedures are omitted for simplicity.

4. Experimental Results

In this section, in order to demonstrate the effectiveness of the proposed RDH method, it is compared with three state-of-the-art PVO-based methods. They are the methods of Peng *et al.* [31], Ou *et al.* [32]

and Qu *et al.* [33]. All the experiments are implemented based on Matlab with the version of 2018a on a tower server (SUPERMICRO LT-7038AX). And, it is worth mentioning that, for a given cover image with a fixed payload, the proposed method can complete the embedding process within four seconds.

The experiments are conducted based on eight standard 512 × 512 sized gray-scale images, including Lena, Baboon, Airplane, Barabra, Lake, Boat, Peppers and Elaine. And the corresponding comparison results are shown with capacity-distortion curves in Fig. 9. Specifically, the embedding capacity increases from 5,000 bits to the largest capacity with a step of 1,000 bits. According to this figure, it is obvious that the embedding performance of the proposed method is superior to other three PVO-based methods. Moreover, the comparison results on PSNR values for fixed embedding capacities with 10,000 bits and 20,000 bits are given in Table 1 and Table 2. Besides, the optimal parameters (t_1^*, t_2^*, t_3^*) of proposed method on different images are also listed in the tables. Note that, for a fixed embedding capacity, the optimal thresholds for different images are different from each other. However, since the computational complexity of the proposed method is acceptable, exhaustive search is adopted for better embedding performance.

As mentioned above, the improved PVO-based RDH proposed by Peng *et al.* [31] achieved performance enhancement upon the original PVO-based RDH method [30]. The cover image is first divided into non-overlapping blocks with equal size, and then secret data is embedded into the largest and smallest pixels of each block. Compared with the original PVO-based RDH, in [31], more smooth blocks are utilized for data embedding by taking the pixel values as well as their locations into consideration. However, its embedding performance can be further improved. According to the comparison results shown in Fig. 9, the proposed method outperforms it on all the test images. In addition, based on Table 1 and Table 2, the proposed method obtains an PSNR increase of 0.90 dB and 0.99 dB for the embedding capacity of 10,000 bits and 20,000 bits, respectively.

The PVO-k RDH method proposed by Ou *et al.* [33] is another improved method of the original PVO [30], which aims to make full use of smooth blocks as well. In this method, the embedding performance is optimized by combining PVO-1 with PVO-2. Although better embedding performance is achieved compared with the improved PVO-based method, there still has much room for improvements. According to Table 1 and Table 2, the proposed method outperforms this method with 0.63 dB and 0.78 dB for the embedding capacity of 10,000 bits and 20,000 bits, respectively.

The PPVO RDH method proposed by Qu *et al.* [33] is quiet different from traditional PVO-based RDH methods [30–32]. They proposed to perform prediction for each pixel to enlarge the embedding capacity as well as gain better embedding performance. Besides, the optimal number of context pixels is obtained by implementing the embedding process for 15 times to test n from 1 to 15, and the marked image with highest PSNR is considered as the final embedding result. The embedding performance of PPVO is superior to previous PVO-based RDH methods. However, by choosing more suitable context pixels as well as make full use of the difference of pixel local complexity, the embedding performance is further improved by the proposed method. Specifically, based on Table 1 and Table 2, the proposed method gets an increase of PSNR by 0.57 dB for an embedding performance of 10,000 bits and 0.44 dB for an embedding performance of 20,000 bits.

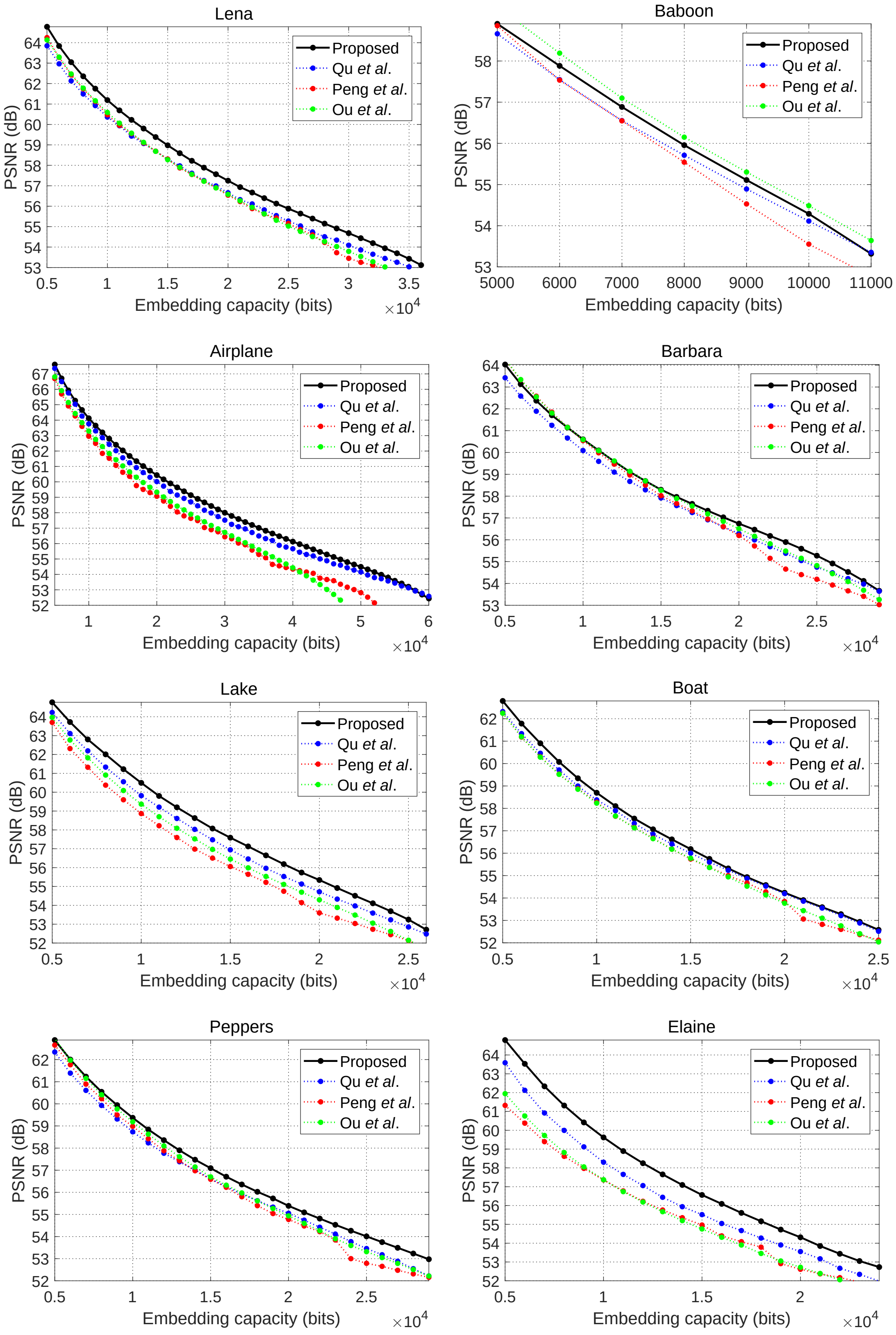


Figure 10: capacity.

Table 1: Comparison of PSNR (dB) among the proposed method and the PVO-based methods of Peng *et al.* [31], Ou *et al.* [32] and Qu *et al.* [33]. The embedding capacity is 10,000 bits. The optimal parameters of the proposed method are also listed.

Images	Peng *et al.*	Ou *et al.*	Qu *et al.*	Proposed Method	
				PSNR	(t_1^*, t_2^*, t_3^*)
Lena	60.49	60.59	60.36	**61.12**	(39,45,73)
Baboon	53.58	54.48	54.11	**54.27**	(337,465,1099)
Airplane	62.97	63.29	63.76	**64.06**	(16,17,22)
Barbara	60.48	60.59	60.08	**60.59**	(17,66,79)
Lake	58.81	59.36	59.81	**60.50**	(54,83,112)
Boat	58.26	58.23	58.37	**58.70**	(78,96,144)
Peppers	58.97	59.18	58.73	**59.32**	(29,92,121)
Elaine	57.37	57.37	58.30	**59.46**	(82,116,142)
Average	58.86	59.13	59.19	**59.76**	

Table 2: Comparison of PSNR (dB) among the proposed method and the PVO-based methods of Peng *et al.* [31], Ou *et al.* [32] and Qu *et al.* [33]. The embedding capacity is 20,000 bits. The optimal parameters of the proposed method are also listed.

Images	Peng *et al.*	Ou *et al.*	Qu *et al.*	Proposed Method	
				PSNR	(t_1^*, t_2^*, t_3^*)
Lena	56.56	56.58	56.65	**57.18**	(52,102,129)
Airplane	59.07	59.33	60.01	**60.40**	(22,31,32)
Barbara	56.20	56.50	56.28	**56.74**	(75,94,121)
Lake	53.53	54.29	54.71	**55.34**	(156,193,227)
Boat	53.83	53.76	54.19	**54.24**	(159,191,266)
Peppers	54.77	54.93	55.05	**55.39**	(96,196,186)
Elaine	52.65	52.71	53.55	**54.21**	(165,334,452)
Average	55.23	55.44	55.78	**56.22**	

5. Conclusion

In this paper, an improved PPVO RDH method is proposed to gain performance enhancement based on the original PPVO [33]. In order to make accurate pixel prediction as well as sharper PEHs, some pixels close to the to-be-predicted pixel, which are not utilized in the PPVO, are involved in the prediction process. Besides, a multi-size context pixels based embedding strategy is proposed, and different number of context pixels are used for the prediction of pixels with different local complexity. In addition, based on the capacity-distortion model, the embedding performance is optimized by reaching a balance between the embedding capacity and the embeddig distortion. Experimental results demonstrate the efficiency of the proposed method through the capacity-distortion performance comparison with some state-of-the-art methods [31–33].

Acknowledgement

This work was supported by the National Key Research and Development of China (No. 2016YFB0800404), the National Science Foundation of China (Nos. 61572052 and U1736213), and the Fundamental Research Funds for the Central Universities (Nos. 2017RC008 and 2018JBZ001).

References

[1] J. B. C. I.J. Cox, M.L. Miller, Digital watermarking.

[2] J. Fridrich, Steganography in digital media.

[3] Y. Q. Shi, X. Li, X. Zhang, H. Wu, B. Ma, Reversible data hiding: Advances in the past two decades, IEEE Access 4 (2016) 3210–3237.

[4] J. Tian, Reversible data embedding using a difference expansion, IEEE Trans. Circuits Syst. Video Technol. 13 (8) (2003) 890–896.

[5] C. Qin, C. C. Chang, Y. H. Huang, L. T. Liao, An inpainting-assisted reversible steganographic scheme using a histogram shifting mechanism, IEEE Trans. Circuits Syst. Video Technol. 23 (7) (2013) 1109–1118.

[6] D. M. Thodi, J. J. Rodrguez, Expansion embedding techniques for reversible watermarking, IEEE Trans. Image Process. 16 (3) (2007) 721–730.

[7] Y. Hu, H. K. Lee, J. Li, De-based reversible data hiding with improved overflow location map, IEEE Trans. Circuits Syst. Video Technol. 19 (2) (2009) 250–260.

[8] X. Li, W. Zhang, X. Gui, B. Yang, A novel reversible data hiding scheme based on two-dimensional difference-histogram modification, IEEE Trans. Inf. Forens. Secur. 8 (7) (2013) 1091–1100.

[9] W. Hong, T. S. Chen, C. W. Shiu, Reversible data hiding for high quality images using modification of prediction errors, J. Syst. Softw. 82 (11) (2009) 1833–1842.

[10] W. Hong, T. S. Chen, Y. P. Chang, C. W. Shiu, A high capacity reversible data hiding scheme using orthogonal projection and prediction error modification, Signal Process. 90 (11) (2010) 2911–2922.

[11] X. Li, B. Li, B. Yang, T. Zeng, General framework to histogram-shifting-based reversible data hiding, IEEE Trans. Image Process. 22 (6) (2013) 2181–2191.

[12] W. Wang, J. Ye, T. Wang, W. Wang, A high capacity reversible data hiding scheme based on right-left shift, Signal Process.

[13] V. Sachnev, H. J. Kim, J. Nam, S. Suresh, Y. Q. Shi, Reversible watermarking algorithm using sorting and prediction, IEEE Trans. Circuits Syst. Video Technol. 19 (7) (2009) 989–999.

[14] P. Tsai, Y. C. Hu, H. L. Yeh, Reversible image hiding scheme using predictive coding and histogram shifting, Signal Process. 89 (6) (2009) 1129–1143.

[15] X. Gao, L. An, Y. Yuan, D. Tao, X. Li, Lossless data embedding using generalized statistical quantity histogram, IEEE Trans. Circuits Syst. Video Technol. 21 (8) (2011) 1061–1070.

[16] X. Li, B. Yang, T. Zeng, Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection, IEEE Trans. Image Process. 20 (12) (2011) 3524–3533.

[17] W. Hong, Adaptive reversible data hiding method based on error energy control and histogram shifting, Opt. Commun. 285 (2) (2012) 101–108.

[18] H. T. Wu, J. Huang, Reversible image watermarking on prediction errors by efficient histogram modification, Signal Process. 92 (12) (2012) 3000–3009.

[19] B. Ou, X. Li, Y. Zhao, R. Ni, Y. Q. Shi, Pairwise prediction-error expansion for efficient reversible data hiding, IEEE Trans. Image Process. 22 (12) (2013) 5010–5021.

[20] I. C. Dragoi, D. Coltuc, Local-prediction-based difference expansion reversible watermarking, IEEE Trans. Image Process. 23 (4) (2014) 1779.

[21] X. Li, W. Zhang, X. Gui, B. Yang, Efficient reversible data hiding based on multiple histograms modification, IEEE Trans. Inf. Forens. Secur. 10 (9) (2015) 2016–2027.

[22] I. C. Dragoi, D. Coltuc, Adaptive pairing reversible watermarking, IEEE Trans. Image Process. 25 (5) (2016) 2420–2422.

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

[24] Z. Ni, Y. Q. Shi, N. Ansari, W. Su, Reversible data hiding, IEEE Trans. Circuits Syst. Video Technol. 16 (3) (2006) 354–362.

[25] M. Fallahpour, Reversible image data hiding based on gradient adjusted prediction, IEICE Electron Exp. 5 (20) (2008) 870–876.

[26] D. Ioan-Catalin, C. Dinu, Local-prediction-based difference expansion reversible watermarking, IEEE Trans. Image Process. 23 (4) (2014) 1779.

[27] D. Ioan-Catalin, C. Dinu, On local prediction based reversible watermarking, IEEE Trans. Image Process. 24 (4) (2015) 1244–1246.

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

[29] B. Ou, X. Li, J. Wang, Improved pvo-based reversible data hiding: A new implementation based on multiple histograms

modification, J. Vis. Commun. Image Represent. 38 (2016) 328–339.

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

[31] F. Peng, X. Li, B. Yang, Improved pvo-based reversible data hiding, Digit. Signal Process. 25 (2) (2014) 255–265.

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

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

[34] S. Weng, Y. Liu, J. S. Pan, N. Cai, Reversible data hiding based on flexible block-partition and adaptive block-modification strategy, J. Vis. Commun. Image Represent. 41 (2016) S104732031630205X.

[35] S. Weng, G. Zhang, J. S. Pan, Z. Zhou, Optimal ppvo-based reversible data hiding, J. Vis. Commun. Image Represent. 48 (2017) S1047320317301062.

[36] W. He, G. Xiong, S. Weng, Z. Cai, Y. Wang, Reversible data hiding using multi-pass pixel-value-ordering and pairwise prediction-error expansion, Information Sciences. 49.

[37] S. Kim, X. Qu, V. Sachnev, H. J. Kim, Skewed histogram shifting for reversible data hiding using a pair of extreme predictions, IEEE Trans. Circuits Syst. Video Technol.