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

Pixel-value-order (PVO) is a high-fidelity reversible data hiding (RDH) method which is commonly used today. Traditional PVO-based RDH methods embedding data into prediction-errors of the largest/smallest pixels within a fixed size block. Later, the pixel-based PVO (PPVO) is proposed to overcome the block constraint by a pixel-by-pixel prediction manner. Each pixel is predicted by the surrounding sorted context pixels in the down right direction and can be used for embedding. However, in this method, some context information in the down left direction is omitted which is benefit for embedding performance, and the number of context pixels of each to-be-predicted pixel is constant, without considering that different local complexity of pixel corresponding to the different suitable context pixels. In this paper, an improved PPVO-based RDH method is proposed with an extended PPVO predictor and the embedding procedure is implemented by a multi-size based embedding method for multiple histogram modification (MHM). First, an extended PPVO predictor is proposed to further utilize the additional surrounding context information in the down left direction, leading to a sharper prediction-error histograms (PEH). Then, we explore the impact of the various sizes of context region through the local complexity consideration. Data is embedded into the prediction-errors with a suitable context region size. The experimental results shows that the proposed method achieves an outstanding performance and is superior to several state-of-the-art PVO-based methods.

Keywords: Reversible data hiding, pixel-value-ordering, adaptive embedding, multiple histograms modification

1. Introduction

In recent days, the tampering and propagation of multimedia data have been getting more and more convenient. In order to verify the integrity and security of information, technologies such as information encryption, fingerprint recognition and data hiding have been extensively studied. Among them, reversible data hiding(RDH) [?] is a technology that embeds secret data into image in an imperceptible way and the decoder can recover the original content lossless from the marked image while extracting the secret data.

Up to now, RDH has been extensively explored and many types of techniques such as difference expansion(DE) [1–5], histogram shifting(HS) [6–9] and prediction-error expansion(PEE) [2, 10–20] have emerged. Among the techniques, the first DE-based method is proposed by Tian *et al.* [1] which expands the difference

---

*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)

between adjacent pixels to achieve embedding procedure. By embedding data into the differences, the high fidelity is guaranteed. HS-based methods achieve embedding by shifting pixels according to the peak-gap distribution of image histogram. However, embedding capacity and distortion is mainly sensitive to characteristic of the histogram. Besides, unlike DE-based and HS-based methods, PEE is the most influential RDH technique due to its balance between distortion and capacity. PEE is first proposed by Thodi and Rodriguez in [3]. In this PEE-based method, pixels are first predicted by adjacent pixels and then accurately predicted pixels are modified by expanding the prediction-errors for embedding, while the others are shifted to ensure the reversibility. Focusing on the pixel prediction and modification of prediction-errors, many methods are proposed, e.g., accurate predictor design for prediction-error histogram(PEH) [3, 4, 6, 10, 21–23], high-dimension histogram modification [5, 16, 19] and multiple histograms modification(MHM) [18, 24, 25].

Among the PEE-based methods, pixel-value-order(PVO) is the most critical technology which provides an accurate prediction method [24–33]. The original PVO-based method [26] is proposed by Li *et al.*. The cover image is first divided into equal size and non-overlapping blocks and data is embedding into the prediction-errors of the largest/smallest pixel value within a fixed size block. Peng *et al.* propose an improved PVO method [27] which considers the original order of the largest/smallest pixel value and the second the largest/smallest pixel value, resulting in the fully utilizing of smooth blocks. Meanwhile, Ou *et al.* propose the PVO-k scheme [28] which modifies the first several largest/smallest pixel values at the same time to utilizing of smooth blocks as well. Later, Ou *et al.* [25] and He *et al.* [32] explore the relationship of pixels in each block. Prediction-errors in the same block are paired to generate a two-dimensional PEH and advisable histogram modification manners are found to further improve the performance. Furthermore, Wang *et al.* [24] and Weng *et al.* [30] further divide the block into smaller sub-blocks, further utilizing the pixels in the smooth region. In addition, to utilize the omitted pixels in the fixed size blocks, Qu *et al.* proposed a pixel-based PVO (PPVO) method [29]. The target pixel is predicted by the the largest/smallest surrounding context pixels in the down left direction and it gets a high embedding capacity by the pixel-by-pixel strategy. It is note that in the PPVO-based method, each pixel is predicted by the context pixels in the down right direction with a constant number of context pixels. However, there are some other pixels in the down left directions is omitted which are closer to the target pixel and may be beneficial for pixel prediction. Besides, number of context pixels for all the pixels is the same without considering the impact of various context pixels on pixels with different local complexity.

In this paper, we try to improve the performance of PPVO-based RDH. First, we find that in the original PPVO RDH method, part of surrounding context pixels do not appreciate in the prediction process, which are correlated with the pixel to-be-predicted. In this work, a extended PPVO predictor is proposed which utilizes pixels in left lower direction of target pixel for prediction as well. And it carries out an accurate prediction process to derive a sharper PEH. Meanwhile, the impact of various size of context region is explored. The context region size is chosen by the local complexity of the target pixel. Pixels in smooth area are predicted with a small context region size and pixels in complex area are predicted with a larger size. The experimental results verify that the proposed method is superior to the several state-of-the-art PVO-based methods.

The rest of the paper is arranged as follow. Section 2 is a brief review of classical PPVO-based RDH method [29]. Section 3 is an introduction of the proposed improved pixel-based PVO method. Section 4 shows the experimental results. Finally, Section 5 concludes the paper.

(a) Context pixels of x by PPVO.
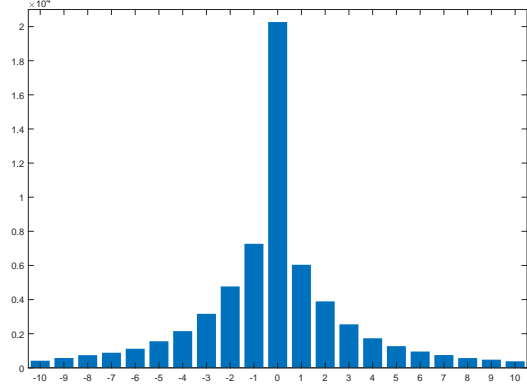
(b) PEH with PPVO predictor on image Lena.

Figure 1: PPVO Context Pixels and show the shaper histogram with $n = 15$.

2. Related Works

In this section, as prior knowledge, we will review the Pixel-based PVO RDH(PPVO) method of Qu *et al.* [29]. And the data embedding and extraction procedures are presented followed.

In the classical PVO-based RDH method such as the original PVO-base method [26] and the improved PVO-based method [27], the cover image is first devided into equal size and non-overlapping blocks. Then, in each block, only two pixels of the largest and smallest value pixels are utilized for embedding, while others pixels are omitted, causing the difficulty of efficient embedding data into smooth region. To breakthrough the block constraint, in [29], Qu *et al.* propose a pixel-based PVO RDH method, which is called PPVO, to utilized the sorted pixels for data embedding. PPVO achieves an pixel-by-pixel data embedding process. Specifically, the method is described as follow.

First of all, to overcome the overflow/underflow problem, pixels with value of 0 are modified to 1 and with value of 255 are modified to 254. And then, for current pixel x, the context pixels defined as the pixels in the lower right direction and form an vector $C = \{c_1, ..., c_n\}$ as shown in Fig. 1(a), where n is the number of context pixels. With a given number of context pixels n, by comparing the values of x and its context pixels, there are four cases for the prediction of x. To simplify the explanation, pixels participating in the embedding process are grouped into four sets, which are

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

where $C_{\max}$ and $C_{\min}$ are the largest and smallest pixel values of context pixels in C, respectively.

- If current pixel x belong to S_1 or S_4, $C_{\max}$ is chosen as the prediction of x denoted as $\hat{x} = C_{\max}$.

- If current pixel x belong to S_2 or S_3, $C_{\min}$ is chosen as the prediction of x denoted as $\hat{x} = C_{\min}$.

Other pixels not within these four sets are skipped and do not participate in the prediction and embedding

3

process. The prediction-error e is computed by

$$e = x - \hat{x}. \tag{1}$$

The PEH of the standard 512×512 gray-scale image Lena is shown in Fig 1(b), it can be seen that a highest frequency locates at bin 0. Thus, based on the characteristic of PEH, prediction-error p is modified to derive the marked prediction-error $\tilde{e}$ as

$$\tilde{e} = \begin{cases} e + b, & \text{if } e = 0, x \in S_1 \cup S_4 \\ e - b, & \text{if } e = 0, x \in S_2 \cup S_3 \\ e + 1, & \text{if } e \geq 1 \\ e - 1, & \text{if } e \leq 1 \end{cases}, \tag{2}$$

where $b \in \{0, 1\}$ is one bit secret data to be embedded. And the pixel x is modified to derive the marked pixel as

$$\tilde{x} = \hat{x} + \tilde{e}. \tag{3}$$

One should be note here that during the embedding process, prediction for pixels is performed in the raster-scan order. And in the extraction process, prediction is implemented in reverse order. For $x \in S_1 \cup S_4$, because the derive $\tilde{x}$ is obtained by increasing the original x with $x \geq C_{\max}$ and shares the same context pixels as original pixel x, so $\tilde{x} \geq C_{\max}$ and $\tilde{x}$ belongs to the same set as x as well. By the same reason, for $x \in S_2 \cup S_3$, the derive $\tilde{x}$ is obtained by decreasing the original x with $x \leq C_{\min}$ leading to $\tilde{x} \leq C_{\min}$ and $\tilde{x}$ belongs to the same set as x after embedding.

Thus, for the decoder, to extract data and recover the original pixel, the prediction value of current marked pixel $\tilde{x}$ is considered in the same way as embedding process. Specifically,

- If current marked pixel $\tilde{x}$ belong to S_1 or S_4, $C_{\max}$ is chosen as the prediction of x denoted as $\hat{x} = C_{\max}$.

- If current marked pixel $\tilde{x}$ belong to S_2 or S_3, $C_{\min}$ is chosen as the prediction of x denoted as $\hat{x} = C_{\min}$.

The the marked prediction-error $\tilde{e}$ is computed by

$$\tilde{e} = \tilde{x} - \hat{x}, \tag{4}$$

and is modified to obtain the prediction-error e by

$$e = \begin{cases} \tilde{e}, & \text{if } e = 0 \\ \tilde{e} - 1, & \text{if } e \geq 1 \\ \tilde{e} + 1, & \text{if } e \leq -1 \end{cases}, \tag{5}$$

And the original pixel value x is recovered by

$$x = \hat{x} + e. \tag{6}$$

Meanwhile, the secret data is extracted that $b = 1$ if $\tilde{e} \in \{-1, 1\}$ and $b = 0$ if $\tilde{e} = 0$.

It should be noted that in the embedding process, the proper n is selected by choosing the best result of $C = \{1, ..., n\}(i \in n, ..., 15)$ exhaustively. And for a given embedding capacity, a embedding selection strategy is carried out by preferentially selecting the pixels in the smooth area. A threshold T is set to
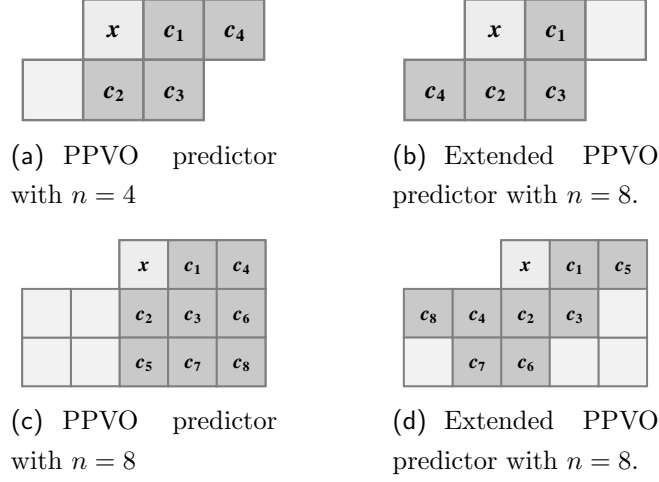
Figure 2: Context pixels of x by extended PPVO predictor.

select the pixels with the complexity less than T to participate in the embedding procedure which can just satisfy the embedding capacity. The complexity of current pixel is obtained by calculating the sum of the absolution differences of the horizontal and vertical of every two adjacent context pixels.

The experimental results show that PPVO gets a large maximum embedding capacity than the previous PVO-based method [26, 27] by pixel-by-pixel embedding. And it obtains a sharper histogram which is causing a result of better performance as shown in Fig. 1.

Finally, there are two oversights should be noticed. First, surrounding pixels of the current x in the down left direction is omitted. These pixels are also close to x and may be helpful for a more accurate prediction. Second, in PPVO-based method, pixels are predicted with the same number of context pixels given in advance. Actually, various context pixels number n on pixels of different local complexity have different effects. These two oversights will be discussed in the proposed method.

3. Proposed Method

In this section, a extended PPVO predictor is first introduced. Then, a multi-size based embedding method for MHM is proposed to further improve the performance. Finally, the embedding procedure and extraction procedure is described.

3.1. Extended PPVO predictor

Compared with PVO-based methods such as PVO[26], IPVO[27] and PVO-k[28], in PPVO, every pixel is predicted for embedding, breaking through the block constraint. And to ensure the reversibility, pixels in lower right direction are utilized as context region to predict the current pixel which is shown in Fig. 2(a) and Fig. 2(c). The prediction is more accurate, however, the surrounding context information is not fully utilized. There are some other pixels are omitted. Clearly, for example, pixels $\{C_4\}$ in Fig. 2(b) and $\{c_4, c_7, c_8\}$ in Fig. 2(d) are closer to current pixel x and can provide more useful information, but they are omitted in pixel prediction stage. The extended PPVO predictor introduces this information and the superiority of extended PPVO predictor is introduced next.

Fig. 3 shows the comparison of normalized PEHs of extended PPVO predictor and original PPVO predictor are displayed on two standard gray-scale 512×512 sized images with $n = 4$ and $n = 8$, respectively. For a reasonable comparison, in Fig. 3(a) and Fig. 3(b), four pixels near x is considered as the context pixels. Fig. 2(a) and Fig. 2(b) show the context pixels selection for the PPVO predictor and the extended PPVO predictor. And in Fig. 3(c) and Fig. 3(d), Fig. 2(c) and Fig. 2(d) show the eight context pixels selection for the PPVO predictor and the extended PPVO predictor.

Obviously, normalized PEHs of extended PPVO predictor have highest peak at 0, and the distributions are sharper, which is beneficial to the final embedding performance. Moreover, we use the proportion of shifted pixels [26] to evaluate characteristic of the histogram theoretically, numerically expressed as

$$\frac{\#\{\text{shifted pixels}\}}{\#\{\text{expanded or shifted pixels}\}} = \frac{\sum_{i \neq 0} \mathrm{H}(i)}{\mathrm{H}(0) + \sum_{i \neq 0} \mathrm{H}(i)}, \tag{7}$$

where H is the PEH and $\mathrm{H}(i)$ is the frequency of pixels whose prediction-errors $p = i$. This evaluation describes the distortion caused by embedding one bit data on H. The smaller the proportion of shifted pixels, the better the characteristics of the PEH, which can bring less distortion. In Fig. 3, for images Lena, and Baboon, with $n = 4$, proportion of of extended PPVO predictor are 0.65, and 0.88 respectively, while value of evaluation are 0.72, and 0.89 of PPVO predictor. This indicates that data embedding on PEH of extended PPVO predictor may carry much less distortion to obtain the better performance. And with $n = 8$, proportion of of extended PPVO predictor are 0.61, and 0.86 respectively, while value of evaluation are 0.7, and 0.88 of PPVO predictor. This indicates that data embedding on PEH of extended PPVO predictor may carry much less distortion to obtain the better performance.

### 3.2. Multi-size based Embedding Method for MHM

In PPVO, before implementing the embedding procedure, number of context pixels n of to-be-predicted x is already set in advance. And for a cover image, every pixel is predicted by the same region of context pixels. Actually, the performance results caused by different number of context pixels are different. In order to explore the impact of n, we consider $n \leq 24$ and, to simplify the presentations, context pixels are group into four sets according to size of the region around the current pixel x, including $C_1 = \{c_1, ..., c_4\}$, $C_2 = \{c_1, ..., c_{10}\}$, $C_3 = \{c_1, ..., c_{18}\}$ and $C_4 = \{c_1, ..., c_{24}\}$, as shown in Fig. 4. Based on the consideration of the complexity of traditional RDH embedding, the comparison of threshold-capacity curves and threshold-proportion curves are shown in Fig. 5(a) and Fig. 5(b) as well. Here the proportion indicates the proportion of shifted pixels refined by (7). And the threshold T is set to choose the smooth pixels whose complexity $\mathrm{NL} < T$ to generate PEH so that all secret data can just be embedded. The complexity of the current pixel is evaluated by the sum of the absolution differences of the horizontal and vertical of every two adjacent pixels belong to the corresponding context. Besides, the capacity is defined as the occurrence of bin 0 of the generated PEH.

In Fig. 5, as an example, we focus on the two size context regions of C_1 and C_2. According to the curves, there are a few observations as follows.
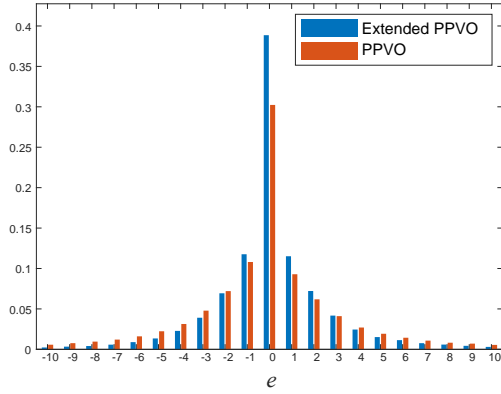
1. In Fig. 5(a), the maximum embedding capacity of C_1 is much larger than C_2. This is because almost pixels in the cover image with small n is used for prediction while part of the pixels are skipped with larger size C_2.
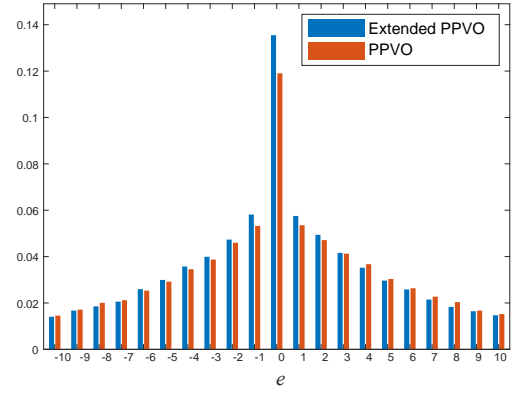
6
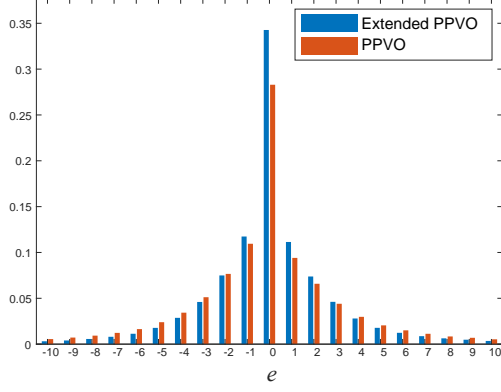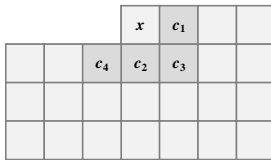
(a) Lena, $n = 4$
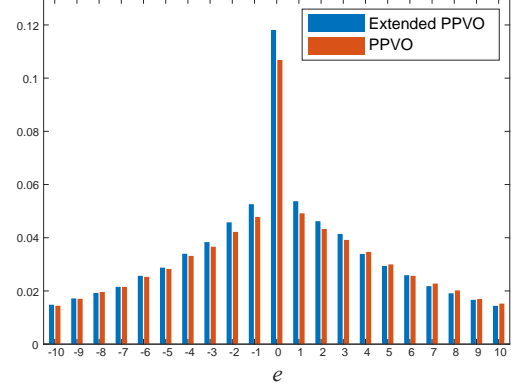
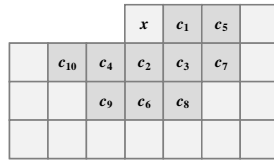(b) Baboon, $n = 4$

(c) Lena, $n = 8$

(d) Baboon, $n = 8$

Figure 3: Comparison of normalized PEH of p of extended PPVO with context pixels in Fig. 2.
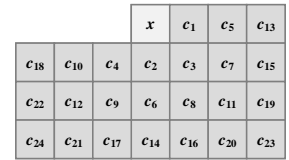

(a) C_1 of $n = 4$

(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.

2. In Fig. 5(a), capacity increases very fast with the increase of threshold T from a small value, especially curve of C_1. This indicates that pixels at smooth region are predicted accurate and are more suitable to be predicted.

3. In Fig. 5(b), with the same threshold T, although the capacity of C_2 is smaller than C_1, proportion of shifted pixels of C_2 is much smaller than C_1. This observation illustrates that among the skipped pixels of C_2, the proportion of pixels that are predicted to be inaccurate is increase, which is no benefit to the performance.

Through the above description, it shows that prediction with context pixels in smaller size context region achieves a large embedding capacity, but distortion also increases due to the pixels to be predicted inaccurately. And for the pixels in larger size context region, lots of inaccurately predicted pixels are skipped and proportion of sifted pixels is much less than the smaller size. But the embedding capacity is limited. Based on these consideration, a strategy of the multi-size based embedding method is given to combine the advantages of both size context.

In the proposed method, pixels on the cover images are divided into three parts by two thresholds t_1 and t_2 ($0 \leq t_1 \leq t_2$). Specifically, we let $X_T = \{x| \text{ NL}(x) < T\}$, which indicates the set of pixels with local complexity less than T. The region of pixels in X_{t_1} are considered to be smooth, and these pixels are predicted by the context pixels of C_1. And the region of pixels in $X_{t_2} \setminus X_{t_1}$ are seem to be normal smooth, which are predicted by the context pixels of C_2. Other pixels are all skipped and do not participate in the embedding procedure. The above strategy is reasonable. In Fig. 5(b), it shows that the distance between the two curves is growing with the increase of threshold. This verify that in the area with higher complexity, more inaccurately predicted pixels are skipped with context pixels in C_2. Therefore, the pixels in smooth regions X_{t_1} is predicted by the context pixels of C_1. This provides much more embedding capacity. And for the pixels in the normal regions $X_{t_2} \setminus X_{t_1}$, context pixels of C_2 are chosen as context pixels to predict, making the distortion as low as possible.

For example, Fig. 7 shows the the embedding procedures for the three types of pixels, where $(t_1, t_2) = (15, 25)$. For the first pixel value $x = 150$, because the complexity NL $= 10$ of the pixel less than the small threshold t_1, the pixel is in a smooth region and context pixels $C = \{148, 150, 150, 148\}$ are chosen. The prediction-error of the pixel value is $p = 0$. One bit secret data $b = 1$ can be embedded by increasing the x by 1 to $\hat{x} = 151$. However, if context pixels of C_2 are chosen for prediction, the pixel wound be skipped because of $\min(C) < x < \max(C)$. For the second pixel $x = 135$, it is in normal region, where the complexity value NL $= 24$ larger than t_1 but less than t_2. The context pixels $C = \{137, 136, 139, 138, 140, 135, 142, 137, 136, 139\}$ are chosen and prediction-error is $p = 0$. One bit data $b = 0$ can be embedded by remaining the pixel value $\hat{x} = x = 135$. If context pixels of C_1 are chosen, the prediction-error wound be $p = 1$ and no one bit data wound be embedded. The third pixel $x = 147$ is in a rough region whose complexity NL $= 34$ larger than t_2 and it is skipped. And for the purpose of finding suitable thresholds, let H_{t_1, t_2} as the histogram of prediction-errors derived from the pixels in X_{t_1} with C_1 and pixels in $X_{t_2} \setminus X_{t_2}$ with C_2. For a given embedding capacity EC, the optimal thresholds (t_1^*, t_2^*) are obtained by

$$
\begin{aligned}
& \underset{0 \leq t_1 \leq t_2}{\arg\min} \quad \frac{\frac{1}{2}\text{H}_{t_1, t_2}(0) + \sum_{i \geq 1} \text{H}_{t_1, t_2}(i)}{\text{H}_{t_1, t_2}(0)} \\
& s.t. \qquad \text{H}_{t_1, t_2}(0) \geq EC
\end{aligned}
\tag{8}
$$

Fig. 8 shows the optimal thresholds (t_1^*, t_2^*) by (8) for some capacities. It is note that the two optimal
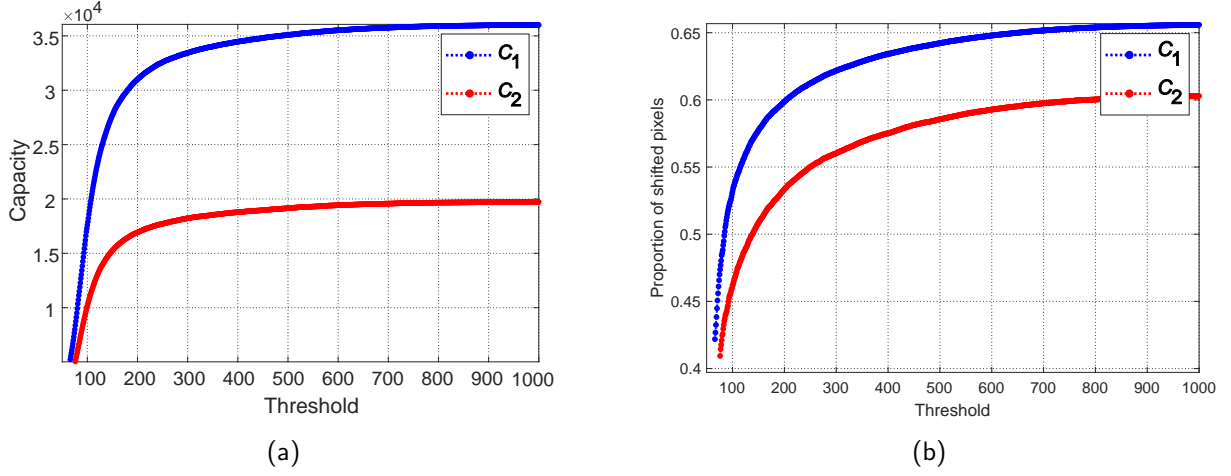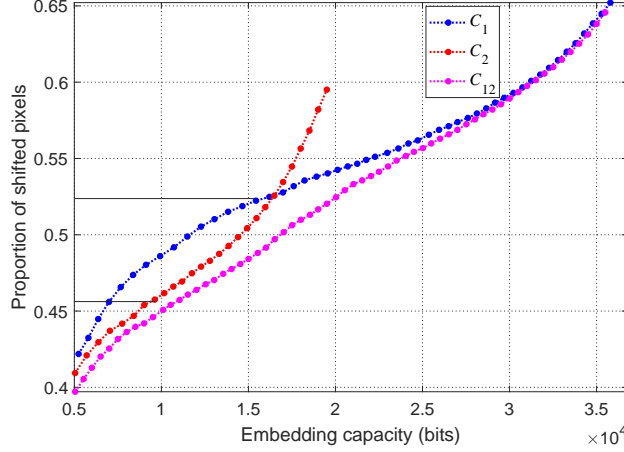
Figure 5: Eval of Lena.

Figure 6: Eval of Lena.

thresholds are positively correlated. However, it is difficult to find the exact relationship between them. So, in the optimization process, we find the optimal parameters by traversing all the values. Fig. 6 shows three capacity-proportion curves, including two curves C_1 in blue and C_2 in red with a given context region, and a curve in pink C_{12} that combines two different size context regions by the proposed method. Obviously, the proposed method achieve an less proportion of shifted pixels than others, which may lead to less embedding distortion.

In addition, more context sizes can be considered to combine to further decrease the proportion. We set N as the number of context regions to be considered. As shown in Fig. 9, for each given capacity, N is from 1 to 4, corresponding the sizes $C_1 \in \{C_1\}$, $C_{12} \in \{C_1, C_2\}$, $C_{123} \in \{C_1, C_2, C_3\}$ and $C_{1234} \in \{C_1, C_2, C_3, C_4\}$, respectively. It shows that although the embedding performance increases with the increase of N, it is already very small when $N = 4$. Therefore, balancing the embedding performance and embedding time complexity, we set $N = 3$ as a suitable size number. Therefore, for the current pixel, there are three complexity thresholds $\{t_1, t_2, t_3\}$ to choose context pixels The smoother the area where the pixel is, the less
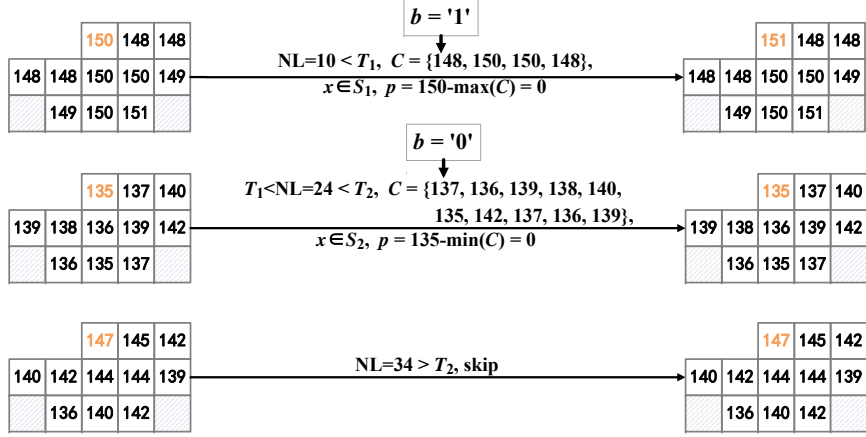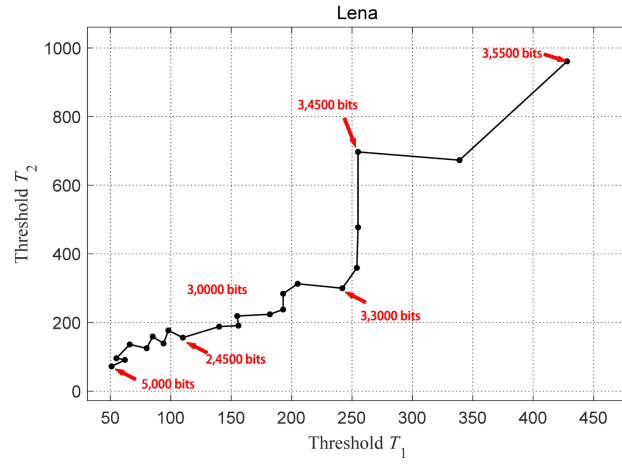
9

Figure 7: Example.
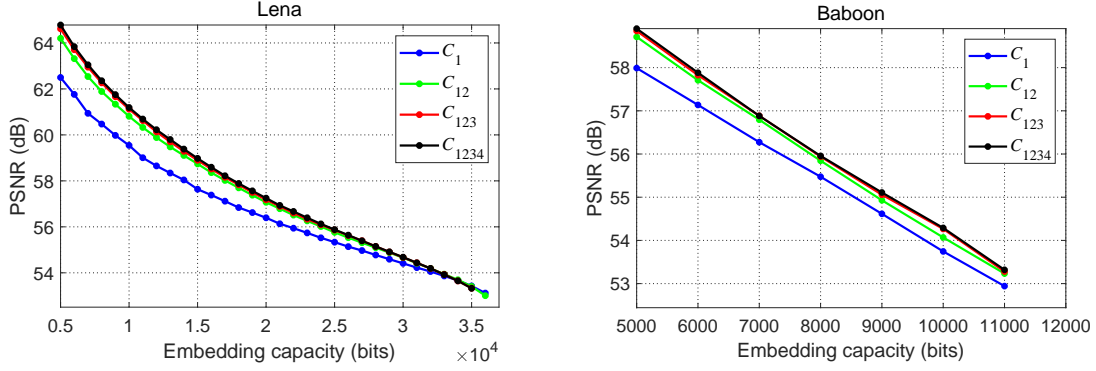

Figure 8: Thresholds.

Figure 9: size.

context pixels are used to predict.

3.3. Data Embedding and Extraction Procedures

The data embedding procedure is divided into the following steps to introduce as follow. Firstly, a location map LM is constructed to solve overflow/underflow problem. For i-th pixel x_i, if $x_i = 0$ or $x_i = 255$, we assign $\text{LM}(i) = 1$. Otherwise, we assign $\text{LM}(i) = 0$. All pixels equals 255 are modified to 254 and 0 are modified to 1. Because LM records all the locations where the overflow/underflow may occur, original pixels can be recovered. Next, for the given several context sizes $\{C_1, C_2, C_3\}$, the complexity $\text{NL}(i)$ of current pixel value x_i is calculated the sum of the absolution differences of the horizontal and vertical of every two adjacent pixels belong to the largest size context C_3. And context pixels to predict x_i are selected by comparing $\text{NL}(i)$ with the given thresholds $\{t_1, t_2, t_3\}(0 \leq T_1 < T_2 < T_3)$, which is similar to the procedure described in section 3.2. Then, x_i is predicted by the chosen context pixels yo derive the prediction-error e by (1) and one bit data is embedded by modifying x_i to $\tilde{x}_i$ by (3). The steps of prediction and embedding are performed pixel-by-pixel until all secret data is embedded and current pixel position is recorded as k_{end}. Finally, for reversibility, some auxiliary information should be used for blind decoding such as

- the complexity thresholds t_1, t_2 and t_3 ($11 \times 3 = 33$ bits),

- the pixel position where data embedding ends k_{end} (18 bits),

- length of compressed location map L (18 bits),

- the compressed location map LM (L bits).

To embed the auxiliary information, LSB of first $69 + L$ bits pixels of the image is recorded to obtain a binary vector V_{LSB} and then replaced by the auxiliary information. V_{LSB} is embedded into pixels after $x_{k_{\text{end}}}$ by the same way as secret data embedding. During the whole embedding procedure, the threshold t_1 is selected from 1 and increased by 1 iteratively and j-th threshold t_j is selected from $t_{j-1} + 1$ and increased by 1 iteratively. The optimal thresholds are obtained by traversing all values of pixel complexities.

The extraction procedure is an opposite process. Firstly, auxiliary information is gotten by reading LSB of first $69 + L$ bits pixels and replaced by LSB of pixels after $x_{k_{\text{end}}}$. Next, for marked pixel value $\tilde{x}$, complexity is computed as the same as embedding procedure by extracting data pixel-by-pixel in reverse

11

order. And, with extracted three complexity thresholds $\{t_1, t_2, t_3\}$, the same context pixels C of $\tilde{x}$ as original pixel value x can be obtained. Then, prediction-error $\tilde{e}$ of the marked pixel value $\tilde{x}$ is computed by (4). And the original pixel value x is recovered by (5). Meanwhile, the secret data b can be extracted that $b = 1$ if $\tilde{e} \in \{-1, 1\}$ or $b = 0$ if $\tilde{e} = 0$. Finally, original image is recovered by LM referring the embedding procedure.

4. Experimental Results

In this section, experimental results of proposed method are presented. The performance is compared with three typical PVO-based methods of Peng *et al.* [27], Ou *et al.* [28] and Qu *et al.* [29]. The proposed method is implemented on MatLab version 2018a on a tower server (SUPERMICRO LT-7038AX) and for a given embedding capacity, embedding procedure can be implemented less than four seconds.

The experimental results are evaluated on eight standard gray-scale 512×512-sized images including Lena, Baboon, Airplane, Barabra, Lake, Boat, Peppers and Eliane. Fig. 9 shows the performance comparison on the eight test images. The capacity increases from 5,000 bits to the largest embedding capacity with a step of 1,000 bits. The figures reveal that the proposed method is superior to other three PVO-basd methods. Specifically, the performance comparison for given capacities of 10,000 bits and 20,000 bits are represented in Table 1 and Table 2. Besides, the optimal parameters (t_1^*, t_2^*, t_3^*) of proposed method are also listed in the tables. It can be seen that the optimal thresholds are different on different test images and differen embedding capacities. Thus, considering that the computational complexity is not high, exhaustively all threshold combinations are feasible.

The improved PVO-based RDH proposed by Peng *et al.* [27] is an improvement of the original PVO RDH [26]. The cover image is first divided into non-overlapping and equal size sub-blocks, and then secret data is embedding into the prediction-errors of largest/smallest value pixels in each sub-block. Compared with the original PVO-based RDH, in the improved PVO-based RDH, the smooth blocks are utilized for data embedding by considering the original order of pixels in each block. Fig. 9 shows that its performance is mediocre compared with other methods. Through Table 1 and Table 2, the proposed method obtain an improvement of PSNR of 0.90 dB and 0.99 dB respectively. The PVO-k proposed by Qu *et al.* [29] is another improvement of the original PVO RDH which aims to make full use of smooth blocks as well. In this method, the blocks with more than one largest/smallest pixel value are utilized by modifying the largest/smallest pixel values concurrent. It shows a better performance than the improved PVO-based method but it is still unsatisfactory. According to Table 1 and Table 2, the proposed method obtain an improvement of PSNR of 0.63 dB and 0.78 dB respectively.

The PPVO-based RDH proposed by Qu *et al.* [29] is different from traditional PVO-based RDH methods. It breaks through the block constraint and prediction process is implemented pixel by pixel. In this method, the embedding procedure is implemented for fifteen times with different number of context pixels from 1 to 15, and number of the best performance is chosen as the final parameter. It means that, whatever the pixel is in a smooth or complex area, the context pixels for prediction is constant. The performance of PPVO achieve a superior performance than traditional PVO-based RDH methods. According to the Table 1 and Table 2, the proposed method considers the impact of the number of context pixels and get a increase of PSNR of 0.57 dB and 0.44 dB respectively.
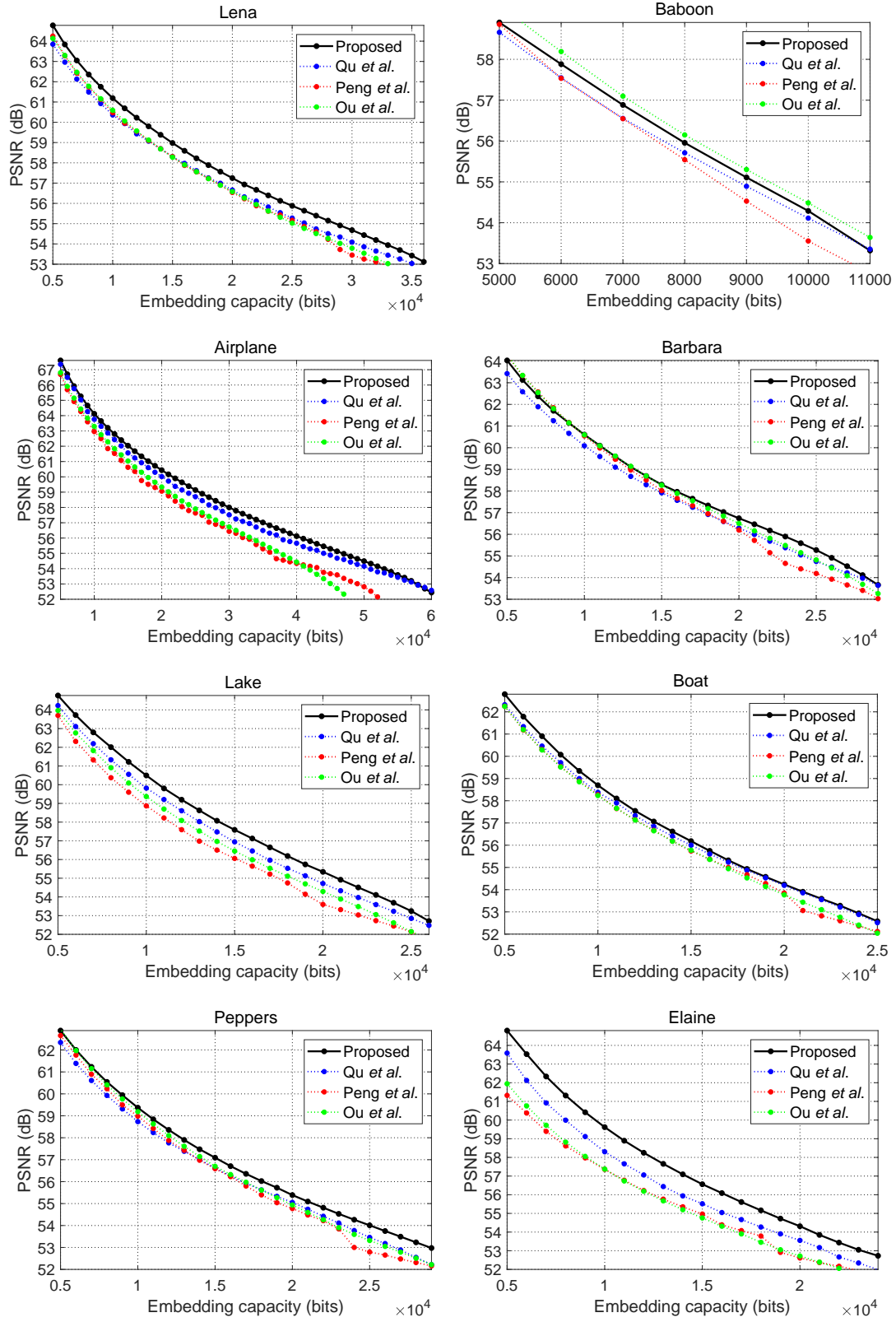
Figure 10: capacity.

13

Table 1: Comparison of PSNR (dB) among the proposed method and the PVO-based methods of Peng *et al.* [27], Ou *et al.* [28] and Qu *et al.* [29]. 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.* [27], Ou *et al.* [28] and Qu *et al.* [29]. 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, based on an extended PPVO predictor and multi-size based embedding method, an extended PPVO-based RDH method is proposed. For each pixel to be predicted, surrounding pixels in left lower and right lower direction are utilized to derive a sharper PEH. And next, the relationship between context region size and local complexity is explored. The pixels in smooth area are predicted by context pixels in a small region size. And for the pixels in complex area, context pixels are chosen by a larger region size. The proposed method is verified that is more satisfactory than other several state-of-the-art PVO-based RDH methods.

Acknowledgement

References

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

[2] 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.

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

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

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

[10] 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.

[11] 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.

[12] 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.

[13] 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.

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

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

[16] 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.

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

[18] 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.

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

[20] 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.

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

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

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

[24] 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.

[25] 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.

[26] 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.

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

[28] 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.

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

[30] 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.

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

[32] 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.

[33] 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.