Information Sciences 467 (2018) 784–799

Contents lists available at ScienceDirect

Information Sciences

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

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

Wenguang He[a], Gangqiang Xiong[a], Shaowei Weng[b], Zhanchuan Cai[c], Yaomin Wang[c,*]

[a] School of Information Engineering, Guangdong Medical University, Guangdong 524023, China
[b] School of Information Engineering, Guangdong University of Technology, Guangdong 510006, China
[c] Faculty of Information Technology, Macau University of Science and Technology, Macau, China

ARTICLE INFO

Article history:
Received 7 October 2017
Revised 25 April 2018
Accepted 29 April 2018
Available online 9 May 2018

Keywords:
Reversible data hiding
Multi-pass pixel value ordering
Pairwise prediction-error expansion
High and low power version of PVO-based pairwise PEE

ABSTRACT

Pixel value ordering (PVO) prediction can achieve remarkable accuracy and thus provide a rather sharp histogram. In addition, the efficiency in histogram manipulation also attracts much attention in recent works. In this paper, a new reversible data hiding scheme based on multi-pass PVO and pairwise PEE is proposed. After dividing the host image into non-overlapped blocks, the largest/smallest two pixels within block are predicted to form a prediction-error pair and finally a 2D prediction-error histogram. Here, the third largest/smallest pixel no longer always serves as predicted value. Once any one error in a pair is shifted, we propose to adaptively re-calculate the other one. For smooth block, location information is considered and then more expandable errors are obtained. For normal block, the shifted pixel is involved in prediction and shifting two errors in a pair without carrying any bit can be avoided. Such multi-pass prediction leads to the so-called high and low power version of PVO-based pairwise PEE. Experimental results verify that their combination can achieve very efficient capacity-distortion trade-off and thus outperform previous PVO-based schemes.

1. Introduction

Data hiding [39] 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 [13], reversible data hiding is quite desirable in some quality sensitive fields such as medical imaging, remote sensing and military where permanent distortion is unacceptable. RDH scheme can be generally evaluated by its capacity-distortion performance. Besides a larger embedding capacity, less embedding distortion for a given EC is also expected. Up to now, many techniques have been devised for RDH, such as lossless compression [2,8], difference expansion (DE) [1,12,34], histogram shifting (HS) [14,15,17,27,35,40], prediction-error expansion (PEE) [3,6,7,9–11,16,18–26,28–33,36] and integer-to-integer transform [4,5,37,38].

Among the aforementioned techniques, PEE [33] is the most effective and extensively exploited one due to its efficient capacity-distortion trade-off. Unlike DE which simply uses the pixel difference for expansion, target pixel is predicted by

* Corresponding author.
E-mail address: 249668530@qq.com (Y. Wang).

https://doi.org/10.1016/j.ins.2018.04.088
0020-0255/

neighboring pixels in PEE and then the prediction-error is utilized for expansion embedding. As the local correlation of a larger neighborhood is exploited, PEE can produce differences with smaller magnitude and thus improve DE significantly. Such feature also motivated lots of efficient prediction methods [6,16,18,28,31,32] to enhance the embedding performance through accuracy improvement. Moreover, PEE can also be developed by various strategies such as context modification [3], adaptive embedding [11,23], two-dimensional histogram modification [24], pairwise PEE [7,19] and multiple histograms modification [25]. It is worth mentioning that prediction-errors are widely sorted in numerous PEE-based schemes. On one hand, prediction-errors with small magnitude can be preferentially utilized to achieve excellent performance for low capacities [9]. On the other hand, various embedding strategies can be applied on different prediction-error groups [23].

HS is another remarkable work of RDH, in which the peak point of intensity histogram is utilized for data embedding while those between the peak point and zero point are shifted. Nowadays HS has been incorporated into PEE and how the histogram is sharply distributed means a lot to embedding performance. For this reason, difference histogram [14] and prediction-error histogram (PEH) [35] are sequentially proposed. The framework of histogram modification based RDH was presented in [15].

Recently, several high-fidelity RDH schemes evolving from PEE, collectively called pixel-value-ordering (PVO) [10,20–22,26,29,30,36], have attracted much attention for their remarkable performance at low capacity. Li et al. [26] firstly proposed novel PVO prediction, in which errors are generated in a block-wise manner and the largest/smallest pixel is predicted by the second 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 [12,16,23,32]. Moreover, the block complexity is measured to achieve embedding primitive selection for extra performance enhancement. Following Li et al.'s work, many PVO-based schemes have been proposed. Peng et al. [29] proposed to introduce the relative location relationship of pixels into predicted value calculation. Then bins 1 and 0 instead of original bin 1 are utilized for data embedding to achieve an improved PVO. In [20], pixels whose values equal to the maximum/minimum are united as one bit data carrier. Suppose k denotes the number of maximum-valued/minimum-valued pixels, original PVO is extended to PVO-k embedding. Peng et al.'s work was further developed in [36] with dynamic blocking strategy which divides block in smooth region into four sub-blocks. The idea of sub-blocks is verified to be better than employing blocks with higher complexity for sufficient EC. Afterwards we [10] further developed the idea of combining various-sized blocks. Qu and Kim [30] provided larger EC while maintaining marked image fidelity by extending PVO into pixel-based PVO, in which each pixel is predicted by its sorted context.

For the aforementioned PVO-based schemes [10,20,26,29,30,36], novelties generally lie in prediction. Specially, after prediction-error generation how PEH is efficiently manipulated also means a lot to embedding performance. From this point of view, the incorporation of advanced prediction method and efficient PEH manipulation method is worthy of in-depth study. In recent works, Ou et al. [21] proposed to incorporate multiple histograms modification with PVO. Later, the incorporation of pairwise PEE and PVO was also considered [22]. Pairwise PEE is a typical higher-dimensional PEH manipulation which differs from conventional methods in processing errors pair by pair. For a pair of errors both of which are expandable, one of the combinations of bits "00", "01" and "10" instead of two bits of data is embedded. In this way embedding distortion can be reduced at the cost of embedding $\log_2 3$ instead of two bits. Pairwise PEE apparently utilizes the correlation of adjacent errors, or the correlation of pixels which produce pairing errors. However, whether in original pairwise PEE [19] which relies on rhombus predictor [32] or PVO-based pairwise PEE [22] which relies on PVO predictor, it can be seen that although pairing errors are jointly modified pixels producing pairing errors are always independently predicted. This exposes the fact that the above incorporations between existing prediction methods and pairwise PEE are superficial and far from efficient, and this may lead to incomplete utilization of pixel correlation. One improvement to be considered is that pixel correlation should be utilized not only at modification stage but at prediction stage, i.e., the correlation of producing pairing errors can lead to a kind of lubricant with which prediction method and pairwise PEE can be better incorporated. In this paper, we focus on exploiting the correlation among predicting the largest/smallest two pixels within block in [22] and then present the idea of multi-pass PVO. The incorporation of multi-pass PVO and pairwise PEE enables flexible optimization. For smooth block, location information is introduced into prediction to obtain more expandable errors. For normal block, the shifted pixel is involved in prediction so that shifting two errors in a pair without carrying any bit can be avoided. Experimental results demonstrate that the research work in this paper leads to really optimized PVO-based pairwise PEE.

The rest of the paper is organized as follows. In Section 2, some related works are briefly introduced. Section 3 presents the proposed scheme in detail. The experimental results and performance comparison with other schemes are shown in Section 4. Finally, Section 5 concludes this paper.

2. Related works

In this section, the basic principle of Peng et al.'s IPVO [29] and its 2D-PEH modification version are firstly reviewed. Then how pairwise PEE is incorporated into PVO is explained and evaluated. For clarity, only the data embedding on the maximum pixel of a block is presented.

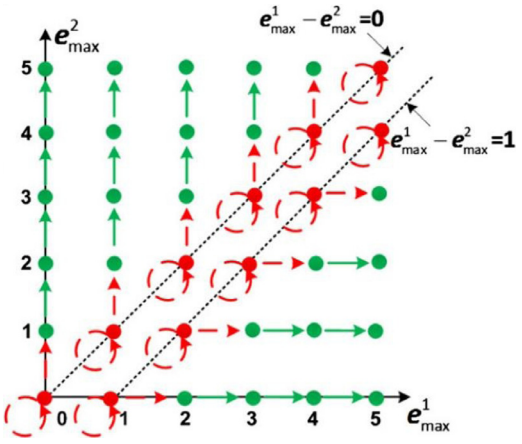

Fig. 1. Modification manner on the 2D-PEH of Peng et al.'s IPVO.

2.1. Peng et al.'s IPVO [29]

PVO generates errors in a block-wise manner. The host image is first divided into non-overlapped blocks. For each block consisted of n pixels, pixels are scanned to form a sequence $(p_1, p_2, \ldots, p_n)$ and next sorted in ascending order to obtain $(p_{\sigma(1)}, p_{\sigma(2)}, \ldots, p_{\sigma(n)})$ where $\sigma : \{1, \ldots, n\} \to \{1, \ldots, n\}$ is the unique one-to-one mapping such that: $p_{\sigma(1)} \le p_{\sigma(2)} \le \cdots \le p_{\sigma(n)}$, $\sigma(i) < \sigma(j)$ if $p_{\sigma(i)} = p_{\sigma(j)}$ and $i < j$. Unlike original PVO which simply uses $p_{\sigma(n-1)}$ to predict $p_{\sigma(n)}$ and only bin 1 is utilized, IPVO renews the prediction-error as

$$e_{max} = p_u - p_v \tag{1}$$

where $u = min(\sigma(n), \sigma(n-1)), v = max(\sigma(n), \sigma(n-1))$. Then the prediction-error e_{max} is no longer always non-negative and data-carrying bins can turn to be bins 1 and 0, i.e., the marked prediction-error e'_{max} is obtained as

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

where $b \in \{0, 1\}$ is a to-be-embedded bit. Accordingly, the maximum is modified to

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

According to (3), only the maximum $p_{\sigma(n)}$ is either increased or unchanged while $p_{\sigma(n-1)}$ is always unchanged in IPVO embedding. In [22], the correlation between $p_{\sigma(n-1)}$ and $p_{\sigma(n)}$ is reinterpreted as two new prediction-errors

$$\begin{cases} e^1_{max} = p_u - p_{\sigma(n-2)} \\ e^2_{max} = p_v - p_{\sigma(n-2)} \end{cases} \tag{4}$$

Apparently, the correlation among e_{max}, e^1_{max} and e^2_{max} is

$$e_{max} = e^1_{max} - e^2_{max} \tag{5}$$

In this way, expandable errors $\{e_{max} | e_{max} \in \{0, 1\}\}$ turn to be expandable error pairs $\{(e^1_{max}, e^2_{max}) | e^1_{max} - e^2_{max} \in \{0, 1\}\}$. Then IPVO embedding can be interpreted using histogram modification for a 2D-PEH as shown in Fig. 1. Such 2D modification manner is described as a reversible 2D mapping, where the mapping arrow is used 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.

2.2. Ou et al.'s PVO-based pairwise PEE [22]

Notice that still only one error (e.g., e^1_{max} or e^2_{max}) which relates to $p_{\sigma(n)}$ is modified while the other one is always unchanged in 2D-PEH modification of IPVO. With the inspiration of modifying both e^1_{max} and e^2_{max}, Ou et al. designed a more efficient modification method which can be described as another reversible 2D mapping as shown in Fig. 2. For simplicity,

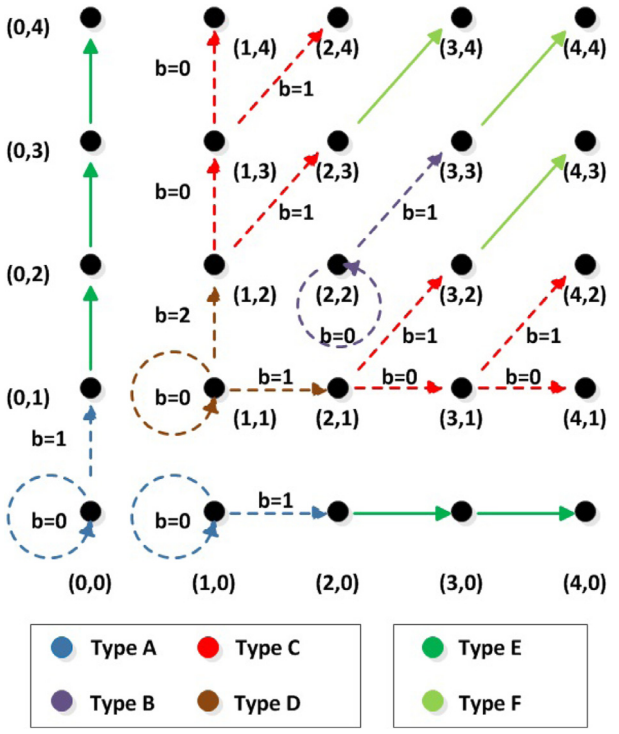

Fig. 2. 2D mapping for Ou et al.'s PVO-based pairwise PEE.

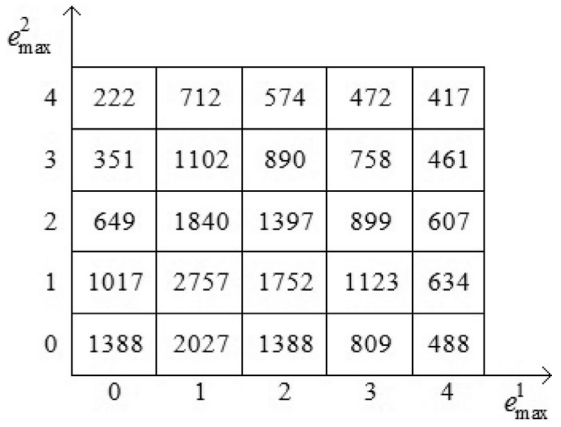

Fig. 3. 2D-PEH generated by Ou et al.'s scheme on image Lena.

the mappings of Figs. 2 and 1 are denoted as f_1 and f_2, respectively. Compared with f_2, expandable pairs in f_1 are redefined as four types:

- A: ($e^1_{max} = 0$ and $e^2_{max} = 0$) or ($e^1_{max} = 1$ and $e^2_{max} = 0$).
- B: $e^1_{max} = 2$ and $e^2_{max} = 2$.
- C: ($e^1_{max} \geq 2$ and $e^2_{max} = 1$) or ($e^1_{max} = 1$ and $e^2_{max} \geq 2$).
- D: $e^1_{max} = 1$ and $e^2_{max} = 1$.

Then the rest Type-E and Type-F pairs are all shiftable.

- E: ($e^1_{max} \geq 2$ and $e^2_{max} = 0$) or ($e^1_{max} = 0$ and $e^2_{max} \geq 1$).
- F: ($e^1_{max} \geq 2$ and $e^2_{max} \geq 3$) or ($e^1_{max} \geq 3$ and $e^2_{max} \geq 2$).

Apparently, the objective of f_1 is to use the bins near (1,1) as many as possible for carrying bits. Fig. 3 shows part of the 2D-PEH on image Lena when the block size is set as 2×3 and only the prediction-errors of maximum pixels are computed. Referring to Fig. 3, it can be seen that f_1 does adapt with the characteristics of 2D-PEH. However, it should be noted that

Table 1
Proportions of all types of pairs from various block groups.

NL	Type-A	Type-B	Type-C	Type-D	Type-E	Type-F
[0, 49]	0.156	0.049	0.276	0.121	0.195	0.203
[50, 99]	0.076	0.035	0.26	0.063	0.16	0.406
[100, 149]	0.029	0.017	0.172	0.022	0.098	0.662
[150, 199]	0.016	0.015	0.145	0.018	0.078	0.728

the distribution of 2D-PEH varies with involved blocks. The average distortion for processing one pair in terms of MSE is $\frac{1}{2}$, 1, $\frac{3}{2}$, $\frac{2}{3}$, 1 and 2 respectively according to pair type. Blocks in rough regions may produce fewer expandable pairs but more shiftable pairs and thus leads to adverse effect on capacity-distortion trade-off. Based on this consideration, the idea of adaptive mapping proposes to modify pairs with $NL \le T_1$ by the mapping f_1 and modify pairs with $T_1 < NL \le T_2$ by the mapping f_2. Here, the noise level NL is taken as the smoothness measurement and calculated as the sum of absolute differences between two consecutive pixels in the context.

To further verify how the distribution of 2D-PEH varies with involved blocks, Table 1 presents the corresponding proportions of all types of pairs from various block groups. It can be obviously seen that with the increase of NL, the proportion of Type-F pair increases significantly while the ones of other types of pairs decrease. So, in view of the severe distortion introduced by processing Type-F pair, it can be inferred that f_1 can only take advantage of PVO on smooth blocks. On the other hand, Table 1 also reminds the new mapping which succeeds f_1 in normal blocks processing the importance of proper handling of Type-F pairs if without renewing current prediction method.

3. Proposed scheme

In this section, a new RDH scheme based on optimized PVO-based pairwise PEE is presented. The section is organized as follows: in Sections 3.1 and 3.2 we first present the so-called high power version of PVO-based pairwise PEE and low power version of PVO-based pairwise PEE by describing the pairwise modification on the maximum pixel pairs. Then the optimal-threshold-determination for the combined embedding is given in Section 3.3. Finally, Section 3.4 gives the detailed embedding and extracting procedures of the proposed scheme.

3.1. High power version of PVO-based pairwise PEE

As known before, the advantage of f_1 lies in the utilization of highest bins for carrying bits. Referring to Table 1, we can see that Type-C pair has the highest proportion if Type-F pair is not included. Then Type-E pair has the second highest one. Here Type-E pair is to be brought into data embedding.

Referring to Fig. 2, for pairs $\{(e^1_{max}, e^2_{max}) | e^1_{max} \ge 2\}$ (except (2,2)), e^1_{max} is always enlarged by 1 and only e^2_{max} is utilized for carrying bits in current mapping f_1. As learned from IPVO, the utilization of e^2_{max} can be improved if the relative location relationship of pixels is considered. Suppose e^1_{max} and e^2_{max} are successively calculated as

$$e^1_{max} = p_u - p_{\sigma(n-2)} \tag{6}$$

$$e^2_{max} = \begin{cases} p_v - p_{\sigma(n-2)}, & \text{if } e^1_{max} < 2 \text{ or } p_u \le p_v \\ p_s - p_t, & \text{else} \end{cases} \tag{7}$$

where $s = min(\sigma(n-2), v), t = max(\sigma(n-2), v)$. It is obvious that we have original e^1_{max} and e^2_{max} when $e^1_{max} < 2$ or $p_u \le p_v$. Then we modify them by f_1. In other cases e^2_{max} will be no longer always non-negative. Then we propose to enlarge e^1_{max} by 1 and embed one bit into e^2_{max} as

$$e^{1\prime}_{max} = e^1_{max} + 1 \tag{8}$$

$$e^{2\prime}_{max} = \begin{cases} e^2_{max} + b, & \text{if } e^2_{max} = 1 \\ e^2_{max} + 1, & \text{if } e^2_{max} > 1 \\ e^2_{max} - b, & \text{if } e^2_{max} = 0 \\ e^2_{max} - 1, & \text{if } e^2_{max} < 0 \end{cases} \tag{9}$$

Accordingly, p_u is enlarged by 1 and the maximum among $p_{\sigma(n-2)}$ and p_v, which is always p_v, is modified as

$$p'_u = p_{\sigma(n-2)} + e^{1\prime}_{max} = p_u + 1 \tag{10}$$

$$p'_v = p_{\sigma(n-2)} + |e^{2\prime}_{max}| = \begin{cases} p_v + b, & \text{if } e^2_{max} \in \{0, 1\} \\ p_v + 1, & \text{else} \end{cases} \tag{11}$$

Since p_u is similarly enlarged by 1 and p_v is increased or unchanged for carrying one bit just like what f_1 did before, reversibility can be guaranteed. However, as e^2_{max} can be taken as $p_{\sigma(n-2)} - p_v$ or $p_v - p_{\sigma(n-2)}$ according to (7), part of once

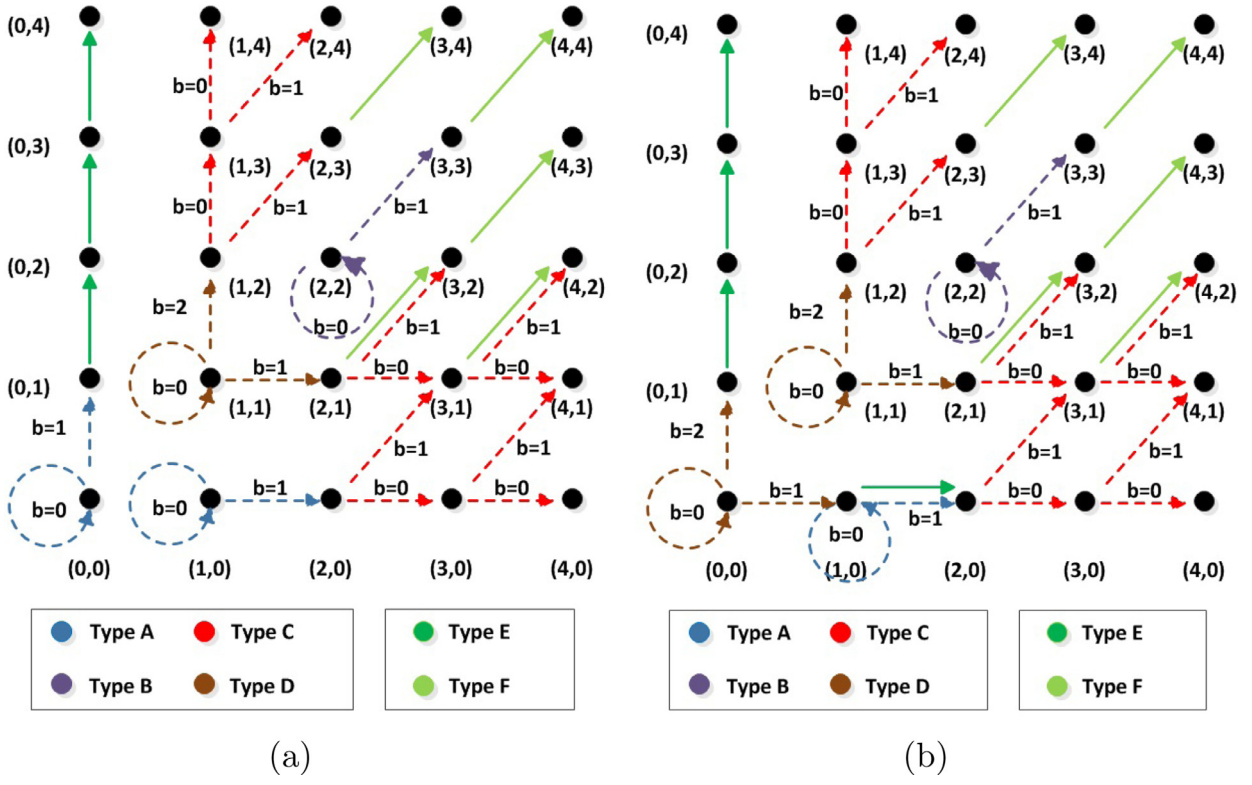

Fig. 4. 2D mapping for high power version of PVO-based pairwise PEE.

expandable pairs with $e^1_{max} \geq 2$ and $e^2_{max} = 1$ may turn to be shiftable ones with $e^1_{max} \geq 2$ and $e^2_{max} = -1$. This prevents us from considering the further utilization of pairs $\{(e^1_{max}, e^2_{max}) | e^2_{max} \geq 2\}$ (except (2,2)), for which e^2_{max} is always enlarged by 1 and only e^1_{max} is utilized for carrying bits. Referring to Fig. 3, bin (0,k) is far lower than bin (1,k) when $k \geq 2$. So we conclude that the utilization of pairs $\{(e^1_{max}, e^2_{max}) | e^2_{max} \geq 2\}$ (except (2,2)) should be maintained. In this way, the effect of our work can be equal to that of the redefinition of error pairs generated by (4):

- A: ($e^1_{max} = 0$ and $e^2_{max} = 0$) or ($e^1_{max} = 1$ and $e^2_{max} = 0$).
- B: $e^1_{max} = 2$ and $e^2_{max} = 2$.
- C: ($e^1_{max} = 1$ and $e^2_{max} \geq 2$) or ($e^1_{max} \geq 2$ and $e^2_{max} = 0$) or ($e^1_{max} \geq 2$ and $e^2_{max} = 1$ and $v < \sigma(n-2)$).
- D: $e^1_{max} = 1$ and $e^2_{max} = 1$.
- E: $e^1_{max} = 0$ and $e^2_{max} \geq 1$.
- F: ($e^1_{max} \geq 2$ and $e^2_{max} \geq 3$) or ($e^1_{max} \geq 3$ and $e^2_{max} \geq 2$) or ($e^1_{max} \geq 2$ and $e^2_{max} = 1$ and $v > \sigma(n-2)$).

The corresponding modification method is described as a new reversible 2D mapping as shown in Fig. 4(a). We denote this new mapping as f_H. In f_H, the Type-F pair involves part of once Type-C pair. This can be compensated by bringing most of Type-E pairs into data embedding. Actually, the new mapping f_H can be further optimized for special cases. Here we focus on constructing the mapping from (0,0) to (1,0) which is discarded in f_1. For pair (0,0), there is always $\sigma(n-2) < u < v$ even after enlarging e^1_{max} or e^2_{max}. For pair (1,0), there could be $\sigma(n-2) < u < v$ or $u < \sigma(n-2) < v$. So, the mapping from (0,0) to (1,0) could be available if shifting pair (1,0) with $\sigma(n-2) < u < v$ to (2,0). In this way, pair (0,0) can be categorized into Type-D and then we have another redefinition of error pairs generated by (4):

- A: $e^1_{max} = 1$ and $e^2_{max} = 0$ and $u < \sigma(n-2)$.
- B: $e^1_{max} = 2$ and $e^2_{max} = 2$.
- C: ($e^1_{max} = 1$ and $e^2_{max} \geq 2$) or ($e^1_{max} \geq 2$ and $e^2_{max} = 0$) or ($e^1_{max} \geq 2$ and $e^2_{max} = 1$ and $v < \sigma(n-2)$).
- D: ($e^1_{max} = 0$ and $e^2_{max} = 0$) or ($e^1_{max} = 1$ and $e^2_{max} = 1$).
- E: ($e^1_{max} = 0$ and $e^2_{max} \geq 1$) or ($e^1_{max} = 1$ and $e^2_{max} = 0$ and $u > \sigma(n-2)$).
- F: ($e^1_{max} \geq 2$ and $e^2_{max} \geq 3$) or ($e^1_{max} \geq 3$ and $e^2_{max} \geq 2$) or ($e^1_{max} \geq 2$ and $e^2_{max} = 1$ and $v > \sigma(n-2)$).

Taking into account that this new mapping is more like a supplement to f_H, we denote it as f_{HS}, as shown in Fig. 4(b).

3.2. Low power version of PVO-based pairwise PEE

As discussed before, for pairs $\{(e^1_{max}, e^2_{max})|e^1_{max} \geq 2\}$ (except (2,2)), e^1_{max} is always enlarged by 1 and only e^2_{max} is utilized for carrying bits in current mapping f_1. As *NL* increases, the significant growth of Type-F pair actually indicates that we are more likely to obtain shiftable e^2_{max} whose value is larger than 1. This reflects the loose correlation between $p_{\sigma(n-2)}$ and p_v. Notice that p_u is surely enlarged by 1, here we propose to turn to utilize the correlation between p_u and p_v. Suppose e^1_{max} and e^2_{max} are successively calculated as

$$e^1_{max} = p_u - p_{\sigma(n-2)} \tag{12}$$

$$e^2_{max} = \begin{cases} p_v - p_{\sigma(n-2)}, & \text{if } e^1_{max} < 2 \text{ or } p_u \leq p_v \\ p_v - (p_u - 2), & \text{else} \end{cases} \tag{13}$$

It is obvious that we have $e^1_{max} \geq 2$ and $p_u > p_v$ in the second case. Then we propose to enlarge e^1_{max} by 1 and embed one bit into e^2_{max} as

$$e^{1\prime}_{max} = e^1_{max} + 1 \tag{14}$$

$$e^{2\prime}_{max} = \begin{cases} e^2_{max} + b, & \text{if } e^2_{max} = 1 \\ e^2_{max}, & \text{else} \end{cases} \tag{15}$$

Accordingly, p_u and p_v are modified as

$$p'_u = p_{\sigma(n-2)} + e^{1\prime}_{max} = p_u + 1 \tag{16}$$

$$p'_v = (p_u - 2) + e^{2\prime}_{max} = \begin{cases} p_v + b, & \text{if } e^2_{max} = 1 \\ p_v, & \text{else} \end{cases}. \tag{17}$$

Notice that $p_u > p_v$, so we have $e^2_{max} \leq 1$. In this way, e^2_{max} can never be shifted without carrying any bit. Similarly for pairs $\{(e^1_{max}, e^2_{max})|e^2_{max} \geq 2\}$ (except (2,2)), suppose e^2_{max} and e^1_{max} are successively calculated as

$$e^2_{max} = p_v - p_{\sigma(n-2)} \tag{18}$$

$$e^1_{max} = \begin{cases} p_u - p_{\sigma(n-2)}, & \text{if } e^2_{max} < 2 \text{ or } p_u \geq p_v \\ p_u - (p_v - 2), & \text{else} \end{cases} \tag{19}$$

It is obvious that we have $e^2_{max} \geq 2$ and $p_u < p_v$ in the second case. Then we propose to enlarge e^2_{max} by 1 and embed one bit into e^1_{max} as

$$e^{2\prime}_{max} = e^2_{max} + 1 \tag{20}$$

$$e^{1\prime}_{max} = \begin{cases} e^1_{max} + b, & \text{if } e^1_{max} = 1 \\ e^1_{max}, & \text{else} \end{cases} \tag{21}$$

Accordingly, p_u and p_v are modified as

$$p'_v = p_{\sigma(n-2)} + e^{2\prime}_{max} = p_v + 1 \tag{22}$$

$$p'_u = (p_v - 2) + e^{1\prime}_{max} = \begin{cases} p_u + b, & \text{if } e^1_{max} = 1 \\ p_u, & \text{else} \end{cases} \tag{23}$$

As $p_u < p_v$, we have $e^1_{max} \leq 1$. In this way, e^1_{max} can never be shifted without carrying any bit too.

Such modification actually embeds bits into bins $(k, k+1)$ and $(k+1, k)$ instead of bins $(1, k+1)$ and $(k+1, 1)$ where $k \geq 1$. This corresponds to the redefinition of Type-C pair. Finally, after discarding the mapping from (2,2) to (2,2) or (3,3), the 2D mapping for low power version of PVO-based pairwise PEE is designed as shown in Fig. 5. We denote this new mapping as f_L and its effect can also be equal to that of the redefinition of error pairs generated by (4):

- A: ($e^1_{max} = 0$ and $e^2_{max} = 0$) or ($e^1_{max} = 1$ and $e^2_{max} = 0$).
- C: $e^1_{max} \geq 1$ and $e^2_{max} \geq 1$ and $|e^1_{max} - e^2_{max}| = 1$.
- D: $e^1_{max} = 1$ and $e^2_{max} = 1$.
- E: ($e^1_{max} \geq 2$ and $e^1_{max} - e^2_{max} \geq 2$) or ($e^2_{max} \geq 2$ and $e^2_{max} - e^1_{max} \geq 2$) or ($e^1_{max} = 0$ and $e^2_{max} = 1$).

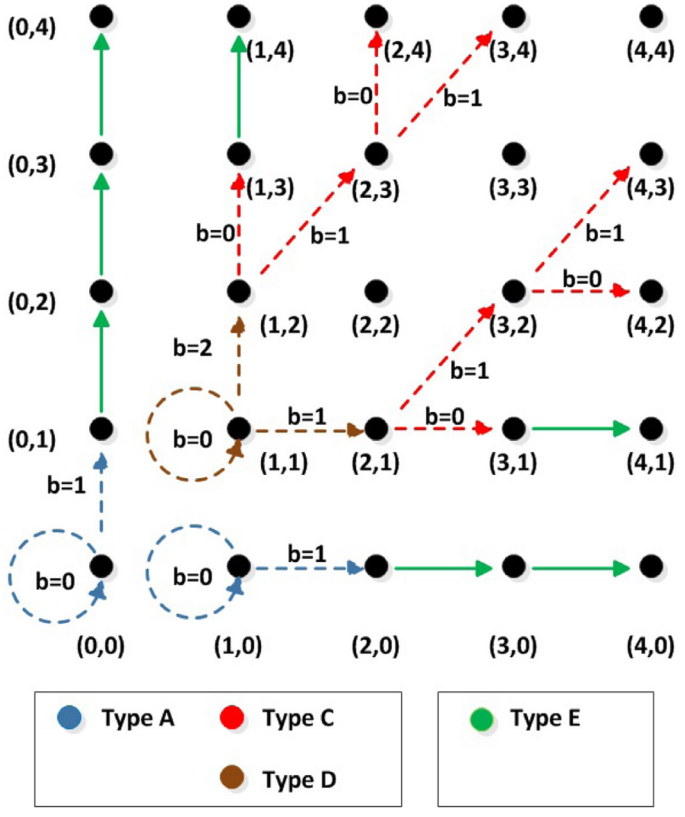

Fig. 5. 2D mapping for low power version of PVO-based pairwise PEE.

3.3. Optimal-threshold-determination for combined embedding

The combination of f_H and f_L forms the proposed scheme. Blocks are first divided into smooth block with $NL \le T_1$, normal block with $T_1 < NL \le T_2$ and rough block with $NL > T_2$. Then error pairs derived from smooth blocks are modified by f_H, the ones derived from normal blocks are modified by f_L. Rough blocks are excluded from data embedding.

Next, for a given EC we have to determine the optimal combination of two thresholds T_1 and T_2, which yields the least distortion in terms of MSE while fulfilling the capacity. Suppose $E_H(t)$ and $D_H(t)$ calculate respectively the capacity and distortion generated by f_H on pairs with $NL = t$, $E_L(t)$ and $D_L(t)$ calculate respectively the capacity and distortion generated by f_L on pairs with $NL = t$. Algorithms 1 and 2 describe the whole calculation procedures.

Finally, the optimal threshold (T_1^*, T_2^*) is determined as

$$\begin{cases} (T_1^*, T_2^*) = \underset{T_1, T_2}{argmin} \sum_{t=0}^{T_1} D_H(t) + \sum_{t=T_1+1}^{T_2} D_L(t) \\ subject \quad to \quad \sum_{t=0}^{T_1} E_H(t) + \sum_{t=T_1+1}^{T_2} E_L(t) \ge EC + 60 + L_{clm} \end{cases} \tag{24}$$

where $60 + L_{clm}$ is the length of auxiliary information and compressed location map. Fig. 6 shows the performance comparison on image Lena by using the single mapping f_H, f_L and their combination, where the block size is 2×3. It can be seen that f_L is superior than f_H with moderate EC, and the advantage shrinks as EC approaches its maximum. The joining of f_H enables maintaining such advantage for longer times. This verifies the importance of processing error pairs dependent on the smoothness.

3.4. Implementation of the proposed scheme

Using the smallest three pixels, one can obtain another available prediction-error pair (e^1_{min}, e^2_{min}) as

$$\begin{cases} e^1_{min} = p_{\sigma(3)} - p_u \\ e^2_{min} = p_{\sigma(3)} - p_v \end{cases} \tag{25}$$

where $u = max(\sigma(1), \sigma(2)), v = min(\sigma(1), \sigma(2))$. Then according to f_H/f_L, e^1_{min} and e^2_{min} will be increased or unchanged by reducing $p_{\sigma(1)}$ and $p_{\sigma(2)}$ while maintaining $p_{\sigma(3)}$ for embedding data. During modification e^1_{min} plays the role of e^1_{max}

```
Input: A sequence of blocks each of which is consisted of
       n Sorted pixels (p_{σ(1)}, p_{σ(2)}, ···, p_{σ(n)})
for each block
    Calculate its noise level as t
    if (e^1_max = 0 and e^2_max = 0) or (e^1_max = 1 and e^2_max = 0)
        E_H(t) = E_H(t) + 1; D_H(t) = D_H(t) + 1/2;
    elseif e^1_max = 2 and e^2_max = 2
        E_H(t) = E_H(t) + 1; D_H(t) = D_H(t) + 1;
    elseif e^1_max ≥ 2 and e^2_max = 0
        E_H(t) = E_H(t) + 1; D_H(t) = D_H(t) + 3/2;
    elseif e^1_max ≥ 2 and e^2_max = 1
        if v < σ(n − 2)
            E_H(t) = E_H(t) + 1; D_H(t) = D_H(t) + 3/2;
        else
            D_H(t) = D_H(t) + 2;
        end
    elseif e^1_max = 1 and e^2_max ≥ 2
        E_H(t) = E_H(t) + 1; D_H(t) = D_H(t) + 3/2;
    elseif e^1_max = 1 and e^2_max = 1
        E_H(t) = E_H(t) + log_2 3; D_H(t) = D_H(t) + 2/3;
    elseif e^1_max = 0 and e^2_max ≥ 1
        D_H(t) = D_H(t) + 1;
    else
        D_H(t) = D_H(t) + 2;
    end
end
```

Algorithm 1. Calculation of '$E_H(t)$' and '$D_H(t)$'.

```
Input: A sequence of blocks each of which is consisted of
       n Sorted pixels (p_{σ(1)}, p_{σ(2)}, ···, p_{σ(n)})
for each block
    Calculate its noise level as t
    if (e^1_max = 0 and e^2_max = 0) or (e^1_max = 1 and e^2_max = 0)
        E_L(t) = E_L(t) + 1; D_L(t) = D_L(t) + 1/2;
    elseif e^1_max ≥ 1 and e^2_max ≥ 1 and |e^1_max − e^2_max| = 1
        E_L(t) = E_L(t) + 1; D_L(t) = D_L(t) + 3/2;
    elseif e^1_max = 1 and e^2_max = 1
        E_L(t) = E_L(t) + log_2 3; D_L(t) = D_L(t) + 2/3;
    elseif e^1_max = 0 and e^2_max = 1
        D_L(t) = D_L(t) + 1;
    elseif e^1_max ≥ 2 and e^1_max − e^2_max ≥ 2
        D_L(t) = D_L(t) + 1;
    elseif e^2_max ≥ 2 and e^2_max − e^1_max ≥ 2
        D_L(t) = D_L(t) + 1;
    end
end
```

Algorithm 2. Calculation of '$E_L(t)$' and '$D_L(t)$'.

while e^2_{min} plays the role of e^2_{max}. Suppose there is an error pair (e^1_{min}, e^2_{min}) to be modified by f_H and we have $e^1_{min} \geq 2$ and $e^2_{min} = 1$, it will be categorized into Type-C if $v > \sigma(3)$, or Type-F if $v < \sigma(3)$. Then f_H first reduces p_u by 1 to achieve $e^{1\prime}_{min} = e^1_{min} + 1$. Next, embed one bit into e^2_{min} by fulfilling $p'_v = p_v - b$ if $v > \sigma(3)$; Otherwise simply reduce p_v by 1.

The whole embedding only enlarges $\{p_{\sigma(n-1)}, p_{\sigma(n)}\}$ and reduces $\{p_{\sigma(1)}, p_{\sigma(2)}\}$ while $p_{\sigma(i)}, i \in \{3, \ldots, n-2\}$ are unchanged. So we have $\{p_{\sigma(1)}, p_{\sigma(2)}\} \leq p_{\sigma(3)} \leq \cdots \leq p_{\sigma(n-2)} \leq \{p_{\sigma(n-1)}, p_{\sigma(n)}\}$ at encoder, and $\{p'_{\sigma(1)}, p'_{\sigma(2)}\} \leq p_{\sigma(3)} \leq \ldots \leq p_{\sigma(n-2)} \leq \{p'_{\sigma(n-1)}, p'_{\sigma(n)}\}$ at decoder. Such invariant order and the nature relative location relationship of pixels guarantee the reversibility. Now the complete data embedding procedure is given below.

Step 1. Location map construction

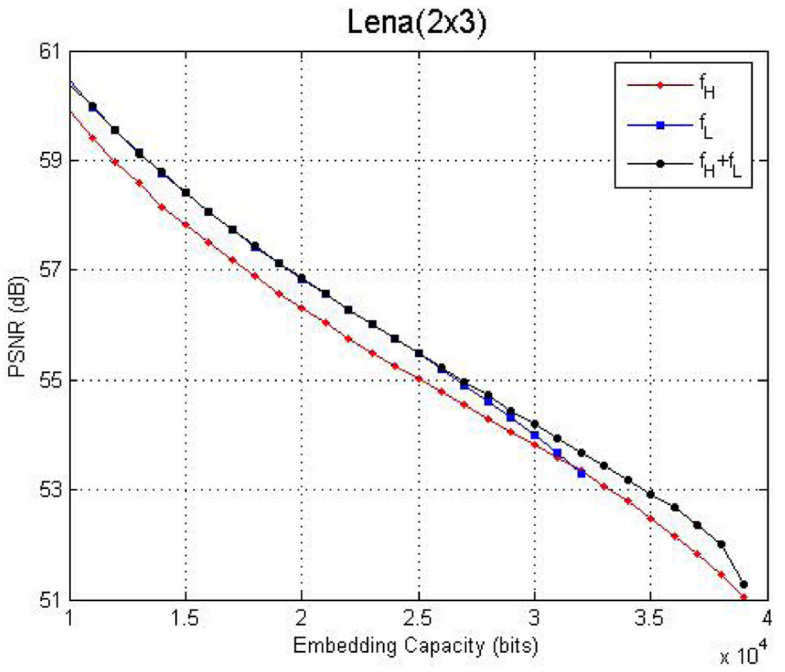

Fig. 6. Performance evaluation by comparing the single mapping of f_H, f_L and their combination.

Overflow/underflow may occur on pixels whose values equal to 0 or 255 for 8 bit gray-scale image. After dividing the host image, we construct a location map (*LM*) in a block-wise manner. A bit 0 in *LM* represents that the block has no pixel with value of 0 or 255 and it is available for payload embedding. Otherwise, a bit 1 represents that the block is unavailable and will be ignored. Using arithmetic coding, *LM* is losslessly compressed into a shorter bit stream.

Step 2. Payload preparation

The first two rows of image are excluded from payload embedding for the reason of auxiliary information transmission. Record the least-significant-bits of first $60 + L_{clm}$ pixels, where L_{clm} is the length of compressed location map. These LSBs and secret message make up the real payload.

Step 3. Payload embedding

Process blocks in raster-scan order to embed payload. For available block, calculate the noise level *NL* and two error pairs (e^1_{max}, e^2_{max}) and (e^1_{min}, e^2_{min}). If $NL \le T_1^*$, modify error pairs by the mapping f_H; if $T_1^* < NL \le T_2^*$, modify error pairs by the mapping f_L; else, keep the pairs unchanged.

Step 4. Auxiliary information embedding

Repeat the embedding until the payload is embedded and then record the end position (i.e., the block index where payload embedding ends). Next, embed the auxiliary information into the first $60 + L_{clm}$ pixels by LSB replacement. Besides the compressed location map, auxiliary information also includes the following components:

- Block size u (2 bits) and v (2 bits).
- Thresholds T_1^* (11 bits) and T_2^* (11 bits).
- End position (18 bits).
- Length of the compressed location map (16 bits).

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

Step 1. Auxiliary information extraction

Extract the LSBs of the first $60 + L_{clm}$ pixels of the marked image to retrieve the auxiliary information, including the values of block size, thresholds, the end position, and the compressed location map.

Step 2. Payload extraction and block recovery

Divide the marked image into non-overlapped blocks and start extraction from the last data-carrying block. For available block, calculate the noise level *NL* and two marked error pairs $(e^{1\prime}_{max}, e^{2\prime}_{max})$ and $(e^{1\prime}_{min}, e^{2\prime}_{min})$. For each error pair, determine its 2D mapping according to *NL*. Then recover its original value by the inverse mapping and extract the data bit.

Step 3. Image recovery

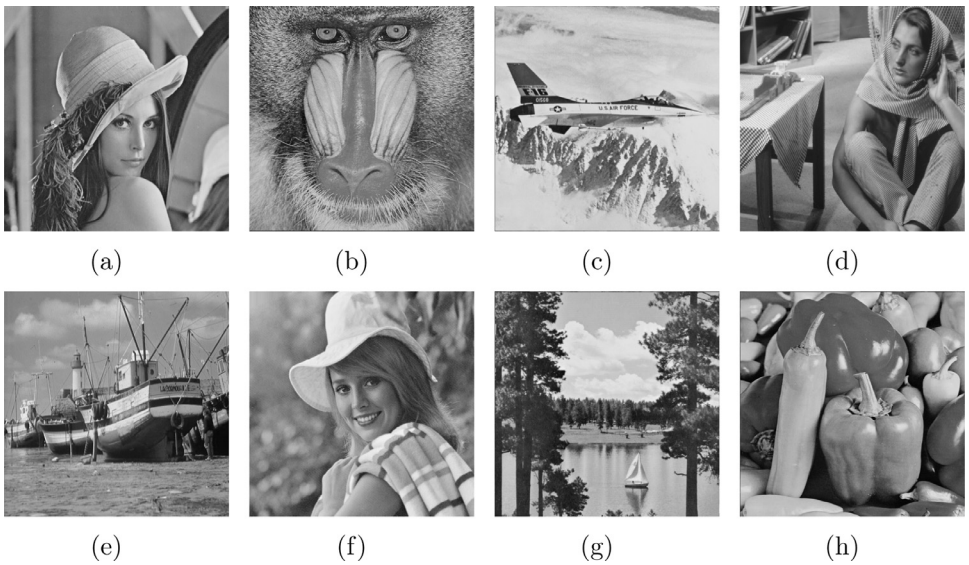

Fig. 7. Original test images: (a) Lena, (b) Baboon, (c) Airplane, (d) Barbara, (e) Boat, (f) Elaine, (g) Lake, (h) Peppers.

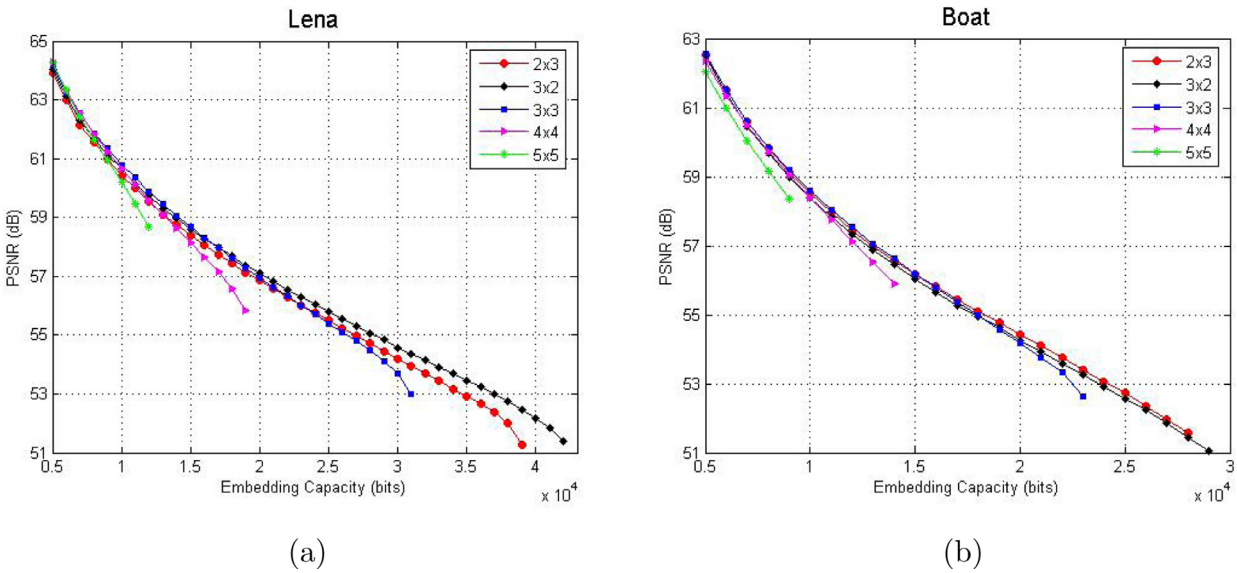

Fig. 8. Performance comparison by using different block sizes.

Separate the extracted data bits into the LSB stream and the secret message. Then replace the LSBs of the first $60 + L_{clm}$ pixels with the LSB stream. Finally, both the host image and secret data are recovered.

4. Experimental results

This section presents experimental results from the proposed scheme. The performance of the proposed scheme will be compared with those of five state-of-the-art schemes [20,22,29,30,32]. For these experiments, eight 512×512 gray-scale images served as test images. Except for Barbara, all images are downloaded from the USC-SIPI image database, as depicted in Fig. 7. As the proposed scheme generates prediction-errors in a block-wise manner, the embedding procedure has to be conducted for different block sizes. Then the one which achieves the best performance is selected. In our scheme, the block size is set as u, $v \in \{2, 3, 4, 5\}$ while 2×2 is abandoned for the reason of insufficient pixels. Fig. 8 shows the embedding performance for several block sizes. It can be seen that the PSNR for a given EC varies with the block size. Unlike original PVO where larger block size generally yields higher PSNR, here the highest PSNR at low capacity is obtained with the block size 3×3 rather than other larger ones. When EC is larger than 16,000 bits, 3×2 becomes the best one on image

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

Image	Lena	Baboon	Airplane	Barbara	Boat	Elaine	Lake	Peppers
Run time	6.00	21.8	6.96	12.97	14.08	5.46	9.25	11.31

Lena whereas 2×3 becomes the best one on image Boat. This confirms the very little relationship between embedding performance and block size and the necessity of flexible partition.

For specific block size, the proposed scheme spends most computation time in determining the optimal thresholds (T_1^*, T_2^*). After calculating $E_H(t)$, $D_H(t)$, $E_L(t)$ and $D_L(t)$, all combinations of (T_1, T_2) are tested. The thresholds (T_1, T_2) which lead to the minimum expected distortion for the given EC are used to embed data. 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 3.0GB RAM. The proposed scheme spends the longest time on Baboon (21.8 s) and the shortest time on Elaine (5.46 s) due to their roughness and smoothness.

Fig. 9 shows the complete performance comparison by varying EC from 5000 bits to the maximum of the proposed scheme with a step of 1000 bits. It can be seen that the proposed scheme has larger maximum EC than previous PVO-based schemes including Peng et al.'s IPVO [29], Ou et al.'s PVO-k [20] and PVO-based pairwise PEE [22]. IPVO and PVO-k are undoubtedly not competitive in EC contribution since at most two bits can be embedded into each block. Compared with PVO-based pairwise PEE, the proposed scheme achieves slight improvement in maximum EC on all images except Baboon. Such improvement is reasonable since in almost all cases not only f_H outperforms f_1 but also f_L outperforms f_2. As shown in Fig. 10, at most 36,000, 35,000, 29,000 and 25,000 bits of data can be embedded respectively into 2×3 sized blocks with $NL \leq 150$ on image Lena. On image Boat, at most 24,000, 23,000, 19,000 and 16,000 bits of data can be embedded respectively into 2×3 sized blocks with $NL \leq 150$. For eight test images, the maximum EC of the proposed scheme is 42,000, 14,000, 52,000, 34,000, 29,000, 28,000, 29,000 and 36,000 bits respectively.

Fig. 9 also confirms that the proposed scheme achieves higher PSNR than IPVO, PVO-k and PVO-based pairwise PEE on all images whatever the EC is. Based on the consideration that they all embed payload block by block, their embedding performances all closely relate to the utilization of pixel blocks. While blocks with $p_{\sigma(n-1)} = p_{\sigma(n)}$ are abandoned in original PVO, IPVO [29] brings them into data embedding through prediction improvement. As discussed in Section 2.1, the improved prediction method which produces no longer always non-negative errors enables embedding bits into extra bin 0, at the cost of turning part of once expandable errors into shiftable. Compared with IPVO, the proposed scheme not only embeds data into more pixels within block but also modifies them by 2D histogram modification instead of 1D histogram modification. Experimental results verify that on all test images the proposed scheme improves IPVO respectively by 0.7 dB, 1 dB, 0.78 dB, 0.78 dB, 0.62 dB, 1.05 dB, 1.18 dB and 0.68 dB on average.

PVO-k [20] similarly optimizes the utilization of pixel blocks on the basis of original PVO. By predicting and modifying the two maximum-valued pixels as a unit, PVO-k successes to embed data into blocks with $p_{\sigma(n-1)} = p_{\sigma(n)}$. However, the idea of modifying multiple pixels for embedding one bits of data evidently corresponds to severer average distortion. As a result, it is not surprising that the performance of PVO-k is similar to that of IPVO. Experimental results verify that on all test images the proposed scheme improves PVO-k respectively by 0.63 dB, 0.76 dB, 0.67 dB, 0.47 dB, 0.62 dB, 1.09 dB, 0.75 dB and 0.5 dB on average.

Ou et al.'s PVO-based pairwise PEE [22] can be treated as an extension of PVO-k in two aspects: one taking the two largest/smallest pixels as independent data carriers, the other modifying them by higher dimensional histogram modification. Compared with PVO-k, the performance of PVO-based Pairwise PEE is far closer to that of the proposed scheme. Moreover, the comparison between PVO-based pairwise PEE and IPVO is of the same significance since part of blocks are actually processed by IPVO in [22]. The above comparisons verify the importance of optimized utilization of pixel blocks, as our research work do. In this paper we denote PVO-based pairwise PEE and IPVO in [22] as f_1 and f_2, then we improve Ou et al.'s work by replacing f_1 with f_H and replacing f_2 with f_L. Referring to Fig. 9, the proposed scheme achieves evident superiority over original PVO-based pairwise PEE, especially on images Baboon, Barbara, Boat and Elaine. Generally the average PSNR improvement is 0.14 dB, 0.19 dB, 0.16 dB, 0.2 dB, 0.2 dB, 0.24 dB, 0.16 dB and 0.13 dB respectively for all test images.

To further address the contributions of the above two replacements, three combinations including f_1 and f_2, f_1 and f_L, f_H and f_L are considered. As shown in Fig. 11, the performance of $f_1 + f_L$ is evidently superior than that of $f_1 + f_2$. f_2 can be treated as the lowest power version of PVO-based Pairwise PEE, where there are only Type-A and Type-E pairs whose average distortions are $\frac{1}{2}$ and 1. The extinction of other types of pairs corresponds to abandoning the advantage of modifying both two errors. By contrast, only Type-B pair is discarded and Type-C pair is redefined in f_L for the purpose of eliminating Type-F pair. Experimental results demonstrate that f_L is more suitable for matching f_1 than f_2. The advantage of f_L can also be measured by calculating the expected distortion. Referring to Table 3, the expected distortion is steadily reduced by 3.07%, 3.5%, 3.91%, 3.05%, 4.62%, 3.15%, 2.5% and 2.21% with the increase of EC by replacing f_2 with f_L on image Lena. On image Boat, the expected distortion is reduced by 4.38%, 4.36%, 3.46%, 4.49% and 4.16% respectively.

Fig. 11 and Table 3 also show that embedding performance can be further promoted by additionally replacing f_1 with f_H, especially when EC approaches its maximum. It is reasonable since lots of blocks with rather high complexity have to be employed if without the extra EC brought by f_H. Suppose the payload size is the maximum EC of $f_1 + f_2$, the PSNR of

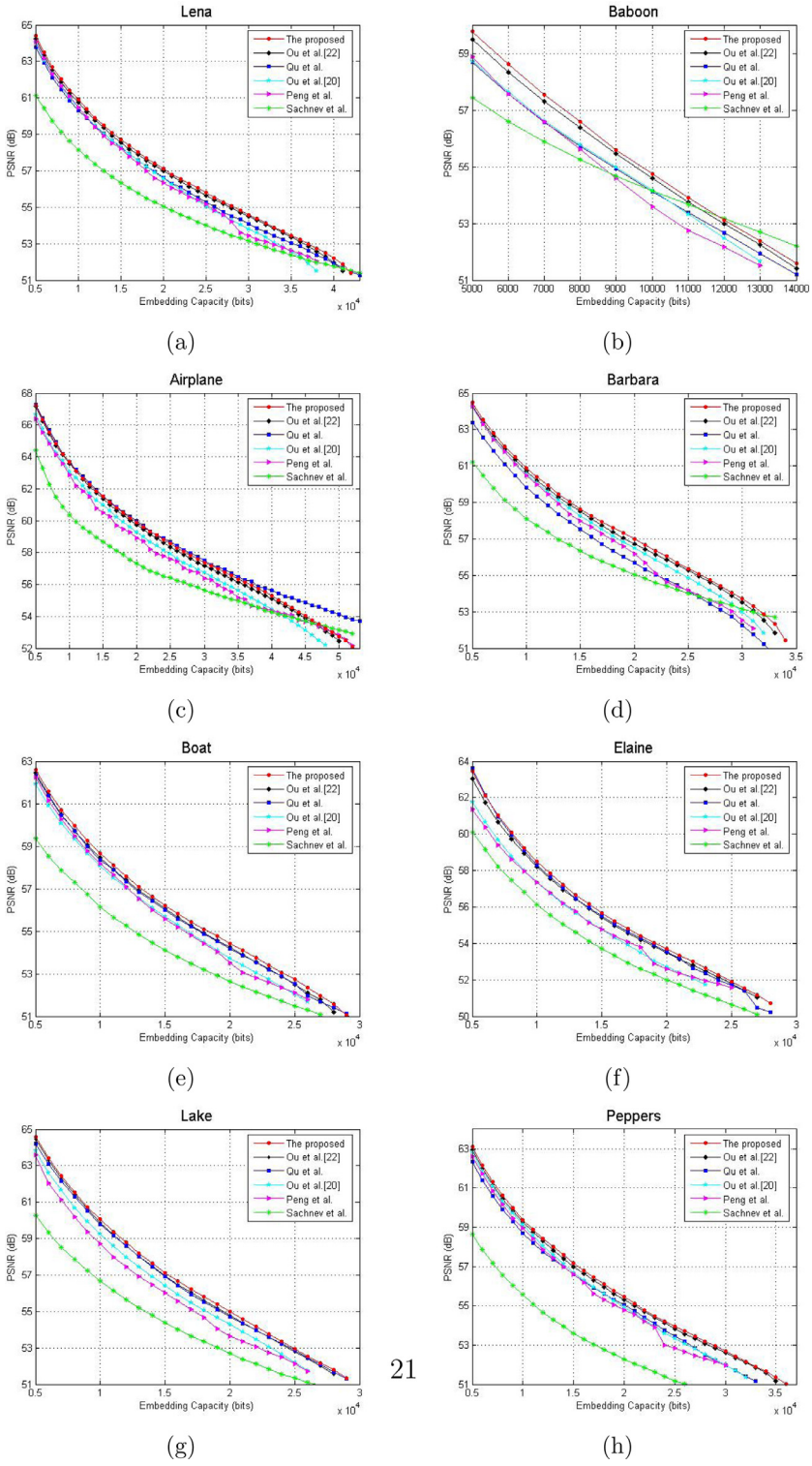

Fig. 9. Performance comparison between the proposed scheme and other related schemes.

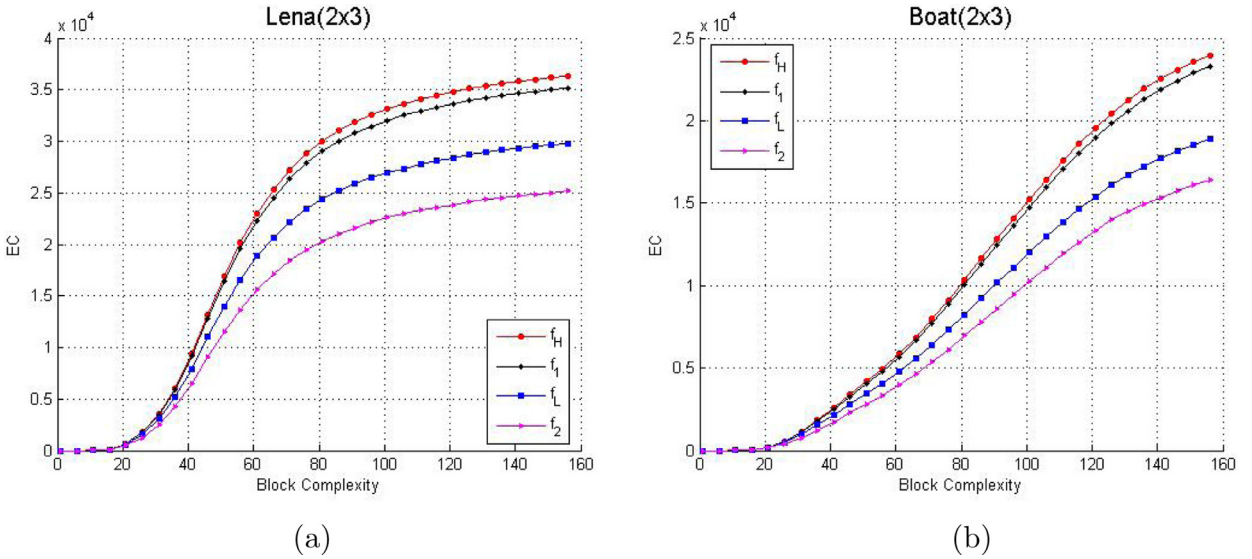

Fig. 10. Performance comparison on embedding capacity.

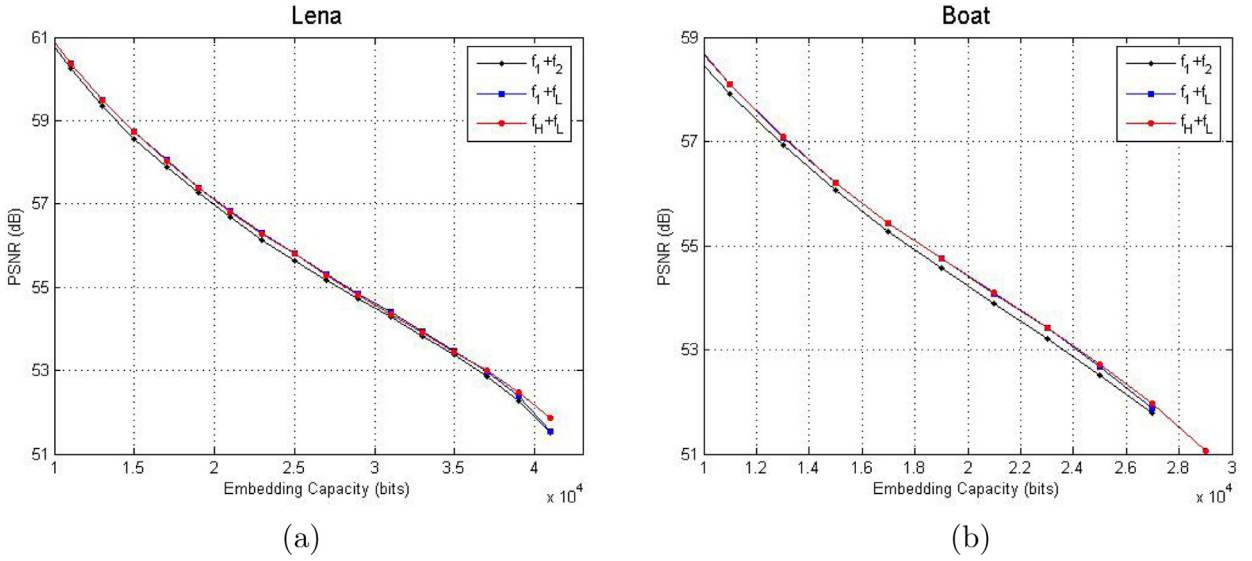

Fig. 11. Performance comparison of flexible combinations.

$f_1 + f_2$ is 51.5 dB whereas the one of $f_H + f_L$ is 51.86 dB on image Lena. On image Boat, $f_H + f_L$ further reduces the expected distortion, i.e., by 4.06%, 4.92%, 3.79%, 4.73% and 5.05% respectively. It should be noted that f_{HS} could be a candidate for taking place of f_1. The embedding performances on images Airplane and Elaine shown in Fig. 9 are actually obtained by $f_{HS} + f_L$. f_{HS} differs from f_H in the manipulation of bins (0,0) and (1,0). Obviously, the more bin (0,0) is higher than bin (1,0), the more likely f_{HS} is chosen.

Schemes of Sachnev et al. [32] and Qu et al. [30] differ from the proposed scheme in embedding payload pixel by pixel. Sachnev et al.'s scheme is a typical PEE-based scheme which predicts each pixel by its four nearest pixels. Moreover, pixels with small local variance are preferentially utilized for embedding. Referring to Fig. 9, our scheme significantly outperforms Sachnev et al.'s in most cases, especially when EC is small. However, the comparative advantage diminishes as EC increases. This can be attributed to the limited maximum EC problem of PVO-based scheme. Experimental results verify that on all test images the proposed scheme improves Sachnev et al.'s respectively by 1.81 dB, 0.81 dB, 1.73 dB, 1.84 dB, 2.04 dB, 1.99 dB, 2.71 dB and 3.29 dB on average.

Qu et al. similarly embedded payload into each pixel, except predicting each pixel by its sorted context. Referring to Fig. 9, the PVO evolving prediction method makes the performance of Qu et al.'s scheme far superior to that of Sachnev et al.'s. Compared with PVO-based schemes, Qu et al.'s scheme has the advantage of utilizing all pixels. This is of great

Table 3
Comparison of expected distortions for various EC.

Image	Scheme	EC							
		5000	10,000	15,000	20,000	25,000	30,000	35,000	40,000
Lena	$f_1 + f_2$	6348	14,283	23,430	33,810	46,311	60,022	77,972	108,236
	$f_1 + f_L$	6153	13,783	22,515	32,779	44,173	58,130	76,024	105,839
	$f_H + f_L$	6171	13,802	22,515	32,767	44,300	58,792	76,363	102,212
Boat	$f_1 + f_2$	9637	24,165	42,056	64,011	95,362	–	–	–
	$f_1 + f_L$	9215	23,111	40,600	61,134	91,397	–	–	–
	$f_H + f_L$	9246	22,975	40,464	60,985	90,544	–	–	–

help to embed data into smooth image. For example, Fig. 9 shows that on image Airplane Qu et al.'s scheme evidently outperforms other schemes including the proposed scheme, especially when EC approaches its maximum. However, on other images, the proposed scheme improves Qu et al.'s respectively by 0.46 dB, 0.71 dB, 1.24 dB, 0.23 dB, 0.2 dB, 0.21 dB and 0.61 dB on average.

5. Conclusion

Histogram shifting has been incorporated into PEE and a sharp histogram can be guaranteed by advanced prediction method. In addition, efficient histogram manipulation such as 2D histogram modification also means a lot to embedding performance. In this paper, we focus on the efficiency in manipulating 2D-PEH and present a new RDH scheme based on optimized PVO-based pairwise PEE. In original PVO-based pairwise PEE, the largest/smallest two pixels which share one predicted value are always modified as a unit. In this paper, they are flexibly predicted and modified and we call this method as multi-pass PVO prediction. Multi-pass PVO enables us to design the high and low power version of PVO-based pairwise PEE. Experimental results verify that the combination of the above two versions of PVO-based pairwise PEE can better utilize the characteristics of 2D-PEH and thus achieve improved performance. The proposed scheme also outperforms several state-of-the-art schemes.

Acknowledgments

The authors would like to sincerely thank the editors and anonymous reviewers for their valuable comments. This work is supported by Medical Scientific Research Fund of Guangdong Province, China (No. A2018186), Science and Technology Development Fund of Macau under Grant 048/2016/A2 and Grant 0012/2018/A1, National NSF of China (No. 61571139, No. 61201393), New Star of Pearl River on Science and Technology of Guangzhou (No. 2014J2200085).

References

[1] A.M. Alattar, Reversible watermark using the difference expansion of a generalized integer transform, IEEE Trans. Image Process. 13 (8) (2004) 1147–1156.
[2] M.U. Celik, G. Sharma, A.M. Tekalp, E. Saber, Lossless generalized-LSB data embedding, IEEE Trans. Image Process. 14 (2) (2005) 253–266.
[3] D. Coltuc, Improved embedding for prediction-based reversible watermarking, IEEE Trans. Inf. Forensics Secur. 6 (3) (2011) 873–882.
[4] D. Coltuc, J.M. Chassery, Very fast watermarking by reversible contrast mapping, IEEE Signal Process. Lett. 14 (4) (2007) 255–258.
[5] D. Coltuc, Low distortion transform for reversible watermarking, IEEE Trans. Image Process. 21 (1) (2012) 412–417.
[6] I.C. Dragoi, D. Coltuc, Local-prediction-based difference expansion reversible watermarking, IEEE Trans. Image Process. 23 (4) (2014) 1779–1790.
[7] I.C. Dragoi, D. Coltuc, Adaptive pairing reversible watermarking, IEEE Trans. Image Process. 25 (5) (2016) 2420–2422.
[8] J. Fridrich, M. Goljan, R. Du, Lossless data embedding-new paradigm in digital watermarking, EURASIP J. Appl. Signal Process. 2002 (2) (2002) 185–196.
[9] 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.
[10] W.G. He, J. Cai, K. Zhou, G.Q. Xiong, Efficient PVO-based reversible data hiding using multistage blocking and prediction accuracy matrix, J. Vis. Commun. Image R 46 (2017) 58–69.
[11] W. Hong, Adaptive reversible data hiding method based on error energy control and histogram shifting, Opt. Commun. 285 (2) (2012) 101–108.
[12] 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.
[13] A. Khan, A. Siddiqa, S. Munib, S.A. Malik, A recent survey of reversible watermarking techniques, Inf. Sci. 279 (20) (2014) 251–272.
[14] S.K. Lee, Y.H. Suh, Y.S. Ho, Reversible image authentication based on watermarking, in: Proc. IEEE ICME, 2006, pp. 1321–1324.
[15] 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.
[16] 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.
[17] Z. Ni, Y.Q. Shi, N. Ansari, W. Su, Reversible data hiding, IEEE Trans. Circuits Syst. Video Technol. 16 (3) (2006) 354–362.
[18] B. Ou, X. Li, Y. Zhao, R. Ni, Reversible data hiding based on PDE predictor, J. Syst. Softw. 86 (10) (2013) 2700–2709.
[19] B. Ou, X. Li, Y. Zhao, R. Ni, Y.Q. Shi, Pairwise prediction-error expansion for efficient reversible data hiding, IEEE Trans. Image Process. 22 (12) (2013) 5010–5021.
[20] 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.
[21] 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.
[22] 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.

[23] 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.
[24] 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.
[25] 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.
[26] X. Li, J. Li, B. Li, B. Yang, High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion, Signal Process. 93 (1) (2013) 198–205.
[27] Z.B. Pan, S. Hu, X.X. Ma, W. L. F, Reversible data hiding based on local histogram shifting with multilayer embedding, J. Vis. Commun. Image R 31 (2015) 64–74.
[28] Q. Pei, X. Wang, Y. Li, H. Li, Adaptive reversible watermarking with improved embedding capacity, J. Syst. Softw. 86 (11) (2013) 2841–2848.
[29] F. Peng, X. Li, B. Yang, Improved pvo-based reversible data hiding, Digit. Signal Process. 25 (2014) 255–265.
[30] X. Qu, H.J. Kim, Pixel-based pixel value ordering predictor for high-fidelity reversible data hiding, Signal Process. 111 (2015) 249–260.
[31] R.M. Rad, K. Wong, J.M. Guo, A unified data embedding and scrambling method, IEEE Trans. Image Process. 23 (4) (2014) 1463–1475.
[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] D.M. Thodi, J.J. Rodriguez, Expansion embedding techniques for reversible watermarking, IEEE Trans. Image Process. 16 (3) (2007) 721–730.
[34] J. Tian, Reversible data embedding using a difference expansion, IEEE Trans. Circuits Syst. Video Technol. 13 (8) (2003) 890–896.
[35] 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.
[36] 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.
[37] X. Wang, X. Li, B. Yang, Z. Guo, Efficient generalized integer transform for reversible watermarking, IEEE Signal Process. Lett. 17 (6) (2010) 567–570.
[38] 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.
[39] 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.
[40] H.T. Wu, J. Huang, Reversible image watermarking on prediction errors by efficient histogram modification, Signal Process. 92 (12) (2012) 3000–3009.