

ORIGINAL RESEARCH PAPER



A novel reversible data hiding based on adaptive block-partition and payload-allocation method

Yang Yang¹ | Genyan Huang¹ | Tianrui Zou¹ | Weiming Zhang²

¹ School of electronics and information engineering, Anhui University, Hefei, China
² School of Information Science and Technology, University of Science and Technology of China, Hefei, China

Correspondence
Yang Yang, School of electronics and information engineering, Anhui University, Hefei, 230601, China.
Email: sky_yang@ahu.edu.cn

Funding information
National Natural Science Foundation of China, Grant/Award Numbers: 61502007, 61572452; China Postdoctoral Science Foundation, Grant/Award Number: 2015M582015; University Natural Science Research Project of Anhui Province, Grant/Award Number: 1608085MF125; Doctoral Scientific Research Foundation of Anhui University, Grant/Award Number: J01001319

Abstract
A reversible data hiding method based on partition is researched because it can effectively reduce shifting distortion in given embedding capacity. However, traditional partition is to divide an image into equal-sized blocks, which cannot be divided reasonably according to the content of the image. In order to achieve dynamic partition and effectively utilize the complexity of the image, this paper proposes a novel reversible data hiding based on adaptive block-partition and payload-allocation method. In this technique, instead of equal partition, adaptive block-partition is proposed to establish multiple histograms by dividing the cover image into several image blocks of different sizes dynamically, and the image blocks of different sizes are processed successively through multiple sorting and implement adaptive payload allocation according to complexity, then the data is embedded into two sides of prediction-error histograms to effectively reduce the shifting distortion. Experimental results show that the proposed method is superior to the state-of-the-art traditional fixed-sized blocking-based reversible data hiding methods.

1 | INTRODUCTION

With the rapid development of digital multimedia technology, the copyright protection technology of digital multimedia content becomes more and more important. In order to protect the information security, data hiding [1] is proposed, which refers to hiding information into the cover medium without being detected. However, most data hiding technologies usually lead to permanent distortion of digital content, therefore, it cannot be restored to the original form after data extraction. But the slight damage of cover medium will cause irreparable loss in some special industries, such as military, medical and so on. To solve this problem, reversible data hiding (RDH) [2] is introduced, which can ensure the privacy and extractability of embedded information, and can completely recover the cover medium without damage.

Nowadays, many RDH algorithms are proposed, including lossless compression [3–6], histogram shifting (HS) [7–10], difference expansion (DE) [11] and prediction error expansion (PEE) [12–15] and so on. The main idea of compression is loss-

less compression of the content of the cover medium, such as the least significant bit (LSB) and embed secret information in the space saved. Fridrich et al. [3] proposed to decompose the gray-image into binary bit plane, and then select the bit plane with the largest redundancy for compression, using the saved space embed information to achieve reversible embedding and extraction of image. However, in this approach the embedding capacity depends mainly on the lossless compression capacity, and the embedding capacity is insufficient. Then Tian [11] proposed a DE technique that divides an image into N pixel pairs and hides 1 bit of data with each legitimate pixel pair. Its embedding capacity can reach 0.5 bpp, but the uncompressed location map is 0.5 bpp, so that it is necessary to reduce the location map to improve the embedding capacity. Therefore, Hu et al. [16] proposed to improve the location map to greatly improve the embedding capacity. Affected by the DE [11], the prediction errors (PE) [17, 18] is proposed to achieve better performance. This extended method is called PEE, which is the current research hotspot and the most powerful technique of RDH. The method uses the pixel prediction mechanism and the

设定偏好

全部接受

全部拒绝

spatial redundancy of natural images, such as median edge detection [19], gradient adjusted prediction [20], interpolation prediction [21], rhombus prediction [22] and least square prediction (LS) [23] etc., to effectively implement RDH based on pixel prediction. In addition to DE, HS is also a far-reaching method. Ni et al. [24] first proposed an HS algorithm that generates a histogram based on the gray value of the image and hides the data reversibly by modifying the histogram. Then, Falahpour et al. [25] proposed that by taking advantage of the spatial characteristics of medical images block, finding the maximum and minimum frequency of pixel intensity histogram for histogram shifting in image. Lee et al. [26] proposed to integrate HS into the PEH. Subsequently, most existing HS based RDH schemes build a single histogram, to take advantages of the correlation among image contents of different texture characteristics, Wang et al. [14, 27] proposed the idea of multiple histograms to achieve better performance.

Recently, a high fidelity RDH method based on pixel value ordering (PVO) is widely studied [28–31]. PVO is a new prediction strategy based on partition. Cover image is divided into several image blocks, and then PVO and PEE are combined to implement information embedding. In 2013, Li et al. [28] first proposed the idea based on PVO, in Li's method, the image is divided into several non-overlapping image blocks equally, and then the pixel values of each image block is sorted in ascending order. When the PE is '1' and '-1' embed the data and the PE is '0' is ignored. However, using only the maximum and minimum values as data bit carriers may lead to embedding capacity deficiency, in order to improve embedding capacity, Peng et al. [29] proposed an improved PVO(IPVO), which use the PE value of '0'. In [30], Ou et al. [30] proposed to extend the PVO method to PVO-K, where K is the number of pixels equal to the maximum/minimum values. Then in order to improve the embedding performance, Wang et al. [31] proposed to partition some smooth image blocks into four blocks according to the complexity of the image. Later on, the methods based on PVO have improved in many aspects [32–35], such as Weng et al. [32] proposed a flexible block-partition and adaptive pixel-modification strategy, more data bits can be embedded into smooth blocks by dividing smooth blocks into smaller sub-blocks. He et al. [36] proposed an optimized PVO-based pairwise PEE and combine with 2D PEH, design the high and low power version of PVO-based pairwise PEE. In [37], Weng et al. [37] proposed an improved PVO scheme based on K-pass, which calculated the PE by studying the position relationship between the largest and second largest pixels (or the smallest and second smallest pixels) to provide a large number of embeddable pixels. And Wu et al. [38] proposed to consider the PEs for the largest and smallest pixel values of each block are jointed as a pair, and then modify the PE pairs based on 2D histogram modification strategy. However, even if the block method is improved, this fixed and equal image partition still cannot efficient use of the smooth region of the image, especially for satisfying the

In this paper, we propose a novel reversible data hiding based on adaptive block-partition and payload-allocation method. Firstly, we propose an adaptive block-partition method by dividing the image into several image blocks with different sizes to utilize the pixel correlation well and establish multiple histograms, thus effectively reduce shifting distortion. Secondly, we use the linear combination of multiple pixel values by integrating the rhombus prediction [22] and LS [23] method for effectively improving the prediction accuracy and having the ability of sorting. Thirdly, we propose an adaptive payload allocation by reasonably allocating the information of each image block according to the complexity to embed more information in the smooth region. Lastly, we propose a new embedding method by embedding on two sides of PEH to effectively reduce shifting distortion. Here, the main contribution of this paper are as follows: (1) Propose an adaptive partitioning method to effectively utilize the correlation between pixels; (2) Propose an adaptive payload allocation method, which embeds information based on image size and complexity instead of equally embedding; (3) Propose a new embedding strategy of two sides embedding to reduce the shifting distortion. Compared with the traditional fixed-sized blocking-based RDH methods, the proposed method can achieve better embedding performance, and its embedding capacity is higher than the traditional fixed-sized blocking-based RDH methods we compared.

The rest of the paper is mainly composed of the following parts: in the Section 2, the method we proposed is introduced in detail; Section 3 is the experiments and results, which gives the experimental results of the proposed method and the comparative analysis with other methods; the last part is the conclusion which will be shown in the Section 4.

2PROPOSED METHOD

Traditional RDH mainly generate single histogram, but it will cause more embedding distortion. So some researchers propose the method of multiple histograms [14, 27] to minimize the length of the PE sequence such as partitioning, which can take advantage of the correlation between image content of different texture features, and then the smooth image block can be preferentially selected for information embedding. For optimizing partitioning performance, we propose an adaptive partitioning method to divide the image into multiple image blocks of different sizes. Then, we assign different sizes of payload to different blocks instead of equally embedding, and we propose an embedding method that can effectively reduce shifting distortion. The framework of the method we proposed is shown in Figure 1.

2.1 | An adaptive block-partition method based on complexity

In order to improve embedding efficiency, some scholars pro-

设定偏好

全部接受

全部拒绝

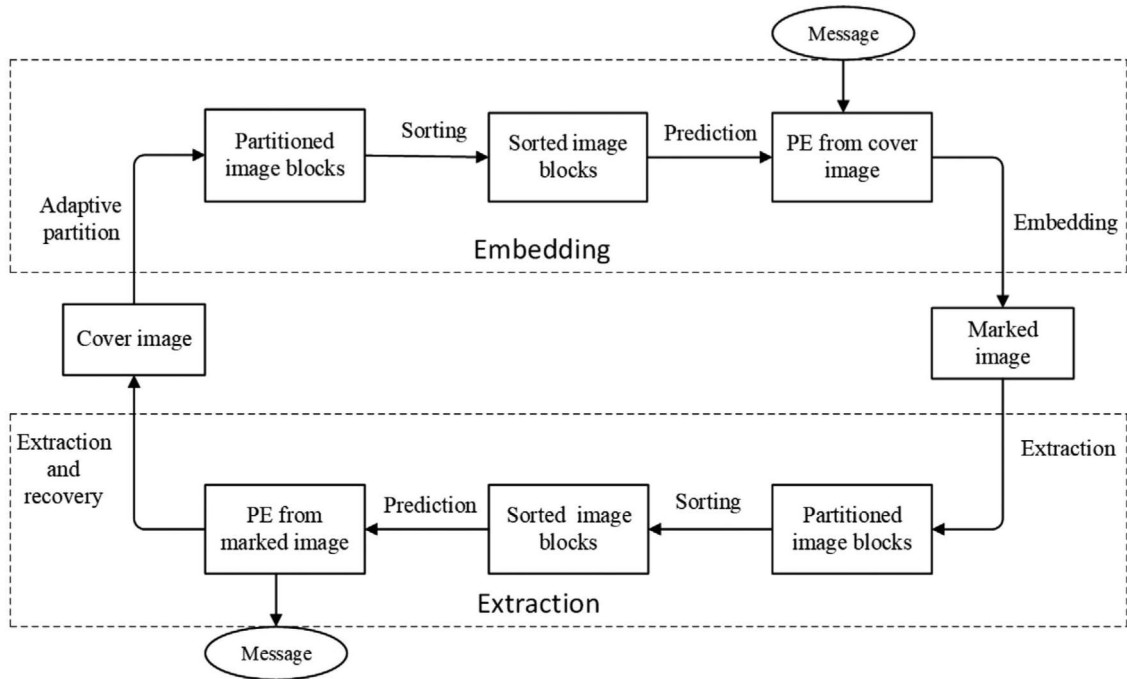


FIGURE 1 Framework of the proposed method

embedding capacity, a second partition is performed for some smooth blocks. However, this method only divides the image equally and does not divide the image reasonably according to the content of the image, so it cannot reflect the texture of the image reasonably. Actually, texture complexity of the image block directly affects the accuracy of the prediction, and the size of the block directly affects the embedding capacity of the information and the visual quality of the marked image. Therefore, in order to distinguish the texture of the image well and to partition the image dynamically, an adaptive block-partition method based on complexity is proposed. The partitioning method we proposed can take full advantage of the complexity to distinguish between smooth and complexity blocks. We know that the pixel values in the smooth region are similar, vice versa, and if a block is smoother than the other blocks, it will have sharper PEH, thus leading to less distortion, so we choose to embed more information in the smooth region to keep the high PSNR value.

In the proposed partition method, we use the ideas of the quadtree. Quadtree is a data structure with up to four subtrees per node. We use the idea to design an adaptive block-partition method based on complexity, which can divide the image into several non-overlapping image blocks of different sizes. Next, we describe the details.

Let $R_0^{(0)}$ of the size $M \times N$ represent the cover image, we find its maximum and minimum pixel values before partitioning, denoted as $R_{\max_{i,j}}$, $R_{\min_{i,j}}$ respectively, the complexity C calculated by Equation (1) is expressed as the difference between

After obtaining the complexity C , compared it with the adaptive threshold T , $T \in [0, 1]$ to determine whether to divide again, since the image $R_0^{(0)}$ is 8 bits gray-image, the threshold T will be multiplied by 255, $R_x^{(n)}$ where n represents the n -th partition, and $x, x \in \{0, 1, 2, 3\}$ represents the x -th image block:

$$R_x^{(n+1)} = \begin{cases} R_0^{(n+1)}, R_1^{(n+1)}, R_2^{(n+1)}, R_3^{(n+1)}; \\ \text{if } C(R_x^{(n)}) \geq 255 \times T \\ R_x^{(n)}; \\ \text{if } C(R_x^{(n)}) < 255 \times T. \end{cases} \quad (2)$$

From Equation (2), $C(R_x^{(n)})$ represents the complexity of image block $R_x^{(n)}$, and a small value of $C(R_x^{(n)})$ indicates the image block is smoother, vice versa. So if $C(R_x^{(n)}) < 255 \times T$, it means that the complexity of image block $R_x^{(n)}$ is small, which means the image block is smooth, then we will not divide it any more. Otherwise, divide it into four blocks $R_0^{(n+1)}, R_1^{(n+1)}, R_2^{(n+1)}, R_3^{(n+1)}$, and then calculate its complexity one by one in the same way until we can no longer divide. By the way, threshold T plays a crucial role in the partitioning effect, we will introduce how to choose the optimal T through adaptive method in Section 2.4.

Previously, we introduce the method based on complexity partitioning, now give an example to explain how to divide the block. As shown in Figure 2, randomly select an 8×8 image

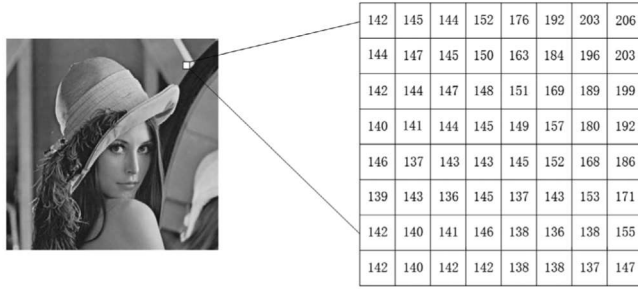


FIGURE 2 Image block from "Lena"

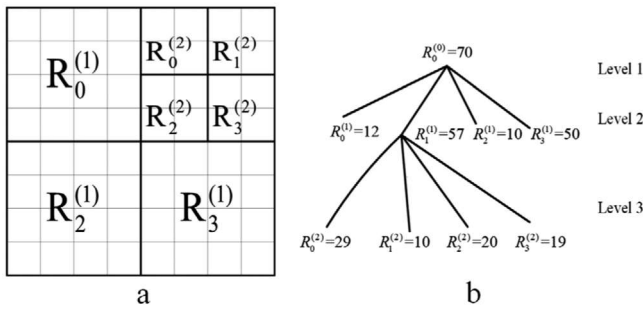


FIGURE 3 Quadtree partition. (a) Partitioned Blocks; (b) corresponding quadtree

1) For a image $R_0^{(0)}$, sort the pixel values to find maximum and minimum values, $R_{\max_{i,j}} = 206$, $R_{\min_{i,j}} = 136$, then calculate with Equation (1): $C(R_0^{(0)}) = R_{\max_{i,j}} - R_{\min_{i,j}} = 70$;

2) According to Equation (2), we find that $C(R_0^{(0)}) \geq 255 \times 0.2$, so divide it into four image blocks, recorded as $R_0^{(1)}$, $R_1^{(1)}$, $R_2^{(1)}$, $R_3^{(1)}$, $C(R_0^{(1)}) = 12$, $C(R_1^{(1)}) = 57$, $C(R_2^{(1)}) = 10$, $C(R_3^{(1)}) = 50$, we find that only $R_1^{(1)}$ needs to be subdivided from the Equation (2);

3) $R_1^{(1)}$ is divided into four image blocks $R_0^{(2)}$, $R_1^{(2)}$, $R_2^{(2)}$, $R_3^{(2)}$ and the complexity of four image blocks is $C(R_0^{(2)}) = 29$, $C(R_1^{(2)}) = 10$, $C(R_2^{(2)}) = 20$, $C(R_3^{(2)}) = 19$ respectively, the complexity is less than 51, so there is no need to subdivide, the method based on complexity partition is completed. Partitioning results as shown in Figure 3.

2.2 | Quad-layered prediction

In this section, we use Quad-layered prediction pattern to make prediction block by block. As show in Figure 4, divide the cover image into four sets [39]: the "Circle" set, the "Triangle" set, the "Square" set, and the "Star" set. Different from other predictions, this prediction incorporates the idea of rhombus prediction [22] and LS [23], using more pixels to make prediction, so

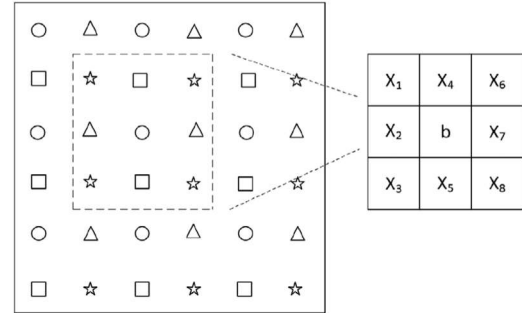


FIGURE 4 Prediction pattern

the first layer, use the surrounding eight pixels from the other three sets to predict the "Circle" set pixel, and then embed the message into PE of the "Circle" set. In the second layer, the modified "Circle" pixels and other surrounding pixel sets will be used to predict the "Triangle" set. After that, the same method will be used to predict and embed the third and fourth layers. Due to the operations in the Quad-layered embedding are basically the same, so we will introduce the prediction on the first layer (the "Circle" set) in detail.

As shown in Figure 4, the "Circle" set pixel $b_{i,j}$ in the cover image are collected as $\mathbf{b} = (b_1; b_2; \dots; b_{N_s})$ from left to right and from top to bottom, where N_s represents the total number of pixels in the "Circle" set of the image block, and the image block is to remove a row (column) on each side based on the cover image block. In order to pursue a sharper PEH, a content pixel predictor is estimated by optimizing the following problem:

$$\begin{aligned} & \text{minimize } \|\mathbf{X}\mathbf{A}^T - \mathbf{b}\| \\ & \text{subject to } \sum_{k=1}^8 A_k = 1 \\ & 0 \leq A_k \leq 1, k = 1, 2, \dots, 8. \end{aligned} \quad (3)$$

The matrix \mathbf{X} and the vector \mathbf{b} are as follows:

$$\mathbf{X} = \begin{bmatrix} X_{11} & X_{12} & \cdots & X_{18} \\ X_{21} & X_{22} & \cdots & X_{28} \\ \vdots & \vdots & \ddots & \vdots \\ X_{N_s1} & X_{N_s2} & \cdots & X_{N_s8} \end{bmatrix} \quad (4)$$

$$\mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{N_s} \end{bmatrix}. \quad (5)$$

In Equation (3), the reason why the L1-norm is used is that the goal of this paper is to optimize the coefficients \mathbf{A} , and the L1-norm is more robust to some outliers. Another advantage of

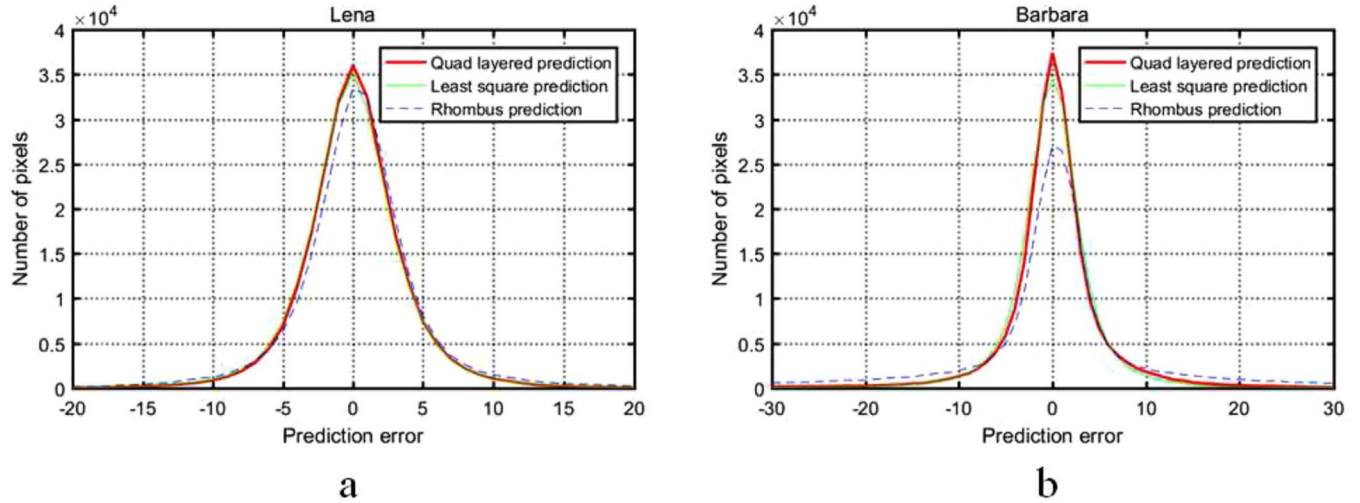


FIGURE 5 Comparison of prediction error histogram performance in “Lena” and “Barbara” images by using Quad-layered prediction, Rhombus prediction and LS prediction. (a) Lena; (b) Barbara

When the prediction coefficients $\mathbf{A} = (A_1, A_2, \dots, A_8)$ is calculated according to Equation (3), and the prediction pixel $\tilde{b}_{i,j}$ is estimated by a weighted sum of eight pixels in the neighbourhood of $b_{i,j}$.

$$\tilde{b}_{i,j} = \sum_{k=1}^8 X_k \cdot A_k. \quad (6)$$

Next, the prediction error $e_{i,j}$ of the current “Circle” set pixel $b_{i,j}$ is expressed as:

$$e_{i,j} = b_{i,j} - \tilde{b}_{i,j}. \quad (7)$$

Simultaneously, the predicted values of all the pixels $\mathbf{b} = (b_1; b_2; \dots; b_{N_s})$ of the “Circle” set in the image block are also obtained by combining the coefficient $\mathbf{A} = (A_1, A_2, \dots, A_8)$ with the surrounding eight pixels. And note that $\mathbf{A} = (A_1, A_2, \dots, A_8)$ remains unchanged in a set.

Generally, Equation (3) is used to make little change to the modification vector \mathbf{b} for the optimized coefficient $\mathbf{A} = (A_1, A_2, \dots, A_8)$, using the modified vector \mathbf{b}' after embedding, Equation (3) is used again to obtain the modified coefficient $\mathbf{A}' = (A'_1, A'_2, \dots, A'_8)$. If we limit the precision of the coefficients to d decimal places. Then, the coefficient residuals vector \mathbf{r} can be expressed as:

$$\mathbf{r} = (\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_8) = \lfloor \mathbf{A} \times 10^d \rfloor - \lfloor \mathbf{A}' \times 10^d \rfloor \quad (8)$$

After getting the coefficient residuals vector, a variable length coding scheme is used to record the coefficient residuals vector. Here, the maximum absolute value of the coefficient residuals

For each coefficient residual r_i , if the r_i equals zero, we use one bit “0” to encode, otherwise we use $t + 2$ bits to encode, where $t + 2$ bits includes a bits “1”, a symbol flag to record the symbol of r_i , and t bits to record the absolute value of r_i . Finally, the encoded bitstream will be embedded as part of the payload. Where the decimal place parameter d and bit length parameter t are set with $d = 2$ and $t = 3$.

Then, the prediction error $e_{i,j}$ is obtained by this method. When the “Circle” set is embedded, the “Triangle” set, the “Square” set, and the “Star” set will continue to be processed in the same way.

To illustrate the performance of the Quad-layered prediction, we compare prediction error histogram in “Lena” and “Barbara” images by using quad-layered prediction, rhombus prediction [22] and LS prediction [23]. The results are shown in Figure 5. It can be observed that the quad-layered prediction displays a better performance than the rhombus prediction [22] and the LS prediction [23].

2.3 | Inter-block sorting and intra-block sorting

Previously, we have partitioned the image according to its complexity, next, the image blocks will be sorted in this section. Sorting is widely applied in RDH, it is proved that this technique can effectively reduce shifting distortion when given an embedding capacity, especially in low embedding capacity. We hope that by sorting, information can be embedded into the smooth region first, and only need to modify some pixels to satisfy the given embedding capacity. Therefore, we introduce inter-block sorting and intra-block sorting.

For inter-block sorting, calculate the complexity of each

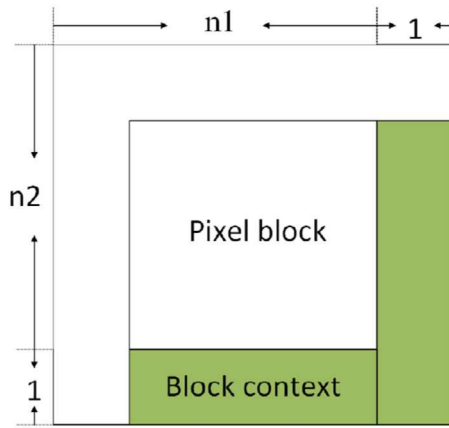


FIGURE 6 Block complexity calculation

shown in Figure 6, which is calculated as the sum of the vertical and the horizontal absolute difference of each two adjacent pixels in the context. Since we do not modify the edge pixels in the prediction and embedding process, therefore, the complexity of image block remains unchanged after data embedding, which ensures the reversibility.

For intra-block sorting, the sorted image blocks are processed sequentially. As shown in Figure 4, we use the method proposed by Sachnev et al. [22] to calculate the local variance $\mu_{i,j}$:

$$\mu_{i,j} = \frac{1}{4} \sum_{k=1}^4 (\Delta X_{(k)} - \Delta \tilde{X}_{(k)})^2 \quad (9)$$

where $\Delta X_{(1)} = |X_{i,j-1} - X_{i-1,j}|$, $\Delta X_{(2)} = |X_{i-1,j} - X_{i,j+1}|$, $\Delta X_{(3)} = |X_{i,j+1} - X_{i+1,j}|$, $\Delta X_{(4)} = |X_{i+1,j} - X_{i,j-1}|$, $\Delta \tilde{X}_{(k)} = (\Delta X_{(1)} + \Delta X_{(2)} + \Delta X_{(3)} + \Delta X_{(4)})/4$.

We arrange the local variances $\mu_{i,j}$ in ascending order when the local variance is obtained. Due to the decoding scheme is inverse of the encoding scheme, the value of local variance $\mu_{i,j}$ remains unchanged after data embedding, the reversibility is guaranteed.

2.4 | Adaptive payload allocation

In general, an image often has different textures, to improve the embedding performance, we assign different sizes of payload to different blocks based on the complexity of the image instead of equally embedding, so how to allocate different sizes of payloads to different blocks is the key problem.

After adaptive partitioning, three types of image blocks are obtained, namely 64×64 , 32×32 , 16×16 , the number of the corresponding blocks is represented as $num64$, $num32$ and $num16$ respectively, the reason for this partition will be explained

and 16×16 respectively.

$$\begin{cases} E_{c64} = EC \times \frac{P_{64}}{P_{64} + P_{32} + P_{16}} \\ E_{c32} = EC \times \frac{P_{32}}{P_{64} + P_{32} + P_{16}} \\ E_{c16} = EC - E_{c64} - E_{c32} \end{cases} \quad (10)$$

$$\begin{cases} P_{64} = x \times num64 \times 64^2 \\ P_{32} = y \times num32 \times 32^2 \\ P_{16} = z \times num16 \times 16^2 \end{cases} \quad (11)$$

In the Equation (11), the coefficients x , y and z represent the embedding proportion of image blocks in different sizes. From the Equations (10) and (11), we can see that $P_{64} : P_{32} : P_{16} = (16x \times num64) : (4y \times num32) : (z \times num16)$, that is, the embedding capacity ratio of each image block in different sizes is $16x : 4y : z$. For embed more information in the smooth regions while satisfying the embedding capacity, set the ranges of x , y , z to $x, y, z \in \{10, 9, 8, 7, 6, 5, 4, 3, 2, 1\}$, and we are adhering to the idea of embedding more information in smooth regions, that is $x > y > z$. However, for different images, image blocks of different sizes will be allocated different embedding capacities, so the optimal x , y , z allocation parameters need to be adaptively determined based on the experimental PSNR.

However, the complexity of image blocks with the same size are still different, especially for a large number of image blocks. Therefore, in order to embed more information in the smooth region, we will further allocate capacity for each sorted image block before embedding. First, divide the capacity E_{c64} , E_{c32} , E_{c16} of each type of image block into two parts: $cap1$ and $cap2$. We know that after inter-block sorting, image blocks will be sorted in ascending order of complexity, so we define the first M image blocks as smoother blocks and allocate $cap1$ embedding capacity, similarly, the rest of the image blocks are complex blocks and we allocate $cap2$ embedding capacity, and then allocate capacity E_{dim}^i , $dim \in \{64, 32, 16\}$ for each image block, i represents the i -th image blocks.

$$\begin{cases} cap1 = \frac{2}{3} E_{cdim} \\ cap2 = E_{cdim} - cap1 \quad dim \in \{64, 32, 16\} \end{cases} \quad (12)$$

$$\begin{cases} \left\lfloor \frac{cap1}{M} \right\rfloor & \text{if } i \leq M \end{cases}$$

According to Equation (13), the value of M is determined by the number of image blocks of each type. In order to achieve the best embedding performance, the final coefficients x, y, z, M must be determined by exhaustive search.

As mentioned in Section 2.1, since the image content is different, the threshold T is different, the result of partitioning is different, and the embedding distortion is different. Therefore, to further improve performance, an adaptive method to choose the optimal T by minimizing the embedding distortion is proposed, as shown in Equation (14).

$$\begin{cases} \text{minimize} & \sum_{i=1}^{\text{num}64} ED_{64}^i + \sum_{i=1}^{\text{num}32} ED_{32}^i + \sum_{i=1}^{\text{num}16} ED_{16}^i \\ \text{subject to} & \sum_{i=1}^{\text{num}64} E_{64}^i + \sum_{i=1}^{\text{num}32} E_{32}^i + \sum_{i=1}^{\text{num}16} E_{16}^i = \text{EC}. \end{cases} \quad (14)$$

Where ED_{dim}^i , $\text{dim} \in \{64, 32, 16\}$ are the corresponding embedding distortion, i represents the i -th image blocks. According to Equation (14), the number of each type of image block greatly affects the embedding result. Therefore, we adopt the adaptive method to find the optimal block-partition method and obtain the most appropriate T by minimizing the embedding distortion.

2.5 | Embedding algorithm

In order to pursue a higher PSNR value, the traditional RDH method based on fixed-size blocks usually leads to a low embedding capacity. It can be found that most of the current methods, such as PEH, PEE or PVO, mainly embed the information in a certain scanning order, without considering the texture information of the image. In fact, the previous methods often embed data into the middle of the PEH, which lead to more shifting distortion in low embedding capacity. Therefore, different from the other RDH methods which embed the data into the middle PEH firstly, we consider embedding on two sides of the PEH because it can effectively reduce shifting distortion, resulting better embedding effect and higher PSNR value in low embedding capacity. This embedding method mainly satisfies two conditions: (1) the bin capacity must satisfy allocated embedding capacity; (2) the bin selected must be in the two sides of PEH.

For the i -th image block, assuming that the embedding capacity is E_{dim}^i , $\text{dim} \in \{64, 32, 16\}$, specific embedding steps are as follows:

- 1) Firstly, the PEH can be easily generated by enumerating the number of PEs that is obtained by the Equation (7). Then express the number of PE = ' f ' as $\text{hist}(f)$ which is the height of bin ' f ';
- 2) Select the most lateral bin with the smallest absolute value as the initial bin, such that Equation (15).

- 3) If the two sides bin cannot reach the E_{dim}^i , we continue to expand to the inside bin to reach it. In order to expand the embedding capacity, we carry out the L round, select the last bins T_p based on the embedding capacity, $T_p \in \{0, 1, 2, \dots, T_m\}$, if f does not appear, we will set it to $\text{hist}(f) = 0$;

$$\left(\sum_{f=-T_p}^{-T_p} \text{hist}(f) + \sum_{f=T_p}^{T_m} \text{hist}(f) \right) > E_{\text{dim}}^i \quad (16)$$

- 4) To get the modified prediction error $e'_{i,j}$, we embed the information according to Equation (17), in which PEs belonging to $e_{i,j} > f$ (or $e_{i,j} < -f$) are shifted to right (left) by a shifting distance s , then data are embedded into the bin $\pm f$ with keeping other bins unchanged.

$$e'_{i,j} = \begin{cases} e_{i,j} + s & \text{if } e_{i,j} > f \\ e_{i,j} - s & \text{if } e_{i,j} < -f \\ e_{i,j} + w & \text{if } e_{i,j} = f \\ e_{i,j} - w & \text{if } e_{i,j} = -f \\ e_{i,j} & \text{otherwise.} \end{cases} \quad (17)$$

Where s is the shifting distances, $e'_{i,j}$ is the modified PEs, and w is the message bit ($w = \{0, 1\}$).

- 5) The marked pixel value $\hat{b}_{i,j}$ expressed by Equation (18).

$$\hat{b}_{i,j} = \tilde{b}_{i,j} + e'_{i,j} \quad (18)$$

For better understand the embedding process, an example is given in Figure 7. Assume that 170 bits pseudo-random sequence are hidden as secret message, and the "0" and "1" have the same probability that is $1/2$. We can see from Figure 7(a), the PEH is generated according to Equation (7); then the initial bins $T_m = 6$ and last bins $T_p = 5$ are obtained based on the given embedding capacity. As shown in Figure 7(b), $\text{PE}_s = \pm 7$ are vacant, according to Equation (17) $\text{PE}_s = \pm 6$ these two bins are used to embed information in Figure 7(b). When the embedding capacity does not meet the preset capacity, we continue to expand inwards. In order to guarantee reversibility, we will move the outer bins to create vacant bin as shown in Figure 7(c). And then, the other data are embedding according to the Equation (17) again as shown in Figure 7(c). Finally, the modified PEH is obtained, as shown in Figure 7(d). From the Figure 7, the PEH is modified two rounds that is $L = 2$. Actually, the capacity E_{dim}^i is proportional to the number of rounds L .

In order to solve the overflow/underflow problem, a location map will be used to deal with pixels that may overflow/underflow, we use "0" to label the value of $0/255$ of the original pixel, and when the modified pixel value is $0/255$, we label it with "1". Then the location map is compressed using the JBIG2 method [41], the compressed location map is denoted as

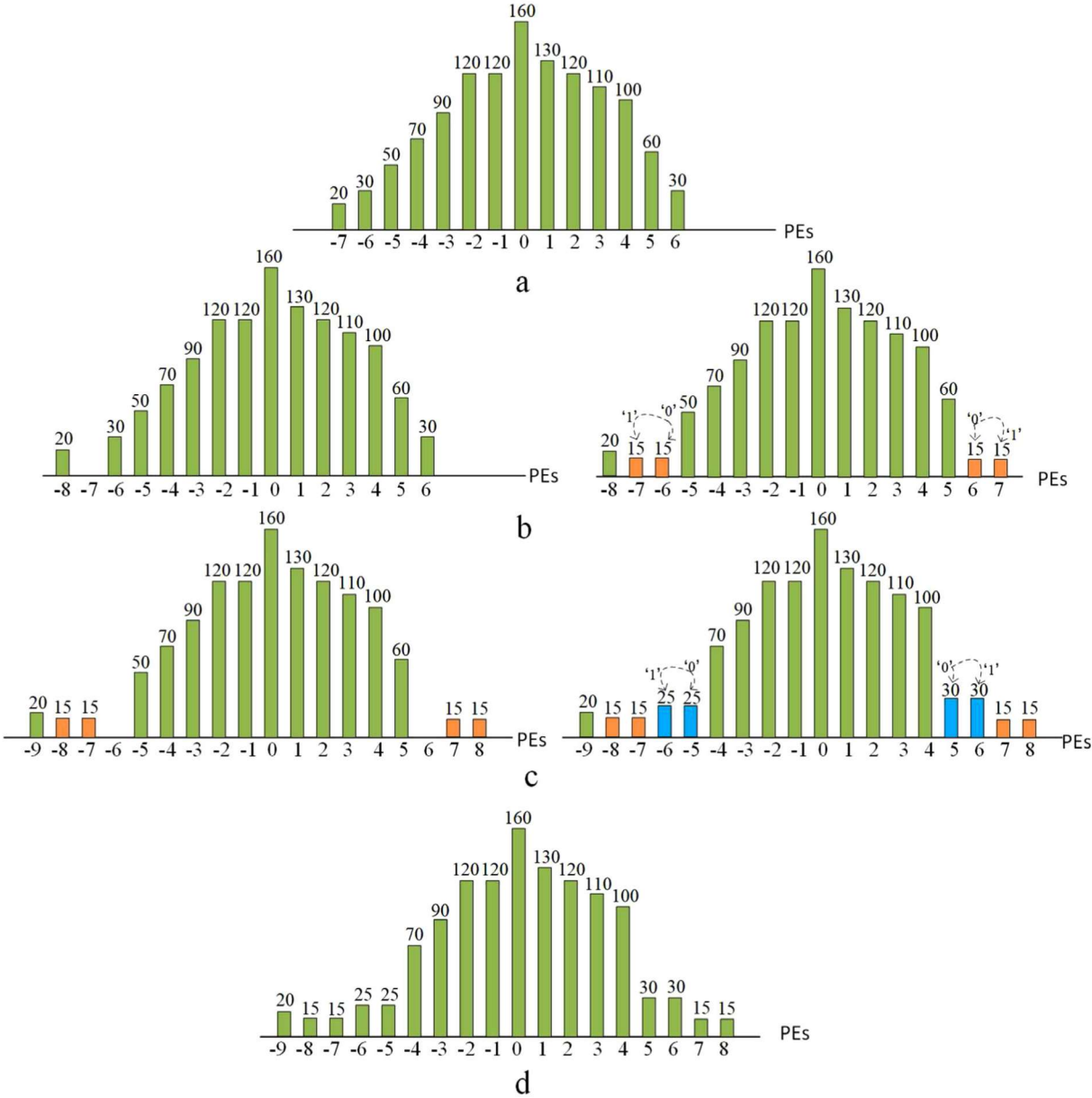


FIGURE 7 An example of two sides of PEH embedding process. (a) Histogram of original prediction errors; (b) first round embedding; (c) second round embedding; (d) modified histogram of prediction errors

into the first 64×64 image block, and record the location of the image block as the LSB's replacement in the first row of the cover image. Simultaneously, the embedding capacity E_{dim}^i , the coefficients x, y, z, M , precision d, t will be replaced as auxiliary information in the first row of the cover image. In addition, the parameter L, r will be used as auxiliary information, therefore, the compressed auxiliary information will replace the LSB of the first row (column) of each image block without first two pixels, the number of LSB planes is instead of the LSB of the first two pixels which have been embedded into image, if there

2.6 | Extraction and recovery processing

After receiving the marked image, data extraction is the reverse process of data embedding. Because the recovery process of the “Star” set, the “Square” set, the “Triangle” set, and the “Circle” set are basically the same, we will take the “Circle” set as an example, and then specific extraction and recovery steps are as follows:

- 1) Read the number of LSB planes from the first two pixels, and then read the first row pixel's LSB planes to get index

该网站存储数据，如为了启用网站基本功能、推广、个性化和分析的cookie。您可随时更改设置或接受默认设置。您可以关闭此横幅以仅继续使用必要的cookie。 [隐私政策](#)

设定偏好

全部接受

全部拒绝

partitioning the image to obtain the same block result as the cover image.

2) Since the edge pixels are not modified, so the block complexity has not changed. After receiving the block complexity in an ascending order, we decode it block by block.

According to Equation (3) obtain the optimal pixel predictor coefficients $\mathbf{A}' = (A'_1, A'_2, \dots, A'_8)$, and combine with the coded residual bit stream \mathbf{r} , recover the original optimal pixel predictor coefficients $\mathbf{A} = (A_1, A_2, \dots, A_8)$. Then obtain the predicted pixel value $\tilde{b}_{i,j}$ of the marked image, the modified prediction error $e'_{i,j}$ is:

$$e'_{i,j} = \tilde{b}_{i,j} - \tilde{b}_{i,j}. \quad (19)$$

Then calculate the local variance $\mu_{i,j}$ by Equation (9), the sorted prediction error $e'_{i,j}$ is obtained.

3) Suppose l is the number of modifications changes from 0 to $L-1$, the modified prediction error goes H from T_p to T_m , decreasing by one each time. At this time, the prediction error $e_{i,j}$ is:

$$e_{i,j} = \begin{cases} e'_{i,j} - l \times s, w = 0 & \text{if } (e'_{i,j} = H + l \times s) \\ e'_{i,j} + l \times s, w = 0 & \text{if } (e'_{i,j} = -H - l \times s) \\ e'_{i,j} - l \times s - 1, w = 1 & \text{if } (e'_{i,j} = H + l \times s + 1) \\ e'_{i,j} + l \times s + 1, w = 1 & \text{if } (e'_{i,j} = -H - l \times s - 1) \\ e'_{i,j} & \text{if } ((e'_{i,j} < H + l \times s) \\ & \text{and } (e'_{i,j} > -H - l \times s)) \end{cases} \quad (20)$$

For the modified prediction errors $e'_{i,j}$ that less than $-T_m$ or larger than T_m , the prediction error $e_{i,j}$ is represented by Equation (21):

$$e_{i,j} = \begin{cases} e'_{i,j} - L \times s & \text{if } e'_{i,j} > T_m + (L-1) \times s + 1 \\ -e'_{i,j} + L \times s & \text{if } e'_{i,j} < -T_m - (L-1) \times s - 1 \end{cases} \quad (21)$$

4) The cover image is recovered to:

$$b_{i,j} = \tilde{b}_{i,j} + e_{i,j}. \quad (22)$$

5) Finally, replace the LSB of the first row (column) pixels by extracted original LSBs, so that the cover image is completely recovered.

3 | EXPERIMENTS AND RESULTS

plane”, “Barbara”, “Boat”, “Elaine”, “Lake”, and “Peppers” are selected as test images are shown in Figure 8. The performance of the proposed method will be demonstrated in the following aspects: (1) In order to explain the effect of threshold T on partition effect, we use different T to introduce partitioning result intuitively; (2) In order to verify the best performance of the adaptive block-partition, we select image blocks of different sizes for comparison; (3) In order to evaluate the proposed method more objectively, we will compare with Weng et al.’s K-PVO [37], Wu et al.’s IPVO-pairwise [38], He et al.’s Muti-pass PVO [36], Weng et al.’s TIPVO [32], Wang et al.’s Dynamic PVO [31], Peng et al.’s IPVO [29], Ou et al.’s PVO-K [30], and Li et al.’s PVO [28] methods.

3.1 | Image partitioning results in different thresholds T

In Section 2, we mentioned that Threshold T plays a crucial role in the partitioning effect, so what does threshold T mean. To show the effect of T , taking “Lena” as an example, four different thresholds T are selected to obtain different partitioning results, as shown in the Figure 9, which can intuitively see the influence of different threshold T .

It can be observed from Figure 9 that different T values lead to different partitioning results. In general, the threshold T is inversely proportional to the number of image blocks after partitioning, the smaller the value of T , the smaller the complexity of the image block is required, that is the value of $C(R_x^{(n)})$ is small, the more image blocks will be obtained after partitioning, such as Figure 9(a). Similarly, with the increases of T , the number of image blocks after partitioning becomes relatively small, as shown in the Figure 9(d).

Previously, we mentioned that the threshold T range is $T \in [0, 1]$, we make a selection of T for each image to ensure that the image block size of each image is within these three types of 64×64 , 32×32 and 16×16 . Then embed the information of each image in an adaptive way, and select the corresponding threshold according to the final embedding distortion. The change range of T of the eight cover images is shown in Figure 10.

3.2 | Experimental results for different block sizes

In the proposed partitioning method, image blocks are divided into three different size categories: 64×64 , 32×32 , 16×16 . The reason why the minimum block is set to 16×16 is that we have to maintain high visual quality and large enough capacity of embedding. If the image block is set too small, too many border pixels cannot be computed in the image block



FIGURE 8 Cover images

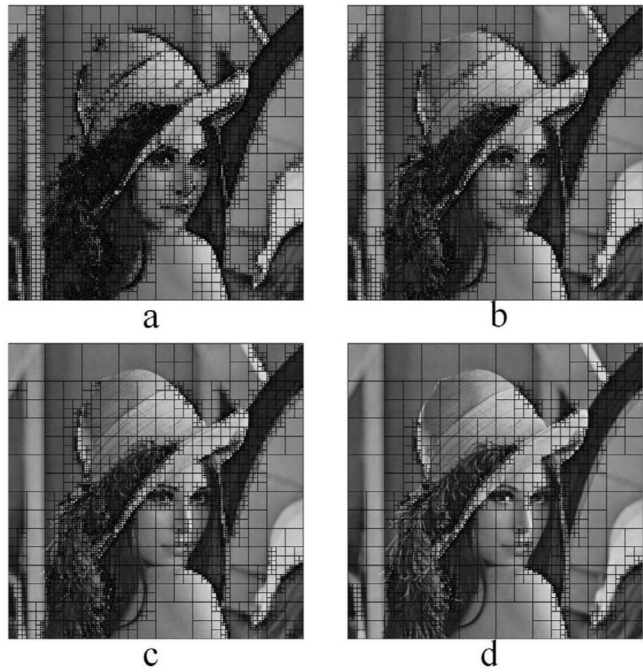


FIGURE 9 Image partitioning results in different thresholds. (a) $T=0.2$; (b) $T=0.3$; (c) $T=0.4$; (d) $T=0.5$

minimum block to be 16×16 , the maximum block to be 64×64 .
To illustrate the advantages of adaptive block-partition, six different partitioning methods are listed, namely 16×16 , 32×32 , 64×64 , 32×32 and 16×16 , 64×64 and 32×32 , and the partition method we proposed. Most of the partition methods compared are equal partition, which are relatively

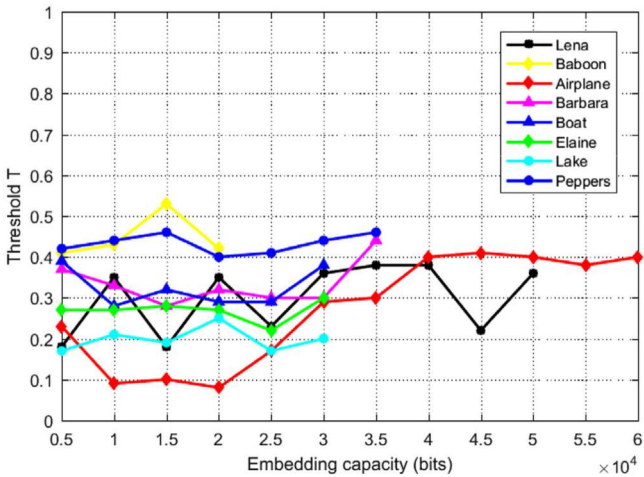


FIGURE 10 Threshold T changes in different images

different numbers of image blocks according to different T , and then given an embedding capacity, we can allocate the embedding capacity reasonably for the image block at each size, and then get the optimal visual quality.

3.3 | Performance comparison

Next, according to the phenomenon in Figure 12, this paper will analyse why the eight methods of PVO have poor performance compared with the proposed method, for the following reasons: Firstly, generally the embedding methods based PVO embed at most 2 bits data in each block, even though

设定偏好

全部接受

全部拒绝

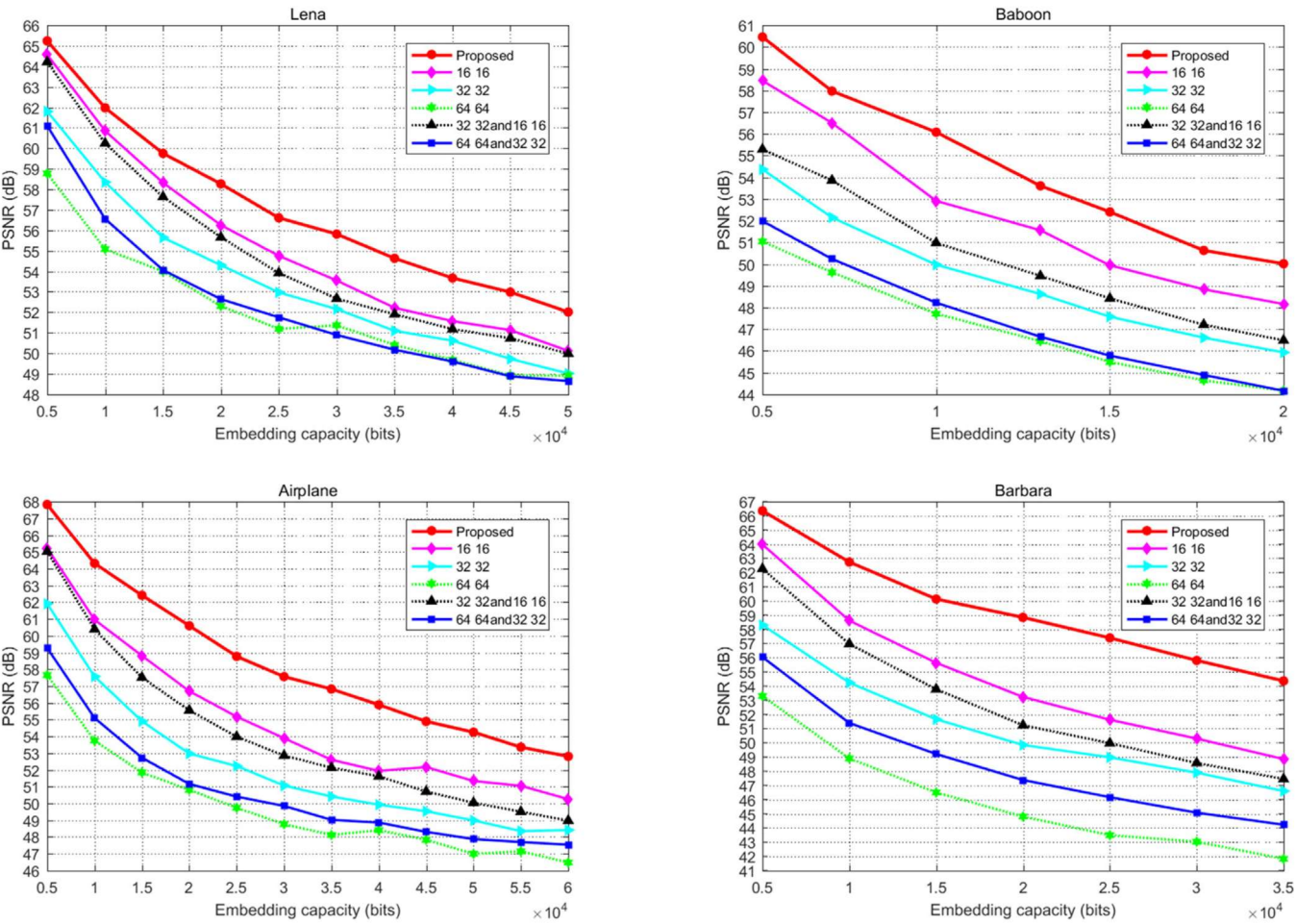


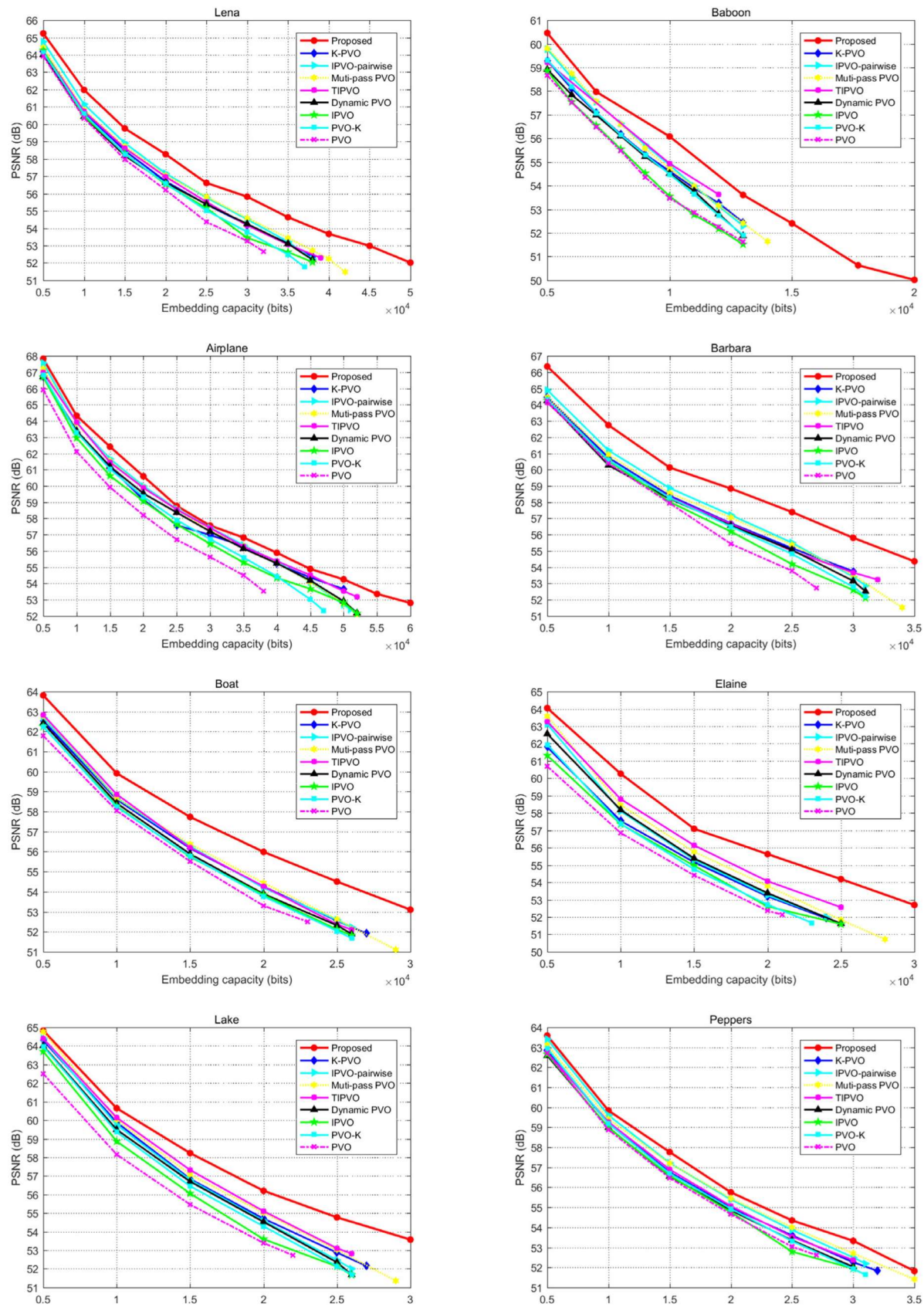
FIGURE 11 Performance comparison for different block sizes

small blocks, but also employ rough blocks for data embedding. However, larger block sizes for higher prediction accuracy, so when the block size is smaller, its prediction accuracy will be greatly reduced. Thirdly, a series of PVO partitioning methods are actually the traditional fixed-sized blocking-based RDH methods. This method does not divide the image reasonably according to the content of the image, and cannot reflect the texture of the image reasonably. It cannot exploit the correlation well between the adjacent pixel values of the image, especially in high embedding capacity. Lastly, most PVO methods embed information on $PE = \pm 1, 0$, for some more complex image blocks, it will lead to more shifting distortion.

The proposed method is better than a series of PVO methods for the following reasons: Firstly, we propose an adaptive block-partition method based on complexity, which can fully consider the characteristics of the image, and divide the image into blocks of different sizes according to the principle of minimum distortion. Secondly, the predictor we use is a linear combination of multiple pixels, which can predict more accurately.

less shifting distortion and greatly improves the visual quality, at the same time, by embedding capacity allocation, the capacity of each image block can be allocated more reasonably, and fully utilize the advantages of the partitioned image block is one of the reasons for maintaining high PSNR.

To further demonstrate the superiority of the proposed method, the average performance of different methods in the same embedding capacity will be compared as shown in Figure 13, and give several tables with the same embedding capacity, such as Tables 1–3. As shown in Figure 12, the embedding capacity starts from 5000 bits, in order to quantify experimental results, three different embedding capacities were selected, namely 5000, 10,000 and 20,000bits, and show the corresponding average PSNR in the table. From the table, it can be seen that the average PSNR of the proposed method is better than that of other methods regardless of the embedding capacity. From Tables 1 and 2, when the EC is 5000 and 10,000 bits respectively, the average PSNR of the proposed method is 64.52, 60.72 dB. In particular, for “Baboon”, we can see clearly from



本网站存储数据，如为了启用网站基本功能、推广、个性化和分析的cookie。您可随时更改设置或接受默认设置。您可以关闭此横幅以仅继续使用必要的cookie。 [隐私政策](#)

设定偏好

全部接受

全部拒绝

TABLE 1 Comparison of PSNR (in dB) between Weng et al.'s K-PVO [37], Wu et al.'s IPVO-pairwise [38], He et al.'s Muti-pass PVO [36], Weng et al.'s TIPVO [32], Wang et al.'s Dynamic PVO [31], Peng et al.'s IPVO [29], Ou et al.'s PVO-K [30], and Li et al.'s PVO [28] methods and the proposed method for an EC of 5000 bits

Image	Lena	Baboon	Airplane	Barbara	Boat	Elaine	Lake	Peppers	Average
Proposed	65.24	60.46	67.82	66.35	63.81	64.05	64.82	63.58	64.52
K-PVO	64.19	59.29	66.72	64.47	62.54	61.80	64.30	62.86	63.27
IPVO-pairwise	64.76	59.77	67.56	64.90	62.70	63.12	64.39	63.41	63.83
Muti-pass PVO	64.42	59.80	67.22	64.45	62.78	63.58	64.76	63.12	63.77
TIPVO	64.23	59.22	66.98	64.13	62.83	63.27	64.39	62.73	63.47
Dynamic PVO	64.00	58.91	66.71	64.24	62.42	62.56	64.00	62.59	63.18
IPVO	64.25	58.86	66.70	64.27	62.25	61.32	63.70	62.66	63.00
PVO-K	64.14	59.32	66.81	64.30	62.25	61.94	63.97	62.95	63.21
PVO	63.92	58.67	65.91	64.22	61.79	60.69	62.50	62.72	62.55

TABLE 2 Comparison of PSNR (in dB) between Weng et al.'s K-PVO [37], Wu et al.'s IPVO-pairwise [38], He et al.'s Muti-pass PVO [36], Weng et al.'s TIPVO [32], Wang et al.'s Dynamic PVO [31], Peng et al.'s IPVO [29], Ou et al.'s PVO-K [30], and Li et al.'s PVO [28] methods and the proposed method for an EC of 10,000 bits

Image	Lena	Baboon	Airplane	Barbara	Boat	Elaine	Lake	Peppers	Average
Proposed	61.98	56.08	64.32	62.74	59.92	60.26	60.64	59.84	60.72
K-PVO	60.71	54.61	63.30	60.74	58.61	57.54	59.89	59.18	59.32
IPVO-pairwise	61.15	54.87	63.88	61.19	58.68	58.11	59.75	59.61	59.65
Muti-pass PVO	60.82	54.72	63.85	60.92	58.67	58.46	60.03	59.42	59.61
TIPVO	60.74	54.93	63.92	60.57	58.87	58.80	60.14	59.27	59.66
Dynamic PVO	60.44	54.52	63.39	60.27	58.42	58.18	59.50	58.99	59.21
IPVO	60.47	53.55	62.96	60.55	58.27	57.36	58.87	58.98	58.88
PVO-K	60.59	54.48	63.29	60.59	58.23	57.37	59.37	59.19	59.14
PVO	60.34	53.47	62.10	60.39	58.05	56.85	58.15	58.88	58.53

“Baboon” is complex, it cannot exploit the advantages of pixel correlation well. In addition, due to the smooth image block is not enough, it’s necessary to use appropriate number of rough blocks to embed data, which will lead to inaccurate prediction. As can be seen from Table 3, the average PSNR of the proposed method is written as 57.33/56.41 dB, where the result on the left is counting “Baboon”, and the result on the right is excluding “Baboon”. In order to show the influence of auxiliary information on the embedding capacity more intuitively, this paper investigate the auxiliary information of different embedding capacities. As shown in Table 4, we enumerate the auxiliary information capacity of eight test images. It can be seen from the Table 4 that the auxiliary information of the proposed method will not increase as the embedding capacity increases, which shows that the proposed method can achieve better embedding performance as the embedding capacity increases. This is because

capacity, large-size image blocks will be added, so the number of image blocks will be reduced. When the embedding capacity increases and the number of image blocks decreases, the number of embedding rounds L and image block index B will decrease, and as the embedding capacity increases, the value of the residual coefficient r will increase. Therefore, regardless of the capacity of the embedded information, the capacity of the auxiliary information does not fluctuate greatly.

4 | CONCLUSION

In this paper, a novel reversible data hiding based on adaptive block-partition and payload-allocation method is proposed. In contrast with the traditional fixed-sized blocking-based RDH methods, an adaptive block-partition based on complexity is taken into account. In addition, by designing a new embedding

设定偏好

全部接受

全部拒绝

TABLE 3 Comparison of PSNR (in dB) between Weng et al.'s K-PVO [37], Wu et al.'s IPVO-pairwise [38], He et al.'s Muti-pass PVO [36], Weng et al.'s TIPVO [32], Wang et al.'s Dynamic PVO [31], Peng et al.'s IPVO [29], Ou et al.'s PVO-K [30], and Li et al.'s PVO [28] methods and the proposed method for an EC of 20,000 bits

Image	Lena	Baboon	Airplane	Barbara	Boat	Elaine	Lake	Peppers	Average
Proposed	58.27	50.02	60.60	58.84	55.99	55.63	56.20	55.75	57.33/56.41
K-PVO	56.71	—	59.16	56.71	54.26	53.19	54.71	54.99	55.68
IPVO-pairwise	57.16	—	59.99	57.21	54.19	53.26	54.59	55.40	55.97
Muti-pass PVO	57.05	—	59.91	57.06	54.43	53.79	55.02	55.45	56.10
TIPVO	56.97	—	59.88	56.67	54.27	54.07	55.11	55.07	56.00
Dynamic PVO	56.63	—	59.54	56.57	53.89	53.38	54.53	54.86	55.63
IPVO	56.54	—	59.07	56.20	53.84	52.61	53.60	54.77	55.23
PVO-K	56.58	—	59.34	56.50	53.76	52.72	54.30	54.94	55.45
PVO	56.21	—	58.20	55.44	53.30	52.37	53.40	54.66	54.80

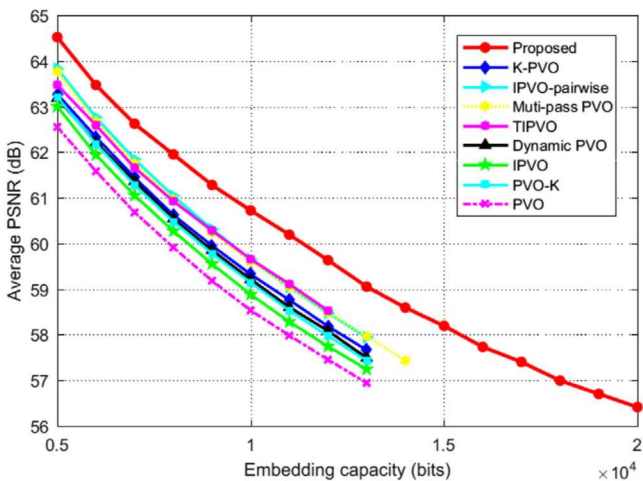


FIGURE 13 Comparison of the average performance of different methods in the same embedding capacity

TABLE 4 The influence of auxiliary information on the embedding capacity of eight test images

Payloads(bits)	10,000	15,000	20,000	25,000	30,000
Lena	2182	2462	2420	2422	2276
Baboon	2149	2049	2173	—	—
Airplane	2477	2370	2564	2497	2399
Barbara	2354	2745	2684	3097	3222
Boat	2209	2494	2484	2702	2845
Elaine	2168	2120	2318	2252	2431
Lake	2272	2176	2514	2615	2703
Peppers	2491	2582	3165	3252	3246

ACKNOWLEDGMENTS

This work was supported in part by the Natural Science Foundation of China under Grants 61502007 and 61572452, the Natural Science Research Project of Anhui province under Grant 1608085MF125, the NO.58 China Post-doctoral Science Foundation under Grant 2015M582015, the Backbone Teacher Training Program of Anhui University and the Doctoral Scientific Research Foundation of Anhui University under Grant J01001319.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

CONFLICT OF INTEREST

No conflict of interest exists in the submission of this manuscript, and manuscript is approved by all authors for publication.

PERMISSION STATEMENT FOR REPRODUCING THE MATERIALS LIKE INDIVIDUAL'S PHOTOGRAPH OR TRADEMARK BRAND

Reproduction the materials like individual's photograph or trademark brand is permitted.

ORCID

Yang Yang  <https://orcid.org/0000-0003-1048-7994>

REFERENCES

1. Cox, I., et al.: Digital Watermarking and Steganography. Morgan kaufmann, Burlington, MA (2007)
2. Khan, A., et al.: A recent survey of reversible watermarking techniques. *Inf Sci* 370, 251–272 (2016)

4. Zhang, X., Zhang, W.: Semantic image compression based on data hiding. *IET Image Proc.* 9(1), 54–61 (2015)
5. Celik, M.U., et al.: Lossless generalized-LSB data embedding. *IEEE Trans. Image Process.* 14(2), 253–266 (2005)
6. Chang, C.-C., et al.: Reversible hiding in DCT-based compressed images. *Inf. Sci.* 177(13), 2768–2786 (2007)
7. Yao, H., et al.: Improved dual-image reversible data hiding method using the selection strategy of shiftable pixels coordinates with minimum distortion. *Signal Process.* 135, 26–35 (2017)
8. Fallahpour, M.: High capacity lossless data hiding based on histogram modification. *IEICE Electronics Express* 7(4), 205–210 (2007)
9. Van Leest, A., et al.: Reversible image watermarking. *Proceedings 2003 International Conference on Image Processing*, 731–734. IEEE, Piscataway, NJ (2003)
10. Jia, Y., et al.: Reversible data hiding based on reducing invalid shifting of pixels in histogram shifting. *Signal Process.* 163, 238–246 (2019)
11. Tian, J.: Reversible data embedding using a difference expansion. *IEEE Trans. Circuits Syst. Video Technol.* 13(8), 890–896 (2003)
12. Shah, M., et al.: Prediction error expansion-based reversible data hiding in encrypted images with public key cryptosystem. *IET Image Proc.* 13(10), 1705–1713 (2019)
13. Xiao, M., et al.: Reversible data hiding based on pairwise embedding and optimal expansion path. *Signal Process.* 158, 210–218 (2019)
14. Wang, J., et al.: Multiple histograms based reversible data hiding by using FCM clustering. *Signal Process.* 159, 193–203 (2019)
15. Thodi, D.M., Rodríguez, J.J.: Expansion embedding techniques for reversible watermarking. *IEEE Trans. Image Process.* 16(3), 721–730 (2007)
16. Hu, Y., et al.: DE-based reversible data hiding with improved overflow location map. *IEEE Trans. Circuits Syst. Video Technol.* 19(2), 250–260 (2008)
17. Wang, S.Y., et al.: Reversible data hiding based on two-dimensional prediction errors. *IET Image Proc.* 7(9), 805–816 (2013)
18. Gao, X., et al.: Lossless data embedding using generalized statistical quantity histogram. *IEEE Trans. Circuits Syst. Video Technol.* 21(8), 1061–1070 (2011)
19. Hong, W., et al.: Reversible data hiding for high quality images using modification of prediction errors. *J. Syst. Softw.* 82(11), 1833–1842 (2009)
20. Wu, X., Memon, N.: Context-based adaptive lossless image code. *IEEE Trans. Commun.* 5(45), 437–444 (1997)
21. Luo, L., et al.: Reversible image watermarking using interpolation technique. *IEEE Trans. Inf. Forensics Secur.* 5(1), 187–193 (2009)
22. Sachnev, V., et al.: Reversible watermarking algorithm using sorting and prediction. *IEEE Trans. Circuits Syst. Video Technol.* 19(7), 989–999 (2009)
23. Dragoi, I.-C., Coltuc, D.: Local-prediction-based difference expansion reversible watermarking. *IEEE Trans. Image Process.* 23(4), 1779–1790 (2014)
24. Ni, Z., et al.: Reversible data hiding. *IEEE Trans. Circuits Syst. Video Technol.* 16(3), 354–362 (2006)
25. Fallahpour, M., et al.: Reversible and high-capacity data hiding in medical images. *IET Image Proc.* 5(2), 190–197 (2011)
26. Lee, S.-K., et al.: Reversible image authentication based on watermarking. *2006 IEEE International Conference on Multimedia and Expo*, pp. 1321–1324. IEEE, Piscataway, NJ (2006)
27. Wang, J., et al.: Multiple Histograms Based Reversible Data Hiding: Framework and Realization. *IEEE Trans. Circuits Syst. Video Technol.* (99), 1–1 (2019)
28. Li, X., et al.: High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion. *Signal Process.* 93(1), 198–205 (2013)
29. Peng, F., et al.: Improved PVO-based reversible data hiding. *Digital Signal Process.* 25, 255–265 (2014)
30. Ou, B., et al.: Reversible data hiding using invariant pixel-value-ordering and prediction-error expansion. *Signal Process. Image Commun.* 29(7), 760–772 (2014)
31. Wang, X., et al.: A novel reversible image data hiding scheme based on pixel value ordering and dynamic pixel block partition. *Inf. Sci.* 310, 16–35 (2015)
32. Weng, S., et al.: Reversible data hiding based on flexible block-partition and adaptive block-modification strategy. *J. Visual Commun. Image Represent.* 41, 185–199 (2016)
33. Li, J., et al.: Generalized PVO-k embedding technique for reversible data hiding. *IJ Network Security* 20(1), 65–77 (2018)
34. Wu, H., et al.: Improved PPVO-based high-fidelity reversible data hiding. *Signal Process.* 167, 107264 (2020)
35. Weng, S., et al.: Dynamic improved pixel value ordering reversible data hiding. *Inf. Sci.* 489, 136–154 (2019)
36. He, W., et al.: Reversible data hiding using multipass pixel-value-ordering and pairwise prediction-error expansion. *Inf. Sci.* 467, 784–799 (2018)
37. Weng, S., et al.: Improved k-pass pixel value ordering based data hiding. *IEEE Access* 7, 34570–34582 (2019)
38. Wu, H., et al.: Improved reversible data hiding based on PVO and adaptive pairwise embedding. *Journal of Real-Time Image Processing* 16(3), 685–695 (2019)
39. Hu, X., et al.: Optimizing pixel predictors based on self-similarities for reversible data hiding. *2014 Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pp. 481–484. IEEE, Piscataway, NJ (2014)
40. Gui, X., et al.: A high capacity reversible data hiding scheme based on generalized prediction-error expansion and adaptive embedding. *Signal Process.* 98, 370–380 (2014)
41. Howard, P.G., et al.: The emerging JBIG2 standard. *IEEE Trans. Circuits Syst. Video Technol.* 8(7), 838–848 (1998)
42. Liu, Y., et al.: A ROI-based reversible data hiding scheme in encrypted medical images. *J. Visual Commun. Image Represent.* 39, 51–57 (2016)

How to cite this article: Yang Y, Genyan H, Tianrui Z, Weiming Z. A novel reversible data hiding based on adaptive block-partition and payload-allocation method. *IET Image Process.* 2021;15:571–585.
<https://doi.org/10.1049/ipr2.12050>

该网站存储数据，如为了启用网站基本功能、推广、个性化和分析的cookie。您可随时更改设置或接受默认设置。您可以关闭此横幅以仅继续使用必要的cookie。 [隐私政策](#)

设定偏好

全部接受

全部拒绝