J. Vis. Commun. Image R. 39 (2016) 12–23

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

High-fidelity reversible data hiding based on pixel-value-ordering and pairwise prediction-error expansion ☆

Bo Ou [a,*], Xiaolong Li [b], Jinwei Wang [c]

[a] College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China
[b] Institute of Computer Science and Technology, Peking University, Beijing 100871, China
[c] School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China

ARTICLE INFO

Article history:
Received 6 February 2016
Revised 1 April 2016
Accepted 10 May 2016
Available online 11 May 2016

Keywords:
Reversible data hiding
Pixel-value-ordering
Pairwise prediction-error expansion
Reversible 2D mapping

ABSTRACT

Pixel-value-ordering (PVO) technique refers to the process of first ranking the pixels in a block and then modifying the maximum/minimum for reversible data hiding (RDH). This paper discusses the PVO embedding in two-dimensional (2D) space and utilizes the prediction-error pair within a block for data embedding. We focus on not only the exploitation of conventional PVO embedding but also its effective implementation in 2D form. The PVO embedding is extended into a 2D form by integrating the pairwise prediction-error expansion, and a reversible 2D mapping adapted to the special distribution of prediction-error pairs is proposed. Moreover, an adaptive mapping selection mechanism is proposed to treat separately rough and smooth prediction-error pairs to further optimize the embedding performance. Experimental results show that the proposed method outperforms the previous PVO-based methods.

1. Introduction

In the digital age, the security of digital works is becoming more and more important with the enormous exchange of information. One of concerns for information security is the content fidelity, and forces a number of applications on the issues of copyright protection, covert communication and authentication [1–11]. To prevent the content from unintended uses, the state-of-the-art approach is to embed information into the medium and provide integrity protection and authentication on the media being transmitted. However, conventional data hiding is mainly concerned on the robustness against to various attacks, and will introduce permanent distortion on the medium because of its inability to recover the original content. Hence, its effectiveness may disappear when the recovery of original content is required. To compensate the weakness, reversible data hiding (RDH) is used to recover both the secret message and the original content, and thereby provides a means to meet a high-level security requirement of applications in the sensitive fields such as military and medical image processing, where the recovery of the original content is desired [4,5].

The goal of RDH is to make the embedded data perceptually invisible to the users while ensuring the resulting modifications to be reversible. So, the challenge of RDH lies in reducing the amount of modifications on the cover media for a given capacity. To some extent, RDH is similar to the lossless compression. Because it aims to transform the original content into a smaller sized representation at first, and then use the saved space to hide the secret message. The key difference is that the transformation process in RDH is required to preserve the image visual appearance without causing perceivable degradations. Up to now, many techniques have been developed for RDH, such as lossless compression [12–14], difference expansion (DE) [15–19], histogram shifting (HS) [20–29], prediction-error expansion (PEE) [30–42] and integer-to-integer transform [43–45].

Among the aforementioned techniques, PEE is the most investigated one due to its capability of capacity and distortion control. It is designed as the integration of DE and HS to better exploit the image redundancy. The embedding performance of PEE is highly dependent on the prediction-error histogram (PEH) on which it is applied, i.e., the more sharply distributes the PEH, the less distortion can be achieved for a given capacity. For this reason, many efforts [31,35,38,46,47] go into the design of advanced predictors that result in accurate predictions. It is worth mentioning that the use of local prediction [47] provides very good results for most of images, at the cost of computing the distinct predictor for every single pixel or pixel group.

☆ This paper has been recommended for acceptance by Zicheng Liu.

* Corresponding author.
E-mail addresses: oubo@hnu.edu.cn (B. Ou), lixiaolong@pku.edu.cn (X. Li), wjwei_2004@163.com (J. Wang).

http://dx.doi.org/10.1016/j.jvcir.2016.05.005
1047-3203/

Another type of PEE-based methods relies on sorting prediction-errors and modifying the small-magnitude ones preferentially. The purpose here is to reduce the overall embedding distortion, by discarding the pixels located in the textured region and turning to use the smooth pixels as many as possible. Sachnev et al. [32] sorted the prediction-errors by the local variances to obtain an excellent performance for low capacities. Li et al. [34] suggested that the prediction-errors with low complexities are able to carry more bits than the others for a given distortion on average. Coatrieux et al. [39] proposed to adaptively modify the pixel according to its local specificity, and determine the most suitable carrier-class for data embedding.

Besides the mainstay of RDH, a recent technique evolving from PEE, called pixel-value-ordering (PVO), is now making RDH easier to exploit image correlations, and can provide good performance at low capacity. Unlike the previous PEE-based methods, PVO-based method predicts a pixel by its nearest gray-level in the neighborhood, and usually embeds data in a block-wise manner. It is based on the nonlinear spatial predictor which yields the prediction according to the ranking result of pixels. Li et al. [48] first proposed a PVO-based RDH, in which the maximum (minimum) pixel in a block is either increased (decreased) or unchanged to hide one bit. Peng et al. [49] improved the embedding mechanism [48] with the help of the spatial information between the pixels. In our previous work [50], we proposed the PVO-*k* algorithm to adaptively modify the block according to the numbers of maximum- and minimum-valued pixels. Recently, instead of embedding data in a block-by-block manner, Qu et al. [51] proposed the pixel-wise PVO to produce a higher capacity while maintaining marked image fidelity. In PVO-based method, a pixel is modified at most by 1, and the lower bound of PSNR for the marked image is guaranteed over 48.13 dB. So, the immediate advantage of PVO embedding is the high-fidelity embedding performance.

Although PVO-based methods have achieved encouraging performances, there are still much room for improvement. Current PVO-based methods are all designed in the one-dimensional (1D) space, and the correlations between the prediction-errors in a block have not been utilized. One improvement to be considered is the high-dimensional representation for PVO embedding. However, the difficulty here is in specifying the high-dimensional modification manner for PVO while ensuring the reversibility. This remains a challenging task. In this paper, we find a new embedding mechanism enabling PVO to fulfill the two-dimensional (2D) modification.

We propose to incorporate the pairwise prediction-error expansion (pairwise PEE [41]) into the design of PVO-based algorithm. We utilize both the intensity and spatial information of ranked pixels for pixel paring within a block, and design a new reversible 2D mapping to adapt with the derived PEH. For each block, the largest two and the smallest two pixels are grouped respectively as two pairs, and then the 3rd largest and the 3rd smallest pixels are used to predict them. After the 2D-PEH is obtained, a reversible 2D mapping is employed to guarantee the reversible modifications of PVO. In particular, to take advantage of the statistical property of derived 2D-PEH, the reversible 2D mapping is designed to put more emphasize on the utilizations of larger-magnitude prediction-errors. Furthermore, in order to achieve adaptive modification on smooth and rough pairs, we propose an adaptive mapping selection mechanism. By measuring the smoothness of pairs, two alternative mapping are utilized to allow a flexible data embedding to further reduce the distortion. The experimental results demonstrate that the proposed method outperforms the previous PVO-based methods [49,50] as well as the typical PEE-based method [32].

The rest of paper is organized as follows. Section 2 briefly reviews the related works for PVO technique. Section 3 presents the proposed method by combining PVO and pairwise PEE with adaptive mapping selection mechanism. Experimental results are provided in Section 4. Finally, Section 5 concludes this paper.

2. Related works

In this section, we first review the basic embedding/extraction mechanism of two previous PVO-based methods [48,49], and then interpret them in the means of 2D-PEH modification.

2.1. Li et al.'s PVO-based method [48]

Unlike the conventional predictors which determine the similarity of two pixels by their spatial distance, PVO-based prediction measures the similarity by their intensities. An example to illustrate this difference is given in Fig. 1. In the figure, to predict the maximum pixel accurately in the 3×3 block, the common way is to use its nearest neighbor for prediction, e.g., its upper or left neighbor, under the assumption that the more closely the two pixels locate, the more similar they are. However, in the complicated region, the most similar pixel to the maximum may locate in a far-away position (as shown in the figure), and the spatial correlation is therefore not reliable for the accurate prediction. To overcome this, the PVO-based prediction defines the optimal context pixel only by the intensity, and can obtain a better prediction for various cases. In [48], both the maximum and minimum pixels are utilized for data embedding. As the modifications for the maximum and minimum pixels are similar, for clarity of presentation, we only describe the data embedding on the maximum pixel of a block as follows.

The cover image I is first divided into non-overlapped blocks with size of $n_1 \times n_2$. For each block, the pixels are scanned in a predefined order as $(p_1, \ldots, p_n)$, where $n = n_1 \times n_2$. Then the sequence is sorted in an ascending order as $(p_{\sigma(1)}, \ldots, p_{\sigma(n)})$, where $\sigma : \{1, \ldots, n\} \to \{1, \ldots, n\}$ denotes the unique one-to-one mapping such that: $p_{\sigma(1)} \leqslant \cdots \leqslant p_{\sigma(n)}, \sigma(i) < \sigma(j)$ if $p_{\sigma(i)} = p_{\sigma(j)}$ and $i < j$. The prediction-error of maximum pixel is computed as

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

where $e_{\max} \geqslant 0$. Then, the marked prediction-error $e'_{\max}$ is obtained as

$$e'_{\max} = \begin{cases} e_{\max}, & \text{if } e_{\max} = 0 \\ e_{\max} + b, & \text{if } e_{\max} = 1 \\ e_{\max} + 1, & \text{if } e_{\max} > 1 \end{cases} \tag{2}$$

where $b \in \{0, 1\}$ is a binary bit. After that, the maximum pixel $p_{\sigma(n)}$ is accordingly changed to

$$p'_{\sigma(n)} = p_{\sigma(n-1)} + e'_{\max} = \begin{cases} p_{\sigma(n)}, & \text{if } e_{\max} = 0 \\ p_{\sigma(n)} + b, & \text{if } e_{\max} = 1 \\ p_{\sigma(n)} + 1, & \text{if } e_{\max} > 1 \end{cases}. \tag{3}$$

Repeat the above modification for blocks one by one until the required capacity is embedded.

At decoders, following the predefined block partition and the scan order, one can obtain the same ranking result for each block. By computing the difference between the marked maximum pixel $p'_{\sigma(n)}$ and the second largest $p'_{\sigma(n-1)}$, one data bit is extracted as 0 and 1 for the cases of $e'_{\max} = 1$ and $e'_{\max} = 2$, respectively. And the original block is recovered by modifying the maximum one $p'_{\sigma(n)}$ in a reverse manner of the embedding process.

In contrast with the conventional PEE-based algorithm where the image is modified pixel-wise, the PVO embedding is processed in a block-wise manner and only the maximum and minimum pixels may be modified in a block. The optimal block size is varied with the capacity. Usually, the larger block size, the higher PSNR

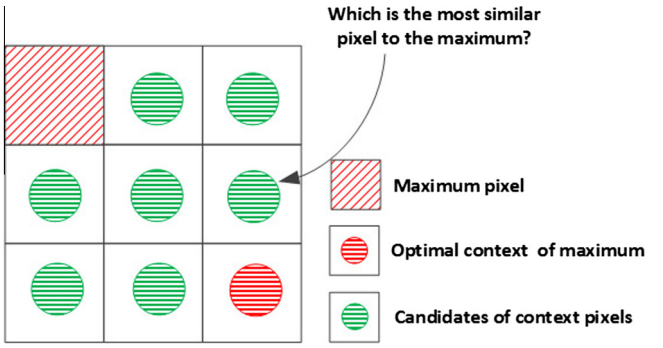

Fig. 1. The optimal context for the prediction of the maximum pixel. It is shown that the determination of the optimal context pixel may not be closely related to the distance between the two pixels.

but the lower capacity. In other words, the embedding distortion can be tuned by adjusting the block size. This enables to compromise the capacity in exchange of the very low embedding distortion.

2.2. Peng et al.'s PVO-based method [49]

Though [48] yields a competitive performance, a drawback is that the bin 0 is not utilized for data embedding. That is, the blocks with $p_{\sigma(n)} = p_{\sigma(n-1)}$ are not used. However, these blocks are usually smooth and beneficial for efficient RDH. To remedy this, Peng et al. [49] proposed to modify the computation of prediction-error by considering the location information. Specifically, instead of computing the prediction-error in a fixed manner, they consider to calculate it according to the spatial order of the largest and second largest pixels as

$$e_{\max} = p_u - p_v \tag{4}$$

where

$$\begin{cases} u = \min(\sigma(n-1), \sigma(n)) \\ v = \max(\sigma(n-1), \sigma(n)) \end{cases}. \tag{5}$$

In this way, the prediction-error $e_{\max}$ would no longer be non-negative, but belongs to $(-\infty, +\infty)$, i.e.,

$$e_{\max} = \begin{cases} p_{\sigma(n)} - p_{\sigma(n-1)} > 0, & \text{if } \sigma(n) < \sigma(n-1) \\ p_{\sigma(n-1)} - p_{\sigma(n)} \leqslant 0, & \text{if } \sigma(n) > \sigma(n-1) \end{cases} \tag{6}$$

The derived PEH is obtained by counting the prediction-errors in all blocks. Then they take the bins 0 and 1 for carrying bits and shift other bins, i.e., the marked prediction-error $e'_{\max}$ is obtained as

$$e'_{\max} = \begin{cases} e_{\max} - b, & \text{if } e_{\max} = 0 \\ e_{\max} + b, & \text{if } e_{\max} = 1 \\ e_{\max} - 1, & \text{if } e_{\max} < 0 \\ e_{\max} + 1, & \text{if } e_{\max} > 1 \end{cases} \tag{7}$$

and the maximum pixel $p_{\sigma(n)}$ is accordingly modified as

$$p'_{\sigma(n)} = p_{\sigma(n-1)} + |e'_{\max}| = \begin{cases} p_{\sigma(n)} + b, & \text{if } e_{\max} \in \{0, 1\} \\ p_{\sigma(n)} + 1, & \text{else} \end{cases}. \tag{8}$$

The recovery and data extraction are similar to that of [48], and are omitted for simplicity.

The comparison of histogram modification manner for [48,49] is shown in Fig. 2, where the modification is represented on the PEH of $e_{\max}$ defined in (4). According to (7), for [49], the prediction-errors 0 and 1 are used to embed data while the other ones are shifted. It can be interpreted as the left sub-image of Fig. 2. By contrast, according to (2) of [48], the difference is that only the prediction-error 1 is expanded, and the one 0 is left unused. Note that although the prediction-error is computed as non-negative in [48], we can still split the prediction-errors into positive and negative parts. The corresponding histogram modification manner is equivalently interpreted as shown in the right of Fig. 2. According to Fig. 2, since the bin 0 has a larger frequency than the bin −1, the method [49] can obtain a higher capacity than [48].

In fact, Peng et al.'s method [49] can also be interpreted using histogram modification for a 2D-PEH as shown in the left of Fig. 3. Here, by using the 3rd largest pixel $p_{\sigma(n-2)}$ as context, one can compute two prediction-errors $e^1_{\max}$ and $e^2_{\max}$ for maximum pixels as

$$\begin{cases} e^1_{\max} = p_u - p_{\sigma(n-2)} \\ e^2_{\max} = p_v - p_{\sigma(n-2)} \end{cases} \tag{9}$$

where p_u and p_v are the largest two pixels defined in (5). In this way, every two prediction-errors derived from the largest two pixels can be modified jointly. More specifically, based on the new defined prediction-error pair, the prediction-error $e_{\max}$ in (4) can be equivalently represented as

$$e_{\max} = e^1_{\max} - e^2_{\max} = p_u - p_v. \tag{10}$$

In this way, the modification on the 1D-PEH of $e_{\max}$ can be equivalently implemented by modifying the 2D-PEH of $(e^1_{\max}, e^2_{\max})$. Such 2D modification manner is described as a reversible 2D mapping, where we can use the mapping arrow to represent the operations of expansion and shifting, i.e., the one-to-many relationship between the original and the marked values denotes the expansion for embedding bits, while the one-to-one mapping is shifting without carrying any bit. Here, specifically, a reversible

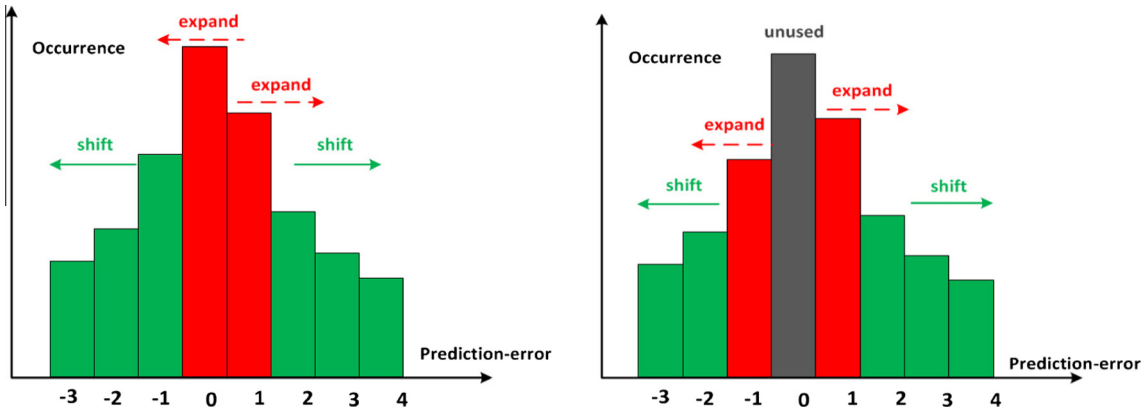

Fig. 2. Modifications of PEH for Peng et al.'s [49] (Left) and Li et al.'s [48] (Right).

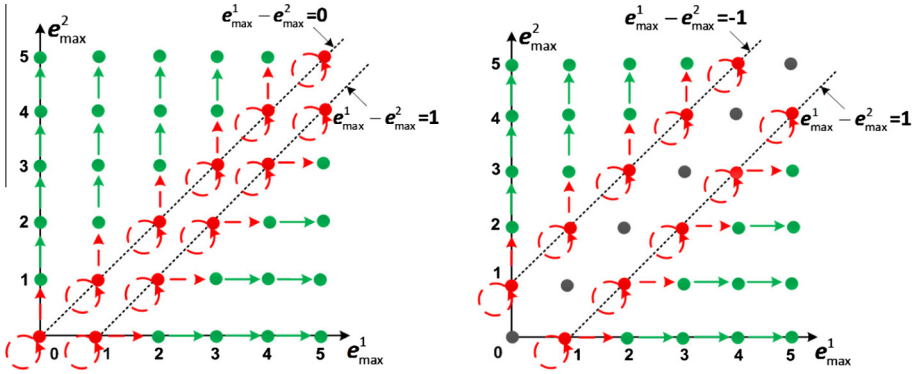

Fig. 3. Modification manner on the 2D-PEH. (Left) Peng et al.'s [49]. (Right) Li et al.'s [48].

2D mapping is a function $f : X \to S(\mathbb{Z}^2)$ such that $f(x_1) \cap f(x_2) = \varnothing$ if $x_1 \neq x_2$, where X is a subset of $\mathbb{Z}^2$ and $S(\mathbb{Z}^2)$ is the power set of $\mathbb{Z}^2$ (i.e., $S(\mathbb{Z}^2)$ is the set composed of all subsets of $\mathbb{Z}^2$). Then, based on (7), the data embedding of [49] can be equivalently implemented in a 2D representation as follows.

- If $e_{max} = 0$, according to (6), we see that $\sigma(n) > \sigma(n-1)$ and $p_{\sigma(n)} = p_{\sigma(n-1)}$. Then, according to (8), $p_{\sigma(n)}$ is either unchanged or modified as $p_{\sigma(n)} + 1$ in data embedding. That is to say, in equivalent embedding in 2D case, the prediction-error pair $(e^1_{max}, e^2_{max}) = (p_{\sigma(n-1)} - p_{\sigma(n-2)}, p_{\sigma(n)} - p_{\sigma(n-2)})$ is either unchanged or modified as $(e^1_{max}, e^2_{max} + 1)$. Notice that, $e^1_{max} = e^2_{max}$, then in this case, each prediction-error pair (k,k) will be modified as (k,k) or $(k,k+1)$ in the 2D-PEH.
- If $e_{max} = 1$, the prediction-error pair is modified as (e^1_{max}, e^2_{max}) and $(e^1_{max} + 1, e^2_{max})$ and one bit is embedded. In the 2D-PEH, each prediction-error pair $(k+1,k)$ is mapped to $(k+1,k)$ or $(k+2,k)$.
- If $e_{max} < 0$, the prediction-error pair is modified as $(e^1_{max}, e^2_{max} + 1)$. In 2D-PEH, each prediction-error pair (k_1,k_2) with $k_1 < k_2$ is mapped to (k_1, k_2+1).
- If $e_{max} > 1$, the prediction-error pair is changed to $(e^1_{max} + 1, e^2_{max})$. Consequently, each prediction-error pair (k_1,k_2) with $k_1 > k_2 + 1$ is mapped to (k_1+1, k_2).

Similarly, by using e^1_{max} and e^2_{max}, the 2D modification manner of [48] can be obtained as shown in the right of Fig. 3. In this case, referring to Fig. 2, the bin 0 is left unused, and the bin -1 is expanded. Consequently, compared with the method [49], the positions of expandable pairs on the 2D-PEH are shifted upwards from the line $e^1_{max} - e^2_{max} = 0$ to the one $e^1_{max} - e^2_{max} = -1$. The prediction-error pairs with $e^1_{max} = e^2_{max}$ (i.e., $e_{max} = 0$) are kept unused, which is the same to the case in the 1D space. From the view-point of 2D-PEH modification, one can see that current employed modification methods are heuristic, and may not be optimal for data embedding. Then, it is possible to improve the embedding performance of PVO by designing a more efficient 2D mapping.

3. Proposed method

In this section, a new PVO-based approach to implement a 2D modification is presented. We first describe the pairwise modification on the maximum pixel pairs, and then propose an adaptive mapping selection mechanism using double-thresholds. After that, the complete algorithm for both maximum and minimum pixel pairs is given at the end of section.

3.1. Pairwise modification for PVO

We follow the same block partition of the previous PVO-based methods. For a $n_1 \times n_2$ sized block, the pixels are sorted in an ascending order by their intensities as $(p_{\sigma(1)}, \ldots, p_{\sigma(n)})$ with $n = n_1 \times n_2$. Then, the largest two pixels $p_{\sigma(n-1)}$ and $p_{\sigma(n)}$ are combined as pair $\mathbf{p}_{max} = (p_u, p_v)$ by their spatial order using (5), and the prediction-errors are computed as $\mathbf{e}_{max} = (e^1_{max}, e^2_{max})$ using (9). Here, by checking which of the two pixels is located first in terms of spatial position, one can determine the order of pixels in a pair reliably. This arrangement is important for decoders to correctly combine pixels into a pair for data extraction and original image restoration. After all prediction-error pairs of blocks are computed, we can obtain the 2D-PEH. Fig. 4 shows the 2D-PEH for the Lena image with block size 2×3. In this figure, the graphical representation of histogram is plotted in the left and the occurrences of histogram bins are given in the right.

One can observe that the 2D-PEH H derived by PVO prediction is specific in two aspects. On one hand, instead of $H(0,0)$, the histogram of PVO is peaked at the value of $(1,1)$. On the other hand, there exists an asymmetrical distribution in H, i.e., $H(k,0) \geqslant H(0,k)$ holds for each $k \geqslant 1$. The two properties do not occur only for Lena image, but are general for various images. We compute the 2D-PEH for seven commonly used images, and show the result in Fig. 5. It is found that the two properties hold as well.

Now, the reasons for the special properties in the PEH of PVO are analyzed as follows. Firstly, in PVO prediction, $(1,1)$ is derived by $p_u = p_v = p_{\sigma(n-2)} + 1$, and $(0,0)$ corresponds to $p_u = p_v = p_{\sigma(n-2)}$. Generally, the prediction-errors of image follow a Laplacian-like distribution with the peak at zero. So in the conventional PEH, the bin 0 is slightly higher than bin 1. But if we impose a constraint that the prediction-error is computed as the absolute value between the pixel and its estimate, the derived histogram would be peaked at 1. In this case, the bin 1 in fact involves the populations of both prediction-errors 1 and -1, and thereby is higher than bin 0. As the prediction-error is also ensured to be non-negative by PVO prediction, the bin 1 is the most populated one in the histogram. This phenomenon, in fact, can also be observed on other images, and holds for most of images [48]. As for the pair histogram, the value $(1,1)$ is inferred as the highest bin.

Secondly, we consider to compare $H(k,0)$ and $H(0,k)$ for $k \geqslant 1$. The pair $(k,0)$ is derived by the largest three pixels $p_{\sigma(n-2)} = p_v = x$ and $p_u = x + k$, while the one $(0,k)$ is referred to $p_{\sigma(n-2)} = p_u = x$ and $p_v = x + k$. For the pixel sequence, no matter where the three pixels are located, one can obtain the relative spatial order among them, which includes three cases: (1) $x \prec x \prec x + k$, (2) $x \prec x + k \prec x$ and (3) $x + k \prec x \prec x$. Here, the symbol $\prec$ denotes

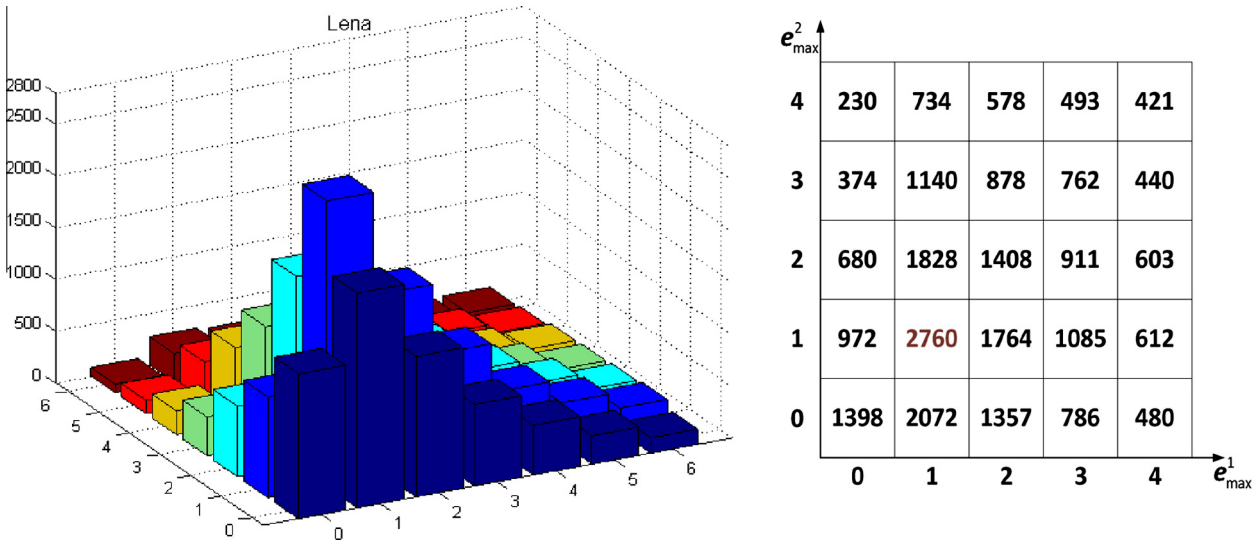

Fig. 4. 2D-PEH generated by the proposed method on Lena image, where the block size is set as 2×3 and only the prediction-errors of maximum pixels are computed. (Left) Graphical representation. (Right) Detailed distribution.

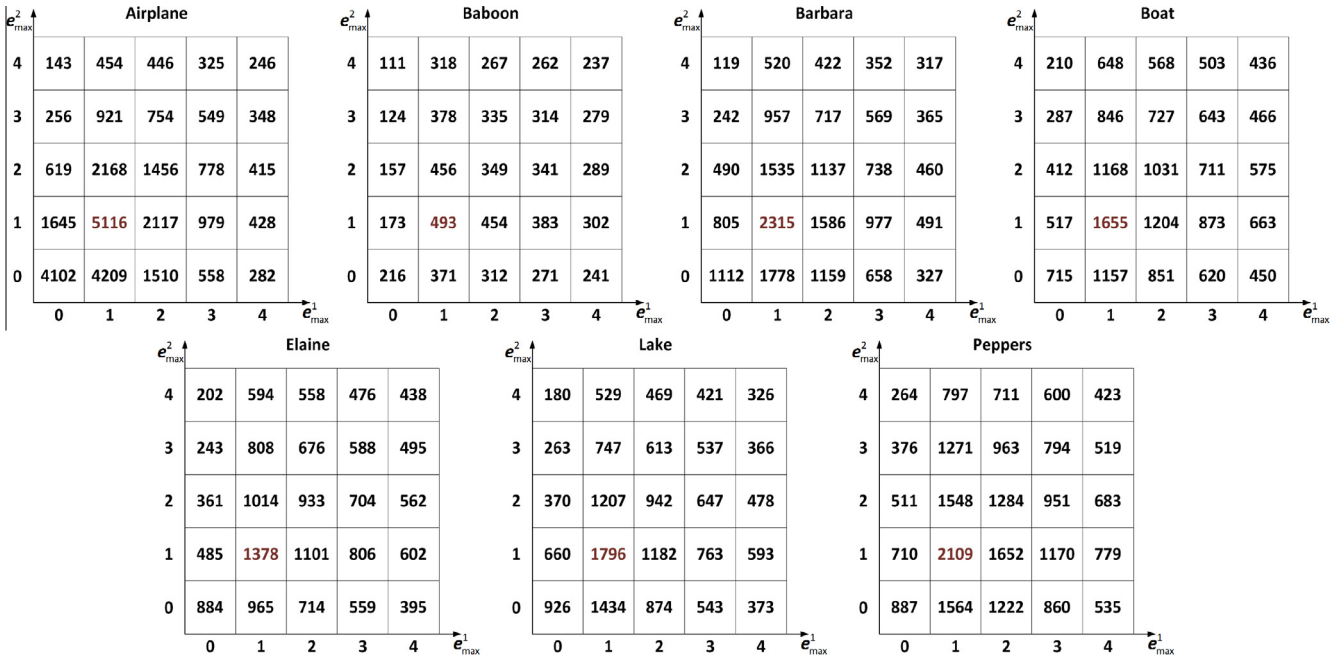

Fig. 5. 2D-PEHs for seven standard images, including Airplane, Baboon, Barbara, Lake, Boat, Elaine and Peppers, where the block size is 2×3 and only the maximum pixels are considered.

the latter element has a larger location than the former one. Intuitively, if we invert the relative orders of case (1), the new permutation is the one in case (3). Case (1) means that $p_{\sigma(n-2)} = p_u = x$ and $p_v = x + k$, and can be deduced a prediction-error pair $(0, k)$. Cases (2) and (3) indicate that $p_{\sigma(n-2)} = p_v = x$ and $p_u = x + k$, and derive a prediction-error pair $(k, 0)$. Notice that the possibilities of case (1) and (3) are equal, then the frequency of pairs with $p_{\sigma(n-2)} = p_v < p_u$ is bigger than that of $p_{\sigma(n-2)} = p_u < p_v$. Consequently, the 2D-PEH bin $H(k, 0)$ has a larger occurrence than its symmetrical counterpart $H(0, k)$, for $k \geqslant 1$. By the analysis, for performance enhancement of PVO, it inspires us to design a new reversible 2D mapping to take advantage of these important characteristics.

3.2. A new reversible 2D mapping for PVO

Now we define the 2D mapping to adapt with this special 2D-PEH as shown in Fig. 6. Our objective is to use the bins near (1,1) as many as possible for carrying bits while leaving the other pairs shiftable. To accomplish this, the prediction-error pair (1,1) is used to carry more than one bit, and the pair which has a prediction-error with value of 1 is used to hide one bit as well. As learned from our previous design [41], we discard the mapping direction from (1,1) to (2,2), and enable the pair (2,2) to carry one bit on average. For the bins (0,0), (1,0) and (0,1), only two can be assigned with two mapping directions for reversibility concerns. Because that (0,0) and (1,0) have a larger frequency than (0,1),

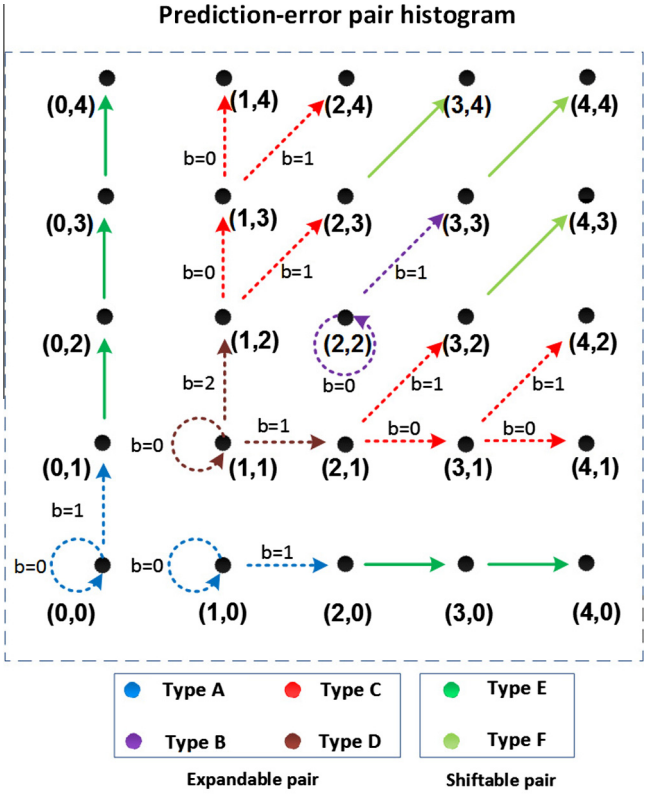

Fig. 6. Proposed reversible 2D mapping for PVO data embedding, where only the design of the first quarter is given as the derived prediction-errors are non-negative. Note that the design for the upper-right zone of 2D-PEH, starting from the point (1,1), is learned from our previous work [41].

the bins (0,0) and (1,0) are defined as expandable while the counterpart (0,1) is shiftable.

For better illustration, the prediction-error pairs are classified as six types as A–F. Among them, the prediction-errors of Types A–D are expanded, while types E and F are shifted. For a prediction-error pair $\mathbf{e}_{\max}$, one can first identify its type and then obtain the marked prediction-error pair $\mathbf{e}'_{\max}$ accordingly. The following examples are given to better illustrate the modifications of prediction-error pair according to Fig. 6.

- If $\mathbf{e}_{\max}=(0,0)$, it is classified as Type A and the marked pair $\mathbf{e}'_{\max}$ is taken as $(0,0)$ and $(0,1)$ to embed 1 bit.
- If $\mathbf{e}_{\max}=(2,2)$, it is classified as Type B. The marked pair is taken as $(2,2)$ and $(3,3)$ to embed 1 bit.
- If $\mathbf{e}_{\max}=(2,1)$, it is classified as Type C. The marked pair is taken as $(3,1)$ and $(3,2)$ to embed 1 bit.
- If $\mathbf{e}_{\max}=(1,1)$, it is classified as Type D and the marked pair is taken as $(1,1),(1,2)$ and $(2,1)$ to embed $\log_2 3$ bits.
- If $\mathbf{e}_{\max}=(0,1)$, it is classified as Type E and it is shifted to $(0,2)$ in data embedding.
- If $\mathbf{e}_{\max}=(2,3)$, it is classified as Type F and it is shifted to $(3,4)$ in data embedding.

Based on the marked prediction-errors, the marked pixels can be obtained as $\mathbf{p}'_{\max}=(p_{\sigma(n-2)},p_{\sigma(n-2)})+\mathbf{e}'_{\max}$. It is noted that, for each block, except the pair of maximum pixels, the other pixels are unmodified in this case, i.e., $p_{\sigma(i)}=p'_{\sigma(i)}$, for $i\in\{1,\ldots,n-2\}$. Moreover, the pixels in maximum pair are guaranteed to be equal to or larger than the others before and after embedding, and the invariant intensity order between the pairs and the other pixels is thus ensured, i.e., $p_{\sigma(1)}\leqslant\cdots\leqslant p_{\sigma(n-2)}\leqslant\{p_{\sigma(n-1)},p_{\sigma(n)}\}$ at encoder and $p'_{\sigma(1)}\leqslant\cdots\leqslant p'_{\sigma(n-2)}\leqslant\{p'_{\sigma(n-1)},p'_{\sigma(n)}\}$ at decoder.

At decoder, since the ranking result between the pairs and the other pixels is the same as that of data embedding, one can find the correct context $p'_{\sigma(n-2)}$ to repeat the above prediction. The original prediction-error pair $\mathbf{e}_{\max}$ is recovered in an inverse 2D mapping of Fig. 6 and the pixel is restored as

$$\begin{aligned}\mathbf{p}_{\max}&=\mathbf{e}_{\max}+(p'_{\sigma(n-2)},p'_{\sigma(n-2)})\\&=(p'_{\sigma(n-2)}+e^1_{\max},p'_{\sigma(n-2)}+e^2_{\max})\end{aligned}\tag{11}$$

where $p'_{\sigma(n-2)}=p_{\sigma(n-2)}$. Accordingly, the data bits are extracted based on Fig. 6 from the marked prediction-error pairs which are derived from the expandable ones.

3.3. Adaptive mapping selection mechanism

The purpose of adaptive mapping selection is to treat the prediction-error pairs differently on the basis of smoothness. For smooth pairs, we adopt the new 2D mapping of Fig. 6 to embed more bits than the conventional one. For moderate textured pairs, we turn to use Peng et al.'s mapping (see the left of Fig. 3) for data embedding, which yields a less capacity than the new one. For the rough pairs, we skip them as most of the corresponding prediction-errors are not expandable. For simplicity, the mappings of Fig. 6 and the left sub-image of Fig. 3 are denoted as f_1 and f_2, respectively.

To achieve adaptive embedding, the smoothness of a pair is measured by the context of block. We consider a $(n_1+2)\times(n_2+2)$ sized window, and use the pixel block's right and bottom neighbors as the context. The same as [50], the noise level NL is taken as the smoothness measurement, and calculated as the sum of absolute differences between every two consecutive pixels in the context, i.e.,

$$NL=\sum d_{ver}+\sum d_{hor}\tag{12}$$

where d_{ver} and d_{hor} denote the vertical and horizontal absolute differences, respectively. An example for the computation of noise level for a 2×3 sized block is given in Fig. 7, where the vertical and horizontal differences are marked with red[1] and blue, respectively.

Next, two thresholds T_1 and T_2 with $T_1<T_2$ are used for the classification of pairs. We only process the blocks whose complexity are less than or equal to the threshold T_2, and skip others. Specifically, the pairs with $NL\leqslant T_1$ are modified according to the mapping f_1, the ones with $T_1<NL\leqslant T_2$ are modified by the mapping f_2. For a prediction-error pair, its applied mapping f is determined as

$$f=\begin{cases}f_1, & \text{if } NL\leqslant T_1\\ f_2, & \text{if } T_1<NL\leqslant T_2\end{cases}.\tag{13}$$

For the parameter adaption of a given capacity EC, we check all possible combinations of T_1 and T_2, and select the ones which fulfill the capacity. After that, the optimal parameters are chosen as the one to yield the least distortion in terms of mean squared error (MSE). That is, the optimal threshold (T_1^*,T_2^*) is determined as

$$\begin{cases}(T_1^*,T_2^*)=\arg\min\limits_{T_1,T_2} D_1(T_1)+D_2(T_1,T_2)\\ \text{subject to } E(f_1)+E(f_2)\geqslant EC\end{cases}\tag{14}$$

where $D_1(T_1)$ and $D_2(T_1,T_2)$ calculate respectively the distortion on the pairs with $NL\leqslant T_1$ and the ones with $T_1<NL\leqslant T_2$, and $E(f)$ represents the capacity for the pairs which follow the mapping f.

[1] For interpretation of color in Fig. 7, the reader is referred to the web version of this article.

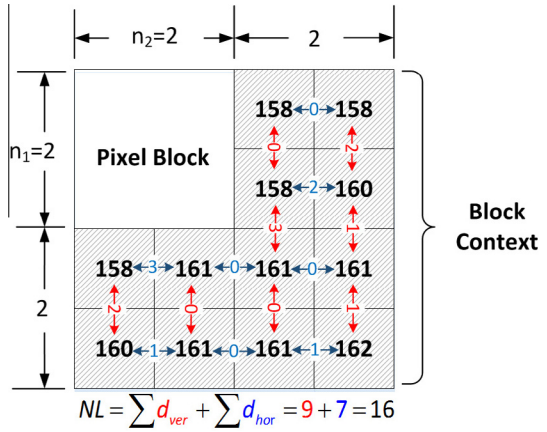

Fig. 7. A 2×3 sized block with its context (shadow pixels), where the noise level NL is computed as the sum of absolute differences between every two consecutive pixels in horizontal and vertical directions.

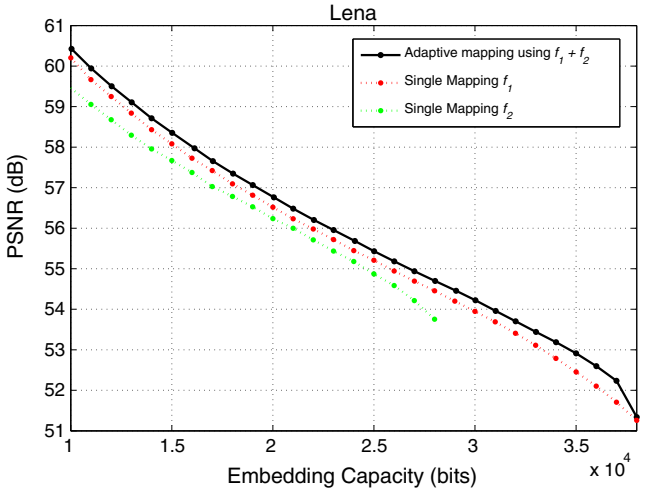

Fig. 8. Performance evaluation by comparing the single mapping of f_1 and f_2 with the adaptive mapping using f_1 and f_2. The mappings f_1 and f_2 are referred to the 2D-PEH modification manners shown in Fig. 6 and the left sub-image of Fig. 3, respectively.

Fig. 8 gives the performance comparison on Lena image by using the adaptive mapping and a single mapping of f_1 and f_2, where the block size is 2×3. The figure shows that the adaptive embedding absorbs the merits of f_1 and f_2, and outperforms an arbitrary one of them no matter what a capacity is. It is verified that the use of adaptive mapping makes the modification of pairs dependent on the smoothness, and treating the smooth and the rough pairs separately can achieve a performance improvement. For the two single mappings, f_1 is superior than f_2 in terms of both PSNR and capacity, because f_1 can better take advantage of the property of PVO, and embeds more bits near the peak. In the proposed method, we adopt this strategy for adaptive embedding on different image areas.

3.4. Implementation of the proposed algorithm

Similar to the modifications on maximum pixels, using the smallest two pixels $p_{\sigma(1)}$ and $p_{\sigma(2)}$, one can obtain the minimum-valued pixel pair $\mathbf{p}_{\min} = (p_u, p_v)$, where the indices u and v are determined as

$$\begin{cases} u = \min(\sigma(1), \sigma(2)) \\ v = \max(\sigma(1), \sigma(2)) \end{cases}. \tag{15}$$

By using $p_{\sigma(3)}$ as prediction context, the prediction-error pair $\mathbf{e}_{\min}$ is computed as

$$\mathbf{e}_{\min} = (e^1_{\min}, e^2_{\min}) = (p_{\sigma(3)} - p_u, p_{\sigma(3)} - p_v) \tag{16}$$

where the derived prediction-errors are non-negative as well. Then, by using the adaptive 2D mapping, modify the marked pairs as $\mathbf{e}'_{\min}$ and obtain the marked minimum pixels as $\mathbf{p}'_{\min} = (p_{\sigma(3)}, p_{\sigma(3)}) - \mathbf{e}'_{\min}$.

The data extraction is the same as that of the maximum pair, and is omitted here for simplicity. After restoring $\mathbf{e}_{\min}$, the original minimum pixel pair is recovered as

$$\mathbf{p}_{\min} = (p_{\sigma(3)}, p_{\sigma(3)}) - \mathbf{e}_{\min} + = \left(p_{\sigma(3)} - e^1_{\min}, p_{\sigma(3)} - e^2_{\min}\right). \tag{17}$$

Now, considering both the maximum and minimum pairs, we have $\{p_{\sigma(1)}, p_{\sigma(2)}\} \leqslant p_{\sigma(3)} \cdots \leqslant p_{\sigma(n-2)} \leqslant \{p_{\sigma(n-1)}, p_{\sigma(n)}\}$ and $\{p'_{\sigma(1)}, p'_{\sigma(2)}\} \leqslant p'_{\sigma(3)} \cdots \leqslant p'_{\sigma(n-2)} \leqslant \{p'_{\sigma(n-1)}, p'_{\sigma(n)}\}$ at encoder and decoder, respectively. Meanwhile, the medium-valued pixels $p_{\sigma(i)}, i \in \{3, \ldots, n-2\}$ are unmodified. The complete illustration of the modification on a block is presented in Fig. 9, where the reversibility is ensured by checking the order between the pairs and the other pixels.

To better illustrate the embedding process, an example of the proposed 2D PVO embedding for the 3×3 sized block is given in Fig. 10. Here, the block's noise level is first calculated and compared with a given threshold. Suppose here the noise level $NL_i = 5$ and the threshold $T_1 = 15$. Because $NL_i \leqslant T_1$, the mapping f_1 is applied to this block. In the figure, the location for each pixel is determined by scanning from left to right and top to bottom. Then the pixels are transformed into one-dimensional sequence in the ascending order of intensity according to the mapping σ. The largest two pixels (i.e., p_4 and p_7) and the smallest two pixels (i.e., p_5 and p_6) are combined into two pairs, respectively. The maximum pixel pair is obtained as $\mathbf{p}_{\max} = (p_4, p_7)$ and the minimum one is $\mathbf{p}_{\min} = (p_5, p_6)$. So we have prediction-error pairs $\mathbf{e}_{\max} = (1, 0)$ and $\mathbf{e}_{\min} = (1, 1)$, which are expandable and belonged to type A and type D according the classification in Fig. 6, respectively. Assume the binary and ternary bits be "1" and "2", the marked pixel pairs are obtained as $\mathbf{e}'_{\max} = (2, 0)$ and $\mathbf{e}'_{\min} = (1, 2)$ according the mapping of Fig. 6. Finally, the pixel pairs (p_4, p_7) and (p_5, p_6) are changed as (167,165) and (159,158), respectively.

For recovery and data extraction, by comparing the noise level with the threshold T_1, the same mapping can be retrieved for decoders. Next, one can sort the pixels into the sequence as shown in the bottom of the figure. The largest two marked pixels p'_4 and p'_7 are taken as the marked pair $\mathbf{p}'_{\max} = (p'_4, p'_7)$. Similarly, the minimum pair is obtained as $\mathbf{p}'_{\min} = (p'_5, p'_6)$. It can be seen that the ranking order between p_5 and p_6 in terms of intensity is exchanged after embedding, but one can still construct the same pair by referring to their spatial orders. Then the prediction-error pairs are computed as $\mathbf{e}'_{\max} = (2, 0)$ and $\mathbf{e}'_{\min} = (1, 2)$, respectively. Based on the inverse 2D mapping shown in Fig. 6, the original prediction-error pairs are recovered and the data bits are extracted as a binary "1" and a ternary bit "2". The recovery of $\mathbf{p}_{\max}$ and $\mathbf{p}_{\min}$ is implemented using the recovered prediction-errors. In this way, the data extraction and the pixel recovery for the block is completed.

Now we describe the procedures of data embedding/extraction in details. To address the overflow/underflow problem, a location map is used to ensure the pixel value falls into the range $[0, 255]$ after embedding. The location map is constructed in a block-wise manner, and initialized to the flag "0" for each block. Then, we scan the blocks in the raster-scan order and change the flag value to "1" if the block has a pixel with a value of 0 or 255. The block marked with "1" is regarded as unavailable and will be ignored during data embedding and extraction. It should be noted that, the location map size depends on the number of blocks. To further reduce the

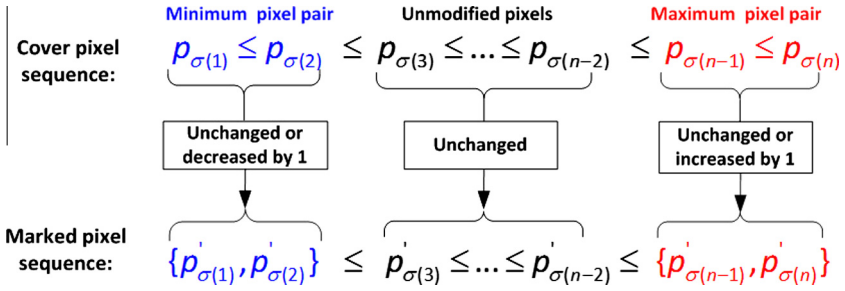

Fig. 9. An illustration to show the change of intensity order before and after modifications, where the minimum and maximum pixel pairs are highlighted by blue and red, respectively. By our method, the intensity order between the pairs and the medium-valued pixels is guaranteed to be invariant. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

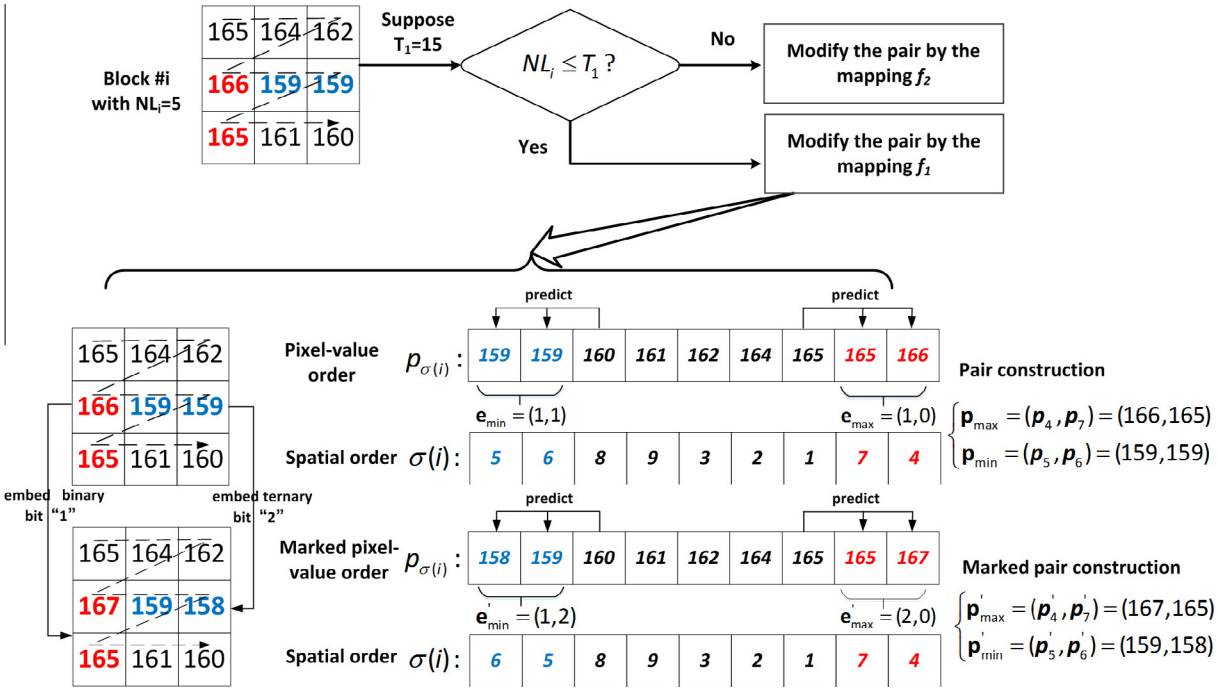

Fig. 10. An example of the proposed PVO embedding on the 3×3 sized block using the adaptive mapping. The alternative mapping of a pair is first determined by checking whether the noise level is equal to or smaller than a given threshold. Then, the modification of the pair follows the employed mapping both at encoder and decoder.

length, the location map is losslessly compressed, and its length is denoted as L bits for better illustration. For a commonly used standard image with size of 512×512, the maximum length of compressed location map is only dozens of bits. Besides the location map, some auxiliary information including the following components should be recorded as well.

- Block size n_1 and n_2 (4 bits).
- Thresholds T_1 and T_2 (20 bits).
- End position (18 bits).
- Compressed location map (L bits).

The auxiliary information is embedded in the first two rows of image by least-significant-bit (LSB) replacement, so that the decoder can extract them in advance to facilitate the data extraction and recovery. The replaced LSBs of pixels will be taken as a part of payload, and embedded into the left region of image. The payload PS consists of two parts: secret message, and the replaced LSBs ($42 + L$ bits). The complete data embedding is given as follows.

- **Step 1. Preparation of auxiliary information:** Empty the LSBs of the first two rows of pixels in the image to prepare the space for auxiliary information. Append the replaced LSBs as a part of payload in order to restore the original first-two-row pixels at decoder.
- **Step 2. Embedding for payload bits:** Construct the location map to mark the unavailable blocks, and compute the corresponding noise level according to (12). The location map is losslessly compressed with the size L.
- **Step 3.** For a block, sort the pixels by intensity. If the flag value is equal to "1", skip this block; else group the largest two and the smallest two pixels into pairs $\mathbf{p}_{\max}$ and $\mathbf{p}_{\min}$, respectively, and compute the prediction-errors $\mathbf{e}_{\max}$ and $\mathbf{e}_{\min}$.
- **Step 4.** For a pair, if the noise level $NL \leqslant T_1$, modify it according to the mapping f_1; if $T_1 < NL \leqslant T_2$, modify it by the mapping f_2; else, keep the pair unchanged. Based on a 2D mapping, modify the prediction-error pair sequence and embed the data bits into the expandable pairs. The marked pixels are obtained by using $\mathbf{e}'_{\max}$ and $\mathbf{e}'_{\min}$.

- **Step 5. Embedding for auxiliary information:** Repeat the embedding until the *PS* is satisfied. Record the compressed location map, the end position, two thresholds $\{T_1, T_2\}$, and the block size $\{n_1, n_2\}$ as the auxiliary information. Embed them into the LSBs of the first-two-row pixels. Finally, the marked image is obtained.

For data extraction and recovery, the blocks are processed in an inverse scan order of data embedding, i.e., from bottom to top, left to right. In this way, the same noise level for each block can be obtained. The corresponding procedure is given as follows.

- **Step 1. Extract auxiliary information:** Extract the LSBs of the first-two-row pixels, and decode them into parameters $\{n_1, n_2\}$, the thresholds $\{T_1, T_2\}$, the end position, and the compressed location map, respectively.
- **Step 2. Payload extraction and block recovery:** Divide the marked image into $n_1 \times n_2$ sized blocks. Then, for each block, check the flag in the location map. For a block, sort the pixels in an ascending order in terms of intensity. If the flag equals to 1, skip the block and go to the next one; otherwise, construct the two pixel pairs $\mathbf{p}'_{\max}$ and $\mathbf{p}'_{\min}$, where the pixel order within a pair is referred to their spatial orders.
- **Step 3.** Compute the marked prediction-error pairs $\mathbf{e}'_{\max}$ and $\mathbf{e}'_{\min}$. For each prediction-error pair, determine its 2D mapping and recover its original value by the inverse mapping. Extract the data bit with the help of identification of its type.
- **Step 4. Recovery of marginal pixels:** Repeat Step 3 until the end position is encountered. Split the extract data bits into the secret message and the LSB stream. Replace the LSBs of the first two rows of pixels with the LSB stream. Finally, both the original image and secret data are recovered.

4. Experimental results

This section evaluates the embedding performance of the proposed method in comparison with three state-of-the-art methods of Sachnev et al. [32], Peng et al. [49] and Ou et al. [50]. All the compared methods focus on the ability of reducing the embedding distortion at low capacity. Among them, Sachnev et al.'s method is a typical PEE-based algorithm, which considers the data embedding in the 1D space and modifies the prediction-errors individually. Peng et al.'s and Ou et al.'s methods are the conventional PVO-based algorithms, and demonstrated high-fidelity performances for low capacity cases. Here, as the proposed method is the extension of PVO embedding, we mainly focus the comparisons with the PVO-based methods [49,50]. Our experiments are all conducted on the gray-scale images.

For PVO-based methods, the trade-off between fidelity and capacity is controlled by adjusting the block size. This inspires us to first investigate the effect of block size on the embedding performance in Fig. 11. In our method, the block size is set as $n_1, n_2 \in \{2, 3, 4, 5\}$, and there are totally 15 candidates. The possible pixel numbers within a block is from 6 up to 25. Note that here, in order to yield two prediction-error pairs in a block, the block size $n = n_1 \times n_2$ must satisfy that $n \geqslant 5$. For a given payload *PS*, the best block size $n_1 \times n_2$ is determined as the one which introduces the least average embedding distortion per bit. However, the relationship between the performance and the block size is not easy to estimate accurately because the correlations of block pixels are implicit, and the optimal size, like other block based data embedding, is highly dependent on the image content. In our experiment, the block-size adaptation is solved by checking all possible combinations, and then finding the best one.

Since some block sizes are very small, they may be redundant and yield similar predictions. In Fig. 11, we only depicts the embedding performances for five different block sizes. It can be seen that the PSNR for a given capacity varies with block size, and the best performance relies on choosing the optimal block partition. At low capacity, almost all the block sizes are able to provide the sufficient capacity. The highest PSNR is obtained with the block size of 3×3 rather than the largest one 5×5, and the number of pixels around 10 is a good choice for the trade-off in this case. The smallest block size is reached until the larger capacity is required. It is found that the horizontal block partition 3×2 is better than the vertical one 2×3 for both the images. This indicates that a more flexible block partition that considers the direction of local correlations may lead to a further improvement on the PVO-based embedding.

Now, we give the performance comparison in terms of distortion-capacity curve on eight 512×512 sized gray-scale images as shown in Fig. 12. The figure shows that the proposed method can yield the highest fidelity of marked image than the other three methods in most cases. As both the accurate prediction and 2D mapping are adopted in our method, it is not surprising that the significant gain in PSNR is obtained compared with the 1D PEE algorithm [32]. It must be noted that the PVO prediction is based on the order statistic in a block, and is sensitive to the abrupt changes of gray-level. So, the effectiveness of PVO prediction is likely to be reduced for the texture image as the neighboring gray-levels within a block are less correlated. It can be seen that for Baboon image, the proposed method is not very effective when the

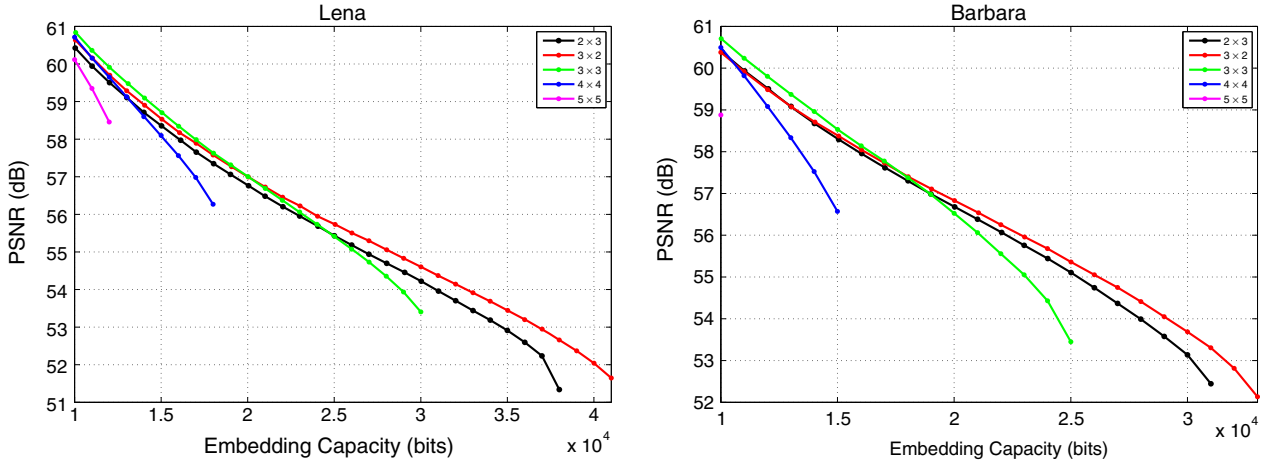

Fig. 11. Performance comparison by using different block sizes.

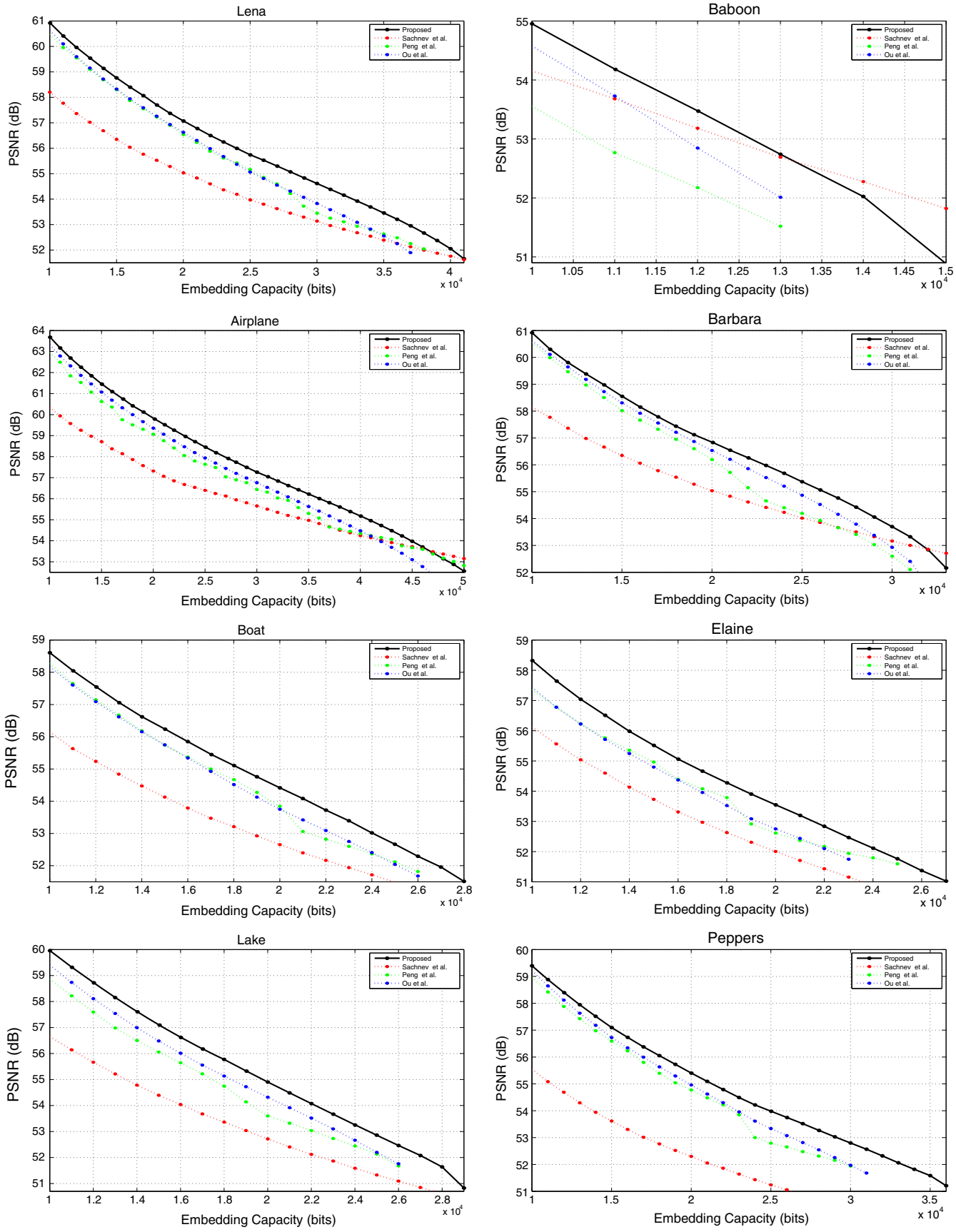

Fig. 12. Performance comparison between our method and four methods of Sachnev et al. [32], Li et al. [19], Peng et al. [49], and Ou et al. [50].

Table 1
Optimal parameters of the proposed method from the capacity of 10,000 to 40,000 bits, where the test image is Lena.

Capacity (bits)	PSNR (dB)	T_1	T_2	n_1	n_2
10,000	60.91	76	92	5	3
15,000	58.75	68	84	5	2
20,000	57.07	61	84	4	2
25,000	55.73	46	77	3	2
30,000	54.60	57	94	3	2
35,000	53.45	70	147	3	2
40,000	52.04	126	379	3	2

Table 2
Optimal parameters on the other test images for the capacity of 10,000 bits.

Images	PSNR (dB)	T_1	T_2	n_1	n_2
Baboon	54.92	145	281	3	2
Airplane	63.68	14	21	2	3
Barbara	60.86	68	80	3	4
Boat	58.61	60	124	2	4
Elaine	58.32	55	108	3	2
Lake	59.92	50	78	2	3
Peppers	59.37	84	119	3	4

Table 3
Performance comparison on Kodak image database between our method and four methods of Sachnev et al. [32], Peng et al. [49], and Ou et al. [50], for a capacity of 10,000 bits.

Image	Sachnev et al.	Peng et al.	Ou et al.	Proposed
kodim01	61.86	62.83	61.40	63.52
kodim02	61.83	64.06	62.92	64.26
kodim03	63.53	65.35	63.97	65.44
kodim04	61.88	63.55	62.58	64.09
kodim05	61.74	61.65	60.74	62.27
kodim06	65.33	65.01	63.26	66.09
kodim07	63.04	65.10	63.74	65.11
kodim08	58.44	56.48	56.60	59.87
kodim09	60.53	63.69	62.87	63.84
kodim10	60.65	63.14	62.00	63.44
kodim11	62.86	65.20	63.55	65.26
kodim12	62.73	64.47	63.49	64.74
kodim13	57.79	53.88	54.74	58.29
kodim14	60.83	61.29	60.66	62.06
kodim15	62.31	64.72	63.18	65.61
kodim16	63.01	64.95	63.54	65.07
kodim17	61.58	63.72	62.71	64.07
kodim18	58.82	60.73	60.44	61.32
kodim19	60.75	63.36	62.68	63.40
kodim20	53.40	64.91	54.06	65.53
kodim21	60.70	63.62	62.88	63.87
kodim22	60.31	62.55	62.11	63.07
kodim23	62.27	64.58	63.45	64.89
kodim24	58.29	61.51	60.54	63.18
Average	61.02	62.93	62.00	63.68

capacity is more than 13,000 bits, and becomes worse than the method [32] at the tail of curve. Compared with the previous PVO-based methods [49,50], our method yields a closed performance for the very low capacity, but gradually obtains a bigger PSNR gain as the capacity goes larger. Once the capacity increases, the small block size has to be chosen. Since the maximum capacity for a block is increased to $\log_2 3$ bits, the proposed mapping enables a larger embedding capacity than the conventional PVO methods. By the proposed 2D mapping, one can still use the large-sized block to provide the required capacity, and fewer rough blocks are utilized for data embedding as the smooth blocks are sufficient to satisfy the large capacity.

In particular, for the case of 10,000 bits, the average PSNRs of the proposed method, Peng et al. [49] and Ou et al. [50] are 59.62, 58.88 and 59.17 dB, respectively. In this case, our gains in PSNR are 0.74 and 0.45 dB, respectively. For 20,000 bits, as Baboon image cannot be embedded with such a high capacity, the average PSNR is computed based on the left seven images. The results for the proposed method and the two compared PVO-based methods are 56.02, 55.23 and 55.47 dB, respectively. The corresponding PSNR gains by using our method are 0.79 and 0.55 dB, respectively. Our gains are mainly due to the utilization of the adaptive 2D mapping, thus allowing to a more flexible data embedding. To show the parameter adaption with capacity, the details of optimal parameters using the proposed method from a capacity of 10,000 to 40,000 bits are given in Table 1. As the capacity increases, the optimal block size goes smaller. Because the mapping f_1 can embed more bits per pixel on average than f_2. For the same block size, to increase the capacity, T_1 is set larger to enable more pairs be modified by f_1. T_2 is thus increased to involve sufficient pairs for f_2 as the frequencies of pairs get less with the increase of noise level. It is noted that, for a fair comparison, the noise level is normalized into the range $[0, 1024]$ for different block sizes. The optimal parameters for the other seven images are given in Table 2, where the capacity is 10,000 bits. For the smooth image, such as Airplane, both T_1 and T_2 are small to fulfill the capacity, and the amount of total modifications is less. While for the rough image Baboon, the double thresholds are set larger to include more pairs for data embedding. Hence, its derived PNSR decreases.

To further validate the adaptability for various images, the proposed method is tested on Kodak image database which contains 24 color images with the size of 512×768 or 768×512. Here, the color images are first transformed into gray-scale version, and then used for test. Table 3 lists the PSNR comparisons between the proposed method and the four compared methods for the capacity of 10,000 bits, where the best two results for an image is marked bold. Given a capacity, the proposed method yields the highest average PSNR. Compared with three methods of Sachnev et al., Peng et al. [49] and Ou et al. [50], the PSNR gains by using the proposed method for the capacity of 10,000 bits are 2.68, 0.75 and 1.68 dB, respectively. It should be noted that several Kodak images have a large number of boundary valued pixels (with values 0 or 255), consuming the embedding space for saving the location map. So the block-wise location map employed in the proposed method and Peng et al. [49] can save more space for data embedding than the pixel-wise location map employed in [50].

5. Conclusions

In this work, we develop the pixel-value-ordering (PVO) based algorithm into the two-dimensional space by jointly considering the largest two and the smallest two pixels in a block. In the 2D space, we observe two special properties of PVO and conduct an analysis on them. Based on the analysis, a new 2D mapping that aims to take advantage of the two properties of PVO is then designed. Moreover, by treating the pairs differently according to their smoothness, an adaptive mapping selection mechanism is proposed to automatically choose the mapping for smooth and the rough pairs. By adaptive embedding, a more flexible modification manner on the derived PEH is conducted, leading to an enhancement of the overall embedding performance. Experimental results on images manifest the proposed method can yield a more satisfactory performance than the previous PVO-based methods [49,50] as well as one other state-of-the-art method [32].

As a future work, a more adaptive 2D embedding mechanism that encapsulates multiple 2D mappings to coincide with the special property of PVO embedding will provide an improved performance. Besides, a theoretical analysis to assess the embedding capability of different block sizes may provides further insights

of PVO embedding into the exploitations of correlations within an image.

Acknowledgement

This work is supported by the National Science Foundation of China (Nos. 61502160, 61572052 and 61272421), the PAPD fund and the CICAEET fund.

References

[1] I. Cox, M. Miller, J. Bloom, J. Fridrich, T. Kalker, Digital Watermarking and Steganography, Morgan Kaufmann, 2007.
[2] J. Fridrich, Steganography in Digital Media: Principles, Algorithms, and Applications, Cambridge University Press, 2009.
[3] H.T. Sencar, N. Memon, Digital Image Forensics: There is More to a Picture than Meets the Eye, Springer Science & Business Media, 2012.
[4] Y.Q. Shi, Z. Ni, D. Zou, C. Liang, G. Xuan, Lossless data hiding: fundamentals, algorithms and applications, Proc. IEEE ISCAS 2 (2004) 33–36.
[5] R. Caldelli, F. Filippini, R. Becarelli, Reversible watermarking techniques: an overview and a classification, EURASIP J. Inf. Secur. 2010 (2010) 2.
[6] B. Li, J. He, J. Huang, Y.Q. Shi, A survey on image steganography and steganalysis, J. Inf. Hiding Multimedia Signal Process. 2 (2) (2011) 142–172.
[7] T. Filler, J. Judas, J. Fridrich, Minimizing additive distortion in steganography using syndrome-trellis codes, IEEE Trans. Inf. Forens. Secur. 6 (3) (2011) 920–935.
[8] J. Fridrich, J. Kodovskỳ, Rich models for steganalysis of digital images, IEEE Trans. Inf. Forens. Secur. 7 (3) (2012) 868–882.
[9] Z. Xia, X. Wang, X. Sun, Q. Wang, Steganalysis of least significant bit matching using multi-order differences, Secur. Commun. Networks 7 (8) (2014) 1283–1291.
[10] Z. Xia, X. Wang, X. Sun, Q. Wang, A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data, IEEE Trans. Parallel Distrib. Syst. 10 (9) (2015) 2016–2027.
[11] J. Li, X. Li, B. Yang, X. Sun, Segmentation-based image copy-move forgery detection scheme, IEEE Trans. Inf. Forens. Secur. 10 (9) (2015) 2016–2027.
[12] J. Fridrich, M. Goljan, R. Du, Lossless data embedding – new paradigm in digital watermarking, EURASIP J. Appl. Signal Process. 2002 (2) (2002) 185–196.
[13] G. Xuan, J. Zhu, J. Chen, Y.Q. Shi, Z. Ni, W. Su, Distortionless data hiding based on integer wavelet transform, IEE Electron. Lett. 38 (25) (2002) 1646–1648.
[14] M.U. Celik, G. Sharma, A.M. Tekalp, Lossless watermarking for image authentication: a new framework and an implementation, IEEE Trans. Image Process. 15 (4) (2006) 1042–1049.
[15] J. Tian, Reversible data embedding using a difference expansion, IEEE Trans. Circuits Syst. Video Technol. 13 (8) (2003) 890–896.
[16] A.M. Alattar, Reversible watermark using the difference expansion of a generalized integer transform, IEEE Trans. Image Process. 13 (8) (2004) 1147–1156.
[17] W.L. Tai, C.M. Yeh, C.C. Chang, Reversible data hiding based on histogram modification of pixel differences, IEEE Trans. Circuits Syst. Video Technol. 19 (6) (2009) 906–910.
[18] 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.
[19] 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.
[20] Z. Ni, Y.Q. Shi, N. Ansari, W. Su, Reversible data hiding, IEEE Trans. Circuits Syst. Video Technol. 16 (3) (2006) 354–362.
[21] S.K. Lee, Y.H. Suh, Y.S. Ho, Reversible image authentication based on watermarking, in: Proc. IEEE ICME, 2006, pp. 1321–1324.
[22] M. Fallahpour, Reversible image data hiding based on gradient adjusted prediction, IEICE Electron. Exp. 5 (20) (2008) 870–876.
[23] L. Luo, Z. Chen, M. Chen, X. Zeng, Z. Xiong, Reversible image watermarking using interpolation technique, IEEE Trans. Inf. Forens. Secur. 5 (1) (2010) 187–193.
[24] Y.-C. Li, C.-M. Yeh, C.-C. Chang, Data hiding based on the similarity between neighboring pixels with reversibility, Digit. Signal Process. 20 (4) (2010) 1116–1128.
[25] H.-T. Wu, J. Huang, Reversible image watermarking on prediction errors by efficient histogram modification, Signal Process. 92 (12) (2012) 3000–3009.
[26] Y.-Y. Tsai, D.-S. Tsai, C.-L. Liu, Reversible data hiding scheme based on neighboring pixel differences, Digit. Signal Process. 23 (3) (2013) 919–927.
[27] 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.
[28] Z. Pan, S. Hu, X. Ma, L. Wang, Reversible data hiding based on local histogram shifting with multilayer embedding, J. Vis. Commun. Image Represent. 31 (2015) 64–74.
[29] H.-T. Wu, J. Huang, Y.-Q. Shi, A reversible data hiding method with contrast enhancement for medical images, J. Vis. Commun. Image Represent. 31 (2015) 146–153.
[30] D.M. Thodi, J.J. Rodriguez, Expansion embedding techniques for reversible watermarking, IEEE Trans. Image Process. 16 (3) (2007) 721–730.
[31] 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.
[32] 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.
[33] 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.
[34] 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.
[35] D. Coltuc, Improved embedding for prediction-based reversible watermarking, IEEE Trans. Inf. Forens. Secur. 6 (3) (2011) 873–882.
[36] Y.-C. Lin, Reversible data-hiding for progressive image transmission, Signal Process.: Image Commun. 26 (10) (2011) 628–645.
[37] W. Hong, Adaptive reversible data hiding method based on error energy control and histogram shifting, Opt. Commun. 285 (2) (2012) 101–108.
[38] B. Ou, X. Li, Y. Zhao, R. Ni, Reversible data hiding based on PDE predictor, J. Syst. Softw. 86 (10) (2013) 2700–2709.
[39] G. Coatrieux, W. Pan, N. Cuppens-Boulahia, F. Cuppens, C. Roux, Reversible watermarking based on invariant image classification and dynamic histogram shifting, IEEE Trans. Inf. Forens. Secur. 8 (1) (2013) 111–120.
[40] C. Qin, C.-C. Chang, Y.-H. Huang, L.-T. Liao, An inpainting-assisted reversible steganographic scheme using histogram shifting mechanism, IEEE Trans. Circuits Syst. Video Technol. 23 (7) (2013) 1109–1118.
[41] 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.
[42] 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.
[43] D. Coltuc, J.M. Chassery, Very fast watermarking by reversible contrast mapping, IEEE Signal Process. Lett. 14 (4) (2007) 255–258.
[44] X. Wang, X. Li, B. Yang, Z. Guo, Efficient generalized integer transform for reversible watermarking, IEEE Signal Process. Lett. 17 (6) (2010) 567–570.
[45] D. Coltuc, Low distortion transform for reversible watermarking, IEEE Trans. Image Process. 21 (1) (2012) 412–417.
[46] X. Ma, Z. Pan, S. Hu, L. Wang, High-fidelity reversible data hiding scheme based on multi-predictor sorting and selecting mechanism, J. Vis. Commun. Image Represent. 28 (2015) 71–82.
[47] I.-C. Dragoi, D. Coltuc, Local-prediction-based difference expansion reversible watermarking, IEEE Trans. Image Process. 23 (4) (2014) 1779–1790.
[48] 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.
[49] F. Peng, X. Li, B. Yang, Improved PVO-based reversible data hiding, Digit. Signal Process. 25 (2014) 255–265.
[50] 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.
[51] X. Qu, H.J. Kim, Pixel-based pixel value ordering predictor for high-fidelity reversible data hiding, Signal Process. 111 (2015) 249–260.