Journal of Visual Communication and Image Representation 49 (2017) 351–360

Contents lists available at ScienceDirect

Journal of Visual Communication and Image Representation

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

Reversible data hiding using multi-pass pixel value ordering and prediction-error expansion☆

Wenguang He, Ke Zhou, Jie Cai, Long Wang, Gangqiang Xiong*

School of Information Engineering, Guangdong Medical University, Guangdong 524023, China

ARTICLE INFO

Keywords:
Reversible data hiding
Multi-pass pixel value ordering
Prediction-error expansion
Optimal combined embedding

ABSTRACT

Pixel value ordering (PVO) prediction has become the most efficient method for high-fidelity reversible data hiding (RDH). In this approach, only the maximum and minimum of pixel block are predicted and modified to embed data and the preservation of pixel value order guarantees the reversibility. To achieve larger embedding capacity and superior performance, more blocks suitable for RDH are utilized in recent improved schemes. However, their performance is still unsatisfactory. In this paper, a novel RDH scheme is proposed by extending original PVO into multi-pass PVO embedding. Specially, the k largest or smallest pixels are taken as independent data bit carriers to fulfill k-pass PVO embedding. Although the pixel value order may change after data embedding, reversibility still can be guaranteed and image redundancy can be far better exploited. Moreover, embedding performance can be further enhanced by optimal combined embedding. Experimental results verify that the proposed scheme outperforms previous PVO-based schemes and some other state-of-the-art works.

1. Introduction

Data hiding [1] can be used in various applications such as copyright protection and authentication by embedding secret message into the host media. Due to the extra ability to recover the host media exactly as well as extracting the embedded data [2], reversible data hiding (RDH) has become an most important branch of data hiding and has been applied to many quality sensitive fields such as medical imaging, remote sensing and military. So far, RDH schemes reported in literatures can be categorized as lossless compression based, histogram modification based and difference expansion based.

Early schemes are mostly based on lossless compression, where embedding space is created by losslessly compressing a feature set of the host image [3,4]. However, the efficiency of lossless compression based RDH is generally limited due to the poor compression ratio. Afterwards, more efficient RDH schemes based on histogram modification and expansion technique have been devised. In 2003, Tian et al. [5] firstly proposed the idea of difference expansion (DE), which embeds one bit of data to the expanded difference between adjacent pixels. Although the embedding capacity (EC) is limited by 0.5 bits-per-pixel in one layer embedding, DE provides larger EC with less distortion when compared with lossless compression and thus has been widely investigated and developed, mainly in the aspects of inter-to-integer transformation [6–8] and prediction error expansion (PEE) [9–11]. Introduced by Thodi et al. [9] in 2007, prediction error expansion (PEE) is commonly treated as the most effective extension of DE where the target pixel is predicted by context pixels and then the prediction error is utilized for expansion embedding. As better predictor produces smaller errors which means less distortion, prediction method plays an important role in PEE-RDH. So far, many efficient prediction methods are investigated, including median edge detection [9,10], rhombus prediction [12], interpolation [13], and gradient adjusted prediction [14] and so on. The aforementioned methods are still continuously studied and developed nowadays, for example, partial differential equation [15], local prediction [16] and checkerboard-based prediction [17]. Besides the improvement of prediction, many advanced errors manipulation strategies have also been proposed in recent years, for example, context modification [18], adaptive embedding [19,20], two-dimensional histogram modification [21–23] and multiple histograms modification [24,25].

The histogram modification technique proposed by Ni et al. in 2006 is another remarkable work of RDH [26]. After the construction of histogram using all pixels' gray-scale values, pixels between the peak bin and zero bin are shifted by one unit to create embedding space. As the maximum modification is at most 1, high-quality marked image can be well guaranteed despite of limited EC. Taking into account that the sharpness of histogram is of great significance to embedding performance, difference histogram [27] and prediction error histogram [28]

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

* Corresponding author.
E-mail address: xionggq@aliyun.com (G. Xiong).

http://dx.doi.org/10.1016/j.jvcir.2017.10.001
Received 13 April 2017; Received in revised form 2 August 2017; Accepted 4 October 2017
Available online 06 October 2017

are expected to achieve improvements. On the other hand, there are also other improvements including optimal expansion bins selection [29], multilevel histogram modification [30] and generalized histogram modification [31]. In [24,25], conventional histogram is decomposed into a set of identifiable histograms to achieve image content dependent embedding.

Higher prediction accuracy is always the fundamental pursuit of PEE-RDH. Recently, Li et al. [32] proposed a novel prediction method namely pixel value ordering (PVO). In this method, the second largest/smallest pixel is used to predict the largest/smallest one in the scope of pixel block. Benefiting from the close correlation of pixels within block, PVO prediction is verified to be more accurate than previous methods [10,12,13,19]. Moreover, the complexity of pixel block is also measured to achieve embedding primitive selection for extra performance enhancement. However, as pointed out by the proposer, using only the maximum and the minimum as data bit carriers and the infeasibility of multi-pass embedding may lead to insufficient EC. Larger EC not only enables more practical applications but also allows one to achieve higher prediction accuracy by using larger blocks.

Following Li et al.'s work, many PVO-based schemes have been proposed. Peng et al. [33] firstly proposed to introduce the relative location relationships of pixels into predicted value calculation. Then bins 1 and 0 instead of original bins 1 and -1 are utilized for data embedding to achieve an improved PVO. In [34], pixels whose values equal to the maximum (or minimum) are united as one bit data carrier. Suppose k denotes the number of maximum-valued (or minimum-valued) pixels, original PVO is extended to PVO-k embedding. Then the optimal combined embedding with $k \in \{1,2\}$ is verified to be better than original PVO which is also the PVO-1 embedding. Wang et al. [35] further developed Peng et al.'s work [33] with dynamic blocking strategy which enables the combination of two various-sized blocks. That is, rough region is divided into large blocks to achieve high PSNR whereas smooth region is divided into small blocks to pursuit sufficient EC. However, although impressive performance is achieved for moderate payload size, its maximum EC is exactly the same as Peng et al.'s [33]. So far, the largest maximum EC related to PVO shows up in Qu et al.'s pixel-based pixel value ordering scheme [36]. For this scheme, all pixels have a chance to embed data and predicted value comes from the maximum or minimum of the sorted context pixels.

Among the aforementioned schemes [32–36], it is noticeable that the maximum modification is commonly at most 1, which is the key to low MSE and high PSNR but also the direct cause of few data bit carriers. To achieve larger EC, one direct way is to adopt smaller block size at the cost of lower prediction accuracy, just as the further division of flat blocks in Wang et al.'s dynamic blocking [35]. By contrast, PVO-k chooses to further exploit pixels correlation with invariant block size. For a block consisted of $n = u \times v$ pixels, all n pixels can be sorted in ascending order to get $(p_{\sigma(1)},p_{\sigma(2)},\cdots,p_{\sigma(n)})$. While block with $p_{\sigma(n)} = p_{\sigma(n-1)}$ is abandoned in original PVO, $p_{\sigma(n)}$ and $p_{\sigma(n-1)}$ are considered together as a unit to embed one bit data and $p_{\sigma(n-2)}$ is employed to predict them in PVO-2 embedding. In this way, more blocks can be utilized for data embedding and EC is thus increased. However, there is still room for improvement based on the consideration that the newly increased EC can be obviously doubled if both $p_{\sigma(n)}$ and $p_{\sigma(n-1)}$ are processed as independent data bit carriers.

In this paper, we extend original PVO into another general form in which not only PVO-1 embedding survives in another form but also the advantage of PVO-2 embedding is doubled. Experimental results demonstrate that the our scheme outperforms previous PVO-based schemes [33,34] and some other state-of-the-art works [12,21]. The rest of the paper is organized as follows. In Section 2, by extending the PVO-k embedding, a new reversible embedding strategy namely k-pass PVO embedding is introduced. For simplicity, only the maximum-direction-modification embedding is introduced. Then by combining the maximum-direction- and minimum-direction- modifications a novel RDH scheme is proposed in Section 3. Experimental results and performance comparisons with other schemes are shown in Section 4. Finally, we conclude the paper in Section 5.

2. A new PVO-based reversible embedding strategy

2.1. Related work: PVO-k embedding [34]

For a block consisted of n pixels, assume that

$$p_{\sigma(1)} \leqslant \cdots \leqslant p_{\sigma(n-k)} < p_{\sigma(n-k+1)} = \cdots = p_{\sigma(n)} \tag{1}$$

where k is the number of maximum-valued pixels which will be modified as a unit for data embedding. That is, for each $p_{\sigma(i)}, i \in \{n-k+1,\cdots,n\}$, the corresponding marked value is obtained as

$$p^w_{\sigma(i)} = \begin{cases} p_{\sigma(i)} + w, & \text{if } x = 1 \\ p_{\sigma(i)} + 1, & \text{if } x > 1 \end{cases} \tag{2}$$

where the prediction error x is

$$x = p_{\sigma(n-k+1)} - p_{\sigma(n-k)} \geqslant 1 \tag{3}$$

and $w \in \{0,1\}$ is a to-be-embedded bit.

Clearly, PVO-1 embedding is just the original PVO [32] since only blocks with $p_{\sigma(n)} > p_{\sigma(n-1)}$ are utilized for data embedding. Then, PVO-k embedding is extended by considering both maximum- and minimum-valued pixels of a block. First, for a block with ascending pixel values $(p_{\sigma(1)},p_{\sigma(2)},\cdots,p_{\sigma(n)})$, if $p_{\sigma(1)} < p_{\sigma(n)}$ assume that

$$p_{\sigma(1)} = \cdots = p_{\sigma(b)} < p_{\sigma(b+1)} \leqslant \cdots \leqslant p_{\sigma(n-a)} < p_{\sigma(n-a+1)} = \cdots = p_{\sigma(n)} \tag{4}$$

where a and b denote numbers of maximum- and minimum-valued pixels, respectively. Then two prediction errors are obtained by using $p_{\sigma(n-a)}$ to predict $p_{\sigma(n-a+1)}$, and $p_{\sigma(b+1)}$ to predict $p_{\sigma(b)}$

$$x = p_{\sigma(n-a+1)} - p_{\sigma(n-a)} \quad \text{and} \quad y = p_{\sigma(b)} - p_{\sigma(b+1)} \tag{5}$$

If $a = k$ or $b = k$, the block will be used for data embedding according to the following four cases:

- Case 1: $a = k, b \neq k$. In this case, only the maximum-valued pixels will be modified and each marked value is determined as

$$p^w_{\sigma(i)} = \begin{cases} p_{\sigma(i)} + w, & \text{if } n-a+1 \leqslant i \leqslant n \quad \text{and} \quad x = 1 \\ p_{\sigma(i)} + 1, & \text{if } n-a+1 \leqslant i \leqslant n \quad \text{and} \quad x > 1 \end{cases} \tag{6}$$

- Case 2: $a \neq k, b = k$. In this case, only the minimum-valued pixels will be modified and each marked value is determined as

$$p^w_{\sigma(i)} = \begin{cases} p_{\sigma(i)} - w, & \text{if } 1 \leqslant i \leqslant b \quad \text{and} \quad y = -1 \\ p_{\sigma(i)} - 1, & \text{if } 1 \leqslant i \leqslant b \quad \text{and} \quad y < -1 \end{cases} \tag{7}$$

- Case 3: $a = b = k$, and $n > 2 \times k$. In this case, both maximum- and minimum-valued pixels will be modified and each marked value is determined as

$$p^w_{\sigma(i)} = \begin{cases} p_{\sigma(i)} + w, & \text{if } n-a+1 \leqslant i \leqslant n \quad \text{and} \quad x = 1 \\ p_{\sigma(i)} + 1, & \text{if } n-a+1 \leqslant i \leqslant n \quad \text{and} \quad x > 1 \\ p_{\sigma(i)} - w, & \text{if } 1 \leqslant i \leqslant b \quad \text{and} \quad y = -1 \\ p_{\sigma(i)} - 1, & \text{if } 1 \leqslant i \leqslant b \quad \text{and} \quad y < -1 \end{cases} \tag{8}$$

- Case 4: $a = b = k$, and $n = 2 \times k$. In this case, only the maximum-valued pixels are modified and each marked value is determined according to Eq. (6).

2.2. K-pass PVO embedding

The major advantage of PVO-k is to provide extra EC by the further

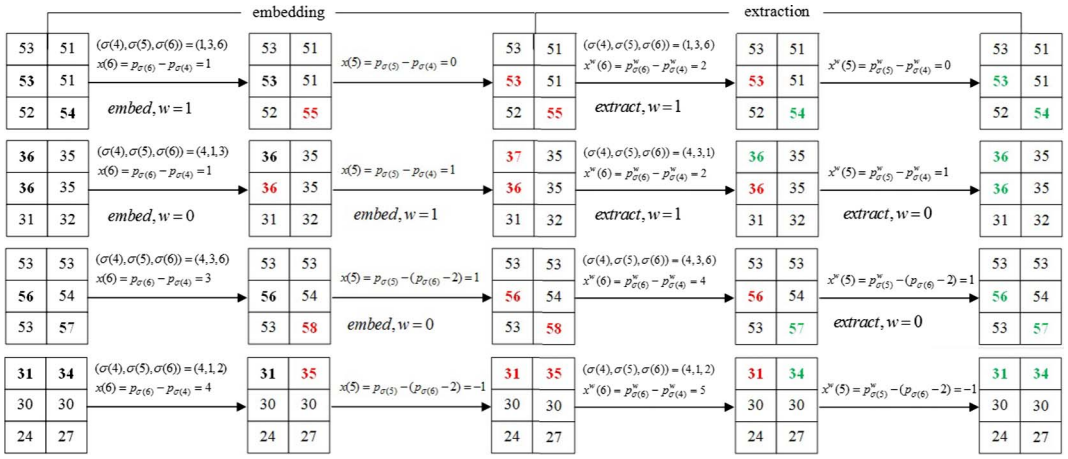

Fig. 1. Illustration of maximum-direction 2-pass PVO embedding and extraction.

utilization of embedding primitives. However, the efficiency of PVO-k is still unsatisfactory, especially when k is large. In PVO-k, the average distortion in terms of mean square error (MSE) for a block is

$$\sum_{i=n-k+1}^{n}(p_{\sigma(i)}-p^{w}_{\sigma(i)})^2=\begin{cases}k/2, & \text{if } x=1\\ k, & \text{if } x>1\end{cases} \tag{9}$$

which is close related to the number of maximum-valued pixels. The larger the k is, the larger the distortion is. One direct solution to this problem is to enlarge the EC contribution of PVO-k from 1 to k and thus achieve a better capacity-distortion trade-off. This motivates the transformation from PVO-k into k-pass PVO.

For each block consisted of n pixels, scan all pixels by raster scan order to form a pixel sequence. Then sort their pixel values $(p_1,p_2,\cdots,p_n)$ in ascending order to obtain $(p_{\sigma(1)},p_{\sigma(2)},\cdots,p_{\sigma(n)})$ where σ: $\{1,\cdots,n\}\rightarrow\{1,\cdots,n\}$ is the unique one-to-one mapping such that: $p_{\sigma(1)}\leqslant p_{\sigma(2)}\leqslant\cdots\leqslant p_{\sigma(n)}$,$\sigma(i)<\sigma(j)$ if $p_{\sigma(i)}=p_{\sigma(j)}$ and $i<j$. Here, the largest k pixels will be selected as data bit carriers respectively in descending order to fulfill the k-pass PVO embedding. For each $p_{\sigma(i)}$,$i\in\{n,n-1,\cdots,n-k+1\}$, the corresponding marked value is obtained as

$$p^{w}_{\sigma(i)}=\begin{cases}p_{\sigma(i)}+w, & \text{if } x(i)=1\\ p_{\sigma(i)}+1, & \text{if } x(i)>1\end{cases} \tag{10}$$

where the prediction error $x(i)$ is

$$x(i)=\begin{cases}p_{\sigma(i)}-p_{\sigma(n-k)}, & \text{if } i=n\\ p_{\sigma(i)}-p_{\sigma(n-k)}, & \text{if } i<n \quad \text{and} \quad x(n)\leqslant 1\\ p_{\sigma(i)}-(p_{\sigma(n)}-2), & \text{if } i<n \quad \text{and} \quad x(n)>1\end{cases} \tag{11}$$

and $w\in\{0,1\}$ is a to-be-embedded bit.

From Eq. (11) we can see that every involved pixel except $p_{\sigma(n)}$ has two candidate predicted values one of which will be selected according to the result of first-pass expansion embedding. The whole procedure of k-pass PVO embedding can be described in Algorithm 1.

Algorithm 1. K-pass PVO embedding

```
Input: Sorted pixel sequence (p_σ(1),p_σ(2),⋯,p_σ(n))
Output: Marked pixel sequence (p^w_σ(1),p^w_σ(2),⋯,p^w_σ(n))
if (p_σ(n) != p_σ(n−k))
{
  x̂ = p_σ(n−k);
  if (p_σ(n)−x̂ > 1)
  {
    p^w_σ(n) = p_σ(n) + 1; x̂ = p_σ(n)−2;
  }
  else
  {
    p^w_σ(n) = p_σ(n) + w;
  }
  for (i = n-1; i > =n-k + 1; i–)
  {
    if (p_σ(i)−x̂ > 1)
    {
      p^w_σ(i) = p_σ(i) + 1;
    }
    else if (p_σ(i)−x̂ = =1)
    {
      p^w_σ(i) = p_σ(i) + w;
    }
  }
}
```

It is clearly seen that 1-pass PVO is just the original PVO which simply use $p_{\sigma(n-1)}$ to predict $p_{\sigma(n)}$. When $k>1$, whether $p_{\sigma(n)}$ is expandable or not if predicted by $p_{\sigma(n-k)}$ determines how the rest involved pixels are predicted. If yes, $p_{\sigma(n-k)}$ continues to serves as predicted value since this may further exploit expandable pixels with high probability. Otherwise, the candidate predicted value derived from $p_{\sigma(n)}-2$ is adopted to avoid large amount of shifted pixels and meanwhile perfectly utilize pixels which used to provide the most suitable predicted value for $p_{\sigma(n)}$.

An example illustrating 2-pass PVO embedding is given in Fig. 1 in which the shadows of PVO-1 embedding and PVO-2 embedding can be seen. For example, in cases 1 and 4, we get exactly the same marked values as PVO-1 embedding. In case 3, data bit carrier turns from $p_{\sigma(n)}$ to $p_{\sigma(n-1)}$ at the cost of shifting $p_{\sigma(n)}$ by 1 when compared with PVO-1 embedding. It is worth to mention that in case 2, the PVO may change after data embedding. However, this has no effect on extraction and image restoration.

2.3. K-pass PVO extraction

The key issue of k-pass PVO is still a generalized invariant PVO (i.e., treating expanded pixels as a unit). This will help to find the same predicted values at the decoder. First, the predicted value for $p^{w}_{\sigma(n)}$ is determined by the unchanged $p_{\sigma(n-k)}$. Then $p^{w}_{\sigma(n)}$ can be judged as shifted or expanded and its original value is recovered. If expanded, $p_{\sigma(n-k)}$ is also the predicted value for other $k-1$ largest pixels. Otherwise, the predicted value is renewed by $p_{\sigma(n)}-2$. The whole

procedure of k-pass PVO extraction can be described in Algorithm 2.

Algorithm 2. K-pass PVO extraction

```
Input: Marked pixel sequence (p^w_σ(1),p^w_σ(2),⋯,p^w_σ(n))
Output: Sorted pixel sequence (p_σ(1),p_σ(2),⋯,p_σ(n)), extracted bit w
if (p^w_σ(n) != p^w_σ(n−k))
{
  x̂ = p^w_σ(n−k);
  if (p^w_σ(n) − x̂ > 2)
  {
    p_σ(n) = p^w_σ(n) − 1; x̂ = p_σ(n) − 2;
  }
  else
  {
    w = p^w_σ(n) − x̂ − 1; p_σ(n) = p^w_σ(n) − w;
  }
  for (i = n-1; i > =n-k + 1; i–)
  {
    if (p^w_σ(i) − x̂ > 2)
    {
      p_σ(i) = p^w_σ(i) − 1;
    }
    else if (p^w_σ(i) − x̂ = =1 —— p^w_σ(i) − x̂ = =2)
    {
      w = p^w_σ(i) − x̂ − 1; p_σ(i) = p^w_σ(i) − w;
    }
  }
}
```

Fig. 1 also presents the extraction procedure of 2-pass PVO. It can be seen that none or only one bit of data is embedded in cases 1,3,4 and invariant PVO is obtained. In case 2, embedding two bits of data changes the PVO. However, it is noticeable that all expanded pixels have the same original values. So, the extracted bits can be perfectly rearranged according to the order in which they are embedded. In other words, they should be ordered according to the nature location of pixels from which they are extracted. Suppose $\{w_1,w_2,w_3\}$ are extracted from $\{p^w_{\sigma(n-2)},p^w_{\sigma(n-1)},p^w_{\sigma(n)}\}$ respectively and we have $\sigma(n-1) < \sigma(n) < \sigma(n-2)$ .Then extracted bits should be ordered as $\{w_1,w_3,w_2\}$.

3. Proposed scheme

In this section, a new RDH scheme is presented by applying k-pass PVO embedding into both maximum-direction- and minimum-direction-modifications. The section is organized as follows: we first describe in Section 3.1 the extension of k-pass PVO embedding by modifying both the k largest and the k smallest pixels. Then, optimal combined embedding is discussed in Section 3.2. Finally, Section 3.3 gives the detailed embedding and extracting procedures.

3.1. Extended k-pass PVO embedding

By employing both the k largest and the k smallest pixels as data bit carriers, k-pass PVO embedding can be performed twice if $n \geqslant 2 \times k + 1$. First, pixels are sorted in ascending order to obtain $(p_{\sigma(1)},p_{\sigma(2)},\cdots,p_{\sigma(n)})$. Then if $p_{\sigma(n)} > p_{\sigma(1)}$, the block will be used for data embedding.

For maximum-direction-modification, $p_{\sigma(n)}$ is firstly predicted by $p_{\sigma(n-k)}$ to fulfill the first pass PVO embedding. If $p_{\sigma(n)}$ is expandable, $p_{\sigma(n-k)}$ continues to serves as predicted value. Otherwise, the new predicted value is derived from $p_{\sigma(n)}-2$. Then $p_{\sigma(n-1)},\cdots,p_{\sigma(n-k+1)}$ are successively selected to fulfill the second, the third until the k-th pass PVO embedding. In the above process, marked values and prediction errors are calculated according to Eqs. (10) and (11).

For minimum-direction-modification, $p_{\sigma(1)}$ is firstly predicted by $p_{\sigma(k+1)}$ to fulfill the first pass PVO embedding. If $p_{\sigma(1)}$ is expandable, $p_{\sigma(k+1)}$ continues to serves as predicted value. Otherwise, the new predicted value is derived from $p_{\sigma(1)} + 2$. Then $p_{\sigma(2)},\cdots,p_{\sigma(k)}$ are successively selected to fulfill the second, the third until the k-th pass PVO embedding. For each $p_{\sigma(i)},i \in \{1,2,\cdots,k\}$, the corresponding marked value is obtained as

$$p^w_{\sigma(i)} = \begin{cases} p_{\sigma(i)} - w, & \text{if } x(i) = -1 \\ p_{\sigma(i)} - 1, & \text{if } x(i) < -1 \end{cases} \tag{12}$$

where the prediction error $x(i)$ is

$$x(i) = \begin{cases} p_{\sigma(i)} - p_{\sigma(k+1)}, & \text{if } i = 1 \\ p_{\sigma(i)} - p_{\sigma(k+1)}, & \text{if } i > 1 \quad \text{and} \quad x(1) \geqslant -1 \\ p_{\sigma(i)} - (p_{\sigma(1)} + 2), & \text{if } i > 1 \quad \text{and} \quad x(1) < -1 \end{cases} \tag{13}$$

and $w \in \{0,1\}$ is a to-be-embedded bit.

3.2. Optimal combined embedding

As shown Fig. 2, we similarly define the complexity of a block as the sum of the vertical and the horizontal absolute difference of every two adjacent pixels in the context. Since blocks are scanned and processed sequentially at encoder and in the reverse order at decoder, the same block complexity can be maintained. Then low complexity blocks are preferentially utilized to achieve improved performance.

Suppose the host image is divided into N blocks and c_i denotes the complexity of the i-th block. We define a function $h_k(e,t)$ to count the prediction errors with magnitude e produced using k-pass PVO embedding as follows:

$$h_k(e,t) = \#\{1 \leqslant i \leqslant N: x = e, c_i = t\} \tag{14}$$

where # denotes the cardinal number of a set and x refers to every obtained prediction error. If without taking into other factors caused consumption, the capacity of k-pass PVO embedding is

$$E_k(T) = \sum_{t=0}^{T-1} (h_k(1,t) + h_k(-1,t)) \tag{15}$$

and the expected value of corresponding distortion in terms of MSE can be formulated by

$$D_k(T) = \frac{1}{2} \sum_{t=0}^{T-1} \sum_{|e|=1} h_k(e,t) + \sum_{t=0}^{T-1} \sum_{|e|>1} h_k(e,t) \tag{16}$$

Then the efficiency of k-pass PVO embedding can be measured by $D_k(T)/E_k(T)$ under the premise of providing sufficient EC. Fig. 3 shows the $EC-D/E$ curves derived from 1-pass PVO embedding, 2-pass PVO embedding and 3-pass PVO embedding on standard test image Lena and Barbara using 3×2 block size. One can see that although 1-pass PVO

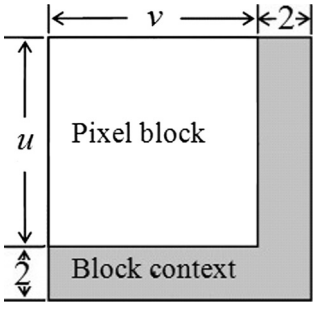

Fig. 2. A $u \times v$ sized block with its context.

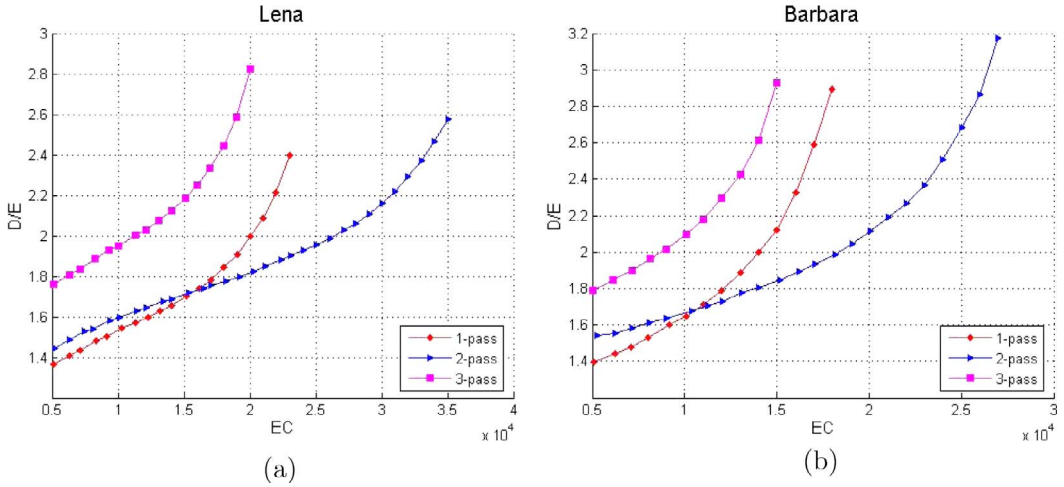

Fig. 3. *EC*−*D*/*E* curves derived from 1-pass PVO embedding, 2-pass PVO embedding and 3-pass PVO embedding using 3 × 2 block size.

embedding has lower $D_k(T)/E_k(T)$ when EC is small, 2-pass PVO embedding is undoubtedly the better choice in most cases. Such observation points out the further improvement of combining 1-pass PVO embedding and 2-pass PVO embedding.

Specially, the proposed scheme uses two thresholds, T_1 and T_2, to fulfill block classification and combined embedding. With the block complexity and two thresholds, blocks are classified and processed as follows:

- Case 1: block with $c_i < T_2$ is classified as a flat block which will be processed using 2-pass PVO embedding. In this case, at most 4 bits of data can be embedded.
- Case 2: block with $T_2 \leqslant c_i < T_1$ is classified as a normal block which will be processed using 1-pass PVO embedding. In this case, at most 2 bits of data can be embedded.
- Case 3: block with $c_i \geqslant T_1$ is classified as a rough block which will be excluded from data embedding.

In this situation, the capacity of combined embedding is

$$E = \sum_{t=0}^{T_2-1} \sum_{|e|=1} h_2(e,t) + \sum_{t=T_2}^{T_1-1} \sum_{|e|=1} h_1(e,t) \tag{17}$$

and the expected value of corresponding distortion in terms of MSE can be formulated by

$$D = \frac{1}{2}\left(\sum_{t=0}^{T_2-1} \sum_{|e|=1} h_2(e,t) + \sum_{t=T_2}^{T_1-1} \sum_{|e|=1} h_1(e,t)\right) + \left(\sum_{t=0}^{T_2-1} \sum_{|e|>1} h_2(e,t) + \sum_{t=T_2}^{T_1-1} \sum_{|e|>1} h_1(e,t)\right) \tag{18}$$

Finally, here comes the optimal thresholds (T_2^*,T_1^*) determination for a given EC. As there is no analytical relationship between T_2^* and T_1^*, they can only be determined by exhaustive search aiming to minimize D/E under the premise of $E \geqslant EC$. Fig. 4 shows the D/E performance comparison for 1-pass PVO embedding, 2-pass PVO embedding, optimized k-pass PVO embedding and optimized PVO-k embedding on standard test image Lena and Barbara using 3 × 2 block size. One can see that Ou et al.'s optimized PVO-k embedding obviously outperforms 1-pass PVO embedding which is also the PVO-1 embedding. Then 2-pass PVO embedding is better than optimized PVO-k embedding when EC is large. Finally, the optimized k-pass PVO embedding achieves the best performance for every EC.

3.3. Implementation for the proposed scheme

First, the detailed data embedding procedure is given below.

Step 1: Over/underflow prevention
As the maximum modification is at most 1 in the proposed scheme, overflow and underflow only occurs on pixels whose values equal to 0 or 255 for eight bit gray-scale image. Here, we modify all pixels with value of 255 or 0 to 254 or 1 and a location map (*LM*) is needed to record those modification to ensure reversibility. A bit 1 in *LM* represents the modification of 255 to 254 or 0 to 1; a bit 0 represents the 254 or 1 is the original pixel value. The other pixels do not need the *LM*. By using arithmetic coding, *LM* is then losslessly compressed into a shorter bit stream denoted as *CLM*.
Step 2: Optimal thresholds determination
Divide the host image into non-overlapped blocks and calculate block complexities. Then fulfill 1-pass PVO prediction and 2-pass PVO prediction to obtain $h_k(e,t),k \in \{1,2\}$. On this basis, determine the optimal T_2^* and T_1^* as discussed in Section 3.2.
Step 3: Data embedding
Scan pixel blocks in raster scan order to fulfill secret message embedding. For block with $c_i < T_2^*$, 2-pass PVO embedding is performed. For block with $T_2^* \leqslant c_i < T_1^*$, 1-pass PVO embedding is performed. Block with $c_i \geqslant T_1^*$ is skipped. The detailed embedding can refer to Eqs. (10)–(13) and Algorithm 1.
Step 4: Auxiliary information and location map embedding
Record the least significant bits of the first 78 + L_{clm} image pixels (denoted as S_{LSB}), where L_{clm} is the length of the compressed location map. Then embed S_{LSB} into the rest part of image and record the index of the last data-carrying block (denoted as E_B).

Replace the LSBs of the first 78 + L_{clm} pixels by the following auxiliary information and the compressed location map:

- value of EC (18 bits)
- values of block size u (2 bits) and v (2 bits),
- values of optimal thresholds T_1^* (10 bits) and T_2^* (10 bits),
- the index of the last data-carrying block E_B (18 bits).
- length of the compressed location map L_{clm} (18 bits).

Finally, the marked image is then generated.

As the inverse process of data embedding, data extraction and image recovery procedure is described as follows.

Step 1: auxiliary information and location map extraction
Read the LSBs of the first 78 pixels of the marked image to retrieve

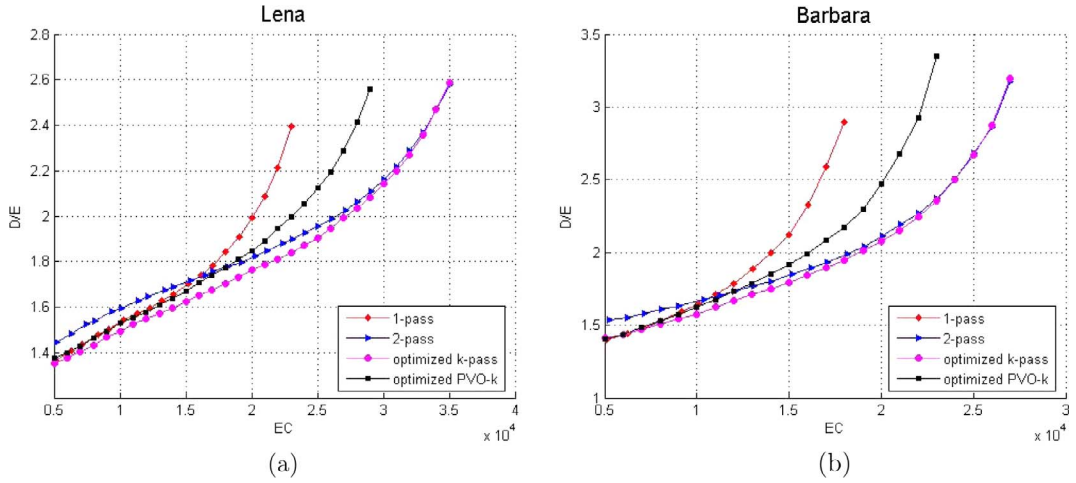

Fig. 4. D/E performance comparison for 1-pass PVO embedding, 2-pass PVO embedding, optimized k-pass PVO embedding and optimized PVO-k embedding.

the auxiliary information, including the values of EC, block size, optimal thresholds, index of the last data-carrying block and the length of compressed location map. Then read the LSBs of the next L_{clm} pixels to retrieve the compressed location map and recover the location map LM by decompression.

Step 2: Sequence S_{LSB} extraction and image restoration

Divide the marked image into non-overlapped blocks and start extracting the embedded sequence S_{LSB} from the last data-carrying block in reverse raster scan order. For each block, the block complexity is firstly calculated. If $c_i < T_2^*$, 2-pass PVO extraction is performed. If $T_2^* \leqslant c_i < T_1^*$, 1-pass PVO extraction is performed. Otherwise, the block is skipped. The detailed extraction can refer to Algorithm 2.

Step 3: Secret message extraction and image restoration

Replace the LSBs of first $78 + L_{clm}$ image pixels by the sequence just extracted. Then use the same method of step2 to extract the embedded message and meanwhile realize the image restoration. Finally, revise the pixel with value 1 and 254 to 0 and 255 according to the decompressed location map respectively.

4. Experimental results

This section presents experimental results from the proposed scheme. The performance of the proposed scheme will be compared with Peng et al.'s IPVO [33], Ou et al.'s PVO-k [34] and two state-of-the-art schemes of Sachnev et al. [12] and Li et al. [21]. For these experiments, eight 512×512 gray-scale images served as test images. Except Barbara, all images are downloaded from the USC-SIPI image database, as depicted in Fig. 5.

Here we first discuss the efficiency of the proposed scheme in terms of being an extension of Ou et al.'s PVO-k embedding. Referring to Fig. 4, when 3×2 block size is adopted, the proposed scheme is verified to introduce less distortion than optimized PVO-k embedding whatever the EC is. Moreover, the gap becomes more and more apparent as EC increases. By using PSNR to evaluate the visual quality of marked image, Fig. 6 shows their actual capacity-distortion performances from which we can see that the superiority of the proposed scheme over Ou et al.'s work is evident. First, larger maximum EC is achieved. On image Lena, the maximum EC reaches up to 35000 bits while that of optimized PVO-k is 29000 bits. Second, there is higher PSNR for every EC and meanwhile the gap becomes more and more apparent as EC increases. For example, on image Barbara the PSNR improvement increases from 0.13 dB for an EC of 10000 bits to 0.75 dB for an EC of 20000 bits.

This can be well explained by comparing the values of thresholds and expected distortion since they have the same block complexity calculation. Table 1 shows the values of two optimal thresholds and expected distortion for embedding 10000 bits and 20000 bits of data using 3×2 blocks. We can see that the proposed scheme always employs fewer blocks to provide sufficient EC and thus achieves less expected distortion. This fully demonstrates the advantage of k-pass PVO embedding. In the following, for a given EC, the embedding procedures of the proposed, Ou et al. and Peng et al. will be carried out for 9 block sizes $(u,v \in \{2,3,4\})$ to achieve the best performance. Specially, since there is no sufficient pixels when using 2×2 blocks, we propose to apply 1-pass PVO embedding instead of 2-pass PVO embedding into minimum-direction-modification for blocks with $c_i < T_2^*$.

Fig. 7 shows the performance comparison of the proposed scheme and all other schemes by varying EC from 5,000 bits to its maximum with a step of 1,000 bits. For the eight test images, the maximum EC of the proposed scheme is 38000, 13000, 50000, 30000, 27000, 27000, 32000 and 24000 bits. Compared with Ou et al.'s scheme, the proposed scheme achieves slight improvement in maximum EC on images Airplane, Barbara, Boat, Lake and Elaine. On the other hand, the proposed scheme also significantly outperforms Ou et al.'s scheme by providing a higher PSNR for every test image whatever the EC is. On all test images, the PSNR improvement ranges from 0.15 to 0.48 dB, 0 to 0.2 dB, 0.24 to 0.83 dB, 0.06 to 0.29 dB, 0.03 to 0.35 dB, 0.12 to 0.68 dB, 0.05 to 0.36 dB and 0.04 to 0.46 dB respectively. Referring to Tables 2 and 3, it can be seen that the proposed scheme improves Ou et al.'s by 0.39 dB on average for an EC of 10000 bits, and 0.27 dB for an EC of 20000 bits. The superiority over Ou et al.'s work is thus verified. However, it is worth to mention that in our experiment border block whose complexity can not be measured by the proposed scheme and Ou et al.'s scheme is unconditionally employed. Otherwise, embedding performance is expected to be slightly promoted at the cost of lower maximum EC.

Unlike Ou et al. taking maximum-valued pixels as a unit, Peng et al. succeeded to reemploy block with $p_{\sigma(n)} = p_{\sigma(n-1)}$ for data embedding and avoid modifying multiple pixels for one bit data embedding. But unfortunately, a major drawback is that part of block with $p_{\sigma(n)} = p_{\sigma(n-1)} + 1$ has to be abandoned. Compared with Peng et al.'s scheme, both block with $p_{\sigma(n)} = p_{\sigma(n-1)}$ and block with $p_{\sigma(n)} = p_{\sigma(n-1)} + 1$ are fully exploited in the proposed scheme. Referring to Fig. 7, the proposed scheme achieves the same or lager maximum EC than Peng et al.'s scheme for every test images except Airplane due to its extreme smoothness. In terms of PSNR, the proposed scheme achieves a more significant superiority in almost all cases. On images Lena, Airplane, Boat and Elaine, the proposed scheme achieves relatively poor performance when EC approaches its maximum (e.g., 36000 bits for Lena, 47000 bits for Airplane, 25000 bits for Boat and 22000 bits for Elaine). It is reasonable since the advantage of k-pass PVO embedding is greatly weakened when using 2×2 block size. However,

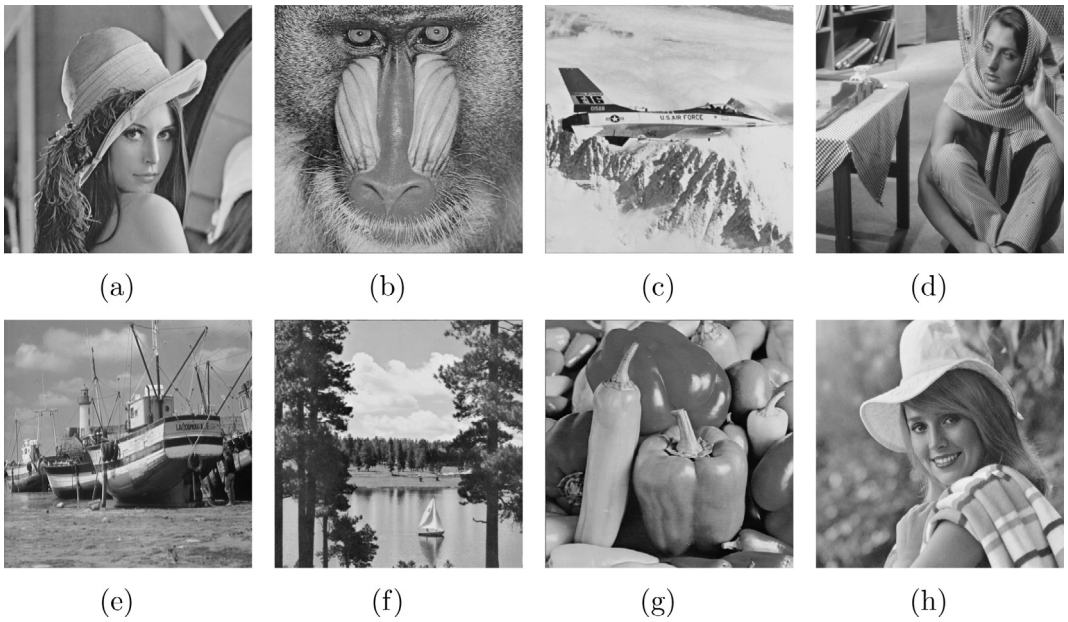

Fig. 5. Original test images: (a) Lena, (b) Baboon, (c) Airplane, (d) Barbara, (e) Boat, (f) Lake, (g) Peppers, (h) Elaine.

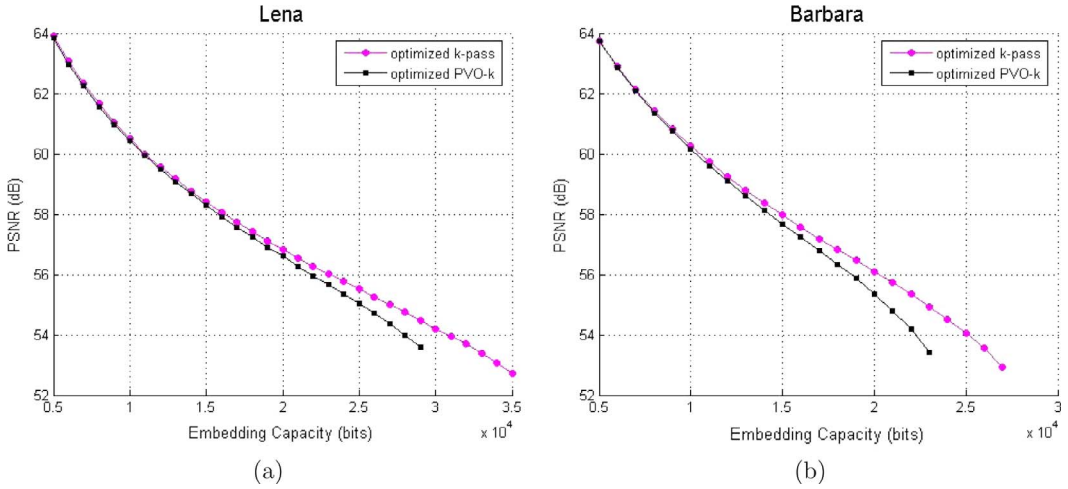

Fig. 6. Capacity-distortion performances of optimized k-pass PVO embedding and optimized PVO-k embedding.

Table 1
Comparison of the values of optimal thresholds and expected distortion.

Image	EC = 10000 bits				EC = 20000 bits			
	k-pass PVO		PVO-k		k-pass PVO		PVO-k	
Lena	{29,49}	15090	{28,51}	15452.5	{43,73}	35569.5	{48,82}	37076.5
Baboon	{244,481}	67450	{372,642}	75837.5	–	–	–	–
Airplane	{17,24}	7801.5	{13,29}	8632.5	{26,36}	19226	{26,46}	22112
Barbara	{35,56}	15862.5	{35,61}	16481.5	{72,108}	41837	{138,210}	49652.5
Boat	{66,96}	26089	{70,109}	27659	{113,214}	69030	{169,453}	76960
Lake	{62,83}	18705	{62,98}	21136.5	{135,187}	60036.5	{219,423}	74605.5
Peppers	{40,77}	22247.5	{46,77}	22255.5	{76,111}	52491	{91,130}	54621
Elaine	{71,119}	29013	{68,129}	31194	{151,265}	83786	–	–

with moderate payload size, the largest PSNR improvement on all test images can reaches up to 0.9 dB, 0.58 dB, 1 dB, 1.09 dB, 0.68 dB, 1.24 dB, 0.95 dB and 0.89 dB respectively. Referring to Tables 3 and 4, it can be seen that the proposed scheme improves Peng et al.'s by 0.39 dB on average for an EC of 10000 bits, and 0.62 dB for an EC of 20000 bits.

Sachnev et al.'s scheme is well known for its high accurate prediction and has been verified better than many state-of-the-art works. For this scheme, prediction error is obtained using rhombus prediction where the four neighboring pixels are used to predict the center one. Moreover, pixels with small local variance are preferentially utilized for embedding. Referring to Fig. 7, our scheme outperforms Sachnev et al.'s in most cases, except when EC approaches its maximum (e.g., 38000 bits for Lena, 11000 bits for Baboon, 47000 bits for

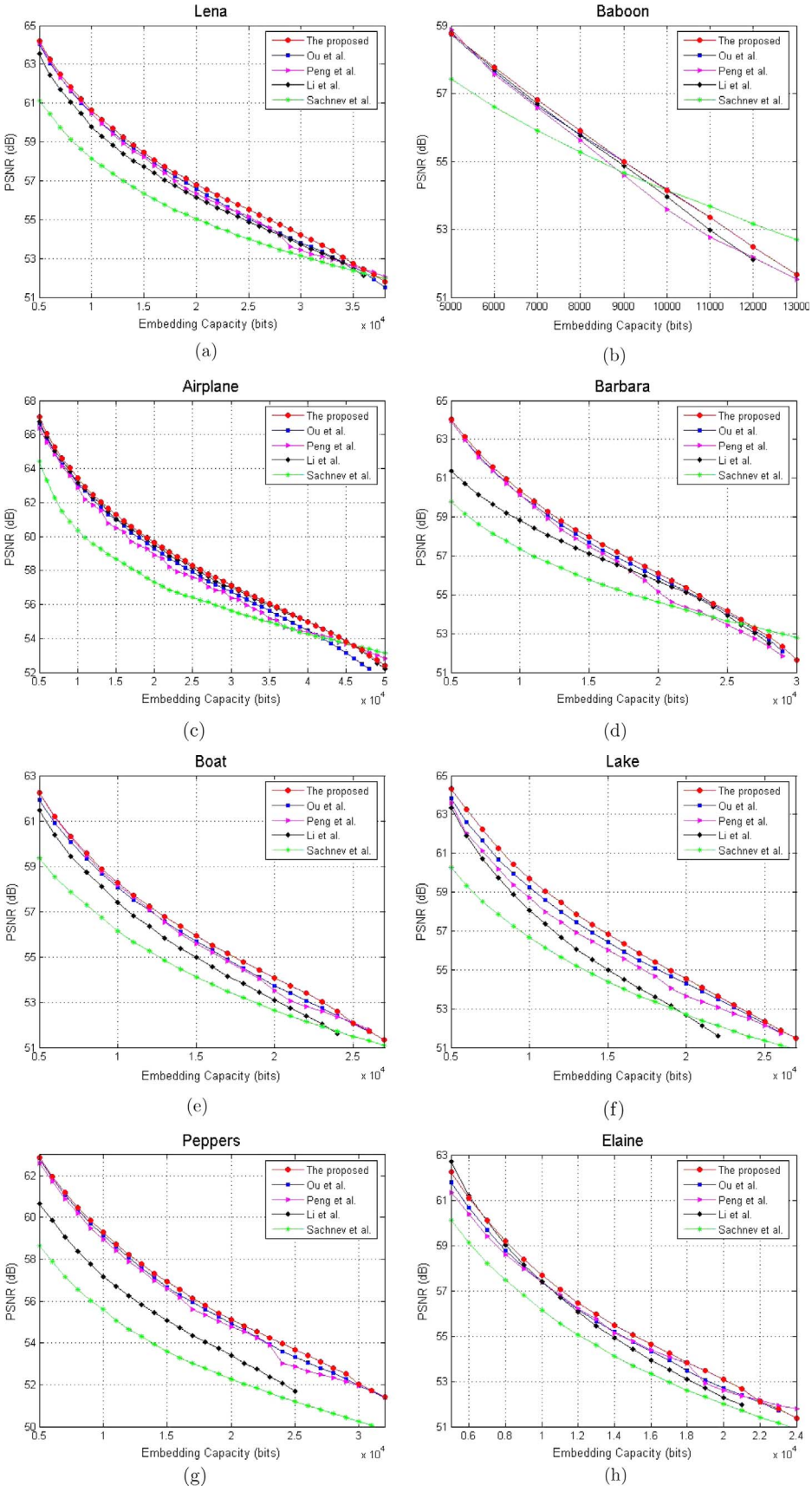

Fig. 7. Performance comparison between the proposed scheme and four schemes of Ou et al. [34], Peng et al. [33], Li et al. [21], and Sachnev et al. [12].

Table 2
Average run time (unit:second) of one loop embedding of the proposed scheme.

Image	Lena	Baboon	Airplane	Barbara	Boat	Lake	Peppers	Elaine
Run time	1.95	9.59	1.65	3.18	3.11	2.4	3.4	2.43

Table 3
Comparison of PSNR (in dB) between the Li et al.[21], Sachnev et al. [12], Peng et al. [33], Ou et al. [34] schemes and the proposed scheme for an EC of 10,000 bits.

Image	Li et al.	Sachnev et al.	Peng et al.	Ou et al.	The proposed
Lena	59.76	58.19	60.49	60.46	60.64
Baboon	53.96	54.14	53.58	54.16	54
Airplane	63.18	60.33	62.9	63.14	63.45
Barbara	58.82	57.37	60.14	60.15	60.37
Boat	57.44	56.16	58.17	58.06	58.28
Lake	58.08	56.65	58.7	59.23	59.71
Peppers	57.11	55.48	58.95	59.16	59.29
Elaine	57.38	56.12	57.36	57.36	57.67
Average	58.22	56.81	58.79	58.97	59.18

Table 4
Comparison of PSNR (in dB) between the Li et al. [21], Sachnev et al. [12], Peng et al. [33], Ou et al. [34] schemes and the proposed scheme for an EC of 20,000 bits.

Image	Li et al.	Sachnev et al.	Peng et al.	Ou et al.	The proposed
Lena	56.12	55.02	56.36	56.6	56.81
Baboon	–	49.36	–	–	–
Airplane	59.42	57.28	58.9	59.26	59.59
Barbara	55.78	54.58	55.17	55.89	56.09
Boat	53.11	52.63	53.51	53.72	54.07
Lake	52.67	52.67	53.62	54.28	54.53
Peppers	53.41	52.31	54.78	54.93	55.1
Elaine	52.29	51.97	52.6	52.71	53.08
Average	54.69	53.78	54.99	55.34	55.61

Airplane and 27000 bits for Barbara). The reason lies in that numbers of rough blocks have to be employed when chasing maximum EC. However, the proposed scheme achieves evident superiority over Sachnev et al.' scheme for moderate EC. Referring to Tables 3 and 4, the proposed scheme improves Sachnev et al.'s by 2.37 dB on average for an EC of 10000 bits, and 1.83 dB for an EC of 20000 bits.

Li et al.'s scheme is an efficient extension of Lee et al.'s work [27] since it succeeded to turn part of errors from to-be-shifted to expandable. Fig. 7 shows that the proposed scheme almost significantly outperforms Li et al.'s for every test image whatever the EC is. Specifically, the average increase in PSNR is about or far more than 1 dB on most images (e.g. 0.96 dB for Barbara, 0.92 dB for Boat, 1.7 dB for Lake and 1.92 dB for Peppers). Tables 3 and 4 also shows that the proposed scheme improves Li et al.'s by 0.96 dB on average for an EC of 10000 bits, and 0.92 dB for an EC of 20000 bits.

For the proposed scheme, the most of computation time is spent in determining the optimal thresholds (T_2^*,T_1^*) by exhaustive search. Moreover, the embedding procedure for a given EC is repeated for several times to determine the best block size. Based on the consideration that for every block 2-pass PVO can surely achieve larger EC than 1-pass PVO, we propose to achieve fast thresholds determination by firstly determine maximum T_1 by taking $T_2 = 0$. Then all combinations of increasing T_2 and decreasing T_1 are tested to get (T_2^*,T_1^*). Table 2 shows the average computation times of one loop embedding for all test images, which are measured on an Intel CPU (i5,2.6 GHz) windows XP PC with 3.0 GB RAM.

Finally, although multi-pass PVO has been verified outperforms Peng et al.'s IPVO [33] and Ou et al.'s PVO-k [34] being an effective extension of original PVO, it should be noted that the performance of PVO-based schemes can be better improved by incorporating outstanding strategies such as pairwise PEE [23] and multiple histograms modification [25]. Therefore, it can be concluded that incorporating advanced embedding strategy into our method is really worthy of investigation in the future study.

5. Conclusion

In this paper, a novel RDH scheme based on multi-pass PVO embedding is proposed. Multi-pass PVO embedding mainly aims to provide more expandable prediction errors and thus lead to improved capacity-distortion trade-off. To this end, the k largest or smallest pixels within block are employed as independent data bit carriers to fulfill k-pass PVO embedding. Finally, by combining 1-pass PVO embedding and 2-pass PVO embedding, optimal combined embedding is presented. In contrast to Ou et al.'s combining PVO-1 embedding and PVO-2 embedding, although the pixel value order may change after data embedding, reversibility still can be guaranteed and independent data bit carrier thinking based multi-pass PVO embedding is experimentally verified better. Moreover, experimental results demonstrate that the proposed scheme also outperforms some state-of-the-art works.

However, although the proposed scheme yields a superior performance while utilizing the same number of blocks as Ou et al.'s scheme, the smoothest blocks with identical values, i.e., $p_{\sigma(1)} = \cdots = p_{\sigma(n)}$, still cannot be utilized for data embedding yet. We argue that the exploitation of these blocks would be a valuable research direction and we will investigate this issue in our future work.

Acknowledgments

This work is supported by the National Scientific Fund of China (No. 61170320).

References

[1] M. Wu, H. Yu, B. Liu, Data hiding in image and video: part II-designs and applications, IEEE Trans. Image Process. 12 (6) (2003) 696–705.
[2] A. Khan, A. Siddiqa, S. Munib, S.A. Malik, A recent survey of reversible watermarking techniques, Inf. Sci. 279 (20) (2014) 251–272.
[3] J. Fridrich, M. Goljan, R. Du, Lossless data embedding-New paradigm in digital watermarking, EURASIP J. Appl. Signal Process 2002 (2) (2002) 185–196.
[4] M.U. Celik, G. Sharma, A.M. Tekalp, E. Saber, Lossless generalized-LSB data embedding, IEEE Trans. Image Process. 14 (2) (2005) 253–266.
[5] J. Tian, Reversible data embedding using a difference expansion, IEEE Trans. Circ. Syst. Video Technol. 13 (8) (2003) 890–896.
[6] A.M. Alattar, Reversible watermark using the difference expansion of a generalized integer transform, IEEE Trans. Image Process. 13 (8) (2004) 1147–1156.
[7] S. Weng, Y. Zhao, J.S. Pan, R. Ni, Reversible watermarking based on invariability and adjustment on pixel pairs, IEEE Signal Process. Lett. 15 (2008) 721–724.
[8] D. Coltuc, Low distortion transform for reversible watermarking, IEEE Trans. Image Process. 21 (1) (2012) 412–417.
[9] D.M. Thodi, J.J. Rodriguez, Expansion embedding techniques for reversible watermarking, IEEE Trans. Image Process. 16 (3) (2007) 721–730.
[10] Y. Hu, H.K. Lee, J. Li, DE-based reversible data hiding with improved overflow location map, IEEE Trans. Circ. Syst. Video Technol. 19 (2) (2009) 250–260.
[11] X. Gao, L. An, Y. Yuan, D. Tao, X. Li, Lossless data embedding using generalized statistical quantity histogram, IEEE Trans. Circ. Syst. Video Technol. 21 (8) (2011) 1061–1070.
[12] V. Sachnev, H.J. Kim, J. Nam, S. Suresh, Y.Q. Shi, Reversible watermarking algorithm using sorting and prediction, IEEE Trans. Circ. Syst. Video Technol. 19 (7) (2009) 989–999.
[13] L. Luo, Z. Chen, M. Chen, X. Zeng, Z. Xiong, Reversible image watermarking using interpolation technique, IEEE Trans. Inf. Forensics Secur. 5 (1) (2010) 187–193.
[14] Q. Pei, X. Wang, Y. Li, H. Li, Adaptive reversible watermarking with improved embedding capacity, J. Syst. Softw. 86 (11) (2013) 2841–2848.
[15] B. Ou, X. Li, Y. Zhao, R. Ni, Reversible data hiding based on PDE predictor, J. Syst. Softw. 86 (10) (2013) 2700–2709.
[16] I.-C. Dragoi, D. Coltuc, Local-prediction-based difference expansion reversible watermarking, IEEE Trans. Image Process. 23 (4) (2014) 1779–1790.
[17] R.M. Rad, K. Wong, J.M. Guo, A unified data embedding and scrambling method, IEEE Trans. Image Process. 23 (4) (2014) 1463–1475.
[18] D. Coltuc, Improved embedding for prediction-based reversible watermarking, IEEE Trans. Inf. Forensics Secur. 6 (3) (2011) 873–882.
[19] 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.

[20] W. Hong, Adaptive reversible data hiding method based on error energy control and histogram shifting, Opt. Commun. 285 (2) (2012) 101–108.
[21] X. Li, W. Zhang, X. Gui, B. Yang, A novel reversible data hiding scheme based on two-dimensional difference-histogram modification, IEEE Trans. Inf. Forensics Secur. 8 (7) (2013) 1091–1100.
[22] 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.
[23] B. Ou, X. Li, J. Wang, High-fidelity reversible data hiding based on pixel-value-ordering and pairwise prediction-error expansion, J. Vis. Commun. Image R 39 (2016) 12–23.
[24] X. Li, W. Zhang, X. Gui, B. Yang, Efficient reversible data hiding based on multiple histograms modification, IEEE Trans. Inf. Forensics Secur. 10 (9) (2015) 2016–2027.
[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 R 38 (2016) 328–339.
[26] Z. Ni, Y.-Q. Shi, N. Ansari, W. Su, Reversible data hiding, IEEE Trans. Circ. Syst. Video Technol. 16 (3) (2006) 354–362.
[27] S.K. Lee, Y.H. Suh, Y.S. Ho, Reversible image authentication based on watermarking, in: Proc. IEEE ICME, 2006, pp. 1321–1324.
[28] 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.
[29] H.-T. Wu, J. Huang, Reversible image watermarking on prediction errors by efficient histogram modification, Signal Process. 92 (12) (2012) 3000–3009.
[30] H. Luo, F.-X. Yu, H. Chen, Z.-L. Huang, H. Li, P.-H. Wang, Reversible data hiding based on block median preservation, Inf. Sci. 181 (2) (2011) 308–328.
[31] 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.
[32] 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.
[33] F. Peng, X. Li, B. Yang, Improved pvo-based reversible data hiding, Digital Signal Process. 25 (2014) 255–265.
[34] 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.
[35] X. Wang, J. Ding, Q. Pei, A novel reversible image data hiding scheme based on pixel value ordering and dynamic pixel block partition, Inf. Sci. 310 (2015) 16–35.
[36] X. Qu, H.J. Kim, Pixel-based pixel value ordering predictor for high-fidelity reversible data hiding, Signal Process. 111 (2015) 249–260.