




Research Article

Reversible Data Hiding Based on Multiple Pairwise PEE and Two-Layer Embedding

Wenguang He , Gangqiang Xiong , and Yaomin Wang 

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

Correspondence should be addressed to Yaomin Wang; 249668530@qq.com

Received 3 September 2021; Revised 19 November 2021; Accepted 20 December 2021; Published 31 January 2022

Academic Editor: Mohamed Amine Ferrag

Copyright © 2022 Wenguang He et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Recent reversible data hiding (RDH) work tends to realize adaptive embedding by discriminately modifying pixels according to image content. However, further optimization and computational complexity remain great challenges. By presenting a better incorporation of pixel value ordering (PVO) prediction and pairwise prediction-error expansion (PEE) technologies, this paper proposes a new RDH scheme. The largest/smallest three pixels of each block are utilized to generate error-pairs. To achieve optimization of the distribution of error pairs, two-layer embedding is introduced such that full-enclosed pixels of each block can be used to determine how to optimally define the spatial location of pixels within block. Then, to modify error pairs with less distortion introduced, the shifted pairing error is involved in the separable utilization of the other one; i.e., it serves as the context for recalculating the other one. Since the recalculation is equivalent to expansion bins selection, various extensions of original pairwise PEE are designed, parameterized, and combined into the so-called multiple pairwise PEE, with which the 2D histogram can be divided into a set of sub-ones for more accurate modification. The experimental results verify the superiority of the proposed scheme over several PVO-based schemes. On the Kodak image database, the average PSNR gains over original PVO-based pairwise PEE are 0.83 and 0.99 dB for capacities of 10,000 and 20,000 bits, respectively.

1. Introduction

REVERSIBLE data hiding (RDH) is a kind of data hiding technology in which some secret message is embedded into the cover media for various purposes such as secret communication and integrity authentication. Moreover, the cover media can be exactly recovered after extracting the hidden data. Compared with traditional data hiding technologies such as digital watermarking whose main concern lies in the robustness against various attacks, RDH is categorized as fragile watermarking technology, and its specific property is to exactly restore the cover media. RDH for uncompressed gray-scale image [1] is the most investigated subject of RDH nowadays. Furthermore, RDH in encrypted domain [2] has also gained increasing attention.

In terms of evaluating a RDH scheme, two aspects including fidelity and capacity are usually considered. To achieve desirable trade-off between them, many novel technologies, such as lossless compression [3, 4], histogram

shifting [5–7], difference expansion [8], prediction-error expansion [9–12], and integer-to-integer transform [13–16], have been devised. Among them, prediction-error expansion (PEE) has attracted much attention since it can better exploit the spatial redundancy in image. For PEE-based RDH, the target pixel is first predicted to generate prediction error. Then, the obtained errors are expanded or shifted to fulfill data embedding. So far, many advanced prediction methods have been devised to achieve optimization of the distribution of errors or a prediction-error histogram (PEH) [17–22]. For prediction-error modification or PEH manipulation, advanced modification methods such as [23–29] have also been proposed to embed the payload by modifying prediction errors as slightly as possible.

Adaptive embedding and high-dimensional PEE are two remarkable achievements in recent years. For example, pixels are adaptively selected for data embedding in [17]. In [29], the obtained errors are classified and embedded with different amounts of data. In [26], the expansion bins are

adaptively selected so as to achieve optimal embedding performance. In [23], a histogram sequence is first derived from decomposing the PEH. Then, for each subhistogram the expansion bins are adaptively selected. High-dimensional PEE additionally exploits the correlation of errors by jointly modifying them [25, 27]. For example, pairwise PEE [27] outperforms conventional PEE by introducing more flexible modification of error pairs; i.e., a pair of expandable errors can only be embedded with one of the combinations of bits “00”, “01”, and “10”. Pairwise PEE was firstly incorporated with rhombus prediction [27]. Afterwards, there was also the incorporation of pairwise PEE and pixel value ordering (PVO) prediction [30]. To enhance pairwise PEE by combining expandable errors into a pair, the strategy of adaptive pairing is proposed later [28]. Based on the observation that adaptive pairing actually aims to achieve optimization of the distribution of error pairs, similar optimization is achieved in [31, 32] where pairwise PEE is better incorporated with PVO prediction. Existing pairwise PEE-based schemes have demonstrated the advantage of pairwise PEE in distortion reduction. On the other hand, PVO prediction has been verified to have high accuracy. Therefore, PVO-based pairwise PEE is of great significance to high-fidelity RDH.

In this paper, a new RDH scheme is proposed by extending PVO-based pairwise PEE into adaptive embedding from two aspects including adaptive error-pair generation and adaptive error-pair modification. The cover image is first divided into shadow and blank blocks to enable two-layer embedding. In this way, the local complexity of each block can be more precisely computed using full-enclosed pixels, and thus a better selection of pixel blocks for data embedding is presented. For error-pair generation, the largest/smallest three pixels of each block are utilized by improved PVO (IPVO) [33] to generate error pairs. For two correlated pixels, the principle of IPVO can be interpreted as predicting the one having relatively smaller location with the other one. Since errors valued 0 or 1 are expandable in IPVO, it is quite necessary to ensure that the to-be-predicted pixel with relatively smaller location has relatively larger gray value. Based on the above analysis, full-enclosed pixels are additionally used to estimate the distribution of pixels within block and to accordingly fulfill spatial location definition. As for error-pair modification, a key observation lies in that most secret data is embedded into error pairs consisting of one or more shiftable errors. For these error pairs, we propose to use the shifted error to adaptively recalculate the other pairing error. Such recalculation can be equivalent to expansion bins selection. In this way, multiple pairwise PEE is designed and parameterized by the selection of expansion bins. Finally, an efficient mechanism to determine the expansion bins aiming for minimized distortion is also proposed.

The rest of the paper is organized as follows. Section 2 briefly introduces some related work including PVO, IPVO, and PVO-based pairwise PEE. The proposed scheme is presented in detail in Section 3. The experimental results are given and discussed in Section 4. This paper is concluded in Section 5.

2. Related Work

2.1. PVO-Based PEE. PVO-based PEE [34] exploits image redundancy in a blockwise manner. After dividing the cover image into nonoverlapped blocks consisting of n pixels, pixels of each block are collected to generate a pixel sequence (p_1, p_2, \dots, p_n) . Next, they are sorted to obtain $(p_{\sigma(1)}, p_{\sigma(2)}, \dots, p_{\sigma(n)})$ where $\sigma: \{1, \dots, n\} \rightarrow \{1, \dots, n\}$ is the unique one-to-one mapping such that $p_{\sigma(1)} \leq p_{\sigma(2)} \leq \dots \leq p_{\sigma(n)}$, $\sigma(i) < \sigma(j)$ if $p_{\sigma(i)} = p_{\sigma(j)}$ and $i < j$. In this way, the prediction error is computed as

$$PE_{\max} = p_{\sigma(n)} - p_{\sigma(n-1)}. \quad (1)$$

To fulfill data embedding, PE_{\max} is modified as

$$PE'_{\max} = \begin{cases} PE_{\max}, & \text{if } PE_{\max} = 0 \\ PE_{\max} + b, & \text{if } PE_{\max} = 1, \\ PE_{\max} + 1, & \text{if } PE_{\max} > 1 \end{cases} \quad (2)$$

where $b \in \{0, 1\}$. Accordingly, $p_{\sigma(n)}$ is modified to

$$p'_{\sigma(n)} = p_{\sigma(n-1)} + PE'_{\max} = \begin{cases} p_{\sigma(n)}, & \text{if } PE_{\max} = 0 \\ p_{\sigma(n)} + b, & \text{if } PE_{\max} = 1 \\ p_{\sigma(n)} + 1, & \text{if } PE_{\max} > 1. \end{cases} \quad (3)$$

2.2. IPVO-Based PEE. In PVO-based PEE, smooth blocks are preferentially selected and a large number of errors valued 0 are thus generated. However, these errors are ignored according to (2), which implies insufficient utilization of smooth blocks. To solve this problem, pixel location is introduced into IPVO-based PEE [33] and the prediction error is renewed as

$$e_{\max} = p_u - p_v, \quad (4)$$

where $u = \min(\sigma(n-1), \sigma(n))$, $v = \max(\sigma(n-1), \sigma(n))$. If $\sigma(n) > \sigma(n-1)$, there is $e_{\max} = p_{\sigma(n-1)} - p_{\sigma(n)} \leq 0$; otherwise, there is $e_{\max} = p_{\sigma(n)} - p_{\sigma(n-1)} \geq 1$. Obviously, the relative order of $\sigma(n-1)$ and $\sigma(n)$ remains unchanged after enlarging $p_{\sigma(n)}$. In this way, the marked prediction error e'_{\max} is computed as

$$e'_{\max} = \begin{cases} e_{\max} - b, & \text{if } e_{\max} = 0, \\ e_{\max} - 1, & \text{if } e_{\max} < 0, \\ e_{\max} + b, & \text{if } e_{\max} = 1, \\ e_{\max} + 1, & \text{if } e_{\max} > 1, \end{cases} \quad (5)$$

and accordingly $p_{\sigma(n)}$ is modified to

$$p'_{\sigma(n)} = \begin{cases} p_{\sigma(n)} + b & \text{if } e_{\max} = 0 \text{ or } e_{\max} = 1, \\ p_{\sigma(n)} + 1, & \text{if } e_{\max} < 0 \text{ or } e_{\max} > 1. \end{cases} \quad (6)$$

2.3. PVO-Based Pairwise PEE. Despite the superiority over PVO-based PEE, the performance of IPVO-based PEE is still limited by considering only the largest two pixels. By involving the 3rd largest pixel, e_{\max} is reinterpreted as $e_{\max} =$

$e_{\max}^1 - e_{\max}^2$ in PVO-based pairwise PEE [30] where e_{\max}^1 and e_{\max}^2 are computed as

$$\begin{cases} e_{\max}^1 = p_u - p_{\sigma(n-2)}, \\ e_{\max}^2 = p_v - p_{\sigma(n-2)}. \end{cases} \quad (7)$$

Obviously, expandable errors $e_{\max} \in \{0, 1\}$ in IPVO-based PEE would turn to error pairs $\{(e_{\max}^1, e_{\max}^2)/e_{\max}^1 - e_{\max}^2 \in \{0, 1\}\}$. In this way, the modification rule of IPVO-based PEE is described as a 2D mapping in Figure 1(a), for which only one pairing error is modifiable.

To better utilize the largest three pixels, another 2D mapping is proposed as shown in Figure 1(b). Correspondingly, p_u and p_v will be modified after embedding. For this new mapping, six types of error pairs or error-pair transforms are defined as shown in Figure 2. Among them, Type-D error pair is the most valuable one since it introduces the least distortion per embedding one bit, followed by Type-A, Type-B, Type-C, Type-E, and Type-F error pairs, respectively. Based on such consideration, it is quite necessary to optimize the distribution of various types of error pairs. For example, Type-D error pair consists of two expandable errors. As we can learn from IPVO, more expandable errors can be generated with pixel location involved.

3. Proposed Scheme

The proposed method is presented in this section. Firstly, the framework of two-layer embedding is introduced and a new calculation of local complexity is presented. Then, error-pair generation and adaptive definition of spatial location are introduced. Finally, multiple pairwise PEE and implementation details are presented.

3.1. Two-Layer Embedding. Existing PVO-based schemes are generally implemented by processing blocks in raster-scan order, i.e., from top to bottom and left to right. For each block, its right and bottom neighbors are always recovered before it at decoder such that they can serve as its context. To obtain a better context so as to achieve better pixel block selection, two-layer embedding is introduced into the proposed scheme. As shown in Figure 3(a), all pixels except boundary ones are divided into nonoverlapped blocks denoted by “shadow” and “blank,” respectively. During data embedding, shadow blocks are first embedded with half of the secret data and blank blocks are embedded with the remaining half later. At decoder, blank blocks are first recovered after data extraction. Then, shadow blocks are similarly processed later.

Referring to Figure 3(a), with the nearest four blank blocks serving as the context of a shadow block, a full-enclosed context is thus constructed. The first advantage of the new framework is to more precisely compute the local complexity of each block. In this paper, a complexity measurement NL_i is computed considering the absolute difference between two consecutive context pixels. As shown in Figure 3(b), NL_i is computed as the sum of all these differences.

3.2. Adaptive Error-Pair Generation. For each selected shadow or blank block, the largest three pixels are utilized to generate an error pair. It has been verified by IPVO that more expandable errors can be derived from two correlated pixels by predicting the one having relatively smaller location. Therefore, with $u = \min(\sigma(n-1), \sigma(n))$, $v = \max(\sigma(n-1), \sigma(n))$, two pairing errors e_{\max}^1 and e_{\max}^2 are computed as

$$\begin{cases} e_{\max}^1 = p_{r1} - p_{s1}, \\ e_{\max}^2 = p_{r2} - p_{s2} \end{cases}, \quad (8)$$

where $r1 = \min(u, \sigma(n-2))$, $s1 = \max(u, \sigma(n-2))$, $r2 = \min(v, \sigma(n-2))$, $s2 = \max(v, \sigma(n-2))$. According to IPVO we have expandable errors $e_{\max}^1 \in \{0, 1\}$ and $e_{\max}^2 \in \{0, 1\}$ here. By combining them into a pair for pairwise PEE, the initial 2D mapping is obtained and shown in Figure 4.

To make a comparison between the new 2D mapping in Figure 4 (denoted by f) and the original one in Figure 1(b) (denoted by f_o), the image Lena is divided into blocks with the size of 4×4 . Then, the numbers of all types of error pairs are computed and presented in Table 1. As it is shown, the new 2D mapping f not only achieves capacity improvement, but also introduces less distortion per embedding one bit.

According to f , all of $(0, 0)$, $(1, 0)$, and $(1, 1)$ are Type-D error pairs. Specifically, the correlation between $p_{\sigma(n-2)}$ and p_u is exploited to generate e_{\max}^1 while the one between $p_{\sigma(n-2)}$ and p_v is exploited to generate e_{\max}^2 . To generate more expandable e_{\max}^1 or e_{\max}^2 such that more Type-D error pairs can be obtained, the probability of $u < v < \sigma(n-2)$ should be promoted. To accomplish this, we propose to adaptively define the spatial location of pixels within block. Suppose the cover image (denoted by I) is divided into a set of $h \times w$ sized blocks, for each of which we propose to use horizontal and vertical gradients to estimate the distribution of pixels within block and thus determine how to collect them. With the upper-left pixel located at (i_0, j_0) , horizontal gradient G_h and vertical gradient G_v are computed as

$$\begin{aligned} G_h &= \frac{1}{h} \sum_{i=i_0}^{i_0+h-1} (I(i, j_0 - 1) - I(i, j_0 + w)), \\ G_v &= \frac{1}{w} \sum_{j=j_0}^{j_0+w-1} (I(i_0 - 1, j) - I(i_0 + h, j)). \end{aligned} \quad (9)$$

For example, every pixel is quite likely to have a right neighbor having smaller gray value when $G_h > 0$. In this case, we should collect pixels from left to right so as to ensure pixel having relatively smaller location has relatively larger gray value. Similarly, every pixel is quite likely to have a bottom neighbor having smaller gray value when $G_v > 0$. In that case, we should collect pixels from top to bottom. By combining G_h and G_v , eight modes of spatial location definition are summarized and presented in Figure 5, taking block with the size of 3×4 as an example. Moreover, how to determine the optimal mode is also given in Figure 5.

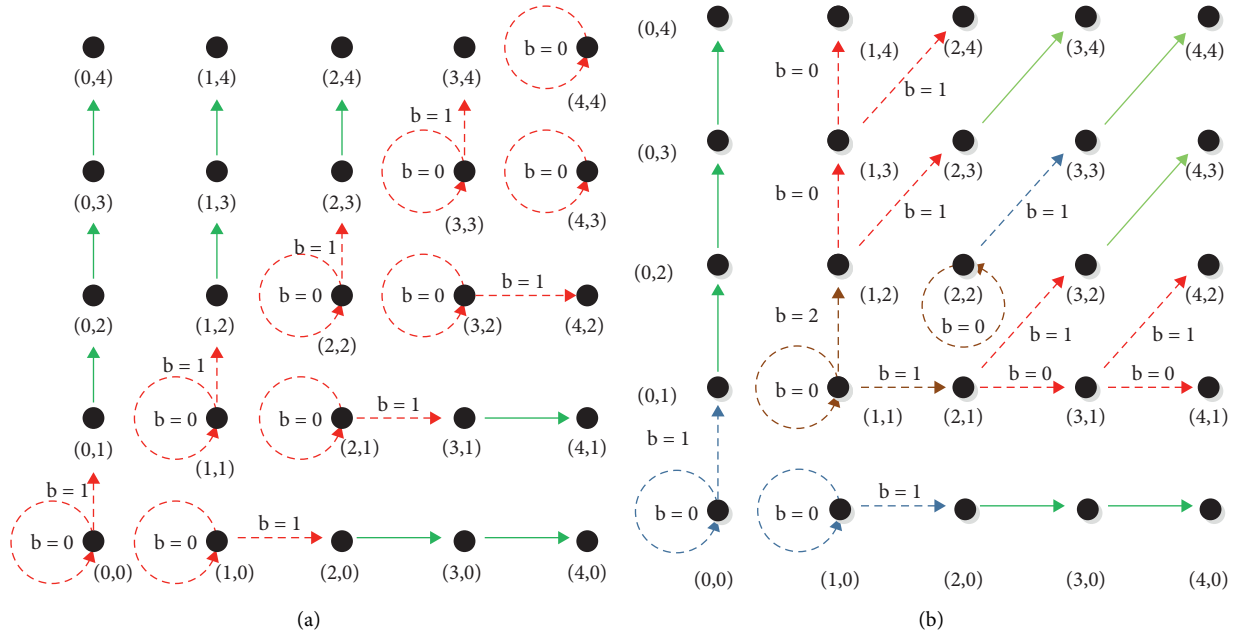


FIGURE 1: 2D mapping of IPVO-based PEE and the one of PVO-based pairwise PEE.

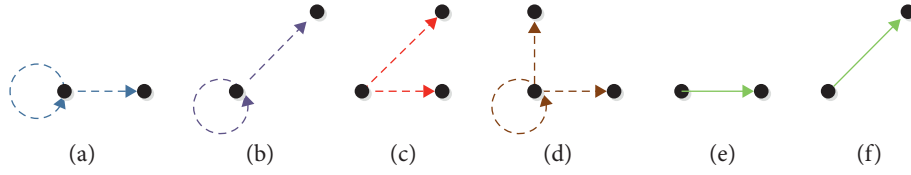


FIGURE 2: Error-pair transforms in PVO-based pairwise PEE.

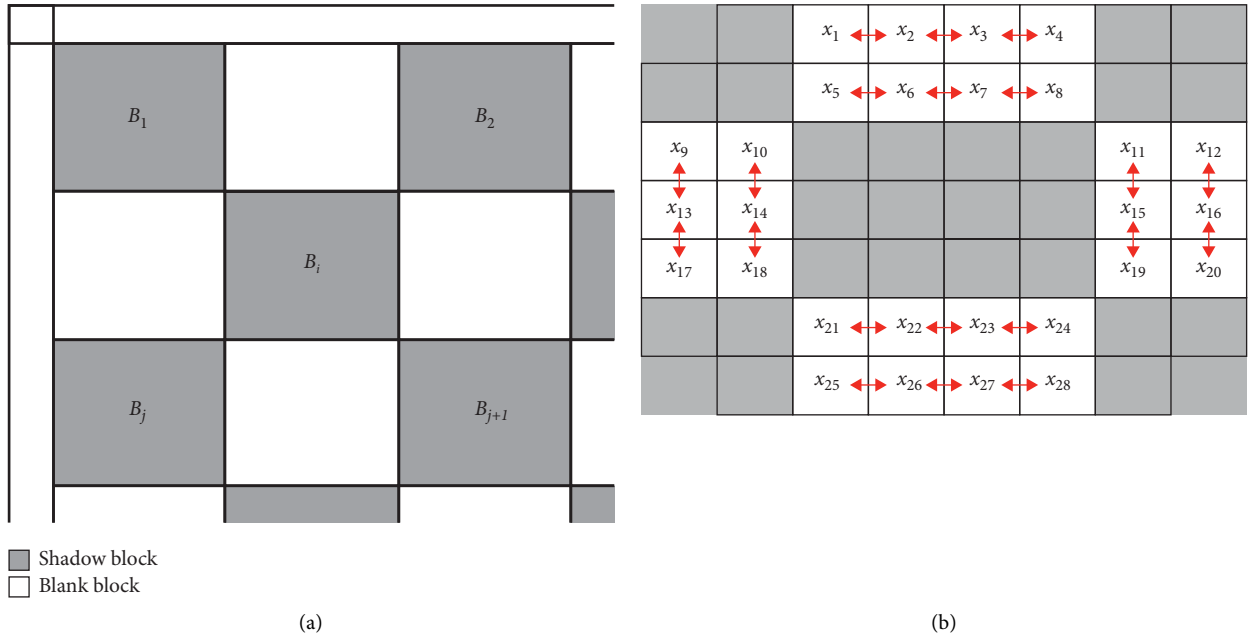


FIGURE 3: Image division and complexity measurement.

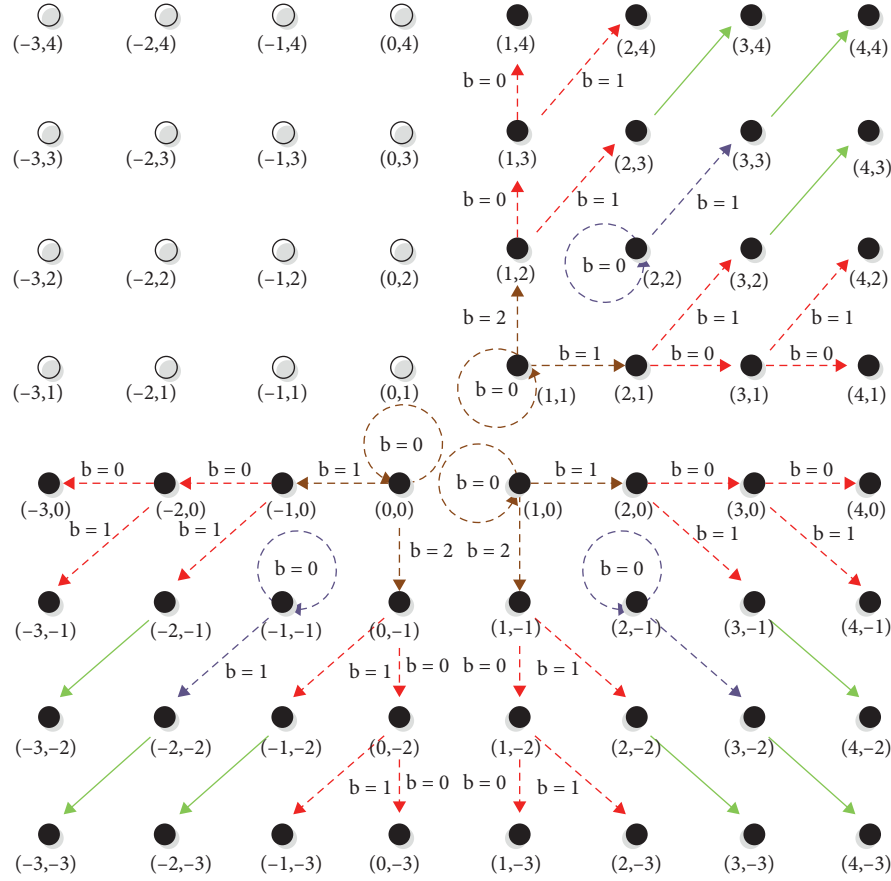


FIGURE 4: Initial 2D mapping for pairwise PEE.

TABLE 1: Performance comparison of various solutions.

Mapping	D	A	B	C	E	F	Capacity	Distortion
f	2444	0	835	6206	0	6645	10 914	25 063
f_o	1497	2283	542	4001	2791	5016	9198	21 506

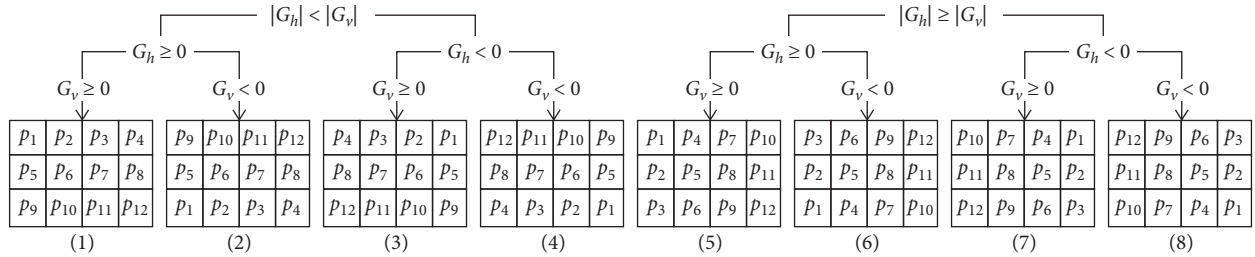


FIGURE 5: Candidate modes of spatial location definition.

Table 2 presents the performance comparison after applying adaptive mode. According to Tables 1 and 2, f_o obtains more Type-A error pairs but fewer Type-E error pairs or specifically more error pairs (1, 0) but fewer error pairs (0, 1). This can be explained by the higher probability of $p_u > p_v$ brought by adaptive mode. As a result, the capacity increases whereas the distortion reduces. It is also seen that adaptive mode is even more beneficial for f . Specifically, we

have more Type-B, Type-C, and Type-D error pairs but fewer Type-F error pairs. So far, with the improvement in error-pair generation, the capacity, which used to be 9198 bits, now turns to be 11 219 bits. Although there is more serious distortion, the proposed scheme introduces less distortion per embedding one bit; i.e., whereas the original performance is measured by 2.338, the new one is measured by 2.213.

TABLE 2: Performance comparison of various solutions after applying adaptive mode.

Mapping	D	A	B	C	E	F	Capacity	Distortion
f	2587	0	855	6264	0	6424	11 219	24 824
f_o	1497	2346	542	4001	2728	5016	9261	21 474

3.3. *Multiple Pairwise PEE*. Notice that the correlation between p_u and p_v is neglected so far. For $\sigma(n-2)$, u , and v , their order can be one of the following three: (1) $\sigma(n-2) < u < v$, (2) $u < \sigma(n-2) < v$, and (3) $u < v < \sigma(n-2)$. Then, in Case (3) we have $e_{\max}^1 = p_u - p_{\sigma(n-2)} \geq 1$ and $e_{\max}^2 = p_v - p_{\sigma(n-2)} \geq 1$. By additionally considering the correlation between p_u and p_v , one can see that e_{\max}^1 is shifted in case of $p_u > p_v$. In this situation how e_{\max}^2 is utilized for data embedding should be related to e_{\max}^1 considering the similarity between them; i.e., the larger e_{\max}^1 is, the more likely e_{\max}^2 is shifted. In this paper, we propose to adaptively determine the predicted value of p_v and recalculate e_{\max}^2 as

$$e_{\max}^N = p_v - \max(p_{\sigma(n-2)}, p_u - \eta), \quad (10)$$

with a parameter η . To fulfill data embedding, e_{\max}^N is modified as

$$e_{\max}^{N'} = \begin{cases} e_{\max}^N + b & \text{if } e_{\max}^N = 1 \\ e_{\max}^N + 1 & \text{if } e_{\max}^N > 1 \\ e_{\max}^N & \text{if } e_{\max}^N < 1 \end{cases}, \quad (11)$$

and accordingly p_v is modified to

$$p_v' = \begin{cases} p_v + b & \text{if } e_{\max}^N = 1 \\ p_v + 1 & \text{if } e_{\max}^N > 1 \\ p_v & \text{if } e_{\max}^N < 1 \end{cases}. \quad (12)$$

Given that $p_u - \eta > p_{\sigma(n-2)}$ (i.e., $e_{\max}^1 = p_u - p_{\sigma(n-2)} > \eta$), we have $e_{\max}^N = p_v - (p_u - \eta)$. That is to say, if $e_{\max}^1 \leq \eta$ the correlation between $p_{\sigma(n-2)}$ and p_v is likely to produce expandable error. Otherwise, we turn to utilize the correlation between p_v and p_u with $p_u - \eta$ serving as the predicted value of p_v . By varying η to cover every similarity between pairing errors, pairwise PEE will become more comprehensive. Similarly, e_{\max}^2 is shifted in case of $p_v > p_u$. In that situation $p_v - \eta$ serves as another candidate predicted value of p_u and we recalculate e_{\max}^1 as

$$e_{\max}^N = p_u - \max(p_{\sigma(n-2)}, p_v - \eta). \quad (13)$$

Then, $e_{\max}^{N'}$ is obtained according to (11) and p_u is similarly modified according to (12). Given that $p_v - \eta > p_{\sigma(n-2)}$ (i.e., $e_{\max}^2 > \eta$), we have expandable error

$e_{\max}^N = p_u - (p_v - \eta) = (p_u - p_{\sigma(n-2)}) - (p_v - p_{\sigma(n-2)}) + \eta = e_{\max}^1 - (e_{\max}^2 - \eta) = 1$. That is to say, one data bit will be embedded into error pairs $\{(e_{\max}^1, e_{\max}^2) | e_{\max}^2 - e_{\max}^1 = \eta - 1\}$. This can be equivalent to expansion bins selection. Table 3 presents the evolution of all types of error pairs from Case (3). The same evolution can be performed on error pairs from Cases (1) and (2) as well. In this way, the original 2D mapping is extended into f_η for $\eta \geq 2$. One can verify that f_η includes the original 2D mapping f as a special case for taking $\eta = 255$.

3.4. *Multiple Pairwise PEE Embedding*. For each block, the minimum pixels are also utilized for data embedding. Let $u = \max(\sigma(1), \sigma(2))$, $v = \min(\sigma(1), \sigma(2))$; another two pairing errors e_{\min}^1 and e_{\min}^2 are calculated as

$$\begin{cases} e_{\min}^1 = p_{r1} - p_{s1} \\ e_{\min}^2 = p_{r2} - p_{s2} \end{cases}, \quad (14)$$

where $r1 = \min(u, \sigma(3))$, $s1 = \max(u, \sigma(3))$, $r2 = \min(v, \sigma(3))$, $s2 = \max(v, \sigma(3))$. (e_{\min}^1, e_{\min}^2) is similarly utilized like (e_{\max}^1, e_{\max}^2), and only $p_{\sigma(1)}$ and $p_{\sigma(2)}$ are reduced. By counting the frequency of error pairs (e_{\max}^1, e_{\max}^2) and (e_{\min}^1, e_{\min}^2) derived from blocks with a given complexity, a 2D histogram sequence h_n for $n \geq 0$ is defined as

$$h_n(e_1, e_2) = \#\{e_{\max/\min}^1 = e_1, e_{\max/\min}^2 = e_2, \text{NL}_i = n\}. \quad (15)$$

So far we have histogram sequence $h_n (n \geq 0)$ and mapping sequence $f_\eta (\eta \geq 2)$. The final step is the selection of histograms and mappings and the determination of parameters $s(n)$ which indicates that the histogram h_n is to be processed by the mapping $f_{s(n)}$. For histograms selection, the one with low complexity is preferentially selected. For mappings selection, any combination of various mappings can be attempted. Suppose that T mappings $\{f_{\eta_0}, f_{\eta_1}, \dots, f_{\eta_{T-1}}\} (\eta_0 < \eta_1 < \dots < \eta_{T-1})$ are activated; $s(n)$ takes values from η_0 to η_{T-1} or 0 indicating that histogram h_n is excluded. The determination of optimal $s(n)$ is described as follows. Firstly, the capacity-distortion performance of f_η in manipulating histogram h_n is considered. Let $A_\eta, B_\eta, C_\eta, D_\eta, E_\eta, F_\eta$ denote the types of error pair defined by f_η . The capacity and distortion, denoted by $EC_{pw}(n, \eta)$ and $ED_{pw}(n, \eta)$, can be formulated as

TABLE 3: All types of error pairs of original pairwise PEE and multiple pairwise PEE.

Type	f	$f_\eta (\eta \geq 2)$
A	None	None
B	$e_{\max}^1 = e_{\max}^2 = 2$ $e_{\max}^1 = 1, e_{\max}^2 > 1$	$e_{\max}^1 = e_{\max}^2 = 2$ $e_{\max}^1 = 1, 1 < e_{\max}^2 \leq \eta$
C	$e_{\max}^1 > 1, e_{\max}^2 = 1$	$1 < e_{\max}^1 \leq \eta, e_{\max}^2 = 1$ $ e_{\max}^1 - e_{\max}^2 = \eta - 1$
D	$e_{\max}^1 = e_{\max}^2 = 1$	$e_{\max}^1 = e_{\max}^2 = 1$
E	None	$ e_{\max}^1 - e_{\max}^2 \geq \eta$
F	Rest	Rest

Input: capacity requirement CR, activated mappings $\{\eta_0, \eta_1, \dots, \eta_{T-1}\}$, performance measurement $EC_{pw}(n, \eta)$ and $ED_{pw}(n, \eta)$;

Output: parameters $s(n)$;

```

(1) Function Optimization (CR,  $\{\eta_0, \eta_1, \dots, \eta_{T-1}\}$ ,  $EC_{pw}(n, \eta)$ ,  $ED_{pw}(n, \eta)$ )
(2)    $n = 0, s(n) = \eta_0$ 
(3)   update ( $n$ )
(4)   While CR is not met do
(5)      $[\eta_{n1}, dp\ b_{n1}] = \min(ED_{pw}(n+1, [\eta_0: \eta_{T-1}]) / EC_{pw}(n+1, [\eta_0: \eta_{T-1}]))$ 
(6)     fixed = false
(7)     for  $i = 0 \rightarrow n$  do
(8)       if record( $i, 1$ ) > 0 and record( $i, 2$ ) <  $dp\ b_{n1}$  then
(9)          $s(i) = \text{record}(i, 1)$ , fixed = true
(10)      update ( $i$ )
(11)    End if
(12)  End for
(13)  If fixed == false then
(14)     $n = n + 1, s(n) = \eta_{n1}$ 
(15)    update ( $n$ )
(16)  End if
(17) End while
(18) Return  $s(n)$ 
(19) End function
(20)
(21) Function update ( $k$ )
(22)    $\eta_{\text{set}} = \text{find}(EC_{pw}(k, : ) > EC_{pw}(k, s(k)))$ 
(23)   record( $k, :$ ) =  $[0, 0]$ 
(24)   If Not Empty( $\eta_{\text{set}}$ ) then
(25)      $[\eta, dp\ b] = \min \left( \frac{ED_{pw}(k, \eta_{\text{set}}) - \min ED_{pw}(k, s(k))}{- \min EC_{pw}(k, s(k))} \right)$ 
(26)     record( $k, :$ ) =  $[\eta, dp\ b]$ 
(27)   End if
(28) End function

```

ALGORITHM 1: Optimal parameters determination for multiple pairwise PEE-based RDH.

$$EC_{pw}(n, \eta) = \sum_{e \in B_\eta} h_n(\mathbf{e}) + \sum_{e \in C_\eta} h_n(\mathbf{e}) + \log_2 3 \sum_{e \in D_\eta} h_n(\mathbf{e}), \quad (16)$$

$$ED_{pw}(n, \eta) = \sum_{e \in \bar{B}_\eta} h_n(\mathbf{e}) + \frac{3}{2} \sum_{e \in C_\eta} h_n(\mathbf{e}) + \frac{2}{3} \sum_{e \in D_\eta} h_n(\mathbf{e}) + \sum_{e \in \bar{E}_\eta} h_n(\mathbf{e}) + 2 \sum_{e \in F_\eta} h_n(\mathbf{e}). \quad (17)$$

Then, the complete determination procedure is presented in Algorithm 1.

An example is given here for better illustration. As shown in Figure 6, for such a 2×3 sized block with $NL_i = 21$, $G_h = 3.5$ and $G_v = -1$ are calculated according to (9) and (10). As a result, mode-6 is applied to obtain $(p_1, p_2, p_3,$

$p_4, p_5, p_6) = (179, 188, 185, 177, 178, 182)$. After sorting, there is $(\sigma(1), \sigma(2), \sigma(3), \sigma(4), \sigma(5), \sigma(6)) = (4, 5, 1, 6, 3, 2)$. Then, prediction errors are calculated as $e_{\max}^1 = p_2 - p_6 = 6$, $e_{\max}^2 = p_3 - p_6 = 3$, $e_{\min}^1 = p_1 - p_5 = 1$, and $e_{\min}^2 = p_1 - p_4 = 2$. Suppose that $s(21)$ is determined as 4; error pairs $(6, 3)$ and $(1, 2)$ are to be processed by f_4 . Apparently, both

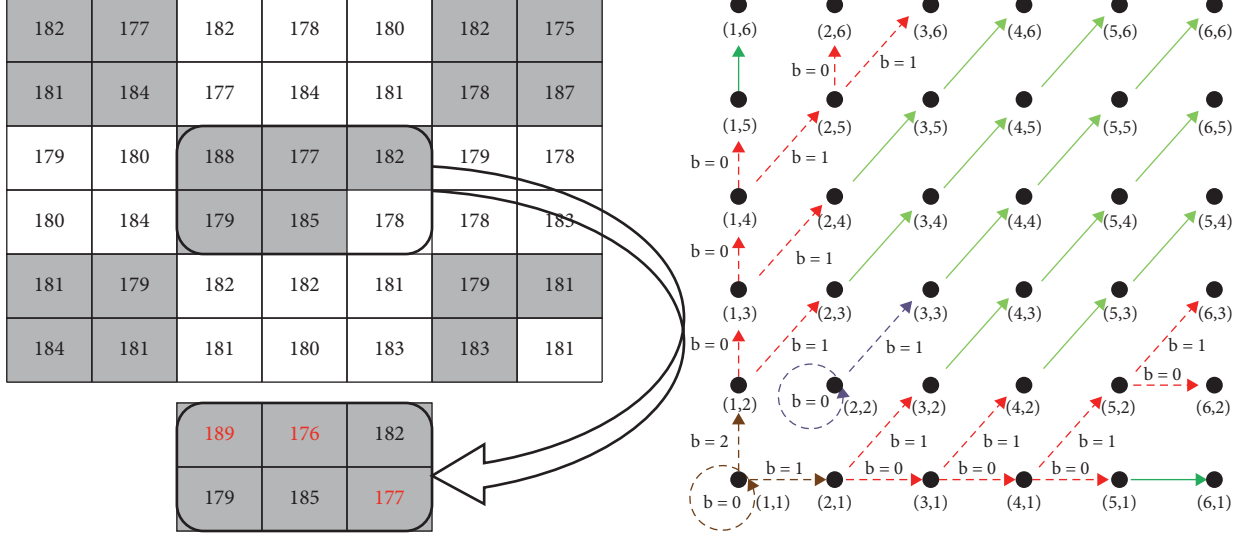


FIGURE 6: Example of multiple pairwise PEE embedding.

error pairs are categorized as Type-C by f_4 . Finally, p_2 is enlarged to 189 while p_3 remains unchanged for embedding one bit valued 0, and p_4 is reduced to 176 while p_5 is reduced to 177 for embedding one bit valued 1.

3.5. Implementation of the Proposed Scheme. With specific block size and parameter T , all pixels except boundary ones are divided into shadow blocks and blank blocks. Then, the embedding and extraction procedures for the shadow layer are given below. Notice that the shadow and blank layers are embedded equally; i.e., each layer is embedded with half of the secret message.

Firstly, to solve the problem of overflow/underflow, pixels are processed in raster-scan order by changing those valued 0 (255) into 1 (254). Here we use a location map record whether a pixel has been changed or not. Then, the location map is compressed using arithmetic coding and we use L_{clm} to denote its length.

Next, optimal $s(n)$ is determined by capacity requirement which refers to the embedding of secret message and auxiliary information. Suppose that histograms with $n < M$ are selected; the transmission of parameters $s(n)$ requires a capacity of $M \times \log_2 T$ bits. Then, the least significant bits (LSBs) of $50 + M \times \log_2 T$ pixels from the first few blocks are recorded. These LSBs and the location map will be embedded along with half of the secret message as the payload.

With the first few ones skipped, shadow blocks are successively processed to embed the payload. For each block, local complexity NL_i and two error pairs (e_{\max}^1, e_{\max}^2) , (e_{\min}^1, e_{\min}^2) are first calculated. If $s(NL_i) \geq 2$, these two error pairs are processed by the 2D mapping $f_{s(NL_i)}$; otherwise, this block is skipped. After embedding the whole payload, the auxiliary information will be embedded by replacing the LSBs of the first $50 + M \times \log_2 T$ pixels. The auxiliary information can be decomposed as follows:

- (i) Block size $h \times w$ (4 bits)

- (ii) Parameters T (4 bits) and M (10 bits)
- (iii) Index of the last processed block (16 bits)
- (iv) Length of the location map (16 bits)
- (v) Parameters $s(n)$ ($M \times \log_2 T$ bits)

At decoder, the auxiliary information is first retrieved from the LSBs of $50 + M \times \log_2 T$ pixels of the first few shadow blocks. Next, with the first few ones skipped, shadow blocks are successively processed. For each block, local complexity NL_i and two marked error pairs $(e_{\max}^{1'}, e_{\max}^{2'})$, $(e_{\min}^{1'}, e_{\min}^{2'})$ are calculated. If $s(NL_i) \geq 2$, these two error pairs are processed by the inverse 2D mapping of $f_{s(NL_i)}$ to extract the embedded data and recover the original pixel values; otherwise, this block is skipped. Finally, the LSB sequence, the compressed location map, and the hidden message are retrieved from the extracted data bits. With the LSB sequence, the LSBs of $50 + M \times \log_2 T$ pixels of the first few shadow blocks are recovered. The location map is decompressed and used to recover a pixel valued 1 (254).

4. Experimental Results

To demonstrate the performance of the proposed scheme, several experiments are conducted in this section. The test images are several gray-scale images with the size of 512×512 . Except Barbara, they are all downloaded from the USC-SIPI1 image database. To achieve the best performance, the embedding procedure is conducted for various block sizes taking $h, w \in \{2, 3, 4, 5\}$. In terms of activated mappings, we consider only the combination of mappings with $\eta \in \{2, 3, \dots, T+1\}$ where T is set as 2, 4, and 8.

To verify the performance of the proposed scheme, some embedding parameters are presented in Table 4. As it is shown, for a given EC of 10,000 bits, the image Lena is divided into 4×3 sized blocks. At error generation stage, mode-8 becomes the most popular one and it is chosen by about 16.5% of blocks in shadow layer and about 16.1% of blocks in blank layer. During shadow-layered embedding

TABLE 4: Parameters of the proposed scheme, for images Lena, Boat, and Lake with different capacities.

Image	Capacity	Block size	T	Layer	M	Mode (1–8)	$f(2-9)$	PSNR
Lena	10,000	4×3	2	Shadow	53	(12.5%, 14.7%, 9%, 10.9%, 10.9%, 15.7%, 10.1%, 16.5%)	(34, 19, 0, 0, 0, 0, 0, 0)	61.15
				Blank	56	(12.4%, 15.2%, 9%, 10.6%, 10.6%, 16%, 10.5%, 16.1%)	(45, 11, 0, 0, 0, 0, 0, 0)	
Boat	20,000	3×3	4	Shadow	140	(14.7%, 13.2%, 11%, 11.7%, 12.7%, 11.9%, 11.1%, 13.3%)	(53, 37, 27, 23, 0, 0, 0, 0)	54.84
				Blank	155	(14.6%, 13.7%, 12%, 12.1%, 12.5%, 11.7%, 10.8%, 12.9%)	(77, 32, 24, 22, 0, 0, 0, 0)	
Lake	30,000	2×3	8	Shadow	216	(11.6%, 14.8%, 12%, 14.4%, 9.3%, 12.9%, 10.3%, 14.4%)	(44, 54, 30, 29, 15, 11, 18, 15)	52.01
				Blank	206	(11.2%, 15.2%, 12%, 14.3%, 9.6%, 12.8%, 10.8%, 14.3%)	(34, 46, 31, 31, 22, 15, 13, 14)	

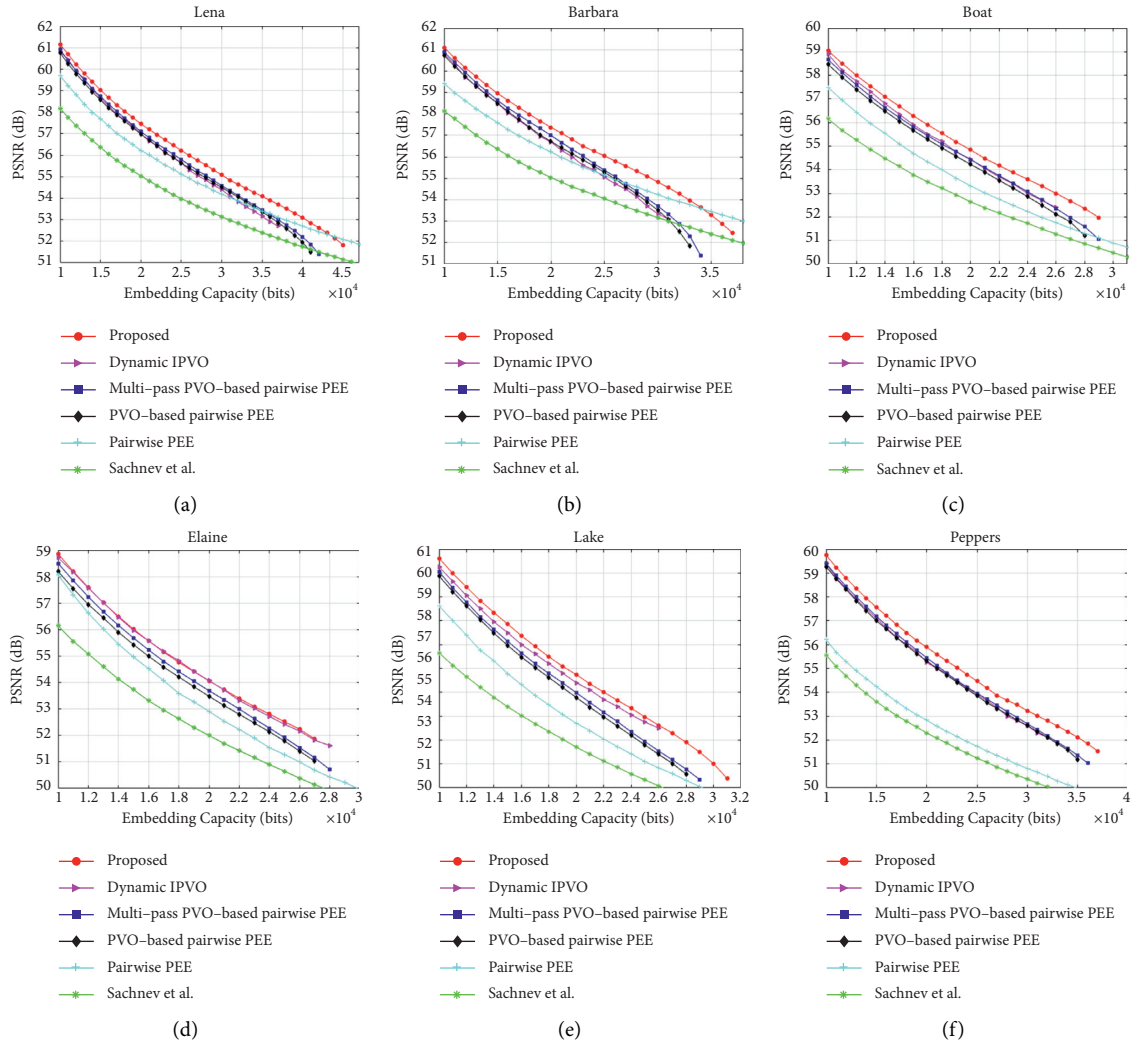


FIGURE 7: Performance comparison with related schemes including Sachnev et al. [17], pairwise PEE [27], PVO-based pairwise PEE [30], multipass PVO-based pairwise PEE [31], and dynamic IPVO [35].

TABLE 5: Comparison of PSNR for the capacity of 10,000 bits, where the unit is dB.

Image	[17]	[27]	[30]	[35]	[31]	Proposed
Lena	58.18	59.70	60.75	60.80	60.91	61.15
Barbara	58.14	59.43	60.73	60.80	60.89	61.10
Boat	56.16	57.49	58.46	58.88	58.66	59.04
Elaine	56.14	58.06	58.20	58.80	58.50	58.86
Lake	56.65	58.62	59.87	60.25	60.05	60.62
Peppers	55.56	56.21	59.27	59.40	59.40	59.75
Average	56.81	58.25	59.55	59.83	59.74	60.07

TABLE 6: Comparison of PSNR for the capacity of 20,000 bits, where the unit is dB.

Image	[17]	[27]	[30]	[35]	[31]	Proposed
Lena	55.04	56.22	56.98	57.00	57.10	57.45
Barbara	55.04	56.23	56.72	56.70	57.00	57.38
Boat	52.64	53.32	54.23	54.45	54.43	54.84
Elaine	52.00	52.91	53.48	54.08	53.70	54.06
Lake	52.70	53.70	54.78	55.40	55.00	55.74
Peppers	52.30	52.83	55.31	55.25	55.45	55.89
Average	53.29	54.20	55.25	55.48	55.45	55.88

histograms with $n < 53$ are selected, of which 34 are processed by f_2 and 19 are processed by f_3 . During blank-layered embedding histograms with $n < 56$ are selected, of which 45 are processed by f_2 and 11 are processed by f_3 . As a result, the highest PSNR reaches up to 61.15 dB.

Then, the proposed scheme is compared with five state-of-the-art schemes [17, 27, 30, 31, 35], as shown in Figure 7. Notice that rhombus prediction is adopted by [17, 27] while PVO prediction is adopted by [30, 31, 35] and the proposed scheme. Figure 7 shows that larger maximum EC is always guaranteed by rhombus prediction. However, PVO-based schemes are generally better in fidelity when capacity is moderate. Rhombus prediction is firstly utilized in 1D PEE [17]. Later, it is incorporated with pairwise PEE to realize 2D PEE [27]. As shown in Figure 7, the comparison of [17, 27] indicates that significant performance improvement is brought by the incorporation with pairwise PEE.

PVO-based pairwise PEE is firstly proposed in [30]. Figure 7 shows that, for some smooth images with large capacity (e.g., Lena with an EC larger than 35,000 bits and Barbara with an EC larger than 27,000 bits), [27] outperforms [30]. In other cases, [30] always outperforms [27]. The proposed scheme can also be regarded as an improved PVO-based pairwise PEE. As shown in Figure 7, the proposed scheme always outperforms [30]. According to Tables 5 and 6, the average gains of the proposed scheme over [30] are 0.52 and 0.63 dB for capacities of 10,000 and 20,000 bits, respectively. The proposed scheme also outperforms [27] in almost all cases. Rhombus prediction guarantees very large capacity on smooth images Lena and Barbara. As a result, on these images the proposed scheme only outperforms [27] with EC not larger than 44,000 and 35,000 bits, respectively.

An improved incorporation of pairwise PEE and PVO prediction is proposed in [31] and it motivates the strategy of multiple pairwise PEE. In [31], the original 2D mapping of [30] is extended to two new ones according to local complexity. However, the gain is limited and the reason mainly

lies in that only error pairs consisting of shiftable error are involved in the 2D mapping evolution. The proposed scheme, by contrast, achieves rather significant performance improvement over [30]. Referring to Tables 5 and 6, the proposed scheme outperforms [31] by 0.33 dB on average for an EC of 10,000 bits and 0.43 dB on average for an EC of 20,000 bits.

The spatial correlation of pixels in a block is similarly exploited in [35]. Although the obtained errors are modified individually, the advantage of [35] is to fully exploit block redundancy by predicting as many pixels as possible. As shown in Figure 7, the proposed scheme outperforms [35] in almost all cases. Referring to Tables 5 and 6, the proposed scheme outperforms [35] by 0.24 dB on average for an EC of 10,000 bits and 0.4 dB on average for an EC of 20,000 bits. Furthermore, there is also significant superiority of the proposed scheme over [35] in capacity.

In summary, the proposed scheme mainly aims to present a better incorporation of pairwise PEE and PVO prediction technologies. Specifically, original PVO-based pairwise PEE [30] is extended into adaptive embedding from two aspects including adaptive error-pair generation and adaptive error-pair modification. To better verify the advantage of adaptive embedding, the comparison of the proposed scheme with [30] on the Kodak2 image database which contains several 512×768 or 768×512 sized images is shown in Figure 8. It shows that the proposed scheme achieves significant superiority over [30] on all kinds of images.

Finally, Table 7 gives the main notations used in this paper and it is worth mentioning that although achievement is achieved through adaptive error-pair generation and adaptive error-pair modification, the proposed scheme still uses only the largest/smallest three pixels in a block for data embedding. In the future work, we will focus on how to predict more pixels in a block and obtain more error pairs.

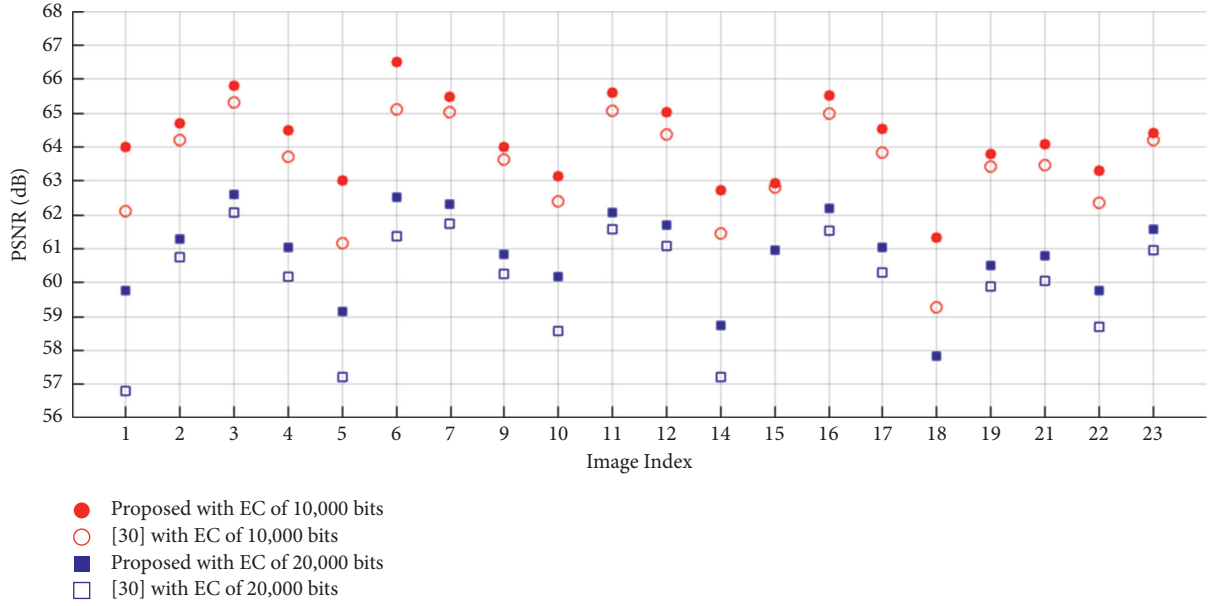


FIGURE 8: Performance comparison between the proposed scheme and [30] on the Kodak image database.

TABLE 7: Main notations used in this paper.

Notation	Description
$p_i, p_{\sigma(i)}$	The i -th pixel in a block before/after sorting
PE_{\max}	Prediction error of PVO-based PEE
e_{\max}	Prediction error of IPVO-based PEE
e_{\max}^1, e_{\max}^2	Prediction error of PVO-based pairwise PEE
e_{\min}^1, e_{\min}^2	
e_{\max}^N	Prediction error after recalculation
NL_i	Complexity measurement of the i -th block
I	Cover image
G_h, G_v	Horizontal gradient and vertical gradient of a block
f, f_o	The 2D mappings in Figures 4 and 1(b)
f_{η}	The parameterized 2D mapping
h_n	2D histogram sequence
$EC_{pw}(n, \eta)$	Capacity performance of using f_{η} to modify h_n
$ED_{pw}(n, \eta)$	Distortion performance of using f_{η} to modify h_n
L_{clm}	Length of the compressed location map

5. Conclusion

In this paper, a novel RDH scheme is proposed by presenting a better incorporation of pairwise PEE and PVO prediction. For error-pair generation, two-layer embedding is applied to obtain full-enclosed pixels. Then, full-enclosed pixels are used to estimate the distribution of pixels within a block and to realize adaptive spatial location definition. In this way, the error-pair generation becomes adaptive, and the distribution of error pairs is thus optimized. For error-pair modification, pairwise PEE is extended to multiple pairwise PEE with which the 2D histogram can be divided into a set of sub-ones for more accurate modification. The experimental results have shown that the proposed scheme outperforms previous state-of-the-art schemes by improving the marked image quality.

Data Availability

Data are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by National Natural Science Foundation of China under Grant 61802074 and in part by Guangdong Basic and Applied Basic Research Foundation (2020A1515010760).

References

- [1] R. Hu and S. Xiang, "CNN prediction based reversible data hiding," *IEEE Signal Processing Letters*, vol. 28, pp. 464–468, 2021.
- [2] I. C. Dragoi and D. Coltuc, "On the security of reversible data hiding in encrypted images by msb prediction," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 187–189, 2021.
- [3] M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, "Lossless generalized-lsb data embedding," *IEEE Transactions on Image Processing*, vol. 14, no. 2, pp. 253–266, 2005.
- [4] J. Fridrich, M. Goljan, and R. Du, "Invertible authentication," *SPIE, Security and Watermarking of Multimedia Contents*, vol. 4314, pp. 197–208, 2001.
- [5] S. K. Lee, Y. H. Suh, and Y. S. Ho, "Reversible image authentication based on watermarking," in *Proceedings of the IEEE International Conference on Multimedia and Expo*, pp. 1321–1324, Toronto, Ontario, Canada, July 2006.
- [6] X. Li, B. Li, B. Yang, and T. Zeng, "General framework to histogram-shifting-based reversible data hiding," *IEEE Transactions on Image Processing*, vol. 22, no. 6, pp. 2181–2191, 2013.
- [7] P. Tsai, Y.-C. Hu, and H.-L. Yeh, "Reversible image hiding scheme using predictive coding and histogram shifting," *Signal Processing*, vol. 89, no. 6, pp. 1129–1143, 2009.
- [8] A. M. Alattar, "Reversible watermark using the difference expansion of a generalized integer transform," *IEEE Transactions on Image Processing*, vol. 13, no. 8, pp. 1147–1156, 2004.
- [9] B. Ou, X. Li, and J. Wang, "Improved pvo-based reversible data hiding: a new implementation based on multiple histograms modification," *Journal of Visual Communication and Image Representation*, vol. 38, pp. 328–339, 2016.
- [10] X. Qu and H. J. Kim, "Pixel-based pixel value ordering predictor for high-fidelity reversible data hiding," *Signal Processing*, vol. 111, pp. 249–260, 2015.
- [11] D. M. Thodi and J. J. Rodriguez, "Expansion embedding techniques for reversible watermarking," *IEEE Transactions on Image Processing*, vol. 16, no. 3, pp. 721–730, 2007.
- [12] H. Wu, X. Li, Y. Zhao, and R. Ni, "Improved ppvo-based high-fidelity reversible data hiding," *Signal Processing*, vol. 167, Article ID 107264, 2020.
- [13] D. Coltuc, "Low distortion transform for reversible watermarking," *IEEE Transactions on Image Processing*, vol. 21, no. 1, pp. 412–417, 2012.
- [14] D. Coltuc and J.-M. Chassery, "Very fast watermarking by reversible contrast mapping," *IEEE Signal Processing Letters*, vol. 14, no. 4, pp. 255–258, 2007.
- [15] X. Xiang Wang, X. Xiaolong Li, B. Bin Yang, and Z. Zongming Guo, "Efficient generalized integer transform for reversible watermarking," *IEEE Signal Processing Letters*, vol. 17, no. 6, pp. 567–570, 2010.
- [16] S. Weng, Y. Zhao, J.-S. Pan, and R. Ni, "Reversible watermarking based on invariability and adjustment on pixel pairs," *IEEE Signal Processing Letters*, vol. 15, no. 1, pp. 721–724, 2008.
- [17] V. Sachnev, H. J. Hyoung Joong Kim, J. Jeho Nam, S. Suresh, and Y. Yun Qing Shi, "Reversible watermarking algorithm using sorting and prediction," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 7, pp. 989–999, 2009.
- [18] L. Lixin Luo, Z. Zhenyong Chen, M. Ming Chen, X. Xiao Zeng, and Z. Zhang Xiong, "Reversible image watermarking using interpolation technique," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 1, pp. 187–193, 2010.
- [19] Q. Pei, X. Wang, Y. Li, and H. Li, "Adaptive reversible watermarking with improved embedding capacity," *Journal of Systems and Software*, vol. 86, no. 11, pp. 2841–2848, 2013.
- [20] B. Ou, X. Li, Y. Zhao, and R. Ni, "Reversible data hiding based on pde predictor," *Journal of Systems and Software*, vol. 86, no. 10, pp. 2700–2709, 2013.
- [21] I.-C. Dragoi and D. Coltuc, "Local-prediction-based difference expansion reversible watermarking," *IEEE Transactions on Image Processing*, vol. 23, no. 4, pp. 1779–1790, 2014.
- [22] R. M. Rad, K. KokSheik Wong, and J. Jing-Ming Guo, "A unified data embedding and scrambling method," *IEEE Transactions on Image Processing*, vol. 23, no. 4, pp. 1463–1475, 2014.
- [23] X. Xiaolong Li, W. Weiming Zhang, X. Xinlu Gui, and B. Bin Yang, "Efficient reversible data hiding based on multiple histograms modification," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 9, pp. 2016–2027, 2015.
- [24] W. Hong, "Adaptive reversible data hiding method based on error energy control and histogram shifting," *Optics Communications*, vol. 285, no. 2, pp. 101–108, 2012.
- [25] X. Xiaolong Li, W. Weiming Zhang, X. Xinlu Gui, and B. Bin Yang, "A novel reversible data hiding scheme based on two-dimensional difference-histogram modification," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 7, pp. 1091–1100, 2013.
- [26] H.-T. Wu and J. Huang, "Reversible image watermarking on prediction errors by efficient histogram modification," *Signal Processing*, vol. 92, no. 12, pp. 3000–3009, 2012.
- [27] B. Ou, X. Li, Y. Zhao, R. Ni, and Y.-Q. Shi, "Pairwise prediction-error expansion for efficient reversible data hiding," *IEEE Transactions on Image Processing*, vol. 22, no. 12, pp. 5010–5021, 2013.
- [28] I.-C. Dragoi and D. Coltuc, "Adaptive pairing reversible watermarking," *IEEE Transactions on Image Processing*, vol. 25, no. 5, pp. 2420–2422, 2016.
- [29] X. Xiaolong Li, B. Bin Yang, and T. Tiejong Zeng, "Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection," *IEEE Transactions on Image Processing*, vol. 20, no. 12, pp. 3524–3533, 2011.
- [30] B. Ou, X. Li, and J. Wang, "High-fidelity reversible data hiding based on pixel-value-ordering and pairwise prediction-error expansion," *Journal of Visual Communication and Image Representation*, vol. 39, pp. 12–23, 2016.
- [31] W. He, G. Xiong, S. Weng, Z. Cai, and Y. Wang, "Reversible data hiding using multi-pass pixel-value-ordering and pairwise prediction-error expansion," *Information Sciences*, vol. 467, pp. 784–799, 2018.
- [32] W. He, Z. Cai, and Y. Wang, "High-fidelity reversible image watermarking based on effective prediction error-pairs modification," *IEEE Transactions on Multimedia*, vol. 23, pp. 52–63, 2021.
- [33] F. Peng, X. Li, and B. Yang, "Improved pvo-based reversible data hiding," *Digital Signal Processing*, vol. 25, pp. 255–265, 2014.
- [34] 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 Processing*, vol. 93, no. 1, pp. 198–205, 2013.
- [35] S. Weng, Y. Shi, W. Hong, and Y. Yao, "Dynamic improved pixel value ordering reversible data hiding," *Information Sciences*, vol. 489, pp. 136–154, 2019.