High-quality reversible data hiding scheme using sorting and enhanced pairwise PEE

Gurjinder Kaur1 | Samayveer Singh1 | Rajneesh Rani1 | Rajeev Kumar2 | Aruna Malik1

1 Department of Computer Science & Engineering, Dr B R Ambedkar National Institute of Technology Jalandhar, Punjab, India
2 Department of Computer Engineering, Delhi Technological University, Delhi, India

Correspondence
Samayveer Singh, Department of Computer Science & Engineering, Dr B R Ambedkar National Institute of Technology Jalandhar, Punjab, India. Email: samayveersingh@gmail.com

Abstract
This paper proposes a novel reversible data hiding (RDH) technique using sorting and pairwise prediction error expansion (PEE) to improve embedding capacity (EC) while retaining the quality of cover image. The proposed scheme traverses alternate pixels of the cover image in a zig-zag order to construct two independent sets for sequential embedding. The pixels of each set are sorted in an increasing order of their rhombus mean followed by a two-pass data embedding by dividing the sets into $1 \times 3$ size blocks based on some predefined criteria. In pass 1, two prediction errors are calculated for the first and the last pixels using their rhombus means; and pairwise mapping is modified and exploited to embed the secret data in such a way that the value of the first pixel is either increased or remains unchanged, and the value of the last pixel is either decreased or remains unchanged. In pass-2, the middle pixel is utilised to predict the first and last pixels and the values of the first pixel and the last pixel are either decreased or remains unchanged and either increased or remains unchanged, respectively. In contrast to some existing recovery-based methods, the proposed pass-2 guarantees to complement the changes made in pass-1, thereby boosting the quality along with increased EC. The experimental results show that the proposed method achieves better embedding performance than the state-of-the-art RDH techniques. More specifically, the proposed method gets an increment by an average of 0.46 dB and 1.01 dB for embedding 10,000 bits and 20,000 bits respectively over its closest prior art.

1 INTRODUCTION

The rapid development in computers and network technology has made it possible to instantly transmit multimedia data over the internet. However, this facility of instant communication comes at a cost of compromising the security of transmitted data. So, to protect the multimedia data from being compromised, additional security mechanisms need to be incorporated into the existing data communication protocols. Information hiding (or data hiding) is one such security mechanism in which the secret data is embedded into a host medium so that the presence of hidden data is imperceptible to everyone. Despite having a strong security edge, the conventional data hiding methods cause permanent damages to the cover media. Due to this reason, these methods are not the best choices for some of the media-sensitive domains such as law enforcement, medical imagery, military, etc. Since these application domains deal with sensitive image operations, an alternative to the conventional data hiding approach is required. To address this concern, the use of reversible data hiding (RDH) seems to be an appropriate option. In RDH, lossless recovery of the cover image is ensured along with the complete extraction of the hidden data. It implies that RDH preserves the integrity of both the cover image and the embedded data. The idea of RDH was pioneered in a US patent [1]. Since then, the growth of research in the field of RDH is increasing monotonically and a broad spectrum of RDH techniques have been proposed in this domain. Various RDH schemes have been devised, paving way for well-known methods, including lossless compression [2, 3], difference expansion (DE) [4, 5], prediction error expansion (PEE) [6–9].

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2021 The Authors. IET Image Processing published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology.
and histogram shifting [10, 11]. Among these methods, PEE is one of the most effective methods due to its embedding performance.

The method of PEE was first introduced by Thodi et al. [6] in which the three-pixel neighbourhood context of a pixel is used to compute its predicted value. The prediction error, calculated using the difference between the original pixel and its predicted value is then expanded to embed the secret data. The PEE method proves to be better than the earliest known variation of expansion-based embedding, that is, the DE method proposed by Tian [4]. The DE technique uses a pair of consecutive pixels and their corresponding difference is expanded to embed the data, which means a pixel pair is treated as a single embedding unit. On the contrary, PEE treats individual pixel as an embedding unit, hence providing scope of increased embedding capacity (EC). Moreover, a high number of small prediction errors are produced by PEE, that construct a sharp prediction error histogram, ultimately enhancing embedding performance. To further stimulate the capacity and quality of the traditional PEE-based RDH, Sachnev et al. [7] introduced a novel PEE scheme by incorporating the sorting of pixels according to their local complexities. They also proposed a new predictor, i.e. ‘rhombus predictor’ to predict the value of a pixel from its four-pixel neighbourhood that led to significantly increase the embeddable prediction errors. Ou et al. [8] introduced pairwise PEE in which a pair of prediction errors is modified simultaneously using 2D prediction error histogram modification strategy. The pairwise PEE exploits the inherent correlation between the pair of prediction errors for a pixel to improve the performance. In pairwise PEE, the expansion of prediction error pair (0,0) is done, such that it becomes (0,0), (0,1) and (1,0) to accommodate \( \log_23 \) data bits into a single pair. Moreover, pairwise PEE expands the error pair (1,1) to (2,2) and itself for additional one bit embedding, whereas this error pair is simply shifted to traditional PEE. Therefore, an increase in EC without disturbing the quality of the image is achieved. To further improve the PEE schemes, various improvements have been proposed in [9, 12].

Li et al. [13] proposed pixel value ordering (PVO), one of the recent advancements of PEE, has gained much popularity and all credit goes to its impressive performance at low and moderate payload capacities. PVO computes the prediction errors using second-minimum and second-maximum pixels in a sorted block and embeds the data into the minimum-valued pixel if the minimum PE is ‘-1’, and maximum-valued pixel if the maximum PE is ‘1’ in a block-wise manner. All other PEs are shifted to guarantee reversibility at the receiver end. Moreover, PVO changes the pixel values only by one at the most, therefore it is able to generate high quality images even in worst case modifications. Peng et al. [14] proposed an extended version of PVO [13] called Improved PVO (I-PVO) and advocated the benefit of considering relative locations of pixels in the original block during error calculation. I-PVO utilises the prediction errors ‘0’ and ‘1’ instead of ‘-1’ or ‘1’ as used in [13]. It resulted in increasing the number of embeddable prediction errors, ultimately leading to enhanced performance. Extending the work of [13, 14], a variety of other similar PVO-based methods [15–24] have come into the picture. The fact of capacity-dependence on the block size is recognised and it is observed that decreasing the block size would increase the capacity. As the PVO-based methods follow a block-wise data hiding approach, a block selection strategy is required to achieve better capacity-distortion ratio. In PVO embedding, the selection of a block is restricted by its complexity value (a measure of roughness/smoothness), where the complexity value is calculated as the difference of middle pixels. As the middle pixels remain unmodified, the receiver also calculates the same complexity during data retrieval. The comparison of complexity value is done with a user-defined threshold to categorise the blocks as embeddable or non-embeddable. Generally, the block is embeddable if its complexity is below a certain value of threshold. Otherwise, the block is named as non-embeddable and skipped. The categorisation of blocks in terms of their complexity is important because non-embeddable blocks could lead to more shifting, hence deteriorating image quality. Thus, block categorisation helps in preventing unnecessary shifting of pixels. However, these methods are more suitable for low to moderate EC as opposed to high capacity.

Recently, Kumar et al. [25] proposed an enhanced pairwise IPVO scheme for attaining high capacity while maintaining a high quality of stego-images. They exploited every pixel of the cover image in a two-phase embedding using rhombus context and pairwise IPVO. They also used a block selection approach based on the rhombus context for three-pixel blocks. In the first pass, the pairwise IPVO for data embedding is used while in the second pass, a recovery-based pairwise embedding is implemented. Although the method achieved almost double the EC than the other PVO based methods, it posed low-quality concerns at smaller payloads. To address the concerns related to increasing the embedding while simultaneously maintaining the image quality, an RDH technique is proposed using the sorting of pixels based on their rhombus context. The proposed scheme utilises a local variance measure to construct blocks of size 1 \( \times \) 3. Also, a novel local complexity measure is used for block selection to prevent unnecessary distortion. The proposed scheme uses the pairwise embedding introduced by Kumar et al. [25] for first-pass embedding, followed by a PEE based embedding in the second pass. The following points highlight the major contributions of the proposed work:

• The proposed scheme uses a zig-zag order traversal of the cover image to construct two independent pixel sets – the plus set and the dot set. Afterwards, a two-pass data embedding is implemented for each of the sets, such that the plus set pixels act as rhombus context for the dot set pixels and vice versa.

• The sorting of pixels is performed in an increasing order of their rhombus means and then 1 \( \times \) 3 size blocks are formed. Reordering of pixels by sorting plays a key role in increasing the EC and minimizing visual degradation as it helps in generating smaller prediction errors. The proposed method is proven to be an effective method for improving the embedding performance as compared to previous methods that do not incorporate sorting. Since the rhombus mean values of the pixels remain unchanged for each set, the same sorting order is generated at the receiver’s side.
• The proposed scheme exploits the local complexity of individual pixels to determine their inclusion into the embedding block. This step ensures that the pixels belonging to a single block come from a common local complexity range that aids in preventing unwanted shifting.

• A two-pass embedding strategy is performed on three-pixel blocks. In pass 1, two prediction errors for the first pixel and the last pixel are computed using their rhombus means. Here, a pairwise embedding is used in which the first pixel is unmodified/increased by 1, and the last pixel is unmodified/decreased by 1. In pass 2, two prediction errors are computed for the first pixel and the last pixel using the middle pixel and PEH is modified for data embedding. Thus, the proposed method successfully overcomes the drawback of recovery-based method of Kumar et al. [25] and other PVO-based methods. In their work, pass-2 does not guarantee recovery of the pixel values modified during pass-1 as in the worst case, a pixel can be modified by ±2. So, the pass-2 in the proposed work is designed specifically to reverse the modifications done in pass-1, with worst case modification of just ±1, hence boosting image quality.

• The middle pixel is also predicted using its rhombus mean and is used for carrying data. Hence, our technique can embed high payloads while simultaneously retaining the image quality.

The remaining paper is structured as follows. In Section 2, related works are discussed. The proposed scheme is explained in Section 3. For performance evaluation and analysis, results are discussed in Section 4. Finally, the concluding remarks are given in Section 5.

2 | RELATED WORKS

In this section, a comprehensive review of the existing RDH-based schemes are discussed. Sachnev et al. [7] proposed an RDH scheme in which the data embedding is performed in a pixel-by-pixel manner using prediction and sorting. They also introduced a ‘rhombus predictor’ such that a pixel is predicted using its rhombus context i.e. four neighbouring pixels. The pixels are divided into two independent sets such that one set is used for embedding while the other is used to calculate predictions. They noticed the performance improvement gain by sorting in Kamstra and Heijmans’ [26] method. Taking inspiration from [26], Sachnev et al. realised that the ordering of pixels needs to be changed to hide more data with less distortion. So, they decided to re-arrange the pixels according to their correlation. Further, they utilised PEE in a pixel-wise manner and experienced much better capacity versus distortion trade-off than another sorting [26] and PEE [6] based techniques of that time. However, the visual quality offered by Sachnev et al. is not at par with today’s well known high-fidelity methods. To overcome the quality-related issues as well as taking advantage of sorting and rhombus context, the proposed method sorts the pixels based on their rhombus means. Furthermore, rhombus prediction is utilised in pass-1 embedding, however, instead of conventional PEE, pairwise PEE has been used in the proposed method to enhance the image quality. The inspiration behind pairwise embedding is taken from the work of Ou et al. [8] in which PEE exploits the inherent correlation between the pair of prediction errors. The pairwise embedding exploits the correlation between the two prediction errors corresponding to maximum and minimum sides. According to a 2D histogram modification scheme used in this method, the error pair (0, 0) can embed log23 data bits by expanding itself to (0, 0), (0, 1) and (1, 0). However, the PE-pair (1, 1) is exploited separately to accommodate 1-bit secret data by expanding itself and (2, 2). The distortion level is brought down significantly, hence providing a high-quality way of data hiding. Xiao et al. [27] and He et al. [28] proposed content-dependent pairwise mappings to further stimulate the performance of pairwise PEE. They challenged the fixed 2D-PEH modification and presented the advantages of flexible histogram modifications. Ou et al. [29] proposed another 2D histogram generation method using geodesic path. In this method, the pixels are adaptively paired according to their local similarity in terms of intensity and spatial distance. It helps in generating prediction error pairs that are more correlated to each other, as opposed to fixed pairing mechanism. A further study of the idea of adaptive prediction error pairing has been conducted by Ou et al. [30] in which they proposed adaptive pixel pairing (APP) and adaptive mapping selection. Its objective is to include only smooth pixels in pairing, while the rough ones are ignored. This decision of adaptively selecting pixel pairs based on their local complexity creates sharper 2D prediction error histogram, hence leading to some performance gains over other pairwise embedding methods.

Pairwise embedding has been further incorporated with PVO [13], that is itself a high-fidelity mechanism for data hiding. PVO-based methods are block-based RDH methods that modify the pixels by ±1 hence resulting in superior visual quality. The pixels in a block are sorted in increasing order of their intensity values. Then two prediction errors are obtained as the differences between smallest and second-smallest pixels, and the differences between the largest and second-largest pixels. Thus, the PEs in the range of 0 to 255 (for maximum valued pixel) and -255 to 0 (for minimum valued pixel) are obtained. In case of maximum prediction error, the peak bin ‘1’ is expanded for embedding and other bins > 1 are shifted to ensure reversibility. Similarly, for the smallest prediction error, the peak bin ‘1’ is expanded to carry data and other bins < -1 are shifted. Peng et al. [14] proposed an I-PVO-based RDH scheme to consider the relative location of pixels in a block during the prediction error calculation. Li et al.’s PVO always generates non-negative differences for the largest pixel and non-positive differences for the smallest pixel. This is overcome by Peng et al. [14] by producing prediction errors in range of -255 to 255. This range enables the encoder to use both ‘0’ and ‘1’ for expansion both for minimum and maximum side predictions. Wang et al. [18] further extended the PVO scheme by experimenting with variable sized blocks. Instead of keeping the block size fixed, they exploited two block sizes i.e. 2 × 2 and 4 × 4 such that smaller blocks are created in smooth regions, and larger size blocks are created in rough regions. This allows an increase in embedding
in highly correlated regions and prevents unnecessary shifting operations in textured areas. The idea of dynamic block partitioning has been studied and further extended in [20, 31, 32] to show improvement. Some of the PVO-based methods such as [33–35] performed partitioning into 1×3 blocks for capacity improvement. It is observed that small block size leads to improved EC but the quality of the stego image is compromised to some extent. Therefore, an efficient block partitioning is needed for an acceptable trade-off between capacity and distortion. Apart from PVO, Dragoi and Coltuc [36] further suggested prediction-error ordering-based RDH method for high-fidelity data embedding. They investigated the significance of prediction error ordering, replacing the conventional PVO by traversing the image into a chessboard pattern. The inclusion of pairwise embedding in PVO schemes is introduced in [12, 22, 37, 38] to improve performance. Dragoi et al. [39] also proposed an improved pairwise PVO-based method in which the blocks are classified on the basis of a local complexity measure, smooth blocks are processed using pairwise PVO, slightly noisy blocks using standard PVO while complex blocks are kept unchanged. Most of the above mentioned PVO methods have been surveyed in [40]. To increase the EC, Kumar et al. proposed two new reversible data hiding schemes using PVO based methods [41, 42]. In [41], Kumar et al. simply reserves a bin of prediction error histogram by initially shifting the pixels so that expansion and contraction can be done to embed additional amount of secret data. Thus, this scheme increases the EC but at the cost of stego-image quality. In [42], Kumar et al. generalised the work of Peng et al. [14] by extending the block size in case of extremely smooth blocks, so that unused pixels such as the middle pixels in [14] are reduced, which in turn increases the EC without much deterioration of the image quality. However, the scope to improve the EC is still there by adopting more efficient block categorization. Also, the regular raster-scan partitioning of the cover image does not promote any improvement. So, the proposed method uses a modified image traversal approach that would calculate the complexity of each pixel based on its neighbouring pixels. The proposed partitioning strategy allows a two-layer embedding in the cover image, ultimately producing improved results.

Kumar et al. [25] recently proposed a high-capacity RDH-based method on enhanced pairwise IPVO using rhombus context. In this method, the cover image is divided into 1×3 blocks by traversing in a zig-zag order. In pass-1, IPVO is utilised for prediction and the PE pairs are modified using 2D-PEH shifting, such that the smallest pixel is decreased/unchanged and the largest pixel is increased/unchanged. In pass-2, the pixels in a block are sorted according to an increasing sequence of their rhombus means and the first and last pixel is used for data embedding using recovery based pairwise PEE. In pass-2, the first pixel is increased/unchanged and the last pixel is decreased/unchanged. Moreover, they also introduce an efficient block categorisation strategy based on the rhombus context and consider pixel-wise complexity in addition to block-level complexity. Although the payload capacity is significantly increased in comparison to existing PVO-based methods, the image quality is inferior at low ECs. The major drawback of Kumar et al.’s method is that they assume that pass-2 embedding would restore the modified pixel values during pass-1 as they assume that pixel with least/highest rhombus mean is most probably the minimum/maximum valued pixel in the block. However, this assumption that would force the worst-case pixel modification to be ±2 might not hold all the time. To overcome the issues pertaining to image quality in Kumar et al.’s method, it is to be ensured that the pass-2 embedding restores the modified pixels during pass-1, hence promoting better performance. Therefore, the two passes have been modified in the proposed method to achieve this objective. In pass-1 of the proposed method, the first pixel is increased/unchanged and the last pixel is decreased/unchanged, while in pass-2, the first pixel is decreased/unchanged and the last pixel is increased/unchanged. Hence, pass-2 is the complement of pass-1 with worst-case modification to be just ±1 thus mimicking the high-fidelity behaviour offered by PVO.

3 PROPOSED METHOD

In this section, the working of the proposed method is explained. An abstract step-by-step demonstration of the proposed method is shown in Figure 1 in which the steps in shaded block is used to calculate its rhombus mean $\mu_{i,j}$ as in

\[
\mu_{i,j} = \frac{\sum_{k=1}^{N} (x_{i+k,j} + x_{i,j+k})}{2N}
\]

where $x_{i,j}$ is the pixel at position $(i,j)$ in the dot sequence, its rhombus context as shown in Figure 3, is used to calculate its rhombus mean $\mu_{i,j}$ as in

\[
\mu_{i,j} = \frac{\sum_{k=1}^{N} (x_{i+k,j} + x_{i,j+k})}{2N}
\]

The dot set pixels are collected in a sequence according to a zig-zag fashion as depicted by the arrows in Figure 2. For each pixel $x_{i,j}$ in the dot sequence, its rhombus context as shown in Figure 3, is used to calculate its rhombus mean $\mu_{i,j}$ as in

\[
\mu_{i,j} = \frac{\sum_{k=1}^{N} (x_{i+k,j} + x_{i,j+k})}{2N}
\]

where $x_{i,j}$ is the pixel at position $(i,j)$ in the dot sequence, its rhombus context as shown in Figure 3, is used to calculate its rhombus mean $\mu_{i,j}$ as in

\[
\mu_{i,j} = \frac{\sum_{k=1}^{N} (x_{i+k,j} + x_{i,j+k})}{2N}
\]

The data embedding and data extraction processes are then explained in Sections 3.2 and 3.3, respectively.

3.1 Pre-processing Stage

3.1.1 Image partitioning

The cover/stego image is partitioned to form a chessboard pattern to obtain two independent pixel sets namely, dot set and plus set, as shown in Figure 2. This alternate-pixel traversal scheme ensures that the changes made to one pixel-set do not affect the other pixel-set during embedding. The data embedding is independently performed in the dot-set pixels, followed by the plus-set pixels. The following procedure applies to dot set pixels and a similar set of steps is repeated for plus set pixels afterwards.

3.1.2 Zig-zag traversal of pixels and rhombus mean calculation

The dot set pixels are collected in a sequence according to a zig-zag fashion as depicted by the arrows in Figure 2. For each pixel $x_{i,j}$ in the dot sequence, its rhombus context as shown in Figure 3, is used to calculate its rhombus mean $\mu_{i,j}$ as in
3.1.3 Sorting of pixels according to rhombus mean

The pixels are sorted in increasing sequence of their rhombus mean so that pixels in a single block are correlated and do not diverge as much to create non-embeddable prediction error scenarios. The rhombus means, \( \mu_{i,j} \) calculated using Equation (1) is supposed to achieve an appropriate sorting as the value of \( \mu_{i,j} \) remains unchanged after data hiding, that would not pose any difficulty at the decoder side.

3.1.4 Block partitioning scheme

The sorted pixel sequence (according to rhombus mean) is segregated into blocks of size 1 × 3. However, the decision to include a pixel into a 1 × 3 block is determined by its noise level. The pixels with smooth neighbourhood are basically chosen for block construction. A local variance measure employed in the Sachnev et al. [7] is used to quantify the noise level in the pixel neighbourhood of the proposed scheme to determine the inclusion of the pixel into an embedding block. According to this method, the noise level NL for pixel \( p_{i,j} \) is computed as follows:

\[
NL(p_{i,j}) = \frac{1}{4} \sum_{k=1}^{4} (\Delta d_k - \Delta d)^2,
\]

(2)

Where, \( \Delta d_1 = |p_{i+1,j} - p_{i,j-1}| \), \( \Delta d_2 = |p_{i,j+1} - p_{i,j-1}| \), \( \Delta d_3 = |p_{i-1,j} - p_{i,j+1}| \) and \( \Delta d = (\Delta d_1 + \Delta d_2 + \Delta d_3 + \Delta d_4)/4 \). The pixels with NL less than a threshold value are chosen to be members of blocks, and others are skipped. If \( t_1 \) is a user-defined threshold, then a 1 × 3 block is formed as follows:

\[
B = \{p_1, p_2, p_3\} \text{ such that } NL(p_1), NL(p_2), NL(p_3) < t_1.
\]

The value of \( t_1 \) is initially chosen as a small random number. Small values of threshold means that smoothly textured blocks are created, and the ones with rough texture are not
considered initially. However, if the requested EC is not achieved, the value of \( t_1 \) is incremented iteratively to increase the number of candidate blocks until the capacity is satisfied. The relationship between the values of the threshold and embedding performance is straightforward. Smaller values of threshold correspond to higher quality but may not be able to embed the required quantity of secret data. On the other side, larger values of threshold can embed relatively large quantity of data but often compromise the image quality. Hence, the smallest possible value of threshold that is able to hide the required payload is considered as the best option.

### 3.1.5 Block selection scheme

As the proposed scheme uses the rhombus predictor for data embedding, a suitable block-selection strategy is required to generate smaller prediction errors and prevent unnecessary distortion caused by large prediction errors. To accomplish this, another pixel-complexity measure is calculated as shown in Equation (3):

\[
I(p_{i,j}) = \frac{|p_{i,j-1} - \mu_{i,j}| + |p_{i-1,j} - \mu_{i,j}| + |p_{i,j+1} - \mu_{i,j}| + |p_{i,j+1} - \mu_{i,j+1}|}{4}.
\]

(3)

Here \( \mu_{i,j} \) is the rhombus mean of the pixel \( p_{i,j} \) and \( p_{i,j-1} \), \( p_{i-1,j} \), \( p_{i,j+1} \), \( p_{i,j+1} \) are the rhombus neighbours of the pixel \( p_{i,j} \) as shown in Figure 2. The measure \( I(p_{i,j}) \) denotes the average of the sum of absolute differences between the rhombus neighbours and rhombus mean of the pixel \( p_{i,j} \). In other words, \( I(p_{i,j}) \) is the measure of ‘average difference’ between the current pixel and its rhombus context. Small values of \( I(p_{i,j}) \) would correspond to high correlation (smooth texture) and large values of \( I(p_{i,j}) \) would correspond to low correlation (rough texture).

Thus, the block \( B = \{p_1, p_2, p_3\} \) is either categorised as a smooth block or probably complex block according to the pixel-complexity measure defined above. The block categorisation is done as follows:

\[
\text{Category}(B) = \begin{cases} 
S & \text{if } I(p_1) < t_2 \text{ & } I(p_3) < t_2 \\
C & \text{otherwise}
\end{cases}
\]

(4)

Here, \( S \) represents a smooth block, \( C \) represents a probably complex block, and \( t_2 \) is a user-defined threshold. In other words, a smooth block is encountered if the pixel-complexity of the first pixel as well as the last pixel are less than a predefined threshold. A random value is chosen for \( t_2 \) in the beginning, and incremented iteratively if the required EC is not achieved. Smooth blocks are only included in data embedding whereas rough ones are skipped. The use of rhombus-context and local variance in the block partitioning and categorisation plays a significant role in improving the capacity-distortion trade-off.

### 3.2 Data embedding

The data is embedded into \( 1 \times 3 \) size smoothly categorised blocks, according to the above-mentioned block selection criteria. As the pixels in a block are already sorted according to their rhombus mean, the proposed embedding scheme exploits this rearrangement of pixels to efficiently hide the secret data. A two-pass data embedding is employed by the proposed method. In pass-1, the first pixel and last pixel are predicted using their respective rhombus mean values. A pairwise PEE is put to use in such a way that the first pixel is increased/unchanged and the last pixel is decreased/unchanged. To embed the data, a 2D-PEH mapping shown in Figure 4 is utilised. In pass-2, the first pixel and the last pixel are predicted using the middle pixel, followed by PEE to embed the data. As a result, the first pixel is decreased/unchanged and the last pixel is increased/unchanged. Thus, the pass-2 embedding is intentionally designed to possibly reverse the modifications done in the pass-1, so that the original pixel values are recovered along with providing additional capacity. Moreover, the middle pixel is also predicted using its rhombus mean to further stimulate the EC. This leads to modifying the pixels just by \( \pm 1 \), hence ensuring high image quality. The pass-1 and pass-2 embedding procedures are explained in detail in the subsequent sections.

#### 3.2.1 Pass-1 data embedding using rhombus predictor

In pass-1 embedding, PEE expansion is utilised for data hiding. Consider a block \( B = \{p_1, p_2, p_3\} \) where \( \mu_1, \mu_2, \mu_3 \) are the rhombus means of the pixels \( p_1, p_2, p_3 \) respectively, such that
\( \mu_1 \leq \mu_2 \leq \mu_3 \). Then the first pixel \( p_1 \) and the last pixel \( p_3 \) are predicted using their rhombus mean \( \mu_1 \) and \( \mu_3 \) respectively. The prediction errors are calculated as follows:

\[
E_{11} = p_1 - \mu_1. \tag{5}
\]

\[
E_{13} = p_3 - \mu_3. \tag{6}
\]

Here, in the prediction error notation \( E_{pq} \), \( p \in \{1, 2\} \) represents the embedding pass-number and \( q \in \{1, 2, 3\} \) represents the pixel that is being predicted. The prediction errors are treated as a pair \((E_{11}, E_{13})\) and the secret data is embedded according to the pairwise mapping inspired from the work of Kumar et al. [25] as shown in Figure 4.

According to this mapping scheme, one block can accommodate \( \log_2 3 \) bits if \((E_{11}, E_{13}) = (0, 0)\), and 1 bit can be embedded if either \( E_{11} = 0 \) or \( E_{13} = 0 \). In this pass, the first pixel \( p_1 \) is either increased by 1 or unmodified and the last pixel \( p_3 \) is either decreased by 1 or unmodified. The marked block \( B' = \{p_1', p_2, p_3'\} \) is fed as input to the pass-2 data embedding.

### 3.2.2 Pass-2 data embedding using middle pixel

The marked block obtained from pass-1 is used in an additional layer of embedding in pass-2. In pass-2, the middle pixel \( p_2 \) that was left unused earlier is used for predicting the values of the first and the last pixels. The two prediction errors for \( p_1' \) and \( p_3' \) using \( p_2 \) are as follows:

\[
E_{21} = p_2 - p_1', \tag{7}
\]

\[
E_{23} = p_2 - p_3'. \tag{8}
\]

Then the secret data is embedded using PEE into the first pixel as well as the last pixel. The goal is to reverse the modifications done in pass-1 so that the original pixel values are recovered, eventually enhancing the stego-image quality. To accomplish this, pass-2 embedding is developed such that the first pixel is either decreased by 1 or unmodified, while the last pixel is either increased or unmodified. In the PEH of \( E_{21} \), the peak bin 0 is expanded to hide data, while the prediction errors \( > 0 \) are shifted to the left side. This is illustrated in Figure 5(a). If \( b_1 \in \{0, 1\} \) is the secret data bit to embed, the first pixel is modified as follows:

\[
p_1'' = \begin{cases} 
  p_1' - b_1 & \text{if } E_{21} = 0 \\
  p_1' - 1 & \text{if } E_{21} < 0 \\
  p_1' & \text{otherwise}
\end{cases} \tag{9}
\]

Thus, the value of the first pixel is either decreased by \( b_1 \) or unchanged. In the PEH of \( E_{23} \), peak bin of 0 is expanded while the prediction errors \( > 0 \) are shifted to the right side. This expansion and shifting are shown in Figure 5(b). If \( b_2 \in \{0, 1\} \) is the secret data bit, the last pixel is modified as follows:

\[
p_3'' = \begin{cases} 
  p_3' + b_2 & \text{if } E_{23} = 0 \\
  p_3' + 1 & \text{if } E_{23} > 0 \\
  p_3' & \text{otherwise}
\end{cases} \tag{10}
\]

Thus, the value of the last pixel is either expanded/shifted by \( b_2 \) or remains unchanged. In addition to the first and third pixels, the second pixel is also utilised for data embedding. The value of \( p_2 \) is predicted using its rhombus mean, and the prediction error is calculated as follows:

\[
E_{22} = p_2 - \mu_2. \tag{11}
\]

Here, \( \mu_2 \) is the rhombus mean of the second pixel \( p_2 \) in the block. If \( b_3 \in \{0, 1\} \) is the secret data bit, then the second pixel is modified as follows:

\[
p_2'' = \begin{cases} 
  p_2 - b_3 & \text{if } E_{22} = 0 \\
  p_2 - 1 & \text{if } E_{22} < 0 \\
  p_2 + b_3 & \text{if } E_{22} = 1 \\
  p_2 + 1 & \text{if } E_{22} > 1
\end{cases} \tag{12}
\]

It means that the value of \( p_2 \) would only be updated by \( \pm 1 \). The final marked block obtained after the two-pass embedding would be \( B'' = \{p_1'', p_2', p_3''\} \) i.e. the first and third pixels are updated twice, while the second pixel is updated once during the entire embedding procedure. However, the second-update of first and last pixels probably reverses the modifications.

---

**Figure 5** Prediction error expansion (PEE) and shifting for pass-2 embedding (green bins are expanded for data embedding, red bins are shifted, blue bins are unchanged), (a) prediction error \( E_{21} \) for the first pixel in a block and (b) prediction error \( E_{23} \) for the third pixel in a block.
performed on them, pretty much ensuring that \( p_1'' = p_1 \) and \( p_3'' = p_3 \).

The above-mentioned data embedding procedure is also repeated for the plus set pixels while using the marked dot set as the rhombus context.

3.2.3 Auxiliary information and location map embedding

To support image recovery and blind data extraction at the receiver end, some additional information is also embedded into the cover image along with the secret payload. This auxiliary information is embedded in the first \((16 + \log_2(L_{\text{CLM}}))\) LSBs of the border pixels, which are not incorporated in the actual data embedding process. The original LSBs are recorded in a binary sequence \( s_{(\text{LSB})} \). The details regarding this additional information are given below:

- **Thresholds:** The user-defined thresholds \( t_1 \) and \( t_2 \) are required for block formation and block selection at the receiver end. So, these values are embedded as auxiliary information. Therefore, a total of 16 bits are required for two threshold values.

- **Location Map:** To avoid the problem caused by underflow/overflow, a location map (LM) is constructed and embedded into the cover image. The LM is a binary sequence where each entry is dedicated to an individual block. For each block \( B_k \), the corresponding entry in the LM is set as follows:

\[
\text{LM} (k) = \begin{cases} 
1 & \text{if } \max(B_k) = 255 \text{ or } \min(B_k) = 0 \\
0 & \text{otherwise.}
\end{cases} \tag{13}
\]

Then the compression of LM is done using arithmetic encoding so that the compressed location map (CLM) can be embedded without affecting the true EC. If \( L_{\text{CLM}} \) represents the length of the CLM, then its upper limit for a \( P \times Q \) size image is calculated as \( L_{\text{CLM}} = \frac{(P-1) \times (Q-1)}{3} \). Then \( \log_2(L_{\text{CLM}}) \) bits are required to store the length of the CLM as well as the CLM itself.

- **Length of the Secret Payload:** The length of the hidden secret data is also embedded for accurate extraction at the decoder side. To represent the length of the secret data, it requires \( \log_2(L_S) \) bits, where \( L_S \) is the length of the secret data \( S \) and calculated as \( L_S = \frac{2 \times (P-1) \times (Q-1)}{3} \) for a cover image of size \( P \times Q \).

- **Finally,** the original LSBs of the border pixels denoted by \( s_{(\text{LSB})} \) are embedded after the secret payload in the next available block.

The pseudo-code of the data embedding is given below.

**ALGORITHM 1** Data embedding

**Inputs:** Sorted block (according to rhombus mean) \( B_k = \{p_1, p_2, p_3\} \), \( t_1, t_2 \): thresholds, \( S \): secret data

**Output:** Stego-block \( B''_k \)

// Block Categorisation and LM construction

Determine the type of block \( B_k \) using Equation (4) and set the corresponding entry in the LM using Equation (13).

// Data embedding

if \( B_k \) is a smooth block:

Step 1: Calculate prediction errors for \( p_1 \) and \( p_3 \) using Equations (5) and (6).

Step 2: Perform data embedding according to pairwise mappings shown in Figure 4 and obtain the block \( B''_k = \{p_1', p_2, p_3\} \).

Step 3: Calculate prediction errors for \( p_1' \) and \( p_2 \) using Equations (7) and (8) and embed the secret data using Equations (9) and (10) to obtain \( \{p_1'', p_2', p_3\} \).

Step 4: Calculate prediction error for the middle pixel \( p_2 \) using Equation (11) and perform secret data embedding according to Equation (12) to obtain the marked value \( \mu_2 \).

Step 5: Output the stego block \( B''_k = \{p_1'', p_2', p_3\} \).

else

Step 6: Skip the block \( B_k \).

end

3.3 Data extraction

For data extraction and image recovery at the receiver side, first \((16 + \log_2(L_{\text{CLM}}))\) LSBs of border pixels are read and then replaced by the \( s_{(\text{LSB})} \) sequence. The original LM is obtained by decompressing the CLM, using arithmetic decoding. The data extraction is followed in a reverse fashion, i.e., data is first extracted from the plus set pixels, followed by an additional extraction layer from the dot set pixels. It means that pixels in the plus set are sorted according to their rhombus mean and blocks are partitioned as mentioned in the data embedding procedure. Also, the two-passes are followed in a reverse sequence. In pass-1, data is extracted from the middle pixels. Then the restored middle pixels are used to predict first and last pixels. In pass-2, the rhombus predictor is employed for data extraction from first and last pixels using pairwise PEE. For marked block \( B''_k = \{p_1'', p_2', p_3\} \) the data extraction procedure is explained as follows.

3.3.1 Pass-1 data extraction using middle pixel

To predict the values of the first and the last pixel, the original value of the middle pixel needs to be restored. To do this, the prediction error for middle pixel \( p_2' \) is calculated as follows:

\[
E_2' = p_2' - \mu_2. \tag{14}
\]

Then the data extraction and pixel recovery are done using rules given in Table 1.

The first and the last pixels are now predicted using the restored value of the original middle pixel \( p_2 \) as shown in
TABLE 1 Pass-1 extraction from middle pixel and recovery rules

| $E'_2$ | $< -1$ | $-1$ | 0 | 1 | 2 | $> 2$ |
|-------|--------|------|---|---|---|------|
| $p_2$ | $p_2'$ + 1 | $p_2' + 1$ | $p_2'$ | $p_2' - 1$ | $p_2' - 1$ |         |
| $b$   | $-$    | 1    | 0 | 0 | 1 | $-$  |

TABLE 2 Pass-1 extraction from first pixel and recovery rules

| $E'_2$ | $< -1$ | $-1$ | 0 | $> 0$ |
|-------|--------|------|---|------|
| $p_1'$ | $p_1' + 1$ | $p_1'$ | $p_1'$ | $p_1'$ |
| $b$   | $-$    | 1    | 0 | $-$  |

Equations (15) and (16). The data extraction rules are illustrated in Tables 2 and 3.

3.3.2 Pass-2 data extraction using rhombus predictor

In pass-2 data extraction, the first and the last pixels of the block are predicted using their corresponding rhombus mean. Two prediction errors are calculated as follows:

$$E'_{21} = p_2 - p_1''.$$  \hspace{1cm} (15)

$$E'_{23} = p_2 - p_3''.$$  \hspace{1cm} (16)

The resultant block $B' = \{ p'_1, p_2, p'_3 \}$ is passed to the pass-2 extraction for further processing.

4 EXPERIMENTAL RESULTS AND DISCUSSIONS

To evaluate the performance of the proposed method and carry out a comparative analysis, the experimental results are demonstrated in this section. The performance is assessed according to two parameters namely, EC and peak-signal-to-noise ratio (PSNR). The EC is defined as the number of bits that the method can hide. On the other side, PSNR is a quality assessment metric measured in decibels (dB). The value of PSNR for a grayscale image is calculated as follows:

$$\text{PSNR} = 10 \log_{10} \left( \frac{255^2}{\text{MSE}} \right).$$  \hspace{1cm} (19)

Here, MSE is the ‘Mean Squared Error’, i.e., the mean of the sum of squared differences between the original cover image and the marked image. The experiments are performed on six
standard grayscale images of size 512 × 512 including Lena, Baboon, F16, Peppers, Boat and Elaine, downloaded from the USC-SIPI dataset. These images are shown in Figure 7. The experiments are implemented in MATLAB and the secret data is generated using pseudo-random number generators. The results of the proposed method are analysed and compared with some existing state-of-the-art methods including [7, 8, 14, 22, 25, 27, 37, 41, 42].

The proposed method is inclined towards achieving high ECs but not giving up on image quality. So, for performance validation on both aspects, i.e., EC and visual quality, the results are compared with the existing well-known methods [7, 8, 14, 22, 25, 27, 37, 41, 42], each of them directed to achieve either higher EC or higher image quality. Pairwise PEE [8], Sachnev et al. [7], Pairwise PEE with optimal expansion path [27], RDH-bin reservation [41] and enhanced pairwise IPVO [25] provide high ECs, however [7, 8, 25] are vulnerable to lower image quality. On the other hand, methods such as [14, 22, 37, 42] using purely PVO predictors are more likely to stand out in terms of image quality, but they struggle to attain higher ECs. Hence, the choice of these methods is practically reasonable to corroborate the performance of the proposed method.

The experimental results for the EC versus image-fidelity trade-off in a graphical representation are shown in Figure 8. These results have been taken by adjusting the pre-defined thresholds i.e. $t_1$ and $t_2$ to suitable values. Initially, the value of the threshold is set to 0 and then incremented according to the capacity requirements. This iterative approach of identifying a suitable value of threshold plays a prominent role in maintaining the high quality of stego images. The superior performance of the proposed scheme is evident from the results. It is observed that the proposed method offers the highest image quality, irrespective of the EC. The gain in EC is quite significant when compared to PVO methods such as Improved PVO [14], pairwise PEE based on adaptive embedding [22], RDH-block extension [42] and multi-pass PVO based pairwise PEE [37].

Improved PVO [14] is a direct extension of Li et al.’s PVO that considers relative pixel location during prediction to stimulate the generation of embeddable prediction errors. Both [22] and [37] utilise pairwise embedding style to enhance the performance using PVO prediction. These are block-based schemes consider using $2 \times 2$ as the minimal block size and use the difference of middle pixels for block selection. This might not be an efficient block-categorisation approach as it ignores the pixel/block neighbourhood. Also, PVO-based methods are only suitable for low to moderate ECs and suffer from visual degradation at higher ECs. So, the proposed scheme partitions the cover image into smaller $1 \times 3$ blocks and uses rhombus context for block categorisation that acts as a stimulus for improving embedding efficiency.

On another note, although high EC is offered by Enhanced pairwise IPVO using rhombus context [25], it suffers from quality problems at low ECs. This method claims to recover the pixel values during pass-2 embedding, which may not always be the case, hence lead to higher distortion. However, the pass-2 in the proposed method is guaranteed to perform pixel-recovery, owing to better image quality. Pairwise PEE [12] and Sachnev et al. [7], viewed as standard methods, also achieve high ECs but they show low PSNR values. Pairwise PEE and optimal expansion path [27] shows an impressive performance among all other existing methods concerned. In this method, the optimal selection of the expansion bins is responsible for enhanced performance. However, the proposed method also shows an efficient capacity-distortion trade off compared to [27]. The proposed method takes inspiration from sorting and prediction in Sachnev et al. [7] method to sort the pixels based on their rhombus mean values such that highly correlated pixels are brought together. Moreover, the proposed method further takes advantage of pairwise PEE incorporated in the pass-1 embedding, where prediction is done using the rhombus context. The pass-2 embedding is intentionally designed to recover the pixel values modified during pass-1 hence restoring image quality. Additional embedding into the middle pixel also boosts the EC in the proposed method.

Referring to the PSNR values for an EC of 10,000 bits and 20,000 bits as shown in Tables 4 and 5 respectively, it is obvious that the proposed method outperforms the existing methods in terms of visual quality for all test images. Among all test images, the highest PSNR obtained by the proposed method is for F16 which possesses a smooth texture. The proposed scheme also does quite well for a rough image like Baboon with PSNR of 57 dB at EC of 10,000 bits, which is better than existing methods. Pairwise embedding with optimal expansion path [27] is reported to have the best average PSNR among other existing schemes considered for comparison. The proposed method gets an increment by an average of 0.46 dB and 1.01 dB for embedding over 10,000 bits and 20,000 bits respectively [27].

To corroborate the visual quality offered by the proposed method, the stego-images, as well as the image intensity histograms (before and after embedding 15,000 bits) are shown in Figures 9–15, respectively. It is clear from Figure 9 that there is no visible mark of the presence of data in the stego-images.
FIGURE 8  Embedding capacity (EC) versus peak-signal-to-noise ratio (PSNR) of proposed method and existing methods [7, 8, 14, 22, 25, 27, 37, 41, 42]
TABLE 4  PSNR values of the proposed method and existing methods [7, 8, 14, 22, 25, 27, 37] at an EC of 10,000 bits

| Images/methods | Proposed method | Enhanced pairwise IPVO [25] | RDH-bin reservation [41] | RDH-block extension [42] | Multi-pass PVO based pairwise PEH [37] | PVO based adaptive pairwise embedding [22] | Pairwise embedding and optimal expansion path [27] | Improved PVO [14] | Pairwise PEE [8] | Sachnev et al. [7] |
|----------------|-----------------|-----------------------------|-------------------------|------------------------|--------------------------------------|---------------------------------------------|-------------------------------------------|------------------|------------------|------------------|
| Lena           | 60.60           | 58.00                       | 56.00                   | 59.50                  | 61.00                                | 61.00                                      | 61.00                                     | 60.00            | 58.20            |
| Baboon         | 57.00           | 54.50                       | 54.30                   | 56.40                  | 54.65                                | 54.90                                      | 56.30                                     | 53.60            | 54.00            |
| F16            | 64.15           | 60.60                       | 57.00                   | 62.00                  | 63.60                                | 63.90                                      | 64.00                                     | 63.00            | 63.00            |
| Peppers        | 58.80           | 56.30                       | 54.03                   | 58.00                  | 59.40                                | 59.60                                      | 59.50                                     | 59.20            | 56.00            |
| Boat           | 59.20           | 56.40                       | 54.00                   | 57.00                  | 58.70                                | 58.60                                      | 58.50                                     | 58.35            | 57.00            |
| Elaine         | 61.30           | 56.70                       | 55.30                   | 57.00                  | 58.30                                | 58.00                                      | 59.00                                     | 57.40            | 58.00            |
| Average        | 60.18           | 57.08                       | 55.10                   | 58.31                  | 59.28                                | 59.33                                      | 59.72                                     | 58.64            | 58.33            | 56.96            |

TABLE 5  PSNR values of the proposed method and existing methods [7, 8, 14, 22, 25, 27, 37] at an EC of 20,000 bits

| Images/methods | Proposed method | Enhanced pairwise IPVO [25] | RDH-bin reservation [41] | RDH-block extension [42] | Multi-pass PVO based pairwise PEH [37] | PVO based adaptive pairwise embedding [22] | Pairwise embedding and optimal expansion path [27] | Improved PVO [14] | Pairwise PEE [8] | Sachnev et al. [7] |
|----------------|-----------------|-----------------------------|-------------------------|------------------------|--------------------------------------|---------------------------------------------|-------------------------------------------|------------------|------------------|------------------|
| Lena           | 57.45           | 55.00                       | 52.10                   | 55.50                  | 57.20                                | 57.30                                      | 57.40                                     | 56.35            | 56.00            | 55.30            |
| F16            | 60.90           | 59.00                       | 53.80                   | 58.00                  | 59.90                                | 60.00                                      | 60.50                                     | 59.20            | 60.80            | 60.00            |
| Peppers        | 56.00           | 53.80                       | 50.50                   | 54.00                  | 55.60                                | 55.40                                      | 55.50                                     | 55.00            | 52.80            | 52.40            |
| Boat           | 55.60           | 53.50                       | 49.50                   | 53.00                  | 54.30                                | 54.20                                      | 54.10                                     | 53.80            | 53.50            | 53.20            |
| Elaine         | 56.30           | 53.00                       | 51.60                   | 52.30                  | 53.70                                | 53.40                                      | 53.70                                     | 52.80            | 53.00            | 52.00            |
| Average        | 57.25           | 54.86                       | 51.50                   | 54.56                  | 56.14                                | 56.06                                      | 56.24                                     | 55.43            | 55.22            | 54.58            |

The intensity histogram of an image represents the distribution of discrete pixel intensity levels from 0 to 255. Thus, the histogram of the stego image should be almost identical to that of the original cover image so that the presence of secret data is not detected easily. It is quite evident from the comparison of image histograms before and after data embedding, that there are no visible changes to the image. These observations are sufficient to support the fact that the proposed scheme offers a high-quality embedding.

**FIGURE 9**  Stego images. (a) Lena, (b) Baboon, (c) F16, (d) Peppers, (e) Boat and (f) Elaine

5 | CONCLUSION

In this paper, a high quality RDH method based on sorting and enhanced pairwise PEE is presented. The proposed scheme performs sorting of the pixels according to their rhombus mean for efficient block construction, eventually aiming to generate smaller prediction errors so that the sharpness of PEH is increased. Then, the data embedding is done in two passes on $1 \times 3$ blocks where each pass exploits a unique predictor. In the first pass, the rhombus predictor is used for a pairwise embedding such that the first/last pixels are increased/decreased. In the second pass, prediction errors are computed using the middle pixel such that the first/last pixels are decreased/increased. Also, the middle pixel is used for data embedding using PEE. The proposed method overcomes the drawbacks of Kumar et al.’s [25] method by designing a recovery-based pass-2. It means that the pass-2 provides
FIGURE 10  Image histograms of Lena. (a) Intensity histogram of cover image and (b) intensity histogram of stego image

FIGURE 11  Image histograms of Baboon. (a) Intensity histogram of cover image and (b) intensity histogram of stego image

FIGURE 12  Image histograms of F16. (a) Intensity histogram of cover image and (b) intensity histogram of stego image
FIGURE 13 Image histograms of Peppers. (a) Intensity histogram of cover image and (b) intensity histogram of stego image

FIGURE 14 Image histograms of Boat. (a) Intensity histogram of cover image and (b) intensity histogram of stego image

FIGURE 15 Image histograms of Elaine. (a) Intensity histogram of cover image and (b) intensity histogram of stego image
guarantee of no further modification to the pixels that are modified in pass-1 instead some of the pixels are recovered and additional data is embedded which in turn guarantees to improve the image quality. Moreover, the proposed block selection strategy efficiently categorises the blocks based on their neighbourhood correlation which further enhances the embedding performance. From the experimental results, it is evident that the proposed scheme has outperformed several well-known RDH methods in terms of capacity-distortion trade-off.

REFERENCES

1. Honsinger, C.W., et al.: Lossless recovery of an original image containing embedded data. US Patents 6278791B1, 21 Aug 2001
2. Fridrich, J., et al.: Lossless data embedding—New paradigm in digital watermarking. EURASIP J. Adv. Signal Process. 2002(2), 986842 (2002)
3. Celik, M.U., et al.: Lossless generalized LSB data embedding. IEEE Trans. Image Process. 14(2), 253–266 (2005)
4. Tian, J.: Reversible data embedding using a difference expansion. IEEE Trans. Circuits Syst. Video Technol. 13(8), 890–896 (2003)
5. Alattar, A.M.: Reversible watermark using the difference expansion of a generalized integer transform. IEEE Trans. Image Process. 13(8), 1147–1156 (2004)
6. Thodi, D.M., Rodriguez, J.J.: Expansion embedding techniques for reversible watermarking. IEEE Trans. Image Process. 16(3), 721–730 (2007)
7. Sachnev, V., et al.: Reversible watermarking algorithm using sorting and prediction. IEEE Trans. Circuits Syst. Video Technol. 19(7), 989–999 (2009)
8. Ou, B., et al.: Pairwise prediction-error expansion for efficient reversible data hiding. IEEE Trans. Image Process. 22(12), 5010–5021 (2013)
9. Dragoi, I.C., Coltuc, D.: Adaptive pairing reversible watermarking. IEEE Trans. Image Process. 25(5), 2420–2422 (2016)
10. Ni, Z., et al.: Reversible data hiding. IEEE Trans. Circuits Syst. Video Technol. 16(3), 354–362 (2006)
11. Li, X., et al.: General framework to histogram-shifting-based reversible data hiding. IEEE Trans. Image Process. 22(6), 2181–2191 (2013)
12. Ou, B., et al.: High-fidelity reversible data hiding based on pixel-value-ordering and pairwise prediction-error expansion. J. Visual Commun. Image Represent. 39, 12–23 (2016)
13. Li, X., et al.: High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion. Signal Process. 93(1), 198–205 (2013)
14. Peng, F., et al.: Improved PVO-based reversible data hiding. Digital Signal Process. 25, 255–265 (2014)
15. He, W., et al.: Reversible data hiding using multi-pass pixel value ordering and prediction-error expansion. J. Visual Commun. Image Represent. 49, 351–360 (2017)
16. Zhao, W., et al.: A higher efficient reversible data hiding scheme based on pixel value ordering. J. Inf. Hiding Multimedia Signal Process. 9(4), 918–928 (2018)
17. Lu, T.-C., et al.: Pixel-value-ordering based reversible information hiding scheme with self-adaptive threshold strategy. Symmetry 10(12), 764 (2018)
18. Wang, X., et al.: A novel reversible image data hiding scheme based on pixel value ordering and dynamic pixel block partition. Inf. Sci. 310, 16–35 (2015)
19. Weng, S., et al.: Reversible data hiding based on an adaptive pixel-embedding strategy and two-layer embedding. Inf. Sci. 369, 144–159 (2016)
20. Di, F., et al.: High-fidelity reversible data hiding by Quadract-based pixel value ordering. Multimedia Tools Appl. 78(6), 7125–7141 (2019)
21. Weng, S., et al.: Pairwise IPVO-based reversible data hiding. Multimedia Tools Appl. 77(11), 13419–13444 (2018)
22. Wua, H., et al.: Improved reversible data hiding based on PVO and adaptive pairwise embedding. J. Real-Time Image Proc. 16(3), 685–695 (2019)
23. Ou, B., et al.: Pixel-value-ordering based reversible data hiding with adaptive texture classification and modification. In: Yoo, C.D., et al. (eds.) Digital Forensics and Watermarking, pp. 169–179. Springer, Cham, Switzerland (2019)
24. Ou, B., et al.: Reversible data hiding using invariant pixel-value-ordering and prediction-error expansion. Signal Process.: Image Commun. 29(7), 760–772 (2014)
25. Kumar, R., Jung, K.-H.: Enhanced pairwise IPVO-based reversible data hiding scheme using rhombus context. Inf. Sci. 356, 101–119 (2020)
26. Kamstra, L., Heijmans, H.J.A.M.: Reversible data embedding into images using wavelet techniques and sorting. IEEE Trans. Image Process. 14(12), 2082–2090 (2005)
27. Xiao, M., et al.: Reversible data hiding based on pairwise embedding and optimal expansion path. Signal Process. 158, 210–218 (2019)
28. He, W., et al.: Reversible data hiding using multi-pass pixel-value-ordering and pairwise prediction-error expansion. Inf. Sci. 467, 784–799 (2018)
29. Ou, B., et al.: High-fidelity reversible data hiding based on geodesic path and pairwise prediction-error expansion. Neurocomputing 226, 23–34 (2017)
30. Ou, B., et al.: Improving pairwise PEE via hybrid-dimensional histogram generation and adaptive mapping selection. IEEE Trans. Circuits Syst. Video Technol. 29(7), 2176–2190 (2019)
31. Weng, S., et al.: Dynamic improved pixel value ordering reversible data hiding. Inf. Sci., 489, 136–154, (2019)
32. He, W., et al.: Efficient PVO-based reversible data hiding using multistage blocking and prediction accuracy matrix. J. Visual Commun. Image Represent. 46, 58–69 (2017)
33. Kaur, G., et al.: A high capacity reversible data hiding technique based on pixel value ordering using interlock partitioning. In: 2020 7th International Conference on Signal Processing and Integrated Networks (SPIN), Amity University, Noida, India, 27-28 Feb. 2020
34. Jung, K.-H.: A high-capacity reversible data hiding scheme based on sorting and prediction in digital images. Multimedia Tools Appl. 76(11), 13127–13137 (2017)
35. Lee, C.-F., et al.: High-capacity reversible data hiding based on star-shaped PVO method. In: Pan, J.S., et al. (eds.): Recent Advances in Intelligent Information Hiding and Multimedia Signal Processing, pp. 280–288. Springer International Publishing, Cham, Switzerland (2019)
36. Dragoi, I.C., Coltuc, D.: Prediction-error-ordering for high-fidelity reversible data hiding. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2019), Brighton, UK, 12-17 May 2019
37. He, W., et al.: Reversible data hiding using multi-pass pixel-value-ordering and pairwise prediction-error expansion. Inf. Sci. 467, 784–799 (2018)
38. Gao, E., et al.: Reversible data hiding based on invariant pixel-value-ordering and pairwise prediction-error expansion. Neurocomputing 226, 23–34 (2017)
39. Dragoi, L.C., Coltuc, D.: Prediction-error-ordering for high-fidelity reversible data hiding. In: 2018 25th IEEE International Conference on Image Processing (ICIP), Athens, Greece, 7-10 Oct. 2018
40. Kaur, G., et al.: A comprehensive study of reversible data hiding (RDH) schemes based on pixel value ordering (PVO). Arch. Comput. Methods Eng. (2020). https://doi.org/10.1007/s11831-020-05912-3
41. Kumar, R., et al.: I-PVO based high capacity reversible data hiding using bin reservation strategy. Multimedia Tools Appl. 79, 22635–22651 (2020)
42. Kumar, R., et al.: High-fidelity reversible data hiding using block extension strategy. In: 2019 34th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC). Jeju, Korea (South), 23-26 June 2019

How to cite this article: Kaur G, Singh S, Rani R, Kumar R, Malik A. High-quality reversible data hiding scheme using sorting and enhanced pairwise PEE. IET Image Processing. 2021;1–15.
https://doi.org/10.1049/ipt2.12212.