Signal Processing 198 (2022) 108577

Contents lists available at ScienceDirect

Signal Processing

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

Reversible data hiding based on adaptive IPVO and two-segment pairwise PEE☆

Ningxiong Mao[a], Fan Chen[b], Hongjie He[a,*], Yaolin Yang[a]

[a] Key Lab of Signal and Information Processing, Southwest Jiaotong University, Chengdu, Sichuan 611756, China
[b] School of Computing and Artificial Intelligence, Southwest Jiaotong University, Chengdu 611756, China

ARTICLE INFO

Article history:
Received 10 January 2022
Revised 20 March 2022
Accepted 31 March 2022
Available online 9 April 2022

Keywords:
Reversible data hiding
Adaptive IPVO
Effective prediction error rate(EPER)
Pairwise PEE

ABSTRACT

Improved pixel value order (IPVO) and pairwise prediction error extension (PEE) strategies can effectively improve the hidden capacity and invisibility of reversible data hiding (RDH) scheme since they can increase the effective prediction error rate (EPER). To further improve EPER, this paper proposes an improved RDH scheme by combining adaptive IPVO (AIPVO) with two-segment pairwise PEE. Different from the fixed predicted value (such as the highest value) for all blocks in the existing IPVO, AIPVO adaptively determine the predicted value of each block according to its own complexity level. To obtain the complexity level of each block, the block complexity level division model is constructed and optimized with EPER as the objective function. AIPVO can effectively improve EPER due to the fact that the pixels with large fluctuate in a block are eliminated by adaptively selecting the predicted value. In the embedding stage, the two-segment pairwise PEE is designed according to the distribution characteristics of the prediction error sequence obtained by AIPVO. Specifically, the prediction error sequence is firstly divided into two segments, and the 2D mappings of the front and back parts use small-magnitude and slightly large-magnitude prediction errors to embed data, respectively. Experimental results show that this paper scheme is better than the latest research results. For example, PSNR of the Lena image reaches 61.55 dB under capacity of 10,000 bit, and the gain is 0.39 dB against the best result in the literature.

1. Introduction

The purpose of reversible data hiding (RDH) is to embed secret data into the host image in an imperceptible way by slightly modifying the pixels of the host image. The key to RDH is to extract secret data without distortion and reconstruct the original host image without distortion at the decoding end [1–3]. Such RDH can meet certain areas where slight distortion of host media is unacceptable, such as military, medical, remote sensing image secret communication and content integrity authentication, etc.

So far, there are many reversible data hiding algorithms, which can be mainly divided into three types. (1). RDH based on lossless compression (LS) [4–8]. The image is compressed to make room for embedding data, but LS algorithms usually provide lower capacity and may cause serious image quality degradation. (2). RDH based on difference expansion (DE) [9–12]. Originally proposed by Tian et al. [9], the least significant bit is vacated to carry data by expanding the difference between adjacent pixels. (3). RDH based on histogram shift (HS) [13–18]. Proposed by Ni et al. [13], the secret data is carried by the peak bin of the histogram, in order to avoid confusing the operation of shifting other bins. In [19], Thodi and Rodriguez used prediction error histogram (PEH) instead of pixel histogram to embed data based on prediction error expansion (PEE). Because the histogram generated by the prediction error is sharp, the expansion of the prediction error can increase the embedding capacity and reduce the distortion. Most of the current RDH algorithms are improved on the basis of PEE. PEE can be divided into two stages: (1). Predict the pixels to obtain accurate pixel prediction values, thereby obtaining a sharp PEH. (2). Design effective rules to expand the PEH to embed data. Because the prediction method is related to the sharp of the PEH, so the pixel value prediction method is very important for PEE. The expansion scheme of the PEH will also affect the capacity and performance of the PEE. Pixel value prediction method and histogram expansion method are the main research directions based on PEE RDH.

On the basis of PEE, in order to make the predicted value more accurate and obtain more small-magnitude prediction errors, many advanced prediction methods are proposed to generate sharp PEH. Such as rhombus predictor [14], multi-predictor [20–21], interpolation-based prediction [22], predicted value is ob-

☆ This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61872303 and U1936113. Corresponding author: Hongjie He. Email address: hjhe@home.swjtu.edu.cn.

* Corresponding author.
E-mail address: hjhe@home.swjtu.edu.cn (H. He).

https://doi.org/10.1016/j.sigpro.2022.108577
0165-1684/

tained by calculating the weights of four adjacent pixels [23], skewed predictor [24], median edge predictor [25] and pixel-based pixel value order prediction (PPVO) [26], and so on. The above prediction method is to obtain accurate prediction values to generate a sharp PEH. On the other hand, it is the research on the expansion scheme of PEH, which can be adapted to select expand bin according to the different embedding capacity to minimize embedding distortion [27–28]. Sort the pixels according to the local complexity, so that the pixels of smooth area are as front as possible, and embed data in the smooth area to reduce invalid shift [29–30]. Use multiple PEH for data embedding, and make full use of different texture regions of the image [31–32]. Recently, pairwise PEE expansion scheme has been proposed [33–35]. Pairwise PEE can further utilize the correlation between prediction errors to effectively improve RDH performance. The prediction error pairing method in pairwise PEE is crucial.

The pixel value order (PVO) prediction technology, which has been widely studied recently, it is an effective RDH method. Li et al. first proposed PVO prediction technology in [36], which divides the image into non-overlapping blocks. The largest/smallest pixel in the block is predicted by the second largest/smallest pixel, and then the largest/smallest pixel is modified for embedding data. Since PVO uses the correlation of neighboring pixel values, so the prediction accuracy of PVO is excellent. In [37], the prediction error is obtained according to the relative spatial position of the pixel and the predicted value, so that the prediction error 0 can be expanded in the negative direction so that the data can be embedded in bin 0 to increase the embedding capacity, also called Improved PVO (IPVO). In [38], a method called PVO-k was proposed to embed data by modifying the k largest/smallest pixels in the block, which can significantly increase capacity. In [39], a dynamic divide block strategy was proposed to adaptive determine block size, using smaller blocks in image smooth area. The performance of the combination of PVO and pairwise PEE is explored in [40–43]. He and Cai [43] proposed a Multi-Pass IPVO Based PEE, which no longer predicts the largest/smallest pixels and embeds data by modifying the largest/smallest pixels. In [43], the largest/smallest pixels are used to predict the non- largest/smallest pixels, the data is embedded by modifying the non- largest/smallest pixels, and two rounds of data embedding are carried out on the same pixel block, and the pixels are modified by reduction/enlargement. In the process of reduction/enlargement the same pixel block, distortion offset occurs to reduce the distortion, and the capacity can be significantly improved. In [43], it also proposed to estimate the distribution of pixel values, to define the spatial position adaptively, and to estimate the distribution of pixel values using the full enveloping context, so as to determine the optimal scanning order in the pixel block.

As can be observed from the above, the existing IPVO schemes mostly use three prediction algorithms. (1) Using the second largest/smallest as the predicted value to predict the largest/smallest pixel is classic. (2) Utilize the median pixel as the predicted value to predict other non-median pixels. (3). Use the largest/smallest value pixel to predict other non- largest/smallest pixels. The prediction values in the above IPVO prediction methods are all selected in a fixed mode, which has a high prediction accuracy for the smooth block. For blocks with large fluctuations in pixel values within the block, large errors will occur. Aiming at this shortcoming, this paper proposed an AIPVO prediction method. In AIPVO, the predicted value is no longer a fixed pattern selection, but is chosen based on the complexity of the block. In this way, the pixel values with large fluctuations in the block can be eliminated as much as possible according to the complexity level of the image block, thereby improving the correlation between the pixel values. How to determine the complexity level of the image block is crucial. We define an EPER for the evaluation index of the prediction error sequence, denoted as R_e. Utilize R_e as the objective function to establish a model for solving the complexity level of the image block. This paper uses pairwise PEE to embed data. We observe that the sorted prediction error sequence is that the front part is mostly small-magnitude prediction errors, and the back part is mostly large-magnitude prediction errors. Therefore, this paper designs two 2D mapping according to different distributions of front and back part of prediction error sequence, respectively. The main contributions of this paper are as follows:

(1) *Novel prediction method.* A new prediction method AIPVO is proposed, which can adaptively determine predicted values according to the complexity level of block and adapt prediction error sequence according to given capacity.
(2) *New evaluation index.* The EPER R_e is defined as an evaluation index of the prediction error sequence, and the complexity level division model of image blocks is established with R_e as the objective function.
(3) *Improve expansion rules.* Designing different 2D mapping rules according to different distributions of prediction error of front and back parts of sorted prediction error sequence can more effective utilize of correlation between prediction errors.

The rest of this paper is organized as follows. Some research progress based on IPVO are briefly introduces in Section 2. Section 3 introduces the proposed scheme in detail. The experimental results as well as the comparison with existing advanced technologies are shown in Section 4. Finally, Section 5 concludes this paper.

2. Related works

This section briefly reviews some research results based on PVO, including improved PVO (IPVO) [37], PVO-based pairwise PEE [40] and Multi-Pass IPVO [43]. Because the process of modifying embedded data for largest pixel and smallest pixel is similar, the following takes largest pixel to modify embedded data as example.

2.1. IPVO [37]

IPVO is an improvement on the initial PVO. Using the spatial position relationship of the largest and second largest pixels, IPVO can embed data in bin 0 and effectively increase capacity. In IPVO, the host image is first divided into non-overlapping pixel blocks, pixels inside the block are $p_1, p_2, \cdots, p_n$, n is the total number of pixels in block, and the pixels in block are sorted in ascending order to get $\{p_{\sigma(1)}, p_{\sigma(2)}, \cdots, p_{\sigma(n)}\}$ and $\{p_{\sigma(1)} \le p_{\sigma(2)} \le \cdots \le p_{\sigma(n)}\}$ where $\sigma:\{1, \cdots, n\} \to \{\sigma(1), \cdots, \sigma(n)\}$ is unique one-to-one mapping. The prediction error in IPVO can be calculated as

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

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

IPVO determines the prediction error based on relative spatial positions of largest and second largest pixels, so that bin 0 can be expanded in negative direction, and bin 0 can also embed data instead of only bin 1. According to the embedding rules of IPVO, the obtained mark prediction error e'_{max} is as follows

$$e'_{max} = \begin{cases} e_{max} - 1 \; if \; e_{max} < 0 \\ e_{max} - b \; if \; e_{max} = 0 \\ e_{max} + 1 \; if \; e_{max} > 1 \\ e_{max} + b \; if \; e_{max} = 1 \end{cases} \tag{3}$$

where b is secret data.

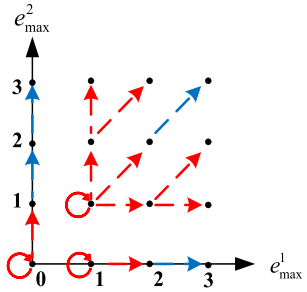

Fig. 1. Reversible 2D mapping in [40].

2.2. PVO-based pairwise PEE [40]

In [40] Ou et al. proposed pairwise PEE based on PVO, which changed from single prediction error expansion to pairwise prediction error expansion. Use $p_{\sigma(n-2)}$ to predict $(p_{\sigma(n)}, p_{\sigma(n-1)})$ to get the prediction error pair (e^1_{max}, e^2_{max}), the specific formula is as follows:

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

Where (u, v) is defined in (2), and two-dimensional prediction error histogram (2D-PEH) can be generated by obtaining (e^1_{max}, e^2_{max}). Then reversible 2D-mapping is carefully designed to modify prediction error pair and embed data. Reversible 2D-mapping is shown in Fig. 1. In Fig. 1, the red arrow represents expansion direction of 2D bin and can embedded data, and in order to avoid confusion, the blue arrow represents the shift direction of the 2D bin and cannot embedded data. The reversible 2D-mapping here is designed based on experience.

2.3. Multi-pass IPVO based PEE [43]

In He and Cai [43], proposed Multi-Pass IPVO through observation, as long as relative spatial position of predicted pixel and predicted value remains unchanged, reversibility can be guaranteed. In this method, using largest pixel in block to predict other non-largest pixels, and prediction error $e_{max}(i)$ can be calculated as follows

$$e_{max}(i) = p_u - p_v \tag{5}$$

$$\begin{cases} u = min\ (\sigma(n), \sigma(i)) \\ v = max\ (\sigma(n), \sigma(i)) \end{cases} i \in \{1, 2, \cdots, n-1\} \tag{6}$$

In Multi-Pass IPVO method, it does not embed data by modifying largest pixels, but by reduce non-largest pixel to embedded data. According to formula (3), marker prediction error $e'_{max}(i)$ can be obtained, and the pixel p'_i can be obtained by formula (7)

$$p'_i = \begin{cases} p_i - b\ if\ e_{max}(i) = 0\ or\ e_{max} = 1 \\ p_i - 1\ if\ e_{max}(i) \langle 0\ or\ e_{max} \rangle 1 \end{cases} \tag{7}$$

Next, we reorder the marked pixels in ascending order to obtain a new pixel sequence $\{p_{\sigma(1)}, p_{\sigma(2)}, \cdots, p_{\sigma(n)}\}$. Then use the smallest pixel in the block to predict other non-smallest pixels, the prediction error $e_{min}(i)$ can be calculated by the following formula

$$e_{min}(i) = p_s - p_t \tag{8}$$

$$\begin{cases} s = min(\sigma(1), \sigma(i)) \\ t = max(\sigma(1), \sigma(i)) \end{cases} i \in \{2, 3, \cdots, n\} \tag{9}$$

Similarly, the marker prediction error $e'_{min}(i)$ can be obtained according to formula (3), and the non-smallest pixels can be enlarged to embed data, therefore marker pixel p'_i as shown in formula (10)

$$p'_i = \begin{cases} p_i + b\ if\ e_{min}(i) = 0\ or\ e_{min} = 1 \\ p_i + 1\ if\ e_{min}(i) \langle 0\ or\ e_{min} \rangle 1 \end{cases} \tag{10}$$

At the decoding end, secret data can be extracted and original pixels can be restored according to mark prediction error. First, extract secret data according to $e'_{min}(i)$ and calculation is as follows

$$b = \begin{cases} e'_{min}(i) - 1\ if\ e'_{min}(i) \in \{1, 2\} \\ -e'_{min}(i)\ if\ e'_{min}(i) \in \{-1, 0\} \end{cases} \tag{11}$$

After extracting secret data b, original pixel value can be restored without distortion.

$$p_i = \begin{cases} p'_i - b\ if\ e'_{min}(i) \in \{-1, 0, 1, 2\} \\ p'_i - 1\ if\ e'_{min}(i) \left\langle -1\ or\ e'_{min}(i) \right\rangle 2 \end{cases} \tag{12}$$

Reorder pixels and calculate marker prediction error $e'_{max}(i)$ according to formulas (5) and (6). According to formula (11), secret data can be extracted, and restored pixels calculated are as follows

$$p_i = \begin{cases} p'_i + b\ if\ e'_{max}(i) \in \{-1, 0, 1, 2\} \\ p'_i + 1\ if\ e'_{max}(i) \langle -1\ or\ e'_{max}(i) \rangle 2 \end{cases} \tag{13}$$

Using the Multi-Pass IPVO method, instead of embedding data at largest/smallest pixels, we embedding data at non-largest/smallest pixels, so the capacity can be significantly increased. And in the process of embedding the data, the first round is to reduce the non-largest pixels, and the second round is to enlarge the non-smallest pixels. This process will produce effect of distortion offset.

3. Proposed scheme

In this section, aiming to overcome the shortcomings of existing IPVO prediction methods, AIPVO prediction method is proposed, then combination of AIPVO prediction method and improved pairwise PEE is described. Firstly, adaptive IPVO prediction method and the establishment and solution of the block complexity level division model is introduced. Second, we describe the improvement of pairwise PEE. Finally, implementation details of the scheme are given.

3.1. Adaptive IPVO prediction method

In [43], the largest/smallest pixels are used to predict other non-largest/smallest pixels, this is effective for smooth blocks. But the error for texture blocks is relatively large. Because the pixel value in the texture block fluctuates greatly, the correlation of the pixel value in the block is low, and the obtained prediction value error is relatively large. In order to overcome this shortcoming, this paper proposes a AIPVO prediction method. Instead of using fixed largest/smallest pixels to predict other non-largest/smallest pixels, the prediction value is determined according to the complexity value of the block. Here we use ascending order to sort the pixels in the block to get $\{p_{\sigma(1)} \le p_{\sigma(2)} \le \cdots \le p_{\sigma(n)}\}$, for a block of size $a \times b$, we will divide into $\{1, 2, \cdots, n-1\}$ $(n = a * b)$ total $n-1$ levels according to block complexity value. If the complexity value of the block belongs to the r-th $(r \in \{1, 2, \cdots, n-1\})$ level, sorted the $(n-r+1)$-th pixel is used as the predicted value, so the prediction error for the block whose complexity level r is calculation by the formula (14).

$$e_{max}(i) = p_u - p_v \tag{14}$$

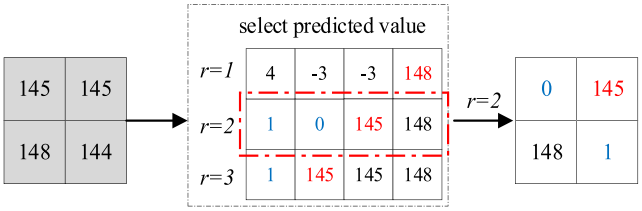

Fig. 2. The selected predicted value and the obtained prediction error of the image block under different complexity levels.

$$\begin{cases} u = min(\sigma(n-r+1), \sigma(i)) \\ v = max(\sigma(n-r+1), \sigma(i)) \end{cases} i \in \{1, 2, \cdots, n-r\} \tag{15}$$

Where r represents the complexity level of the block.

As shown in Fig. 2, we take a 2×2 ($n = 4$) size block as an example Fig. 2 shows the prediction error obtained when r is a different value. When $r = 1$ is the same as the prediction method in [43]. The prediction error $\{-3, -3, 4\}$ can be obtained by using the largest pixel as the predicted value, which not only cannot provide capacity but will produce the invalid shift. When $r = 2$, we take the second-largest pixel as the predicted value, and the prediction error is $\{1, 0\}$, providing 2 bit capacity. When $r = 3$, the prediction error is $\{1\}$, providing 1bit capacity. As can be seen, using the second-largest pixel as the prediction value is the best selected. How to determine the complexity level r of each block is the key problem we need to solve next.

3.2. Block complexity level model

Next, we need to determine the complexity level of the block according to the complexity value of the block. The smaller the block complexity value, the smaller the corresponding r value. Determine the complexity level of each block by dividing the block complexity value. For a block of size $a \times b$, there are at most $n-2$ complexity thresholds denoted as $\{T_1, T_2, \cdots, T_{(n-2)}\}$, where $\{C_1 \le T_1 \le T_2 \le \cdots \le T_{(n-2)} \le C_m\}$, C_1,C_m are the largest and smallest of all block complexity values. According to the complexity threshold, we can divide the complexity value of the block into $n-1$ intervals $\{[C_1, T_1), [T_1, T_2), \cdots, [T_{(n-2)}, C_m]\}$, and the r value corresponding to each interval is $\{r = 1, r = 2, \cdots, r = n-1\}$, respectively. If the block complexity value belongs to $[C_1, T_1)$, then $r = 1$ and the predicted value is $p_{\sigma(n)}$. If the block complexity value belongs to $[T_1, T_2)$, then $r = 2$ and the predicted value is $p_{\sigma(n-1)}$, in the same way, the block complexity value belongs to $[T_{(n-2)}, C_m]$, then $r = n-1$ and the predicted value is $p_{\sigma(2)}$. How to find the optimal complexity threshold $\{T_1, T_2, \cdots, T_{(n-2)}\}$ to optimize the prediction error sequence is a problem that needs to be solved.

The block complexity division model is an optimization problem. First, we need to determine an optimization objective value. The model optimizes the prediction error sequence. Performance of the prediction error sequence can be evaluated by measuring the prediction error rate R_e. The goal of the prediction method is to pursue high R_e. The prediction error sequence is used for data embedding in the pairwise prediction error extension (PEE). Suppose that an effective prediction error is determined to be $\{-3, -2, -1, 0, 1, 2\}$. Therefore, the calculation method of R_e in this paper is shown in formula (16).

$$R_e = \frac{count(\{-3 \le E \le 2\})}{count(E)} \tag{16}$$

$count(E)$ denotes the total number of prediction errors and $count(\{-3 \le E \le 2\})$ denotes the total number of effective prediction errors. E is the prediction error sequence and can be expressed as $E = \{e([C_1, T_1)), e([T_1, T_2)), \cdots, e([T_{(n-2)}, C_m])\}$ Each element $e([C_1, T_1))$ represents the prediction error obtained by performing the first-level ($r = 1$) prediction and $e([T_{(n-2)}, C_{max}])$ represents the prediction error obtained by performing the $(n-1)$-level ($r = n-1$) prediction.

The goal is to make R_e as large as possible while satisfying the capacity. The block complexity level division model can be formulated as

$$\begin{cases} \left(T_1^*, T_2^*, \cdots, T_{n-2}^*\right) = arg_{(T_1, T_2, \cdots, T_{(n-2)})} \max R_e \\ subject\ to\ count(\{-3 \le E \le 2\}) \ge EC \end{cases} \tag{17}$$

In (17), $R_e(T_1, T_2, \cdots, T_{(n-2)})$ is the effective prediction error rate. $\{T_1^*, T_2^*, \cdots, T_{n-2}^*\}$ is the optimal complexity threshold set. EC is the number of secret data to be embedded.

3.3. Solving block complexity level model

The following is how to solve the model (17) to obtain the optimal complexity threshold $\{T_1^*, T_2^*, \cdots, T_{n-2}^*\}$. We tried many solution methods such as genetic algorithm in intelligent optimization algorithm, PSO, dynamic programming, etc., but the results were not satisfactory. Finally, we use the local threshold segmentation method to solve the model. We first calculate the complexity value of all shadow blocks and arrange the blocks with ascending order of complexity value. After sorted, a set of complexity values can be obtained, denoted as $C = \{C_1, C_2, \cdots, C_j, \cdots, C_M\}$ and $\{C_1 < C_2 < \cdots < C_M\}$, and one complexity value may correspond to multiple blocks. Blocks with the same complexity value have the same complexity level (the same value of r). We denote the EPER corresponding to each complexity value as R_C and $R_C = \{R_C(1), R_C(2), \cdots, R_C(j), \cdots, R_C(M)\}$. We define a skip EPER threshold set denoted as R_S and $R_S = \{0,\ 0.05,\ 0.1, 0.15, \cdots, R_S(i), \cdots, 0.95\}$. We start with the smallest complexity value C_1 to calculate the EPER $R_C(1)$. If $R_C(1)$ of the complexity value C_1 is greater than or equal to $R_S(i)$, the complexity level of the complexity value C_2 continues to be 1 ($r = 1$). Otherwise, the complexity level of the complexity value C_2 is increased by one to $r = 2$, and the first complexity threshold is denoted as $T_1 = C_1$. The following complexity thresholds $\{T_2, \cdots, T_{n-2}\}$ can be obtained in a similar manner. Details are shown in the flowchart of Fig. 3. K denotes the total number of elements in the set R_S and M that in the set C.

The optimal complexity threshold set $\{T_1^*, T_2^*, \cdots, T_{n-2}^*\}$ can be obtained through the following steps 1 to 5(In the revised manuscript), and the optimal prediction error sequence can be determined after $\{T_1^*, T_2^*, \cdots, T_{n-2}^*\}$ is obtained through formula (14), (15).

Step 1. According to the block complexity value formula (20), the complexity values of all shadow blocks can be obtained. After the complexity values are sorted in ascending order, the corresponding block sorting can be obtained, and the sorted complexity value set C can be obtained.

Step 2. Starting from $C_1$with the smallest complexity level ($r = 1$), the prediction error corresponding to the complexity value C_1uses the largest pixel value as the prediction value to calculate the prediction error, and blocks with the same complexity value select the prediction value in the same pattern. According to formulas (14)(15), the prediction errors of blocks with the same complexity value are calculated and $R_C(1)$ is obtained.

Step 3. Compare $R_C(1)$ with the given skip threshold $R_S(i)$, if $R_C(1)$ is greater than or equal to $R_S(i)$, return to Step 2 to calculate the $R_C(2)$ of the next complexity value. Otherwise, increase the complexity level r by increase one level ($r = r + 1$) and then return to Step 2 to calculate the $R_C(1)$ of the complexity value C_1, and record the complexity value and save it to the $i-th$ solution θ_i.

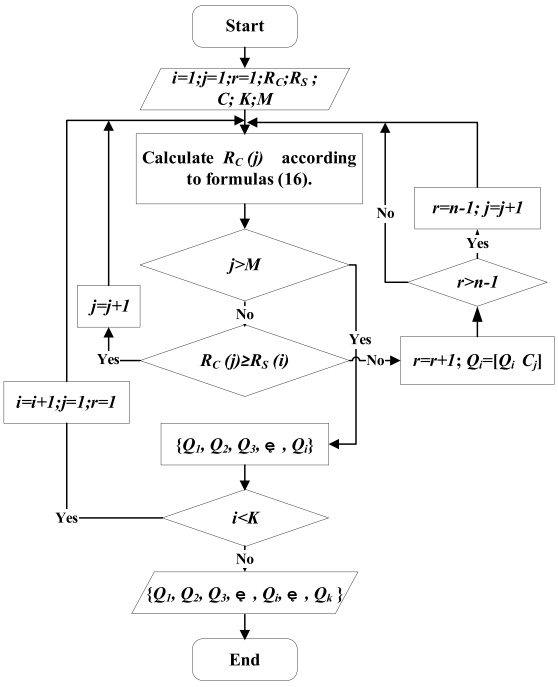

Fig. 3. Solving block complexity level flowchart.

Step 4. When the block corresponding to the largest complexity value C_m is calculated, a solution θ_i of the block level division corresponding to $R_S(i)$ can be obtained. θ_i contains multiple complexity values, but it is less than or equal to $n-2$. The skip threshold $R_S(i)$ will be updated to $R_S(i+1)$and then return to Step 2 to calculate the solution of the $(i+1)-th$ block complexity level division, and finally a complexity threshold solution set $\{\theta_1, \cdots, \theta_i, \cdots, \theta_k\}$ is obtain (k is The total number of elements in R_S).

Step 5. Select a solution with the highest EPER R_e as the optimal solution $\theta^* = \{T_1^*, T_2^*, \cdots, T_{n-2}^*\}$ of the block complexity level division model in the solution set $\{\theta_1, \cdots, \theta_i, \cdots, \theta_k\}$.

3.4. Second round of embedding

As above, we can obtain optimal complexity threshold θ^* to divide blocks into complexity levels, and obtain optimal prediction error sequence e_{max} through AIPVO. After optimal prediction error sequence is obtained, the segment adaptive pairwise PEE of this paper is used for data embedding, pixels are modified according to formula (7). After the prediction error sequence e_{max} data embedding, this paper will also adopt a two-round embedding strategy of Multi-Pass IPVO in paper [43]. Reorder the pixels to obtain a new pixel sequence $\{p_{\sigma(1)} \le p_{\sigma(2)} \le \cdots \le p_{\sigma(n)}\}$ and calculate prediction error $e_{min}(i)$, where the complexity threshold is same as the θ^* of first round embedding. For the block of rth ($r \in \{1, 2, \cdots, n-1\}$) complexity level, the prediction error is calculated as follows.

$$e_{min}(i) = p_s - p_t \tag{18}$$

$$\begin{cases} s = min(\sigma(r), \sigma(i)) \\ t = max(\sigma(r), \sigma(i)) \end{cases} i \in \{r+1, r+2, \cdots, n\} \tag{19}$$

Then we embed data to pixel according to prediction error $e_{min}(i)$, but this modification is to enlarge pixel and modify it as in formula (10). Similar to the paper [43], two-round embedding strategy is adopted, so the embedding process will produce a distortion offset effect. Let's take an example of a 2 × 2 block size.

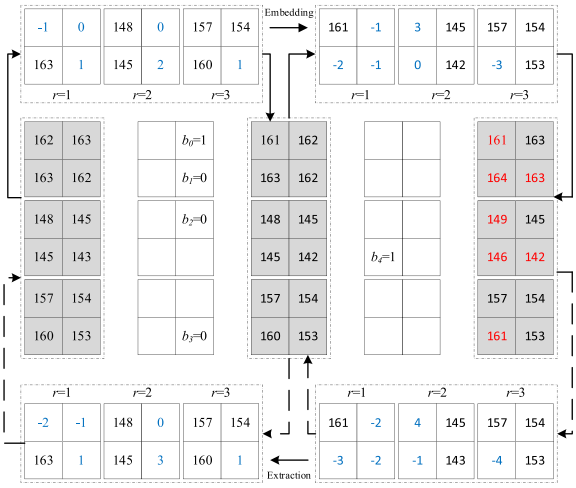

Fig. 4. AIPVO process of three different complexity level blocks.

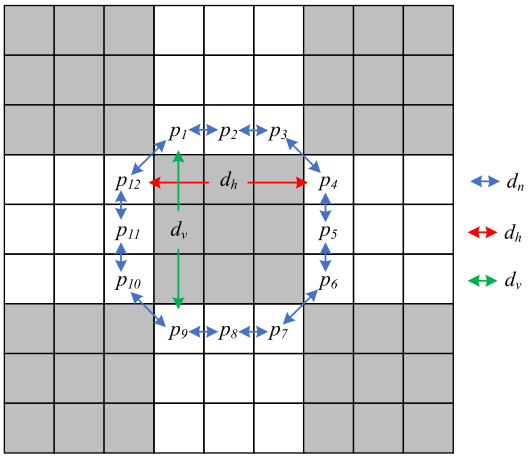

Fig. 5. Context for calculating of block complexity value.

Fig. 4 shows the AIPVO process of three different complexity levels of blocks, with the traditional 0, 1 as the embeddable prediction error. The embedded capacity of AIPVO is 5 bits, the capacity of Multi-pass IPVO [43] is 3 bits, and the capacity of traditional IPVO is 3 bits, as shown in Fig. 4. The AIPVO method in this paper has obvious advantages.

For calculation of complexity value of the block, this paper adopts the full-enclosed complexity calculation method, which can fully describe the texture structure of the block. As shown in Fig. 5, taking 3 × 3 shadow block as an example, the block complexity value is calculated as shown in formula (20). d_n Represents the absolute difference between adjacent pixels. d_h represents the absolute difference between on the left and right sides. d_v represents the absolute difference between on the upper and lower sides. We an see that this paper adopts checkerboard mode, dividing the blocks into shadow blocks and blank blocks. First, shadow blocks are embedded data, and then blank blocks are embedded data.

$$C = \sum d_n + \sum d_h + \sum d_v \tag{20}$$

3.5. Adaptive 2D mapping of segmented pairwise PEE

Above, we can get optimal prediction error sequence e_{max}through AIPVO prediction method. We need how to design the modification rules of the prediction error to achieve

Table 1
Different image and block size under 2500bit capacity, ratio of prediction error $\{-3, -2, -1, 0, 1, 2\}$.

Image	Block size	e_{max}	0–0.2	0.2–0.4	0.4–0.6	0.6–0.8	0.8–1
Lena	2 × 2	$-3 \le e \le 2$	0.88	0.76	0.53	0.39	0.21
	2 × 2	$e < -3 \mid e \rangle 2$	0.12	0.14	0.47	0.61	0.79
Baboon	2 × 2	$-3 \le e \le 2$	0.53	0.38	0.23	0.17	0.12
	2 × 2	$e < -3 \mid e \rangle 2$	0.47	0.62	0.77	0.83	0.88
Lena	4 × 4	$-3 \le e \le 2$	0.89	0.91	0.88	0.73	0.50
	4 × 4	$e < -3 \mid e \rangle 2$	0.11	0.09	0.12	0.27	0.50
Baboon	4 × 4	$-3 \le e \le 2$	0.67	0.57	0.39	0.30	0.25
	4 × 4	$e < -3 \mid e \rangle 2$	0.33	0.43	0.61	0.70	0.75

excellent embedding performance. Based on recent research, pairwise PEE has proven to achieve excellent performance. This paper will also use pairwise PEE to modify prediction error for data embedding. The two key issues of pairwise PEE are: (1). How to perform effective prediction error pairing to obtain excellent 2D-PEH. (2). How to design effective reversible 2D-mapping.

By observing prediction errors sequence sorted according to complexity value, we find that distribution of prediction errors in sequence is inconsistent. The front part of prediction error sequence is mostly small-magnitude prediction errors, and back part is mostly large-magnitude prediction errors. The front and back part of the prediction error sequence are denoted as e^f_{max} and e^b_{max}, respectively. As shown in Table 1, the ratio of small-magnitude prediction error in e_{max} each part, optimal e_{max} sequence obtained by AIPVO with 2 × 2 block size of Lena image under 1250 bit capacity. Table 1 shows that the prediction error $\{-3, -2, -1, 0, 1, 2\}$ numbers for the first part 0–0.6 of the prediction error sequence have a greater ratio, while the large-magnitude prediction error numbers for the back half of the 0.6–1 proportion have a larger ratio. Therefore, this paper designs different reversible 2D-mapping for front and back part of prediction error sequence, denoted as $MP_f$and MP_b, respectively. The front part mainly uses a small-magnitude prediction error as expanded bin, and the back part uses a slightly large-magnitude prediction error as expanded bin, so that the correlation between prediction errors is further fully utilized.

Next, we need to divide the prediction error sequence into two parts, front part prediction error employ 2D mapping MP_f, and back part prediction error employ 2D mapping MP_b. We use the traversal search method to determine optimal split point Pr. Here $Pr \in \{0.05, 0.1, 0.15 \cdots, 0.95\}$, respectively calculate the ratio of the prediction error $\{-3, -2, -1, 0, 1, 2\}$ in the front $\{0.05, 0.1, 0.15 \cdots, 0.95\}$ of prediction error sequence, and select the split point Pr with the largest ratio to obtain the optimal split point Pr^*. In this way, we can find the optimal split point to segment prediction error sequence into front and back part, and perform the corresponding 2D mapping respectively for data embedding.

In papers [34] and [35], prediction error pairing method was studied. In [34] and [35], in order to increase the capacity, the prediction error is divided into two sets: extended sets and shift sets, and then local adaptive pairing is performed according to position. This paper will use global adaptive pairing method to directly perform prediction error pairing on prediction errors sequence sorted by complexity value. Taking the first part e^f_{max} of e_{max} as an example, the prediction errors are divided into two sets as shown in Eq. (21). In formula (21), E^f_d represents that the extended prediction error that can be embedded data, and S^f_d represents that the shift prediction error that does not embedded data. Refer to the paper [35] and [38], where $d = 3$. Then according to global adaptive pairing method proposed in this paper, prediction errors in same set are paired. Fig. 6 an example based on global adaptive pairing, where e_{max} is optimal prediction error sequence. In this

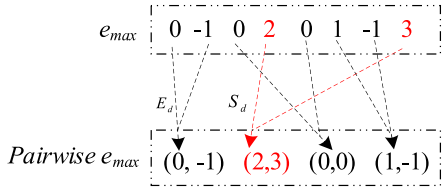

Fig. 6. The global adaptive pairing of this paper.

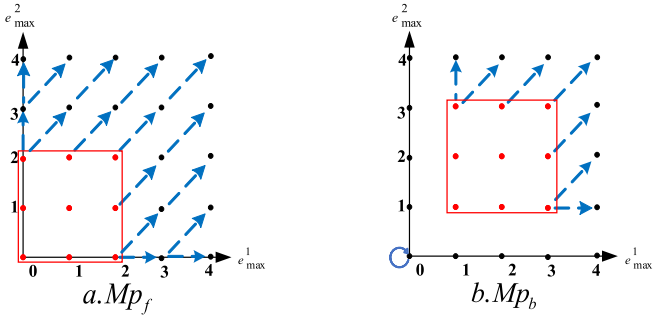

Fig. 7. 2D mapping of the prediction error sequence front and back parts.

way, prediction error pairing can be completed.

$$\begin{cases} E^f_d = \left\{e^f_{max} : \ -d \le e^f_{max} \le d-1\right\} \\ S^f_d = \left\{e^f_{max} : \ e^f_{max} > d-1 \ or \ e^f_{max} < -d\right\} \end{cases} \tag{21}$$

2D mapping MP_f and MP_b specific details are shown Fig. 7. Because four quadrants of 2D mapping are symmetrical, here we take first quadrant as an example. The red arrow represents embedded data, and blue arrow represents shift. Fig. 7a shows 2D mapping of front part of the prediction error sequence. Here, red bin will be used as extended bin. Fig. 7b shows the 2D mapping of back part of the prediction error sequence. Here we mainly use slightly larger prediction error as expanded bin (red dot indicates). The back part of prediction error sequence is also divided into two sets as shown in formula (23), where $d = 3$, and prediction errors in same set are paired. The reference paper [35] adopts the general search method to adaptive 2D mapping, so that the optimal 2D mapping can be solved.

$$\begin{cases} E^b_d = \{e^b_{max} : \ -d \le e^b_{max} \le 2-d \ or \\ d-2 \le e^b_{max} \le d\} \\ S^b_d = \ \{e^b_{max} : \ e^b_{max} \langle -d \ or \ e^b_{max} \rangle d \\ or \ -d-2 < e^b_{max} < d-2\} \end{cases} \tag{22}$$

3.6. Complete process

In the above, we introduced the use of AIPVO to obtain the prediction error sequence and carefully designed two different 2D mapping to modify prediction error and embed data. Next, we

will briefly describe complete proposed scheme. First, this paper adopts checkerboard mode, and divides block into shadow block and blank block as shown in Fig. 5. Because the operation on shadow block is similar to operation on the blank space, here we take data embedding of shadow pixel block as an example.

(1) Pretreatment. In order to prevent the overflow/underflow of the pixel value, a marked bitmap needs to be generated. For pixels with a pixel value 0 and 255 is marking, data cannot be embedded, and marking bitmap is compressed to obtain a short bit stream, which is denoted as *LM*.
(2) Generate prediction errors. Eliminate the surrounding row/column, collect shadow blocks from left to right and top to bottom in the order, use AIPVO method to get optimal prediction error sequence e_{max}, and sort e_{max} according to complexity value.
(3) Embedded data. After the optimal prediction error sequence is obtained, the optimal split point Pr^* can be solved. The prediction error sequence is divided into front and back two parts, and then prediction error is paired using global adaptive pairing method, and the prediction error is modify and embedded data according to 2D mapping Mp_f,Mp_b, and finally calculate $e_{min}(i)$ to the second round data embedding.
(4) Auxiliary information. After shadow block is embedded data, auxiliary information is generated, including optimal complexity threshold θ^*, optimal split point Pr^*, embedding end position, and mark bitmap *LM*. These auxiliary information will be embedded first when blank block is embedded data. The process of embedding data for blank blocks is similar to that for shadow pixel blocks. After blank block is embedded data, the auxiliary information is embedded in first line of test image by LSB replacement.

At decoding end, first extract auxiliary information of blank block, then extract secret data in blank block and restore pixel value of blank block. Then extract secret data of shadow block and restore pixel value of shadow pixel block. Extraction is reverse process of embedding. The specific details of the extraction are as follows

(1) Firstly, auxiliary information. auxiliary information of blank block is extracted including: optimal complexity threshold θ^*, optimal segmentation point Pr^*, end position of the embedding, length of auxiliary information of shadow block, and mark bitmap *LM*.
(2) Calculate prediction error and sorted. After extracting auxiliary information, we can calculate the corresponding mark prediction error e'_{min} according to formula (18), (19), and calculate the block complexity value, sort e'_{min} according to block complexity value.
(3) Extract secret data and restore original pixel values. After getting the mark prediction error e'_{min}, extracting the secret data and restoring the pixel value according to 2D mapping and formula (11), (12). We calculate e'_{max} for the second round of secret data extraction and recovery pixel value. There are similar operations for blank blocks.

4. Experimental results

This section will mainly discuss performance of this paper scheme and its comparison with other advanced schemes, including Li et al. [36], Peng et al. [37], Ou et al. [38], Jia et al. [23], Kim et al. [24], Wang et al. [31], Ou et al. [35], Wu et al. [42], He and Cai [43]. Six 512 × 512 classic gray-scale images named Lena, Baboon, Boat, Elaine, Lake, Pepper will be tested. In this paper, the block sizes a and b are taken as $\{2, 3, 4, 5 \mid a * b \rangle 2\}$, and finally the optimal value that produces the highest PSNR is used for data embedding.

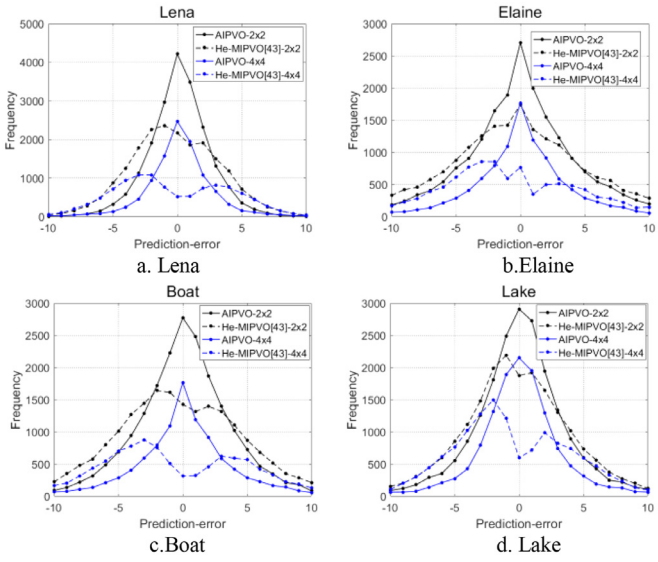

Fig. 8. Comparison of histograms under different block sizes.

4.1. Performance analysis of AIPVO and two-segment pairwise PEE

In this section the AIPVO prediction method and the advantages of two-segment pairwise PEE will be verified. The following will analyze effectiveness of AIPVO method in this paper.

Prediction error histogram (PEH) sharp. Fig. 8 shows a comparison of PEH with different block sizes, the block sizes are 2 × 2 and 4 × 4. The prediction error sequence used here has the same length, its first 20,000 prediction errors, and the prediction error of AIPVO is obtained with capacity of 1250bit. We can observe that the prediction histogram obtained by AIPVO is sharp for each block size, which is conducive to data embedding. The AIPVO prediction method in this paper has a higher accuracy, as may be seen intuitively.

Effective prediction error rate (EPER). At different block sizes and capacities, Table 2 compares the EPER R_e between the prediction method of AIPVO and He and Cai [43]. R_e is calculated as shown in formula (16). The larger the R_e, the sharper the PEH and the better the embedding performance. Table 2 shows that the R_e of the AIPVO method is greater than the He and Cai [43]. prediction method for the same capacity and different block size. The AIPVO method produces fewer invalid shifts when embedding the same capacity of secret data, and has better embedding performance. Especially for the Baboon image, the R_e of AIPVO is nearly twice as high as that of He and Cai [43], indicating that the AIPVO prediction method has a better effect on texture images.

Ineffective shift. In order to further verify advantages of the AIPVO method, Table 3 compares the length of the prediction error sequence used in the first round of embedding of the different capacity and blocks size in the shadow block. In Table 3, the prediction error $\{-3, -2, -1, 0, 1, 2\}$ is used to embed data based on IPVO. The shorter the prediction error sequence used under the same capacity means the less the invalid shift and the smaller the distortion. When the capacity is 2500 bit, the prediction error sequence length used by the AIPVO method is much smaller than the prediction method of He and Cai [43], so the IPVAO method will produce fewer invalid shifts. At 5000bit capacity, we can observe that the number of prediction errors used by the He and Cai [43] method is larger than that of the AIPVO method, which means that the AIPVO method produces less invalid shift. Under the same capacity, the larger the block size, the longer the prediction error sequence used by the He and Cai [43] prediction method, which

Table 2
Comparison of R_e under different block sizes.

Image	Method	2500				5000			
		2 × 2	3 × 3	4 × 4	5 × 5	2 × 2	3 × 3	4 × 4	5 × 5
Lena	AIPVO	**0.82**	**0.77**	**0.85**	**0.83**	**0.80**	**0.80**	**0.75**	**0.79**
	He and Cai [43].	0.61	0.48	0.37	0.30	0.57	0.43	0.28	0.19
Baboon	AIPVO	**0.48**	**0.48**	**0.47**	**0.46**	**0.38**	**0.38**	**0.34**	**0.33**
	He and Cai [43].	0.24	0.16	0.12	0.02	0.19	0.11	0.1	0.08
Boat	AIPVO	**0.72**	**0.79**	**0.78**	**0.79**	**0.65**	**0.69**	**0.75**	**0.76**
	He and Cai [43].	0.47	0.36	0.26	0.20	0.42	0.26	0.18	0.11
Elaine	AIPVO	**0.77**	**0.78**	**0.77**	**0.75**	**0.62**	**0.65**	**0.59**	**0.61**
	He and Cai [43].	0.66	0.52	0.36	0.25	0.44	0.29	0.21	0.12
Lake	AIPVO	**0.81**	**0.75**	**0.79**	**0.79**	**0.76**	**0.74**	**0.79**	**0.76**
	He and Cai [43].	0.76	0.67	0.53	0.44	0.62	0.47	0.34	0.25
Peppers	AIPVO	**0.71**	**0.75**	**0.75**	**0.79**	**0.67**	**0.72**	**0.72**	**0.69**
	He and Cai [43].	0.44	0.35	0.27	0.22	0.42	0.32	0.24	0.19

Table 3
IPVO embedding uses the length of prediction error sequence with 2500 bit and 5000 bit capacity.

Image	Method	2500				5000			
		2 × 2	3 × 3	4 × 4	5 × 5	2 × 2	3 × 3	4 × 4	5 × 5
Lena	AIPVO	**3487**	**3719**	**3324**	**3370**	**7097**	**7177**	**6639**	**6776**
	He and Cai [43].	4945	6556	9274	12,385	10,606	15,065	22,159	32,403
Baboon	AIPVO	**6620**	**6576**	**6750**	**6923**	**17,922**	**18,022**	**21,213**	**22,030**
	He and Cai [43].	13,906	27,753	55,411	121,778	36,634	113,321	120,916	124,826
Boat	AIPVO	**4064**	**3604**	**3642**	**3618**	**9129**	**8349**	**9040**	**8953**
	He and Cai [43].	6349	9716	15,676	24,820	15,785	26,954	48,031	80,977
Elaine	AIPVO	**3746**	**3467**	**3739**	**3823**	**9680**	**9115**	**10,202**	**10,077**
	He and Cai [43].	4472	6009	9301	16,637	14,800	24,905	43,518	80,138
Lake	AIPVO	**3528**	**3864**	**3629**	**3614**	**7595**	**7865**	**7288**	**7540**
	He and Cai [43].	3804	4401	5750	7393	9553	13,418	20,701	33,242
Peppers	AIPVO	**4115**	**3829**	**3847**	**3605**	**8784**	**8066**	**8084**	**8427**
	He and Cai [43].	7273	9876	14,778	20,594	15,724	23,009	36,482	56,257

Table 4
Optimal solutions of different image parameters R_s and θ^*.

Image	10,000		20,000	
	R_S	θ^*	R_S	θ^*
Lena	0.6	(11, 16)	0.5	(20, 36)
Baboon	0.25	(11, 39)	0.05	(99, 169)
Boat	0.5	(13, 20)	0.35	(27, 46)
Elaine	0.5	(20, 26)	0.4	(31, 49)
Lake	0.45	(16, 23)	0.35	(28, 36)
Pepper	0.55	(4, 15)	0.4	(19, 44)

indicates that the larger the block, the greater the error of the He and Cai [43] prediction method.

Table 4 shows the optimal parameters selected in the first round embedding of different image shadow blocks under the 2 × 2 block size. The R_e value of the Baboon image is the lowest in Table 4. This is because Baboon is a texture image, low prediction accuracy resulting in a low R_e value. Generally, the higher the complexity of the image texture, the lower the value of R_s, and the value of R_s will gradually decrease as the capacity increases.

Whether to use two-segment pairwise PEE. Fig. 9 shows the use of a single 2D mapping, segmented adaptive 2D mapping Mp_f, Mp_b and AIPVO method, He and Cai [43] prediction method of the two-two combination of performance comparison. In this paper, the combination of AIPVO and segmented 2D mapping Mp_f,Mp_b, the combination of AIPVO and single 2D mapping, the combination of He prediction method and segmented 2D mapping Mp_f, Mp_b, and the combination of He and Cai [43] prediction method and single 2D mapping are recorded separately is Proposed, Pro-OneMap, He-TwoMap, He-OneMap. The PSNR of the smoother image is not substantially different under large capacity, as seen in

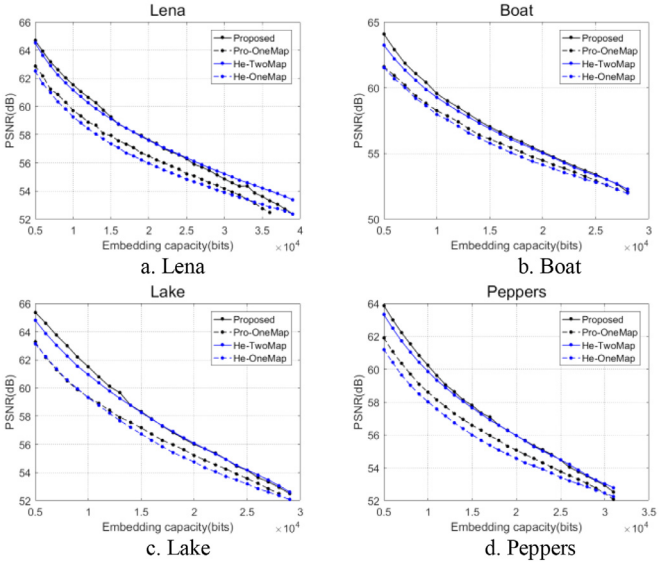

Fig. 9. Performance comparison between only using mapping Mp_f and using segmented mapping $Mp_f$and Mp_b.

Fig. 9. This is because the number of expanded 2D bins has been increased. Almost all the prediction errors obtained by using AIPVO in a small capacity can be formed into expanded 2D bins to embed data. In large capacity, the use of AIPVO will reduce the total number of prediction errors, which will reduce performance. We can observe that the PSNR of Proposed and Pro-OneMap are higher than He-TwoMap and He-OneMap respectively, which proves that AIPVO is better than He and Cai [43]. This also verifies the effectiveness of the AIPVO prediction algorithm again.

Table 5
The block size under the best PSNR with different capacities.

Image		Lena	Baboon	Boat	Elaine	Lake	Pepper
10,000 bit	*PSNR*	61.55	55.74	59.56	59.63	61.52	60.24
	$a \times b$	5 × 5	4 × 3	2 × 5	4 × 3	3 × 4	4 × 3
20,000 bit	*PSNR*	57.63	–	55.11	54.42	55.99	55.96
	$a \times b$	3 × 3	–	2 × 3	3 × 2	3 × 2	2 × 3

Table 6
PSNR comparison between this paper scheme and other advanced schemes with 10,000 bits.

Image	[23]	[24]	[31]	[35]	[42]	[43]	Proposed
Lena	59.08	60.08	60.99	60.42	61.15	61.16	**61.55**
Baboon	54.74	55.97	55.81	**56.03**	54.86	55.11	55.74
Boat	56.61	57.63	58.97	58.13	58.67	59.27	**59.56**
Elaine	57.83	58.39	58.52	58.50	58.09	59.50	**59.63**
Lake	57.89	58.22	59.20	59.18	59.74	60.95	**61.52**
Pepper	55.63	56.94	58.67	57.02	59.60	59.87	**60.24**
Average	56.96	57.87	58.69	58.21	58.68	59.31	**59.71**

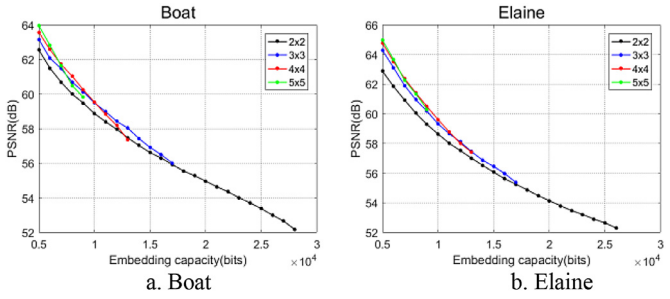

Fig. 10. Performance comparison of different block sizes with Boat and Elaine.

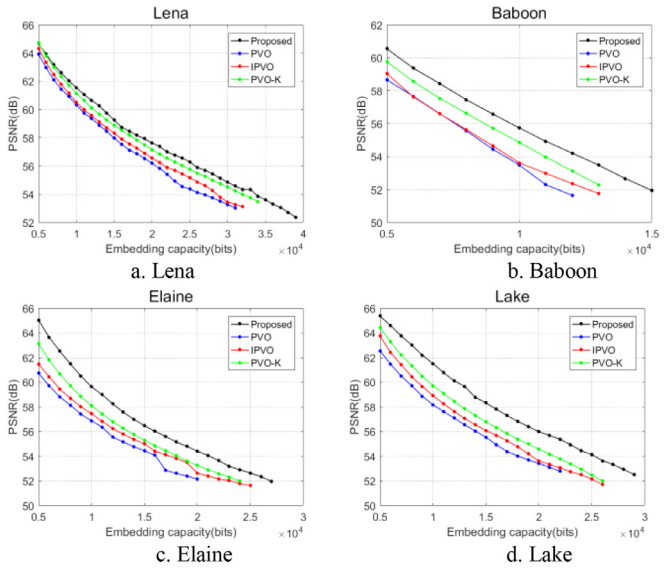

Fig. 11. PSNR comparison of proposed method and other three PVO methods.

4.2. Comprehensive performance analysis

The effect of block size on performance. The block size of block will affect the accuracy of predicted value, as well as the sequence of operations on the block. Fig. 10 shows impact of Boat and Elaine images on performance under different block sizes with 5000 bit. In Fig. 10 we can observe that the larger the block, the better the performance, but large blocks have limited embedding capacity. The main reason is that the larger the block, the more accurate the complex value of the block, and the very smooth block can be used preferred. But it is only effective for very smooth blocks, and the prediction accuracy of normal blocks and texture blocks is very low. Natural images rarely have very smooth images, so the large block capacity is low. Small blocks generally have higher prediction accuracy, and can utilize different texture regions of the image, which can significantly increase the capacity. Table 5 shows that when the embedding capacity grows, the block size gradually decreases. This is because as the capacity increases, more small-magnitude prediction errors are required, and small blocks can improve the prediction accuracy of texture area. Generally, large blocks are selected for small capacity, and small blocks are selected for large capacity.

Performance comparison with existing PVO algorithms. Fig. 11 shows that the proposed AIPVO method achieves better visual quality at all embedding payload in four test images. In Lena, when the embedding capacity is 10,000 bits, the PSNR of AIPVO is 0.40 dB higher than PVO-K, which is 0.47 dB when the embedding capacity increases to 20,000 bits. In Elaine, when the embedding payload is 10,000 bits, the PSNR by AIPVO is 1.54 dB higher than that by PVO-K, which is 1.16 dB when the pure payload increases to 20,000 bits. Experimental results show that the embedding capacity of the proposed method is also enhanced. In Lena, the embedding capacity of proposed method is 5000 bits larger than that in PVO-K, 5000 bits larger than that in IPVO.

Performance comparison with the latest existing RDH scheme. Fig. 12 shows comparison of the PSNR of this paper algorithm and other advanced algorithms at each capacity point. Comparing 6 test images, we can observe that the performance of this paper algorithm surpasses other advanced algorithms. Tables 6 and 7 show the PSNR comparison between this paper algorithm and other advanced algorithms at 10,000 bit and 20,000bit capacities. Whether it is 10,000 bit or 20,000 bit, this paper algorithm is superior to other advanced algorithms in all test images. This paper algorithm is most similar to algorithm in the paper [43] and [43] is the latest research result. The comparative paper [43] has an average gain of 0.4 dB and 0.13 dB at 10,000 bit and 20,000 bit capacities, respectively. The above data fully illustrates the superiority of this paper algorithm.

In Wu et al. [42], the IPVO prediction method is combined with the adaptive 2D mapping. Refer to Tables 6 and 7, compared to Wu et al. [42] the average gain of this paper algorithm is 1.02 dB and 1.02 dB, respectively. The IPVO prediction method in Wu et al. [42] utilized the second largest/smallest pixel to predict largest/smallest pixel, so the prediction accuracy is very high. However, there is no double-round embedding in Wu et al. [42] that

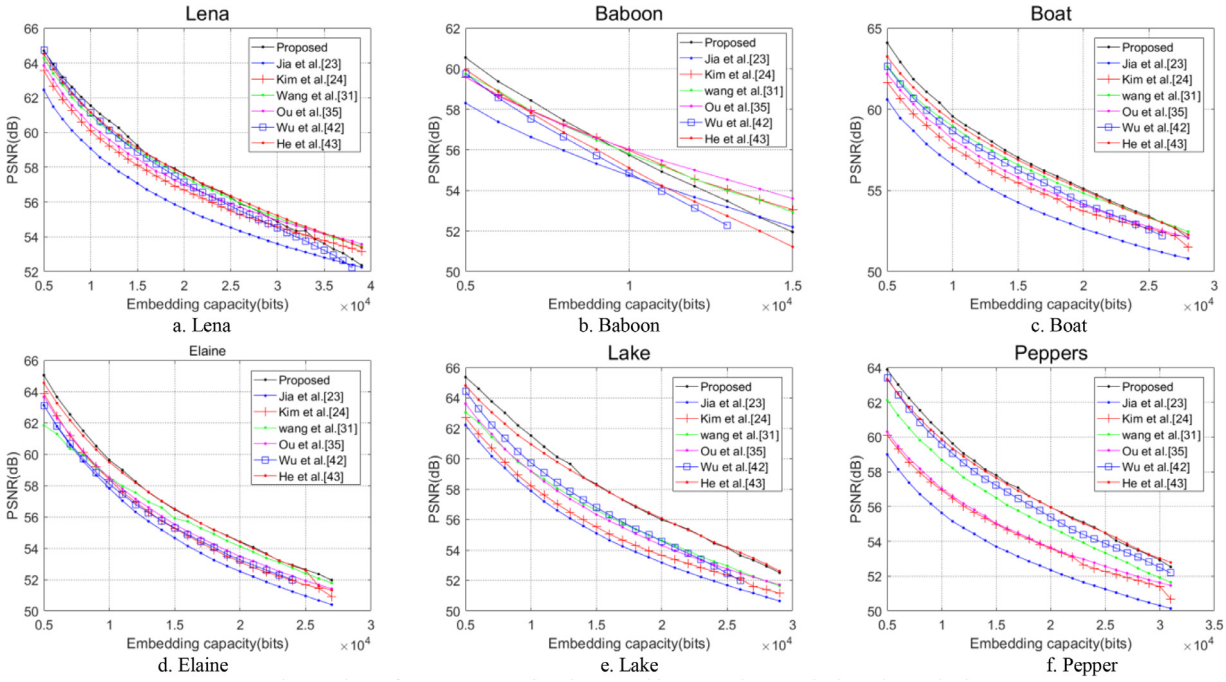

Fig. 12. The performance comparison between this paper scheme and other advanced schemes.

Table 7
PSNR comparison between this paper scheme and other advanced schemes with 20,000 bits.

Image	[23]	[24]	[31]	[35]	[42]	[43]	Proposed
Lena	55.62	56.70	57.47	57.00	57.16	57.60	**57.63**
Boat	52.63	53.73	54.83	54.09	54.19	55.02	**55.31**
Elaine	52.52	53.13	54.12	53.47	53.25	54.39	**54.62**
Lake	53.17	53.65	54.64	54.31	54.59	56.08	**56.19**
Pepper	52.35	53.60	54.80	53.64	55.39	**55.97**	55.96
Average	53.26	54.16	55.17	54.50	54.92	55.81	**55.94**

will not produce distortion offset. The complexity calculation in this paper is closed enclosing that can accurately describe the texture complexity of pixel block, while the complexity calculation in Wu et al. [42] is a half-open context. Then in the embedding process, this paper designs two different 2D mapping based on the different distributions of prediction errors in the prediction error sequence, which can make full use of the correlation between the prediction errors.

In Ou et al. [35], adaptive 2D mapping is used. Refer to Tables 6 and 7 compared with Ou et al. [35] the average gain of this paper algorithm is 1.49 dB and 1.44 dB, which is significantly better than the algorithm in Ou et al. [35]. The accuracy of the traditional rhombus prediction in [35] is lower than that of AIPVO method, so the prediction error sequence obtained by AIPVO method is better than the prediction error sequence in [35], mean the same capacity embedded but invalid shifts are less. The prediction error pairing method paper [35] is a local adaptive pairing based on position. The pairing method in this paper is to directly pair according to the sorted prediction error sequence, which is a global adaptive pairing, which has a higher capacity.

In [31], clustering algorithm is used to obtain multiple histograms. Through multiple histograms, different texture areas of the image can be fully utilized. However, referring to Fig. 12, Tables 6 and 7, the PSNR of this paper algorithm is much higher than literature [31]. The reason is similar to the above explanation.

Performance testing on Kodak image database. In order to further verify the performance of this paper algorithm, tested in a larger image set Kodak. Because [43] is the latest research result, and algorithm in [43] is similar to in this paper algorithm, and both use multi-pass embedding, so [43] is used here as a comparison literature. The comparison result is shown in Fig. 13. In Fig. 13, we can observe that the PSNR of the algorithm in this paper is inferior to the algorithm of He and Cai [43] on some images, but these images are mainly smooth images, which cannot utilize advantage of the AIPVO of this paper. Under the capacity of 30,000 bit, the 13th image is an image of complex texture. The capacity of the He and

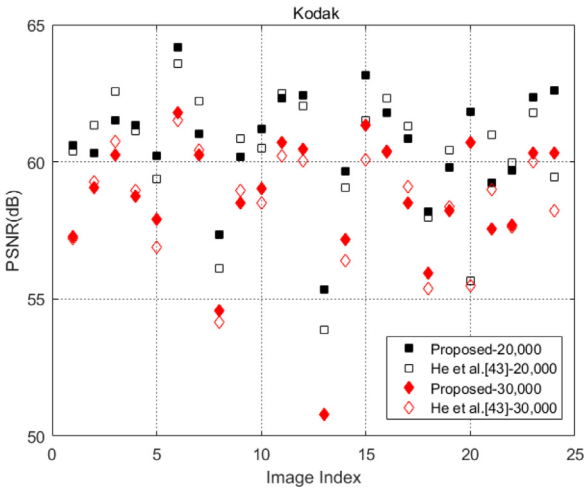

Fig. 13. Performance comparison on Kodak image database between proposed scheme and [43].

Cai [43] algorithm is less than 30,000 bit, but the algorithm in this paper can reach 30,000 bit capacity, which shows that the algorithm in this paper has a higher capacity. Although the PSNR of the algorithm in this paper is inferior to the algorithm [43] on some smooth images, the overall performance is superior than the algorithm [43]. Compared the algorithm [43], the average PSNR gain of the proposed algorithm are 0.71 dB and 0.50 dB for 20,000 bit and 30,000 bit capacity, respectively. The overall performance can prove the superiority of this paper algorithm.

5. Conclusion

This paper proposes a high-performance RDH scheme. This paper adopts the AIPVO prediction method, which can adaptively determine the predicted value according to the complexity value of the block. The EPER R_e of the evaluation index of a prediction error sequence is defined, and the optimal block complexity level division model is solved by using R_e as the objective function. AIPVO can eliminate the pixels with large fluctuations in the block and further utilize the correlation of the pixel values at block within. According to the distribution of prediction errors in the prediction error sequence, the prediction error sequence is divided into two parts. By traversing the search method to find the optimal split point, the prediction error sequence is divided into front and back two parts. Design two different 2D maps for the front and back part of the prediction error sequence, respectively. The correlation of prediction errors can be further exploited. This paper scheme inherits the advantages of multi-pass embedded distortion offset. Experimental results show that this paper RDH scheme is superior to other advanced RDH schemes.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Ningxiong Mao: Conceptualization, Methodology, Software, Validation, Writing – original draft, Visualization. **Fan Chen:** Funding acquisition, Resources. **Hongjie He:** Writing – review & editing, Project administration, Funding acquisition. **Yaolin Yang:** Investigation, Formal analysis, Data curation.

References

[1] Y.Q. Shi, Z. Ni, D. Zou, C. Liang, G. Xuan, Lossless data hiding: fundamentals, algorithms and applications, in: Proceedings of the IEEE International Symposium on Circuits and Systems (IEEE Cat. No. 04CH37512), 2, 2004, pp. II–33.

[2] Y.Q. Shi, Reversible data hiding, in: Proceedings of the International Workshop on Digital Watermarking, Springer, 2004, pp. 1–12.

[3] R. Caldelli, F. Filippini, R. Becarelli, Reversible watermarking techniques: an overview and a classification, EURASIP J. Inf. Secur. (2010) 1–9.

[4] G. Xuan, J. Chen, J. Zhu, Y.Q. Shi, Z. Ni, W. Su, Lossless data hiding based on integer wavelet transform, in: Proceedings of the IEEE Workshop on Multimedia Signal Processing, 2002, pp. 312–315.

[5] M.U. Celik, G. Sharma, A.M. Tekalp, E. Saber, Reversible data hiding, in: Proceedings of the International Conference on Image Processing, 2, 2002 II–II.

[6] G. Xuan, Y.Q. Shi, Z. Ni, J. Chen, C. Yang, Y. Zhen, J. Zheng, High capacity lossless data hiding based on integer wavelet transform, in: Proceedings of the International Symposium on Circuits and Systems (IEEE Cat. No. 04CH37512), 2, 2004, pp. II–29.

[7] M.U. Celik, G. Sharma, A.M. Tekalp, E. Saber, Lossless generalized-LSB data embedding, IEEE Trans. Image Process. 14 (2005) 253–266.

[8] M.U. Celik, G. Sharma, A.M. Tekalp, Lossless watermarking for image authentication: a new framework and an implementation, IEEE Trans. Image Process. 15 (2006) 042–1049.

[9] J. Tian, Reversible data embedding using a difference expansion, IEEE Trans. Circuits Syst. Video Technol. 13 (2003) 0–896.

[10] A.M. Alattar, Reversible watermark using the difference expansion of a generalized integer transform, IEEE Trans. Image Process. 13 (2004) 7–1156.

[11] I.C. Dragoi, D. Coltuc, Local-prediction-based difference expansion reversible watermarking, IEEE Trans. Image Process. 23 (2014) 1779–1790.

[12] S. Xiang, Y. Wang, Non-integer expansion embedding techniques for reversible image watermarking, EURASIP J. Adv. Signal Process. 2015 (2015) 1–12.

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

[14] 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 (2009) 989–999.

[15] G. Feng, L. Fan, Reversible data hiding of high payload using local edge sensing prediction, J. Syst. Softw. 85 (2012) 392–399.

[16] T.C. Lu, C.Y. Tseng, K.M. Deng, Reversible data hiding using local edge sensing prediction methods and adaptive thresholds, Signal Process. 104 (2014) 152–166.

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

[18] X. Hu, W. Zhang, X. Li, N. Yu, Minimum rate prediction and optimized histograms modification for reversible data hiding, IEEE Trans. Inf. Forensics Secur. 10 (2015) 653–664.

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

[20] X. Ma, Z. Pan, S. Hu, L. Wang, High-fidelity reversible data hiding scheme based on multi-predictor sorting and selecting mechanism, J. Vis. Commun. Image Represent. 28 (2015) 71–82.

[21] S. Hiary, I. Jafar, H. Hiary, An efficient multi-predictor reversible data hiding algorithm based on performance evaluation of different prediction schemes, Multimed. Tools Appl. 76 (2017) 2131–2157.

[22] L. Luo, Z. Chen, M. Chen, X. Zeng, Z. Xiong, Reversible image watermarking using interpolation technique, IEEE Trans. Inf. Forensics Secur. 5 (2009) 187–193.

[23] Y. Jia, Z. Yin, X. Zhang, Y. Luo, Reversible data hiding based on reducing invalid shifting of pixels in histogram shifting, Signal Process. 163 (2019) 238–246.

[24] S. Kim, X. Qu, V. Sachnev, H.J. Kim, Skewed histogram shifting for reversible data hiding using a pair of extreme predictions, IEEE Trans. Circuits Syst. Video Technol. 29 (2018) 3236–3246.

[25] Y. Hu, H.K. Lee, J. Li, De-based reversible data hiding with improved overflow location map, IEEE Trans. Circuits Syst. Video Technol. 19 (2008) 250–260.

[26] H. Wu, X. Li, Y. Zhao, R. Ni, Improved PPVO-based high-fidelity reversible data hiding, Signal Process. 167 (2020) 107–264.

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

[28] J. Wang, J. Ni, X. Zhang, Y.Q. Shi, Rate and distortion optimization for reversible data hiding using multiple histogram shifting, IEEE Trans. Cybern. 47 (2016) 315–326.

[29] W. Pan, G. Coatrieux, N. Cuppens, F. Cuppens, C. Roux, Reversible watermarking based on invariant image classification and dynamical error histogram shifting, in: Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2011, pp. 4477–4480.

[30] C. Qin, C.C. Chang, Y.H. Huang, L.T. Liao, An inpainting assisted reversible steganographic scheme using a histogram shifting mechanism, IEEE Trans. Circuits Syst. Video Technol. 23 (2012) 1109–1118.

[31] J. Wang, N. Mao, X. Chen, J. Ni, C. Wang, Y. Shi, Multiple histograms based reversible data hiding by using FCM clustering, Signal Process. 159 (2019) 193–203.

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

[33] 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 (2013) 5010–5021.

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

[35] B. Ou, X. Li, W. Zhang, Y. Zhao, Improving pairwise pee via hybrid-dimensional histogram generation and adaptive mapping selection, IEEE Trans. Circuits Syst. Video Technol. 29 (2018) 2176–2190.

[36] 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 (2013) 198–205.

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

[38] 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 (2014) 760–772.

[39] 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.

[40] 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 Represent. 39 (2016) 12–23.

[41] S. Weng, Y. Shi, W. Hong, Y. Yao, Dynamic improved pixel value ordering reversible data hiding, Inf. Sci. 489 (2019) 136–154.

[42] H. Wu, X. Li, Y. Zhao, R. Ni, Improved reversible data hiding based on PVO and adaptive pairwise embedding, J. Real Time Image Process. 16 (2019) 685–695.

[43] W. He, Z. Cai, An insight into pixel value ordering prediction-based prediction-error expansion, IEEE Trans. Inf. Forensics Secur. 15 (2020) 3859–3871.