IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 26, 2019, accepted September 9, 2019, date of publication September 16, 2019, date of current version October 11, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2941500

Reversible Data Hiding Scheme Based on Quad-Tree and Pixel Value Ordering

JIAN-JUN LI[1], CHIN-FENG LEE[2], CHIN-CHEN CHANG[3], (Fellow, IEEE), JIANG-YI LIN[3,4], AND YUN-HE WU[1]
[1]Department of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China
[2]Department of Information Management, Chaoyang University of Technology, Taichung 41349, Taiwan
[3]Department of Information Engineering and Computer Science, Feng Chia University, Taichung 40724, Taiwan
[4]School of Computer and Information Engineering, Xiamen University of Technology, Xiamen 361024, China

Corresponding author: Chin-Chen Chang (alan3c@gmail.com)

ABSTRACT This study proposes a reversible data hiding (RDH) scheme based on pixel value ordering (PVO). RDH schemes based on PVO technique generally realize the prediction procedure by employing the sorted pixels in a block-wise manner, and then embedding secret data by expanding the prediction errors. Since these operations are carried out based on the block, and pixel intensity has spatial correlation such that the neighboring pixels have similar values, the block patterns and sizes certainly affect the embedding performance of the PVO-based RDH method. The original PVO-based RDH schemes use fixed block size. However, due to the differences in image complexity, fixed-size blocks have many limitations. Thus far, several methods that break the limitations have been proposed, such as setting a threshold value, and determine the size from the complexity of the block, using sliding window instead of fixed blocks. These methods significantly enhanced the embedding performance of PVO-based RDH schemes, but this can still be improved. The proposed scheme introduces the quad-tree structure, which combines the advantages of several previously proposed methods to design a flexible block patterns, thus fully utilizing the characteristics of the image itself in embedding secret data. Experimental results indicate that the proposed scheme is flexible and better than previous approaches.

INDEX TERMS Pixel value ordering, prediction error expansion, reversible data hiding, quad-tree.

I. INTRODUCTION

Data hiding refers to concealing data for secret data communication or information authentication. Image data hiding technology conceals messages into meaningful images, referred to as cover images, without giving attention to malicious individuals. Image data hiding focuses on embedding as large payload and as high image quality as possible after messages are hidden. However, the large embedding capacity and high image quality create a trade-off situation, since a large payload always incurs serious image distortion.

Reversible data hiding technology completely recovers the original cover image after the embedded secret data have been exactly extracted. This no-distortion data hiding technology is a widespread research topic in the academic field of data hiding, as well as attracting interest from scholars in the fields of medical image and military image processing.

Many RDH schemes have been proposed in recent years. The two most representative reversible data hiding approaches are difference expansion (DE) and histogram shifting (HS). DE schemes utilize the correlation between pixels to derive a difference between adjacent pixels, or design a prediction method to obtain predicted pixel values, and then calculate the differences between the original pixel values and the predicted pixel values. The final purpose is to expand these error values to embed secret data. Tian [1] presented the original DE method, which expands differences between the adjacent pixels to embed secret data. This method needs to establish a location map to indicate the expandable and un-expandable pixels for fixing the over/under-flow problem. Because the location map introduces additional information overhead, the embedding capacity is fairly low. To enhance this method, some proposed

The associate editor coordinating the review of this manuscript and approving it for publication was Malik Jahan Khan.

schemes [2]–[4] aim to reduce the amount of extra information, by shrinking the location map. Several RDH schemes based on pixel value prediction error expansion (PEE) [5]–[8] have also been proposed. Those methods all predict the pixel value, and then adopt the prediction error, rather than the difference between pixels, to embed secret data. Whether to reduce the size of the location map or to embed secret data based on PEE, the performances of the original DE method are significantly improved.

Ni *et al.* developed another RDH technology, histogram shifting (HS), in 2006 [9]. The HS method modifies each pixel value by at most 1. Therefore, the camouflage images in RDH schemes based on HS typically have very high quality. In the method of Ni *et al.* [9], a histogram of the pixel values is established first, and then the pixel values with most/least frequency are identified as peak/zero points. The pixel values are then moved between the peak and zero points towards zero points by one space, thus providing a place beside the peak point for embedding secret data. However, since the amount of embedded secret data is directly determined by the number of peak pixels, the embedding capacity of the HS method is always low for natural images. Accordingly, several improved HS methods [10]–[14] have been proposed, and are principally aimed at raising the number of peak points in the histogram. Among them, Hong *et al.* [12] (2009) generated the histogram using the prediction error instead of the original pixel value, as in the approach of Ni *et al.* [9], and thus significantly improved the embedding capacity. Because most of the pixel values of the prediction error histogram are "0", the schemes in [13]–[14] have more accurate prediction methods than [12], and thus have improved performance. In addition, sorting [15] or pixel selection [16] can also improve performance. These methods preferentially deal with the prediction error in smooth regions, as this can be accurately predicted, thus creating more histogram peaks in such regions. Embedding secret data in these smooth areas with higher priority can reduce unnecessary modification of pixel values, also reducing distortion of the camouflage images.

Li *et al.* (2013) [17] proposed a novel RDH scheme based on pixel value ordering (PVO), which divides the image into blocks for embedding secret data. First, the pixel values in each block are ordered, then, only maximum value and minimum value can be predicted, enabling secret data to be embedded into them. The prediction error is derived as the difference between the maximum value and the second-largest value, or between the minimum value and the second-smallest value. The sorted pixel values in the prediction procedure demonstrate that the Li *et al.* method has much better prediction accuracy than previously proposed methods, such as MED (median-edge-detector) [18]–[19] and GAP (gradient-adjusted-predictor) [20]–[21]. This finding also implies that the quality of the camouflaged images in [18]–[21] would be significantly improved. The RDH scheme in [17], chooses the maximum/minimum pixel values to embed the secret information only when the prediction errors are "1" or "-1", and thus ignores the "0" prediction errors. However, "0" prediction errors are very common in natural images. Therefore, Peng *et al.* [22] developed an enhanced PVO scheme (called IPVO), which first tags each pixel with a number corresponding to its position in that block. The numbered and sorted pixels are taken to be predicted, meaning that the prediction error "0" can also be employed in the embedding phase. Additionally, Ou *et al.* [23] proposed an extension of PVO named PVO-K. Their approach takes K maximum values or K minimum values as an entire unit to deal with, and also significantly enhances the performance of the PVO method by adjusting the size of K. As can be seen, the PVO scheme is a special case of the PVO-K scheme when $K = 1$. In 2018, Li *et al.* proposed a generalized PVO-K method (GePVO-K) [24]. Unlike the PVO-K method, in a block only one bit can be hidden at a time on K consecutive maximum or minimum pixel values. GePVO-K method can embed K-bit secret data into K consecutive maximum or minimum pixel values so that to take advantage of pixels with maximum or minimum values to increase payload.

Because the PVO-based schemes consider a block as a unit to predict and embed secret data, the block size directly affects the prediction accuracy and the embedding capacity. The above methods always adopt the block with fixed size, so that the smooth region cannot be fully used to conceal information. Therefore, Wang *et al.* [25] presented a novel block division method, in which a 4×4 block may be divided into four 2×2 blocks depending on the block complexity. Consequently, the performances in terms of embedding capacity and image quality are significantly improved. The PVO, IPVO, PVO-K, and Wang *et al.'s* schemes hide data in a block-wise manner. However, the pixel-based pixel value ordering (PPVO) method of Qu and Kim [26] applies a sliding window rather than a block to predict and embed secret data, thus breaking the block constraints whereby only two bits of secret data can be embedded into each block, and significantly improving the embedding capacity of the PVO-based schemes, particularly for relatively smooth images. However, using the sliding window on cover images with complex regions may negatively affect the quality of the camouflaged images owing to the reduced prediction accuracy, because it may modify too many pixel values.

The PVO-based information hiding method has been increasingly investing in research in recent years. Lu et al. (2018) [27] design an adaptive threshold generation mechanism for different images to achieve high image quality. Lee *et al.* [28] proposed the "star-shaped" PVO method, which fully exploits the similarity of adjacent pixels in each 3×3 block to maximize the embedding of secret data. The advantage of this method is that even when performing multiple layers of embedding. It also maintains proper image quality. Not only does it have excellent embedding capacity in smooth images, but its embedded capacity is superior to previous PVO series methods even for complex images. Li *et al.* [29] introduced a dynamic pixels-value-ordering

(D-PVO) method that predicts the cover pixels by sorting their cross-over neighbors and exploits two end pixels or median pixels to perform the prediction. Data are hidden using the prediction error expansion to achieve higher fidelity of the stego-images. Weng et al (2019) [30] proposed an improved k-pass PVO by considering the location relationship. Some of the pixels in the block are considered along with the neighborhood around the block to increase the estimation accuracy of the local complexity. In addition, three largest and three smallest pixels are involved in data hiding to improve the embedding capacity. Lee *et al.* [31] proposed an overlapping PVO (OPVO) method. The OPVO method takes full advantage of the correlation between adjacent pixels in a natural image to achieve that the embedding capacity is more than twice that of previous PVO series methods.

This work proposes a RDH scheme based on quad-tree and pixel value ordering, which dynamically determines the size of the partition block and selects the appropriate embedding PVO- based method according to the complexity of the image. The proposed scheme achieves maximum efficiency in the embedding procedure flexibly in the disassembly and integration according to the characteristics of the cover image. In continuous smooth areas, sliding windows of PPVO are adopted to embed secret data; complex areas either have data embedded in large blocks or are skipped. In order to achieve the best performance, the proposed scheme utilizes the flexibility of the quad-tree structure, combined with several advantages of PVO-based schemes, to maximize the utilization of the image characteristics.

The remaining parts of this paper are arranged as follows. In Section 2, several PVO-based RDH schemes are briefly introduced. Section 3 elaborates the details of the proposed scheme, and presents the steps of the data embedding and extraction procedures and their flowchart. Section 4 describes the experimental results and analyses, and compares several PVO-based approaches with the proposed scheme. The conclusions of this paper are presented in Section 5.

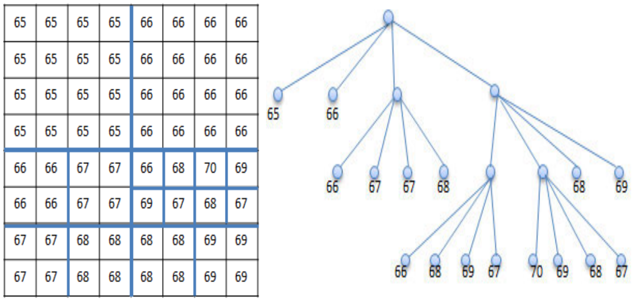
II. RELATED WORKS

This section first introduces the data structure of quad tree, and an improved PVO-based RDH method, IPVO [22]. Two schemes that improve the PVO block division method, Wang *et al.*'s scheme [25] and PPVO [26], are then briefly introduced.

A. QUAD-TREE STRUCTURE

Quad-tree is a tree data structure. Its basic principle is recursive information decomposition, with each node having at most four child nodes. For instance, the two-dimensional spatial is generally subdivided to four quadrants or regions, and the information in each area is stored in the quad-tree nodes. The common usage of quad-tree is image segmentation, which can decompose the $2^n \times 2^n$-sized image into four equal-sized sub images, each of which is marked as a node. If the node does not fulfill the decomposition conditions, then it is marked as a leaf node; if it does satisfy the given conditions, then it is further divided into four child nodes. The division is performed recursively in accordance with the above rules for division, until none of the leaf nodes can be further decomposed.

The most commonly used decomposition condition for image segmentation based on the quad-tree is that the pixel values in the current node are exactly the same. For instance, Fig. 1 shows the image segmentation process using quad-tree structure, assuming that the image size is 8×8. First of all, check whether all pixel values are identical. As the pixel values are obviously not all the same, divide the original image into four equal-sized sub-images, and then check each sub-image for identical pixel values, until each node has the same pixel values. Finally, a quad-tree structure of the original image is obtained. Notably, the decomposition conditions can be determined from different requirements to attain the desired segmentation results.



65	65	65	65	66	66	66	66
65	65	65	65	66	66	66	66
65	65	65	65	66	66	66	66
65	65	65	65	66	66	66	66
66	66	67	67	66	68	70	69
66	66	67	67	69	67	68	67
67	67	68	68	68	68	69	69
67	67	68	68	68	68	69	69

FIGURE 1. **An example of quad-tree construction.**

B. IPVO METHOD

The RDH scheme proposed by Li *et al.* [17] provides a novel prediction method, which adopts the ordered sequences of pixel values in prediction procedure, and significantly improves the prediction accuracy compared to the HS method. However, it can embed secret data only when the prediction errors are 1's or -1's, and disregards the other regularly appearing prediction errors 0's. Peng *et al.* [22] proposed an improved scheme, named IPVO, to tackle this shortcoming.

Like PVO, IPVO also first divides the cover image into blocks. For each block, the pixels are numbered in a zigzag manner, and then sorted in ascending order, assuming that the results are $B\{x_{\pi(1)}, x_{\pi(2)}, \ldots, x_{\pi(n_1 \times n_2)}\}$. The prediction errors of the maximum and minimum value are calculated as Formula (1).

$$\begin{cases} d_{\max} = x_{u_1} - x_{v_1} \\ d_{\min} = x_{u_2} - x_{v_2}, \end{cases} \tag{1}$$

where

$$\begin{cases} u_1 = \min(\pi(n1 \times n2), \pi(n1 \times n2\text{-}1)) \\ v_1 = \max(\pi(n1 \times n2), \pi(n1 \times n2\text{-}1)) \\ u_2 = \min(\pi(1), \pi(2)) \\ v_2 = \max(\pi(1), \pi(2)). \end{cases} \tag{2}$$

After the prediction errors are obtained, the secret data are embedded using the HS method [9]. If the calculated prediction errors are 0's or 1's, then secret data $b \in \{0, 1\}$ can be embedded. Otherwise, the prediction error values need to be moved to provide the embedded space. If the prediction errors are greater than 1, then are moved to the right by one space; if the prediction errors are less than 0, then are moved to the left by one space. In summary, the prediction errors of the maximum and minimum values can be adjusted based on Formula (3) and Formula (4).

$$d'_{\max} = \begin{cases} d_{\max} + b & \mathit{if}\ d_{\max} = 1 \\ d_{\max} + 1 & \mathit{if}\ d_{\max} > 1 \\ d_{\max} - b & \mathit{if}\ d_{\max} = 0 \\ d_{\max} - 1 & \mathit{if}\ d_{\max} < 0. \end{cases} \tag{3}$$

$$d'_{\min} = \begin{cases} d_{\min} + b & \mathit{if}\ d_{\min} = 1 \\ d_{\min} + 1 & \mathit{if}\ d_{\min} > 1 \\ d_{\min} - b & \mathit{if}\ d_{\min} = 0 \\ d_{\min} - 1 & \mathit{if}\ d_{\min} < 0. \end{cases} \tag{4}$$

Finally, the maximum and minimum pixel values are adjusted according to the prediction errors. The specific modification method is presented in Formula (5) and Formula (6).

$$\begin{aligned} x'_{\pi(n1\times n2)} &= x_{\pi(n1\times n2)} + \left|d'_{\max}\right| \\ &= \begin{cases} x_{\pi(n1\times n2)} + b & \mathit{if}\ d_{\max} = 1 \\ x_{\pi(n1\times n2)} + 1 & \mathit{if}\ d_{\max} > 1 \\ x_{\pi(n1\times n2)} + b & \mathit{if}\ d_{\max} = 0 \\ x_{\pi(n1\times n2)} + 1 & \mathit{if}\ d_{\max} < 0. \end{cases} \end{aligned} \tag{5}$$

$$\begin{aligned} x'_{\pi\ (1)} &= x_{\pi\ (1)} - \left|d'_{\min}\right| \\ &= \begin{cases} x_{\pi\ (1)} - b & \mathit{if}\ d_{\min} = 1 \\ x_{\pi\ (1)} - 1 & \mathit{if}\ d_{\min} > 1 \\ x_{\pi\ (1)} - b & \mathit{if}\ d_{\min} = 0 \\ x_{\pi\ (1)} - 1 & \mathit{if}\ d_{\min} < 0. \end{cases} \end{aligned} \tag{6}$$

Formula (5) and Formula (6) indicate that embedding the secret information increases the maximum pixel value and decreases the minimum pixel value, and thus does not change the order of pixel values. Therefore, the data extraction process, adopts the same method as embedding phase for prediction after numbering and sorting the pixels in the block, and can extract the secret information completely and restore the pixel values correctly, as depicted in Formula (7) and Formula (8).

$$x_{\pi(n1\times n2)} = \begin{cases} x'_{\pi(n1\times n2)} - 1 & \mathit{if}\ d_{\max} > 2 || d_{\max} < -1 \\ x'_{\pi(n1\times n2)},\ b = 0 & \mathit{if}\ d_{\max} = 1 || d_{\max} = 0 \\ x'_{\pi(n1\times n2)} - 1, b = 1 & \mathit{if}\ d_{\max} = 2 || d_{\max} = -1. \end{cases} \tag{7}$$

$$x_{\pi\ (1)} = \begin{cases} x'_{\pi\ (1)} + 1 & \mathit{if}\ d_{\min} > 2 ||\ \mathit{if}\ d_{\min} < -1 \\ x'_{\pi\ (1)},\ b = 0 & \mathit{if}\ d_{\min} = 1 || d_{\min} = 0 \\ x'_{\pi\ (1)} + 1, b = 1 & \mathit{if}\ d_{\min} = 2 ||\ \mathit{if}\ d_{\min} = -1. \end{cases} \tag{8}$$

C. WANG ET AL.'S METHOD

PVO-based RDH schemes always utilize the fixed-size blocks as units to perform the embedding and extracting procedures. However, different-sized blocks are appropriate regions with different complexity. The method proposed by Wang *et al.* [25] first divides the cover image into 4×4-sized blocks, then processes each block according to the complexity of this block. The block complexity *NL* value is derived as follows. First, divide the block into four 2×2-sized sub-blocks, and then sort the four pixel values in each resulting sub-block. Only the maximum and minimum values in the block can be adjusted during the embedding process. Therefore, the unchanged values, namely the second-largest values $x_{\pi\ (3),i} (i \in \{1, 2, 3, 4\})$ and the second-smallest values $x_{\pi\ (2),i} (i \in \{1, 2, 3, 4\})$, are recorded to determine *NL* using Formula (9). Specifically, *NL* is the difference between the maximum value among the four second-largest values and the minimum value among the four second-smallest values.

$$\begin{cases} NL = \max(x_{\pi\ (3),i}) - \min(x_{\pi\ (2),i}), \\ i \in \{1, 2, 3, 4\}. \end{cases} \tag{9}$$

Two thresholds T_1 and T_2 are set before the secret data are embedded. If $NL \leq T_1$, then the current block is regarded as a smooth block. In this case, the block is divided into four sub-blocks. If $T_1 < NL \leq T_2$, then the current block is regarded as a normal block, and secret data can be directly embedded in it; if $NL > T_2$, then the current block is regarded as a complex block, and its prediction value is difficult to obtain accurately, so embedding data needlessly modifies the pixel values. Therefore, these blocks are skipped. After dividing the blocks, Peng *et al.*'s method is adopted to embed and extract the secret data.

Wang *et al.*'s scheme judges whether to divide the current block into sub-blocks based on the complexity of the block. It can prioritize the use of 2×2-sized blocks in the smooth regions, and 4×4-sized blocks in the complex regions. Therefore, Wang *et al.*'s method performs significantly better than simple IPVO.

D. PPVO METHOD

The RDH schemes with PVO-based predictor in a block-by-block manner produce high-fidelity marked images. However, a block-wised predictor has boundary restrictions, preventing it from fully employing the smooth regions of the cover image. If the sorted pixel values can be adopted to improve the prediction accuracy, with no block limitation, then the performance of the PVO-based RDH schemes is further improved.

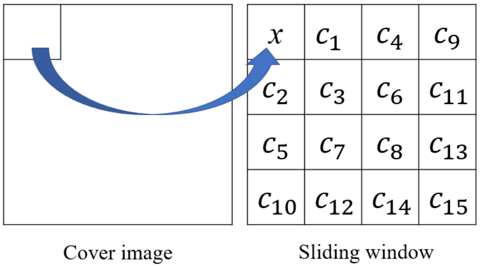


FIGURE 2. **Schematic diagram of sliding window.**

Qu and Kim [26] replaced the fixed block with a sliding window, so that the pixel values in the sliding window can still be used to derive the prediction value after ordering, and the dynamic moving characteristics of the sliding window also eliminates the block limitations. As displayed in Fig. 2, assume that the size of the window is 4×4. For the pixel in the upper left corner, a pixel window ω is defined as $\omega = [x, C]$ and is sorted in the ascending order of the pixel values in the pixel vector $C = (c_1, c_2, \ldots, c_{15})$ to predict the pixel x. The prediction process is as follows:

Case1: If the maximum and minimum values in the pixel vector C are not equal, then compare x with them. Calculate the prediction results according to Formula (10).

Case2: If the maximum and minimum values in the pixel vector C are equal, then denote the value as VC, and then compare x with VC. Calculate the predicted value according to Formula (11).

$$\hat{x} = \begin{cases} \min(x) & \textit{if } x \le \min(x) \\ \max(x) & \textit{if } x \ge \max(x) \\ \textit{skip} & \textit{if } \min(x) < x < \max(x). \end{cases} \tag{10}$$

$$\hat{x} = \begin{cases} VC & \textit{if } x = VC = 254 \\ \textit{skip} & \textit{if } x \ne VC = 254 \\ VC & \textit{if } x \le VC \ne 254 \\ \textit{skip} & \textit{if } x > VC \ne 254. \end{cases} \tag{11}$$

The prediction error can be derived from the predicted value. Then move the sliding window to the right by one pixel at a time, when sliding to the right end of the cover image, starting from the next line. By analogy, if the size of the window is 4×4, then each pixel has a predicted value except for the right three columns and the bottom three rows.

III. PROPOSED SCHEME

This section proposes a RDH scheme based on PVO using a dynamic partition method based on the quad-tree structure. The overview of the proposed scheme is introduced in Section *A*, and then the detailed embedding and data extraction procedures are described in Sections *B* and *C*. Section *D* has an example to illustrate the proposed scheme.

A. OVERVIEW OF PROPOSED METHOD

The conventional PVO-based RDH methods all take the block as a unit to predict, embed and extract secret data. Since the secret information that can be embedded in each block is limited, so the size of the blocks directly determines the embedding capacity of the cover image, and further affects the quality of the camouflage image. Hence, the block method plays a crucial role in the PVO-based RDH methods.

The mentioned PVO-based RDH methods demonstrate that, for the same cover image, larger blocks can embed less secret information and obtain a camouflage image with better quality, while smaller blocks can embed more secret information and obtain a camouflage image with poor quality. This phenomenon can be explained as follows. The size and number of the blocks in the cover image are inversely proportional, and each block can embed a fixed amount of secret information onto the maximum/minimum pixels. Therefore, larger blocks result in a smaller embedding capacity, which means that fewer pixels are modified, leading to a higher quality camouflage image. Conversely, an image with smaller blocks will have a larger embedding capacity.

In addition to the block size, the characteristics of the cover image itself also influence the performance of the PVO-based RDH method. Regarding the smooth regions, because the pixel values are similar, using smaller blocks can obtain a sufficiently accurate predictive value. A smaller block size also makes more blocks available for information embedding, thus obviously increasing the embedding capacity. However, for relatively complex regions, the pixels in the block always have large differences, so the embedding requirements normally cannot be fulfilled. Using larger blocks in these regions significantly reduces unnecessary pixel value modification. Therefore, to better enhance the performance of the PVO-based RDH method, the process of region smoothness judgment and block size selection needs to be optimized. The ultimate goal is to use as many small blocks as possible in the smooth regions, and as many large blocks as possible in the complex regions, to embed secret information, and finally to determine an optimal balance between the quantity of camouflage image and the embedding capacity.

The proposed scheme introduces the concept of quad-tree, and dynamically divides the cover image based on the quad-tree structure. The proposed scheme adaptively determines the depth of the quad-tree structure on the basis of the complexity of each region. Accordingly, smaller blocks are always used in the smooth regions, and the larger blocks are used in complex regions. Furthermore, the threshold value to determine whether to continue to divide can be adjusted based on the required embedding capacity.

The PPVO method [26] adopts a sliding window instead of fixed blocks. The remaining pixels within the sliding window can still be reused after processing the pixel in the left up corner of the sliding window. The dynamic mobile characteristics of PPVO also eliminate the block limitation. Therefore, PPVO achieves high performance, especially for

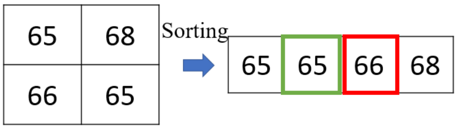


FIGURE 3-1. Calculation method of *NL* for a 2×2-sized block.

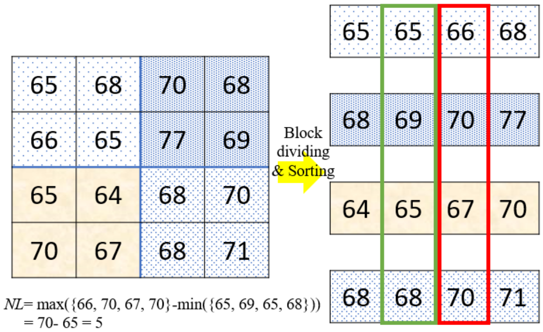


FIGURE 3-2. Calculation method of *NL* for a 4×4–sized block.

images with many continuously smooth areas. This study utilizes the advantages of the PPVO method to embed the secret information in continuously smooth regions to improve use of similar adjacent pixels.

The proposed scheme take the Noisy Level (*NL*) as the basis of the complexity of the block and adaptively divide the cover image. The value of *NL* is determined as follows. If the block size is 2×2, as shown in Fig. 3-1, *NL* equals the second-largest value minus the second-smallest value; when the block size is 4×4, as depicted in Fig. 3-2, the red box contains the second-largest values; the green box contains the second-smallest values, and *NL* is the difference between the maximum value in the red box and the minimum value in the green box; when the block size is larger than 4×4, *NL* is the average value of all *NL* values in each sub-block. Generally, traditional PVO-based RDH methods use the *NL* values to determine whether a block is appropriate for embedding secret information. Moreover, the *NL* determined using the proposed method can also be used to determine the dividing depth of each block. Therefore, the depth is obtained from two threshold values as in the following cases:

Case1: If the size of the block $S > 8 \times 8$, then it is divided into two cases.
Case1-1: If $NL \leq T_1$, the block is in a continuous smooth region, then the PPVO method is adopted to embed secret information.
Case1-2: If $NL > T_1$, the block utilizing the quad-tree partition is processed to smaller sub-blocks.
Case2: If the size of the block $2 \times 2 < S \leq 8 \times 8$, then it is divided into three cases.
Case2-1: If $NL \leq T_1$, then the block is a relatively smooth block, and therefore is further divided.
Case2-2: If $T_1 < NL \leq T_2$, then the block is a normal block, and it does not need to be divided. Therefore, the IPVO method [22] is adopted to embed secret information.
Case2-3: If $NL > T_2$, then the block is a complex block, and cannot continue to be divided. Therefore, the block is not suitable for embedding secret information, and is skipped. The pixel values in this block are not changed.
Case3: If the size of the block $S = 2 \times 2$, then it is divided into two cases.
Case3-1: If $NL \leq T_2$, then the block is a relatively smooth block. Peng *et al.*'s method is adopted to embed secret information.
Case3-2: If $NL > T_2$, then the block is a complex block. Therefore, it is not suitable for embedding secret information, and is skipped.

For the blocks that can be applied to embed secret information, giving embedding priority to larger blocks improves the quality of the camouflage image. Therefore, after building the quad-tree, a breadth search method is chosen to traverse the leaf nodes (blocks) for embedding secret information. These pixel values are then modified according to Formula (5) and Formula (6). To further optimize the proposed scheme, the dividing range $(L_1 - L_2)$ of the quad-tree can be set to improve its flexibility (assuming that the size of the cover image is $H \times M$, $0 \leq L_1 < L_2 \leq \log_4(H \times M) - 1$). Fig. 4 shows the hierarchical definition of the proposed scheme. Obviously, the method proposed by Li *et al.* [24] is a special case of the proposed scheme when $L_1 = \log_4(H \times M) - 2$ and $L_2 = \log_4(H \times M) - 1$. For instance, if the size of cover image is 512×512, then 4×4-sized blocks and 2×2-sized blocks are adopted to embed secret information. Additionally, the case of $L_1 = L_2 = 1$, $T_1 = 255$, implies using a sliding window to traverse the whole image to embed secret information. Thus, PPVO method is also a special case of the proposed scheme.

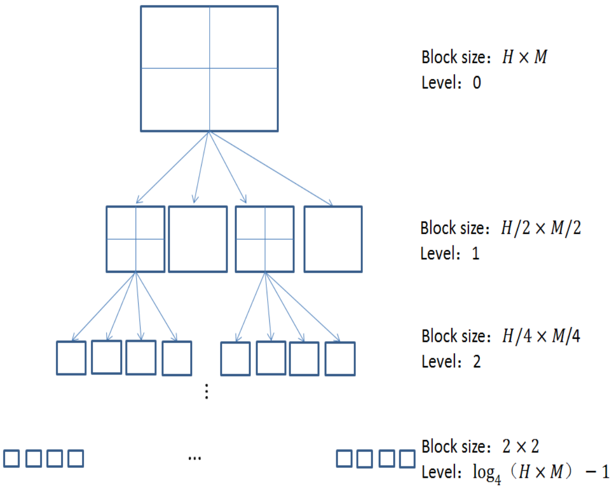


FIGURE 4. Definition of the levels in quad-tree structure.

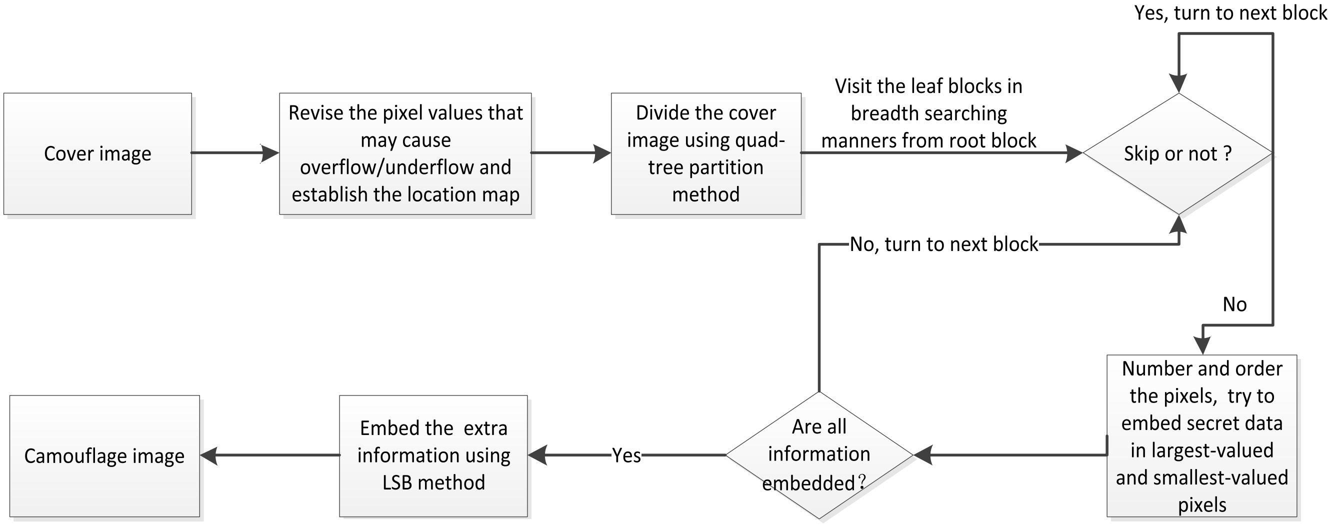


FIGURE 5. The flow chart of the data embedding procedures.

TABLE 1. Extra information.

Extra information	Purposes	Memory required
Complexity thresholds T_1 and T_2	Determine how to deal with the current block	16 bits
Division level L_1 and L_2	Correctly divide the camouflage image	$2\lceil \log_2(\log_4(H \times M)) \rceil$ bits
Compressed location map	Record the position of the pixels that may cause overflow/underflow	L bits
Length of compressed location map	Correctly extract the location map	$\log_2(H \times M)$ bits

B. EMBEDDING PROCEDURE

The embedding procedure of the proposed scheme is presented as follows.

Like all PVO-based RDH methods, the proposed scheme first has to manage overflow and underflow, that is, handle the pixel values 0 and 255. Then the cover image needs to be dynamically adaptively divided to embedding the secret information. In order to extract the secret information and restore the cover image, some additional information needs to be recorded and embedded, as listed in Table 1. The embedding steps are described as follows. Fig. 5 depicts the flowchart of the secret information embedding procedures.

Step1: Scan through the cover image of each pixel value to establish a location map. If the pixel value is 0 or 255, then it is modified to 1 or 254 for avoiding the pixel under/over-flow after secret data are concealed, and records its position for pixel restoration. Otherwise, keep the pixel value unchanged. The location map is then compressed by using the arithmetic coding, and the length of the compressed location map is recorded as L.

Step2: Use the quad-tree structure to divide the cover image for embedding secret information. Then traverse the leaf node (blocks) through breadth-searching, and determine the *NL* values of each block. For the blocks that can embed secret information, if the block size $S > 8 \times 8$, the PPVO method in Section 2.3 is selected for embedding secret information; otherwise, sort the pixel values in the block and then embed secret information according to Formula (5) and Formula (6).

Step3: Record the first $\left(16 + 2\lceil \log_2(\log_4(H \times M)) \rceil + 2\log_2(H \times M) + L\right)$ least significant bits (LSBs) of the cover pixels. Then, employ the simple LSB substitution method proposed to embed additional information (as shown in Table 1) into these pixels. Finally, embed the record LSBs into the remaining blocks after all the secret information is hidden.

C. DATA EXTRACTION PROCEDURE

To correctly extract the secret information and restore the pixels into the cover image, the embedding process needs to be run in reverse. The data extraction and image recovery procedure which is reversed to the embedding process also obtains the same quad-tree as the original embedding process. Only the maximum and minimum values in the block can be adjusted during the embedding process. Therefore, the unchanged values, namely the second-largest values and the second-smallest values, are used to determine NL.

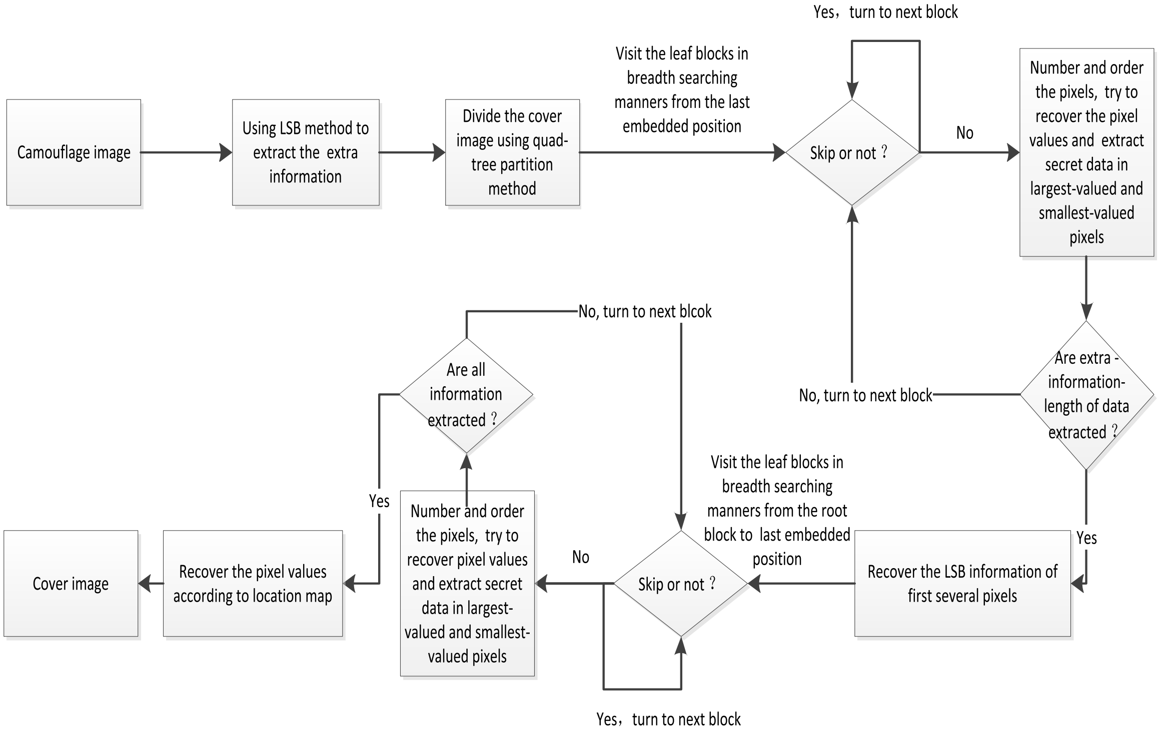


FIGURE 6. **The flowchart of the data extraction procedure.**

So the NL of each block is also invariant. According the block complexity NL, we can reconstruct the same quad-tree, so the structure invariant helps the exact data can be extracted and stego-pixels return to their original values. The detailed extraction steps are given as follows. Fig. 8 shows the flowchart of data extraction procedures.

Step1: Read the first $\left(16 + 2\left\lceil \log_2(\log_4(H \times M))\right\rceil + 2\log_2(H \times M) + L\right)$ LSBs of the pixels in the camouflage image, to obtain two threshold values T_1 and T_2; the partition level range of the quad-tree L_1 and L_2; the position of the pixel that finally used to embed the secret information, and the compressed location map. Then, decompress the compressed location map to obtain the original LM.

Step2: Similar to the embedding procedures, the quad-tree partition divides the camouflage image into blocks. Because the pixel values that are utilized to calculate the NL values and then divide the cover image remain unchanged during the embedding process, the reverse embedding process obtains the same quad-tree as the original embedding process. First, $(16+2\left\lceil\log_2(\log_4(H \times M))\right\rceil+2\log_2(H \times M)+L)$ bits of secret information are extracted starting from the final embedding position, and are then applied to recover the first $(16 + 2\left\lceil\log_2(\log_4(H \times M))\right\rceil + 2\log_2(H \times M) + L)$ LSBs of the pixels in the camouflage image. The leaf nodes (blocks) are then traversed from the root node to the final embedding block by breadth searching. For the blocks that contain the embedded secret information, if the block size $S > 8 \times 8$, then extract secret information and restore the pixel values using Qu *et al.*'s method; otherwise, extract secret information and modify the pixel values according to Formula (7) and Formula (8).

Step3: Traverse the pixels $x_i (i \in \{1, 2, ..., H \times M\})$ in the camouflage image, and restore the pixel values that may overflow/underflow according to the uncompressed location map LM. If $LM_i = 0$, keep the pixel value unchanged; if $LM_i = 1$, then the pixel value 1 is changed to 0, and the pixel value 254 is modified to 255.

D. EXAMPLE OF DATA EMBEDDING AND EXTRACTION WITH IMAGE RECOVERY

To better demonstrate the proposed method, an example of embedding secret data and data extraction with image recovery is given below, with a current block size of 8×8. The first step is to divide the given block into 4×4-sized sub-blocks and calculate each sub-block complexity value, $NL_1 = 8$, $NL_2 = 4$, $NL_3 = 6$, and $NL_4 = 2$. The average NL value of four 4×4-sized sub-blocks is 5. Assume $T_1 = 5$ and $T_2 = 7$. Since $NL \leq T_1$, the block is further divided into four 2×2-sized sub-blocks as displayed in Fig. 7-1. As can be seen from Fig. 7-2, the first sub-block is skipped with no secret data embedded since$NL_1 = 8 > T_2$. With $NL_2 = 4 \leq T_1$, the second sub-block can be divided into four 2×2-sized sub-blocks. The four 2×2-sized sub-blocks have their noise levels

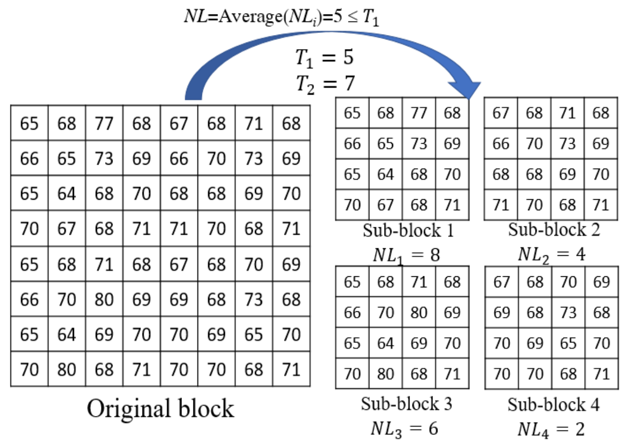


FIGURE 7-1. Calculate NL_i values of each 4 × 4 sub-block.

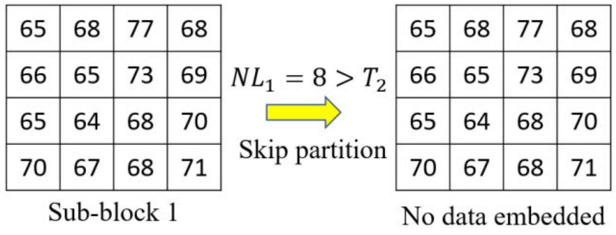
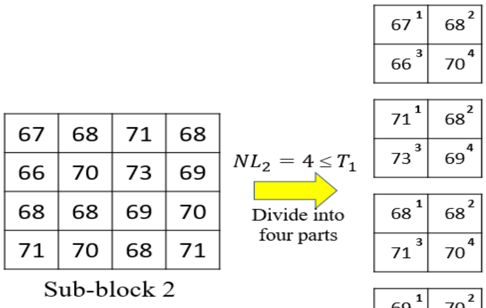


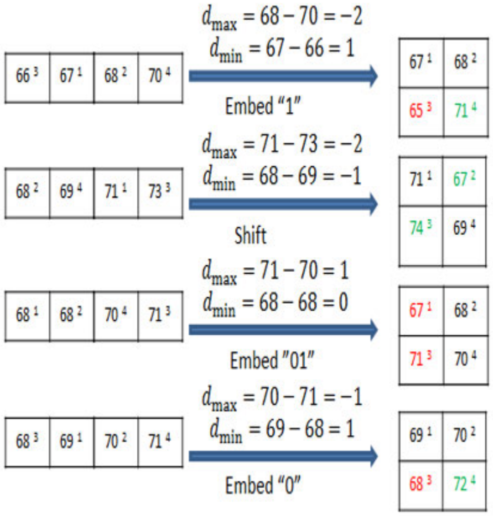
FIGURE 7-2. No data embedded in the first 4×4-sized sub-block.

as 1, 2, 3, and 1, respectively. Obviously, secret data can be directly embedded in all the smaller and smooth sub-blocks using Peng *et al.*'s IPVO method, as shown in Fig. 7-3. The noise level value of the third sub-block has $T_1 < NL_3 \leq T_2$. According to Case 2-2, $d_{max} = 80 - 80 = 0$ and $d_{min} = 65 - 64 = 1$; therefore, the secret information "10" is directly embedded using IPVO method as shown in Fig. 7-4. The fourth sub block is handled as the second sub-block, as shown in Fig. 7-5. Finally, as presented in Fig. 7-6, the various parts of the original block need to be merged after processing them.

The data extraction process is similar to the embedding procedure, and is divided into six parts, which are depicted in Fig. 8-1 to Fig. 8-6. The given block is in the root of a quad-tree in which the root has exactly four children as shown in Fig. 8-1. As can be seen from Fig. 8-2 to Fig. 8-5, since the complexity value corresponding to each sub-block remains unchanged during the embedding procedure, the four sub-blocks are separately divided, sorted, or skipped as same as the embedding procedure. After that, the secret data are extracted and the original pixel values are recovered. At last, the four parts of the original block is merged as presented in Fig. 8-6.



(a) Sub-block 2 is divided into four 2×2-sized sub-blocks



(b) secret data embedded in the smaller and smooth 2×2-sized sub-blocks using Peng *et al.*'s IPVO method

Fig.7-3. Deal with the second 4×4 sub-block

FIGURE 7-3. Deal with the second 4 × 4 sub-block. (a) Sub-block 2 is divided into four 2×2-sized sub-blocks. (b) secret data embedded in the smaller and smooth 2×2-sized sub-blocks using Peng *et al.*'s IPVO method.

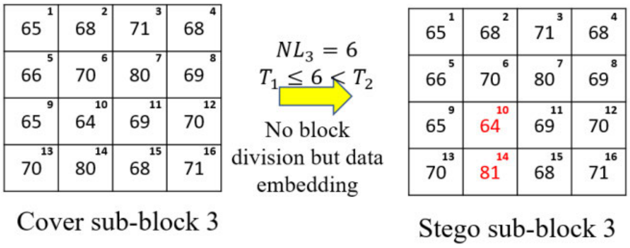


FIGURE 7-4. Data "10" embedded into the third 4×4 sub-block.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

In this section, the experimental results of the proposed scheme are presented and analyzed. In order to more comprehensively test the proposed method, eight standard gray scale test images with different complex degree, "Lena", "Baboon", "Airplane", "Barbara", "Peppers", "Boat", "Elaina" and "Lake", as shown in Fig. 9, are tested. All the experimental results are obtained based on the MATLAB R2010a platform. In order to quantify the performance of the proposed method, the embedding capacity (EC), peak signal-to-noise ratio (PSNR) and other parameters are measured in the experiment. Among them, EC refers to the

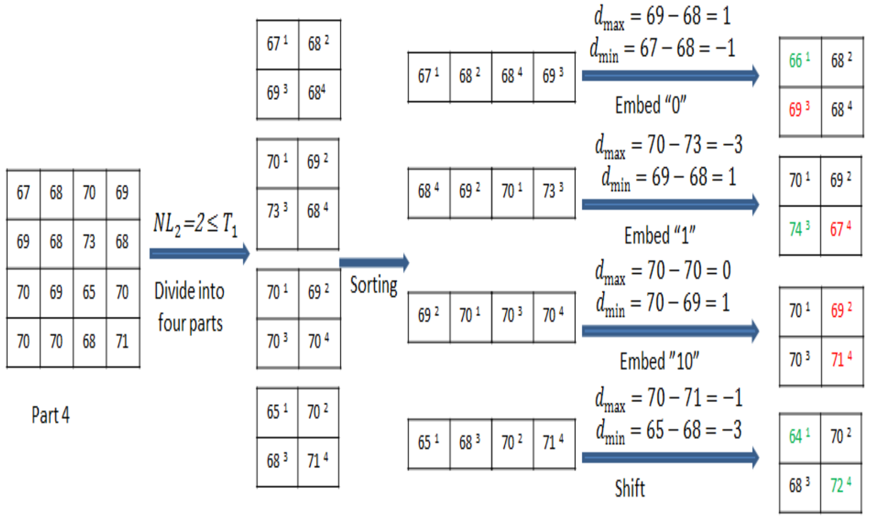


FIGURE 7-5. Deal with the fourth 4 × 4 sub-block.

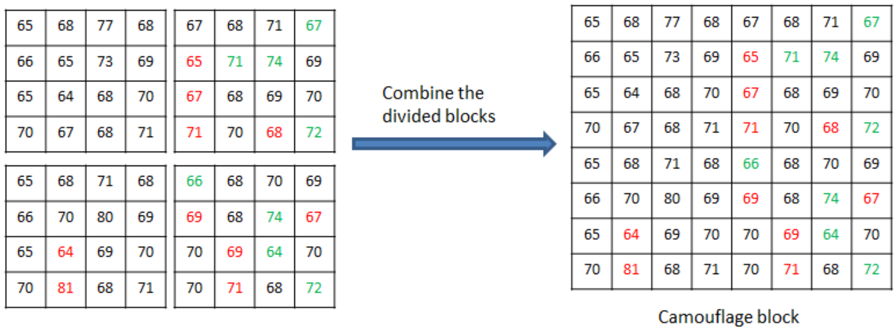


FIGURE 7-6. Combine the four sub-blocks.

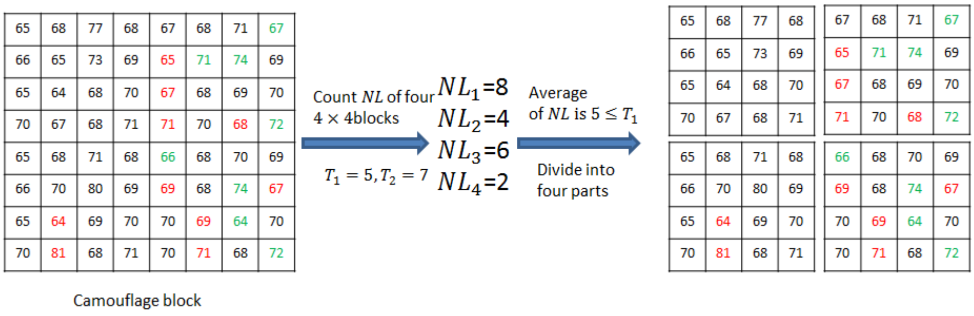


FIGURE 8-1. Calculate NL values of each 4 × 4 sub-block.

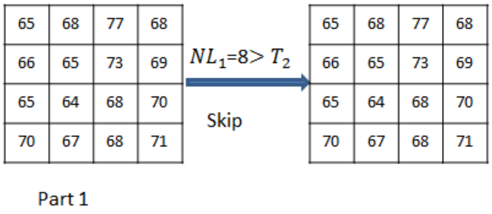


FIGURE 8-2. Deal with the first 4 × 4 sub-block.

number of secret data that is embedded in the test images, PSNR is a kind of objective criteria for the evaluation of the image, the greater the PSNR, the better the quality of the image. PSNR can be calculated as the Formula (12) and (13), where in, H and W represents the height and width of the cover image, respectively, I represents the original cover image, and I' represents the camouflage image.

$$PSNR = 10 \times \log_{10}\left(\frac{255^2}{MSE}\right). \tag{12}$$

$$MSE = \frac{1}{H \times W}\sum_{i=1}^{H}\sum_{j=1}^{W}\left(I_{(i,j)} - I'_{(i,j)}\right)^2. \tag{13}$$

Fig. 10 presents visually the quad-tree data map of "Lena". Obviously, the smooth regions with similar texture

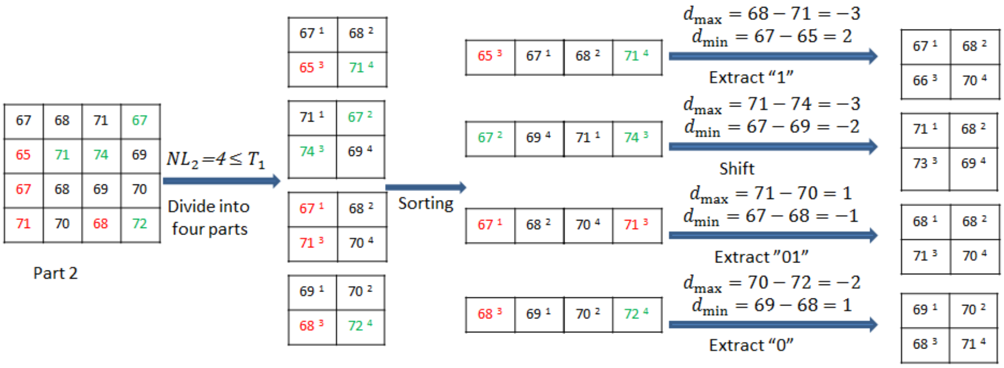


FIGURE 8-3. Deal with the second 4 × 4 sub-block.

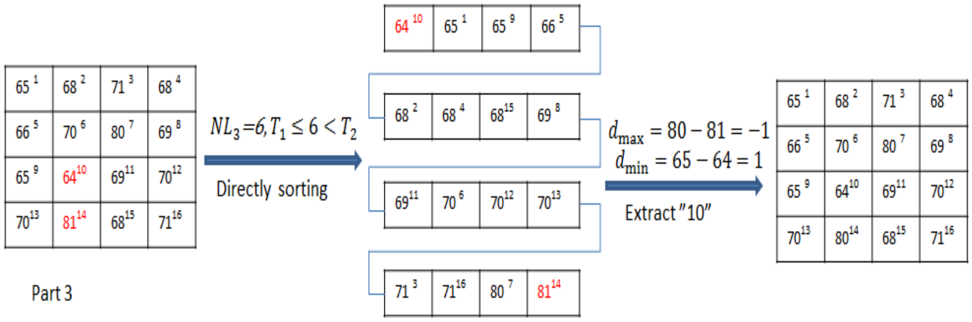


FIGURE 8-4. Deal with the third 4 × 4 sub-block.

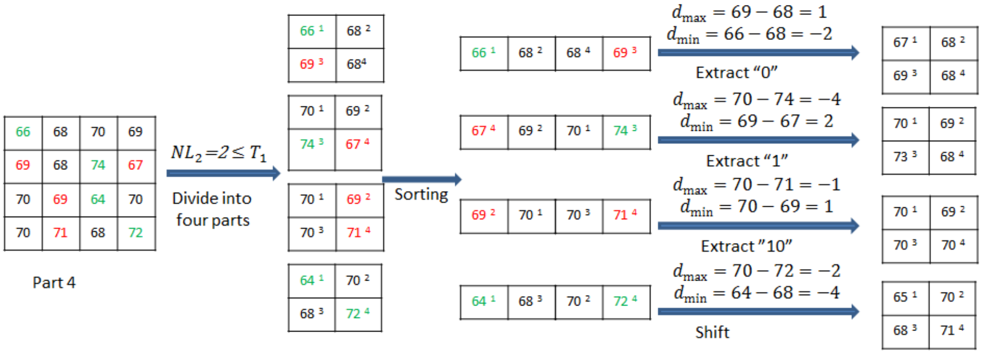


FIGURE 8-5. Deal with the fourth 4 × 4 sub-block.

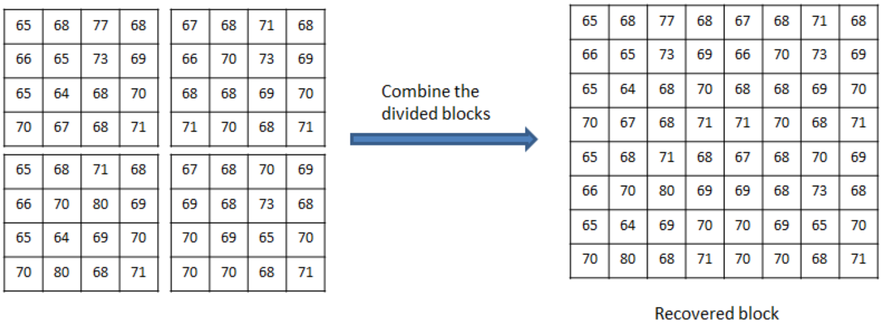
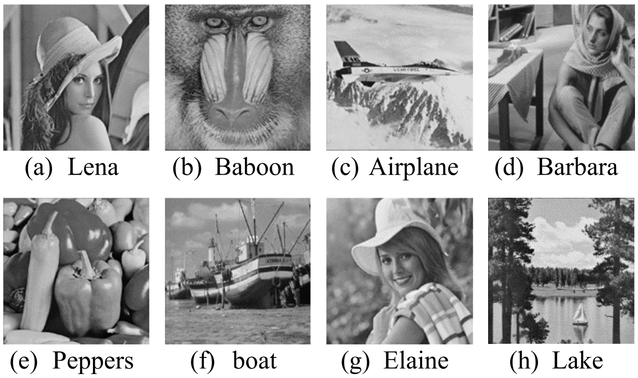


FIGURE 8-6. Combine the four sub-blocks.

correspond to the larger block, while in the complex regions, especially regions that have more edge features, the corresponding blocks are smaller. Since the performance of the PVO-based RDH schemes is directly determined from the accuracy of the prediction method, the smooth regions are more suitable for embedding secret information due to the smaller difference between the pixel values, which means that the predicted values in these regions are usually more


(a) Lena (b) Baboon (c) Airplane (d) Barbara

(e) Peppers (f) boat (g) Elaine (h) Lake

FIGURE 9. Test images.

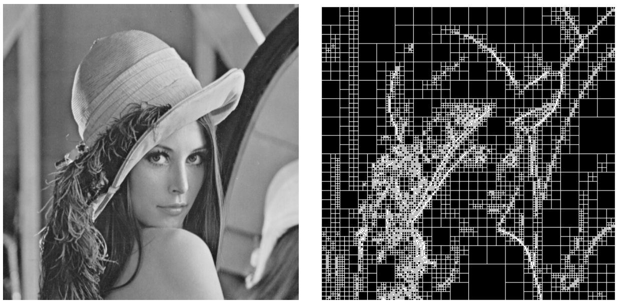


FIGURE 10. Quad-tree map of "Lena."

TABLE 2. Performance comparison between the proposed scheme and other four PVO-based schemes when EC=5000bits.

Images	PVO	IPVO	PPVO	Wang *et al.*'s method	Proposed scheme
Lena	63.5	64.2	64.0	64.0	64.4
Baboon	58.7	58.8	59.2	58.8	59.2
Airplane	65.2	66.8	67.0	66.8	67.4
Barbara	63.5	64.2	64.0	64.0	64.3
Peppers	62.3	62.5	63.0	62.4	63.2
Boat	61.8	62.2	62.4	62.3	62.6
Elaine	60.8	61.2	64.0	62.7	64.3
Lake	61.8	63.8	64.0	64.0	64.2
Average	62.4	62.9	63.4	63.1	63.7

accurate. Therefore, the sliding window is utilized to for embed the secret information in the smooth and large regions (please refer to Case 1-1), because of its ability to maximize the use of the pixels in the block. However, the small complex blocks are embedded directly or skipped. The proposed scheme exploits the benefits of the quad-tree structure when dividing the image into different sized blocks according to the regional complexity, thus improving the performance by effectively exploring the characteristics of the cover image itself.

Fig. 11 presents the performance comparison of the experimental results of five PVO-based RDH schemes on the basis of eight standard gray-scale test images. Since the proposed scheme employs two hierarchical level values L_1 and L_2, as well as two thresholds T_1 and T_2, to explore the best performance, all of their combinations were tested in the embedding phase. The data in Fig. 11 clearly demonstrate that Wang *et al.*'s method [25] and PPVO method [26] performed best among the four tested schemes, all of which are special cases of the proposed scheme. Therefore, the performance of the proposed method cannot be worse than the other comparison methods. Fig. 11 also indicates that the proposed scheme is similar to several other PVO-based RDH schemes, in that the quality of the camouflage image gradually deteriorates with increasing embedding capacity. The proposed method has better PSNR under the same embedding capacity than the other four methods, especially when the embedding capacity is small. The proposed scheme has similar maximum embedding capacity in a single cover image to the PPVO method, and both have much better performance than the other three methods. Significantly, the embedding capacity is much greater when the cover image smoother; for instance, for the most complex image "Baboon," the proposed scheme had a maximum embedding capacity 2000 bits of secret data greater than PVO, IPVO and Wang *et al.*'s method; however, for the smoothest image "Airplane," the proposed method embedded 31,000 more bits of secret data than PVO method; and 17,000 more than IPVO and Wang *et al.*'s method. Moreover, a more effective use of the smoothing regions can further enhance the performance of the RDH schemes based on a series of PVO methods.

To present a detailed performance comparison of the PSNR of several PVO-based RDH schemes, Tables 2–4 list their PSNR values when at embedding capacities of 5000bits, 20000bits and 10000bits. Table 2 shows that the proposed scheme had the best PSNR value, at 1.3db higher than PVO, 0.8db higher than IPVO, 0.3dB higher than PPVO and 0.6dB higher than Wang *et al.*'s method. Fig. 11 clearly reveals that the differences of PSNR values among the five tested schemes gradually narrowed with rising embedding capacity. As indicated in Table 3, the proposed method still had the best PSNR value when 10,000 bits of secret data were embedded, but by smaller margins than Table 2. The proposed method was only 1.2dB higher than PVO, 0.7dB higher than IPVO, 0.1dB higher than PPVO and 0.4dB higher than Wang *et al.*'s method. From Table 4, the performance advantages of the proposed method in terms of PSNR further decreased when the embedding capacity was further increased. However, the proposed scheme had a higher maximum embedding capacity is gradually unfolded, as shown in Fig. 11 and Table 4. When the embedding capacity achieved a certain level, the other methods could not obtain a PSNR value, because they had reached their maximum embedding capacity.

The proposed scheme divides the cover image into blocks of various-sizes adaptively based on *NL* of the blocks using

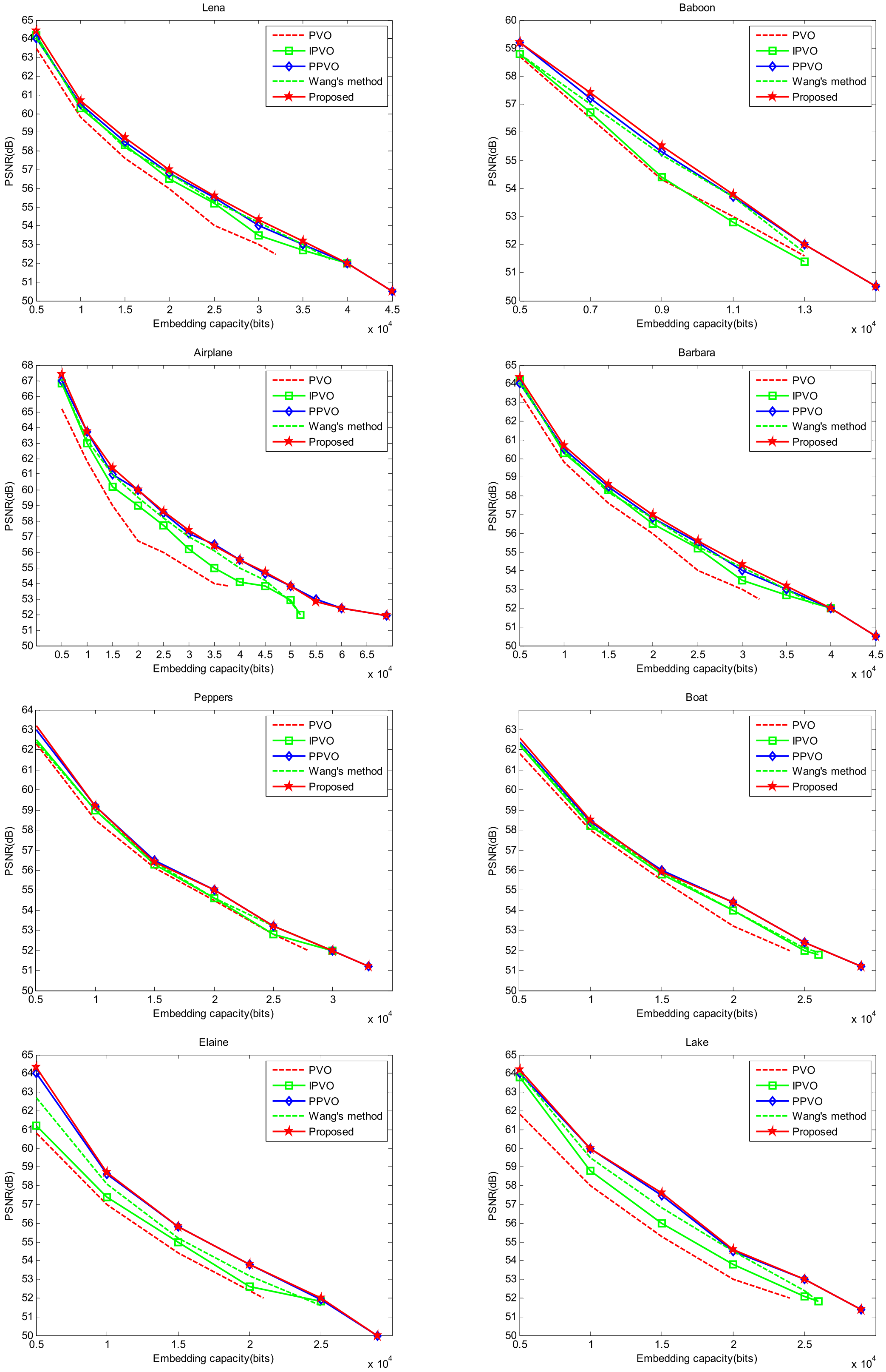


FIGURE 11. **Performance comparison between the proposed method and other four PVO-based schemes.**

a quad-tree structure and then uses either IPVO or PPVO method for each block. The PVO and IPVO methods do not change the second-largest and second-smallest values of each block and thus left the *NL* value of each block unchanged, but this is not the case for the PPVO method. In our proposed method, for a block that carries data by the PPVO method,

TABLE 3. Performance comparison between the proposed scheme and other four PVO-based schemes when EC=10000bits.

Images	PVO	IPVO	PPVO	Wang *et al.*'s method	Proposed scheme
Lena	59.8	60.3	60.5	60.4	60.7
Baboon	53.6	53.6	54.5	54.4	54.7
Airplane	61.8	63.0	63.7	63.2	63.7
Barbara	59.8	60.3	60.5	60.4	60.7
Peppers	58.5	59.0	59.2	59.0	59.2
Boat	58.0	58.2	58.4	58.3	58.5
Elaine	57.0	57.4	58.6	58.1	58.7
Lake	58.0	58.8	60.0	59.5	60.0
Average	58.3	58.8	59.4	59.1	59.5

TABLE 4. Performance comparison between the proposed scheme and other four PVO-based schemes when EC=20000bits.

Images	PVO	IPVO	PPVO	Wang *et al.*' s method	Proposed scheme
Lena	56.0	56.5	56.8	56.8	57.0
Baboon	-	-	-	-	-
Airplane	56.7	59.0	60.0	59.5	60.0
Barbara	56.0	56.5	56.8	56.8	57.0
Peppers	54.5	54.6	55.0	54.6	55.0
Boat	53.2	54.0	54.4	54.0	54.4
Elaine	52.4	52.6	53.8	53.2	53.8
Lake	53.0	53.8	54.5	54.5	54.6
Average	54.5	55.2	55.9	55.6	56.0

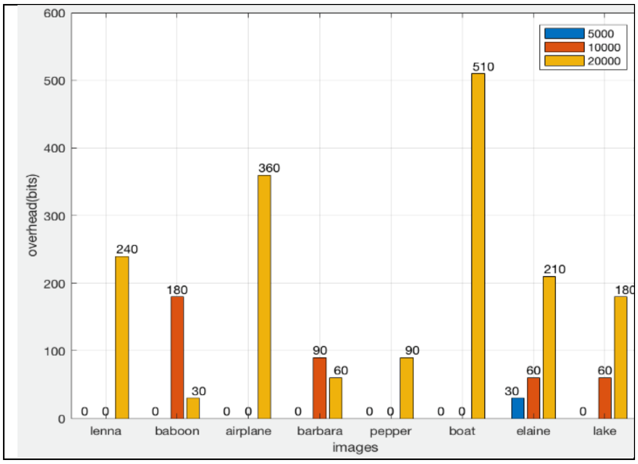
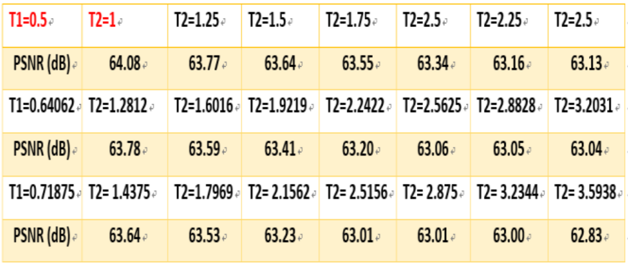


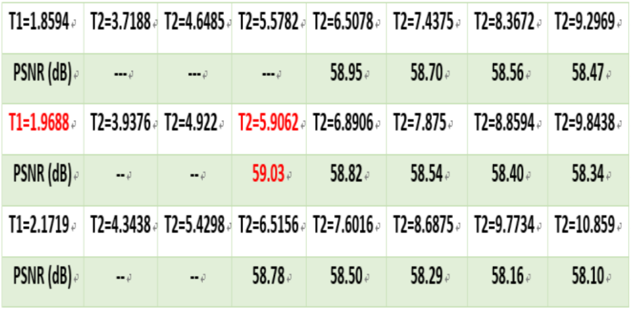
FIGURE 12. Number of bits of additional information for block recovery for various images when the pure payload EC = 5000, 10000, and 20000 bits.

TABLE 5. The PSNR values under various thresholds T_1 and T_2 for "Lean" image.



T1=0.5	T2=1	T2=1.25	T2=1.5	T2=1.75	T2=2.5	T2=2.25	T2=2.5
PSNR (dB)	64.08	63.77	63.64	63.55	63.34	63.16	63.13
T1=0.64062	T2=1.2812	T2=1.6016	T2=1.9219	T2=2.2422	T2=2.5625	T2=2.8828	T2=3.2031
PSNR (dB)	63.78	63.59	63.41	63.20	63.06	63.05	63.04
T1=0.71875	T2= 1.4375	T2=1.7969	T2= 2.1562	T2= 2.5156	T2= 2.875	T2= 3.2344	T2= 3.5938
PSNR (dB)	63.64	63.53	63.23	63.01	63.01	63.00	62.83

TABLE 6. The PSNR values under various thresholds T_1 and T_2 for "Baboon" image.



T1=1.8594	T2=3.7188	T2=4.6485	T2=5.5782	T2=6.5078	T2=7.4375	T2=8.3672	T2=9.2969
PSNR (dB)	---	---	---	58.95	58.70	58.56	58.47
T1=1.9688	T2=3.9376	T2=4.922	T2=5.9062	T2=6.8906	T2=7.875	T2=8.8594	T2=9.8438
PSNR (dB)	--	--	59.03	58.82	58.54	58.40	58.34
T1=2.1719	T2=4.3438	T2=5.4298	T2=6.5156	T2=7.6016	T2=8.6875	T2=9.7734	T2=10.859
PSNR (dB)	--	--	58.78	58.50	58.29	58.16	58.10

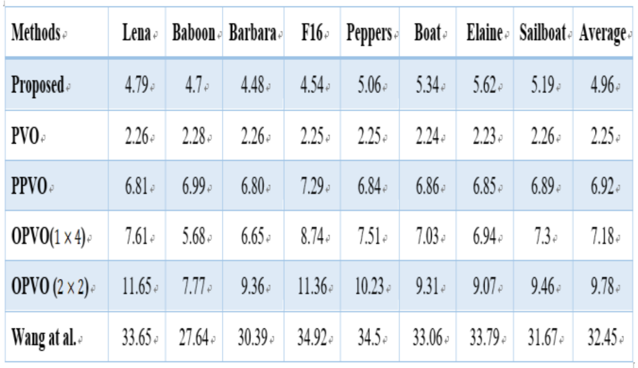
if the *NL*value of the block after data embedding exceeds T_1, the overhead information corresponding to the block needs to be recorded for block restoration. Therefore, for a block that require an additional message as recovery, our method uses 30 bits to record the starting coordinates (x, y), the width w, and the height h of that block, where x, y, w and h needs 8, 8, 7 and 7 bits, respectively. Fig. 12 below shows the number of bits of block information that need to be additionally recorded for block restoration under various images.

We also conducted an experiment to show how the two thresholds affect the performance in terms of embedding capacity and image quality. As a result of the experiment, various thresholds T_1 and T_2 are listed in Tables 5 and 6 for test images "Lean" and "Baboon", respectively.

Take a number of suitable T_2 for testing (such as taking $T_2 = T_1 \times \{2, 2.5, 3, 3.5, 4, 4.5, 5\}$) and choose the best T_1 and T_2. Taking Lena and Baboon as examples, the embedding capacity is 5,000 bits. The values of each T_1 and T_2 and the corresponding PSNR are calculated as shown in the following Tables 5 and 6. From the experimental results, it can be observed that under the same T_1, the PSNR with different T_2 values does not change much (the variation range is about 1 dB). Therefore, the proposed method can find the fittest solution from these combinations.

Table 7 shows the runtime comparisons (in seconds) in the case where the entire image is embedded with secret data to the maximum capacity. Our proposed method takes an average of 4.96 seconds in performing the embedding procedure, which is faster than others except PVO method because the PVO method proposed by Li *et al.* is simple and non-overlapping, thus taking only 2.25 seconds on average to embed secret data.

TABLE 7. **Comparisons of runtime for PVO-based, Wang's and proposed methods for eight standard images.**



Methods	Lena	Baboon	Barbara	F16	Peppers	Boat	Elaine	Sailboat	Average
Proposed	4.79	4.7	4.48	4.54	5.06	5.34	5.62	5.19	4.96
PVO	2.26	2.28	2.26	2.25	2.25	2.24	2.23	2.26	2.25
PPVO	6.81	6.99	6.80	7.29	6.84	6.86	6.85	6.89	6.92
OPVO(1 × 4)	7.61	5.68	6.65	8.74	7.51	7.03	6.94	7.3	7.18
OPVO (2 × 2)	11.65	7.77	9.36	11.36	10.23	9.31	9.07	9.46	9.78
Wang at al.	33.65	27.64	30.39	34.92	34.5	33.06	33.79	31.67	32.45

V. CONCLUSION

This work develops a dynamic partitioning method based on quad-tree structure to enhance the PVO-based RDH schemes. The proposed scheme utilized a quad-tree structure, in which the cover image is dynamically divided by comprehensively and flexibly exploring the characteristics of the cover image itself. Additionally, the proposed scheme combines the advantages of several excellent PVO-based RDH methods. It can adaptively choose the appropriate processing method for a block according to the complexity of the block itself, to optimize performance by maximizing the use of the image characteristics.

The proposed scheme is an extension of Wang *et al.*'s method [24] and PPVO, both of which are special cases of the proposed scheme. Specifically, Wang *et al.*'s method is the special case when $L_1 = \log_4 (H \times M) - 2$ and $L_2 = \log_4 (H \times M) - 1$ in the proposed scheme; PPVO is the special case of the proposed scheme when $L_1 = L_2 = 1$ and $T_1 = 255$. The proposed method improves on those two methods by not only breaking the block limitations of the conventional PVO-based RDH schemes, but also utilizes the quad-tree structure, so that the image can be flexibly divided and characteristics of the image itself can be fully exploited to embed secret data. Experimental results can certify that the proposed scheme performs better than several most cutting-edge PVO-based RDH schemes [22]–[25], both in the quality of camouflage image and the maximum embedding capacity.

REFERENCES

[1] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 8, pp. 890–896, Aug. 2003.

[2] S. Weng, Y. Zhao, J. S. Pan, and R. Ni, "Reversible watermarking based on invariability and adjustment on pixel pairs," *IEEE Signal Process. Lett.*, vol. 15, pp. 721–724, Nov. 2008.

[3] H. J. Kim, V. Sachnev, Y. Q. Shi, J. Nam, and H.-G. Choo, "A novel difference expansion transform for reversible data embedding," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 3, pp. 456–465, Sep. 2008.

[4] M. Liu, H. S. Seah, C. Zhu, W. Lin, and F. Tian, "Reducing location map in prediction-based difference expansion for reversible image data embedding," *Signal Process.*, vol. 92, no. 3, pp. 819–828, Mar. 2012.

[5] I.-C. Dragoi and D. Coltuc, "Local-prediction-based difference expansion reversible watermarking," *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1779–1790, Apr. 2014.

[6] X. Li, B. Yang, and T. Zeng, "Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection," *IEEE Trans. Image Process.*, vol. 20, no. 12, pp. 3524–3533, Dec. 2011.

[7] B. Ou, X. Li, Y. Zhao, R. Ni, and Y.-Q. Shi, "Pairwise prediction-error expansion for efficient reversible data hiding," *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 5010–5021, Dec. 2013.

[8] D. M. Thodi and J. J. Rodriguez, "Expansion embedding techniques for reversible watermarking," *IEEE Trans. Image Process.*, vol. 16, no. 3, pp. 721–730, Mar. 2007.

[9] Z. Ni, Y.-Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 3, pp. 354–362, Mar. 2006.

[10] M. Arabzadeh and M. R. Rahimi, "Reversible data hiding scheme based on maximum histogram gap of image blocks," *KSII Trans. Internet Inf. Syst.*, vol. 6, no. 8, pp. 1964–1981, Aug. 2012.

[11] M. Fallahpour and M. H. Sedaaghi, "High capacity lossless data hiding based on histogram modification," *IEICE Electron. Express*, vol. 4, no. 7, pp. 205–210, 2007.

[12] W. Hong, T.-S. Chen, and C.-W. Shiu, "Reversible data hiding for high quality images using modification of prediction errors," *J. Syst. Softw.*, vol. 82, no. 11, pp. 1833–1842, 2009.

[13] I. F. Jafar, K. A. Darabkh, R. T. Al-Zubi, and R. A. Al Na'mneh, "Efficient reversible data hiding using multiple predictors," *Comput. J.*, vol. 59, no. 3, pp. 423–438, Mar. 2016.

[14] E. N. Jaara and I. F. Jafar, "Reversible data hiding based on histogram shifting of prediction errors using two predictors," in *Proc. IEEE Jordan Conf. Appl. Elect. Eng. Comput. Technol. (AEECT)*, Nov. 2015, pp. 1–5.

[15] V. Sachnev, H. J. Kim, J. Nam, S. Suresh, and Y. Q. Shi, "Reversible watermarking algorithm using sorting and prediction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 7, pp. 989–999, Jul. 2009.

[16] X. Hu, W. Zhang, X. Hu, N. Yu, X. Zhao, and F. Li, "Fast estimation of optimal marked-signal distribution for reversible data hiding," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 5, pp. 779–788, May 2013.

[17] X. Li, J. Li, B. Li, and B. Yang, "High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion," *Signal Process.*, vol. 93, no. 1, pp. 198–205, 2013.

[18] Y. Hu, H.-K. Lee, and J. Li, "DE-based reversible data hiding with improved overflow location map," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 2, pp. 250–260, Feb. 2009.

[19] X. Wu and N. Memon, "Context-based, adaptive, lossless image coding," *IEEE Trans. Commun.*, vol. 45, no. 4, pp. 437–444, Apr. 1997.

[20] M. Fallahpour, "Reversible image data hiding based on gradient adjusted prediction," *IEICE Electron. Express*, vol. 5, no. 20, pp. 870–876, 2008.

[21] M. J. Weinberger, G. Seroussi, and G. Sapiro, "The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS," *IEEE Trans. Image Process.*, vol. 9, no. 8, pp. 1309–1324, Aug. 2000.

[22] F. Peng, X. Li, and B. Yang, "Improved PVO-based reversible data hiding," *Digit. Signal Process.*, vol. 25, no. 2, pp. 255–265, 2014.

[23] B. Ou, X. Li, Y. Zhao, and R. Ni, "Reversible data hiding using invariant pixel-value-ordering and prediction-error expansion," *Signal Process., Image Commun.*, vol. 29, no. 7, pp. 760–772, 2014.

[24] J.-J. Li, Y.-H. Wu, C.-F. Lee, and C.-C. Chang, "Generalized PVO-K embedding technique for reversible data hiding," *Int. J. Netw. Secur.*, vol. 20, no. 1, pp. 65–77, 2018.

[25] X. Wang, J. Ding, and Q. Pei, "A novel reversible image data hiding scheme based on pixel value ordering and dynamic pixel block partition," *Inf. Sci.*, vol. 310, pp. 16–35, Jul. 2015.

[26] X. Qu and H. J. Kim, "Pixel-based pixel value ordering predictor for high-fidelity reversible data hiding," *Signal Process.*, vol. 111, no. 1, pp. 249–260, Jun. 2015.

[27] T.-C. Lu, C.-Y. Tseng, S.-W. Huang, and T. N. Vo, "Pixel-value-ordering based reversible information hiding scheme with self-adaptive threshold strategy," *Symmetry*, vol. 10, no. 12, p. 764, Dec. 2018.

[28] C.-F. Lee, J.-J. Shen, and Y.-C. Kao, "High-capacity reversible data hiding based on star-shaped PVO method," in *Recent Advances in Intelligent Information Hiding and Multimedia Signal Processing* (Smart Innovation, Systems and Technologies), vol. 109. Cham, Switzerland: Springer, 2019, pp. 280–288.

[29] T. Li, Y. Ke, M. Zhang, Y. Lei, and Y. Ding, "High-fidelity reversible data hiding using dynamic prediction and expansion," *J. Electron. Imag.*, vol. 28, no. 1, Jan. 2019, Art. no. 013013.

[30] S. Weng, Y. Chen, B. Ou, C.-C. Chang, and C. Zhang, "Improved K-pass pixel value ordering based data hiding," *IEEE Access*, vol. 7, pp. 34570–34582, 2019.

[31] C.-F. Lee, J.-J. Shen, Y.-C. Kao, and S. Agrawal, "Overlapping pixel value ordering predictor for high-capacity reversible data hiding," *J. Real-Time Image Process.*, vol. 16, no. 4, pp. 835–855, Aug. 2019.

JIAN-JUN LI received the B.Sc. degree in information engineering from the Xi'an University of Electronic Science and Technology, Xi'an, China, the M.Sc. degree in electrical and computer from The University of Western Ontario, Canada, and the Ph.D. degree in electrical and computer from the University of Windsor, Canada. He is currently a Chair Professor with the School of Computer Science and Technology, Hangzhou Dianzi University. He is also the Director of the Institute of Graphic and Image. His research interests include micro-electronics, and audio, video, and image processing algorithms and implementation.

CHIN-FENG LEE received the Ph.D. degree in computer science and information engineering from National Chung Cheng University, Taiwan. She is currently a Professor of information management with the Chaoyang University of Technology, Taichung, Taiwan. Her research interests include steganography, image processing, information retrieval, and data mining.

CHIN-CHEN CHANG received the B.Sc. degree in applied mathematics and the M.Sc. degree in computer and decision sciences from National Tsing Hua University and the Ph.D. degree in computer engineering from National Chiao Tung University.

From 1989 to 2005, he was with National Chung Cheng University, where he is currently a Chair. He has been a Professor with the Department of Information Engineering and Computer Science, Feng Chia University, since February 2005. He was an Associate Professor with National Chiao Tung University, a professor with National Chung Hsing University, and a Chair Professor with National Chung Cheng University. He was a Visiting Researcher with Tokyo University, Japan, and a Visiting Scientist with Kyoto University, Japan. During his service in Chung Cheng University, he was the Chairman of the Institute of Computer Science and Information Engineering, the Dean of the College of Engineering, the Provost and the Acting President with Chung Cheng University, and the Director of the Advisory Office in the Ministry of Education, Taiwan. On numerous occasions, he was invited to serve as a Visiting Professor, a Chair Professor, an Honorary Professor, an Honorary Director, an Honorary Chairman, a Distinguished Alumnus, a Distinguished Researcher, and a Research Fellow by universities and research institutes. His current research interests include database design, computer cryptography, image compression, and data structures. He is also a Fellow of the IEE, U.K. He has won many research awards and honorary positions by and in prestigious organizations in both national and international levels. In his early years of career development, he consecutively won the Outstanding Talent in Information Sciences of China, the AceR Dragon Award of the Ten Most Outstanding Talents, the Outstanding Scholar Award of China, the Outstanding Engineering Professor Award of China, the Distinguished Research Awards of National Science Council of China, the Top Fifteen Scholars in Systems and Software Engineering of the *Journal of Systems and Software*, and so on.

JIANG-YI LIN received the B.S. and M.S. degrees in computer science and technology from Fuzhou University, Fujian, China, in 2005 and 2008, respectively. He is currently pursuing the Ph.D. degree with the Multimedia and Secure Networking Laboratory (MSN lab), Department of Information Engineering and Computer Science, Feng Chia University, Taichung, Taiwan. His research interests include image processing, secret sharing, and steganography.

YUN-HE WU received the B.S. degree in computer science and technology from the Henan University of Science and Technology, Henan, China, in 2014, and the M.S. degree in computer science and technology from Hangzhou Dianzi University, Zhejiang, China. His research interests include data hiding, video coding/decoding, and image processing.

• • •