International Journal of System Dynamics Applications
Volume 11 • Issue 2

Robust RDH Technique Using Sorting and IPVO-Based Pairwise PEE for Secure Communication

Aruna Malik, National Institute of Technology, Jalandhar, India
Rajeev Kumar, Kyungil University, South Korea

ABSTRACT

Reversible data hiding (RDH) is used extensively in information-sensitive communication domains to protect the integrity of hidden data and the cover medium. However, most of the recently proposed RDH methods lack robustness. Robust RDH methods are required to protect the hidden data from security attacks at the time of communication between the sender and receiver. In this paper, the authors propose a robust RDH scheme using IPVO-based pairwise embedding. The proposed scheme is designed to prevent unintentional modifications caused to the secret data by JPEG compression. The cover image is decomposed into two planes, namely HSB plane and LSB plane. As JPEG compression most likely modifies the LSBs of the cover image during compression, it is best not to hide the secret data in LSB planes. So, the proposed method utilizes a pairwise embedding to embed secret data into the HSB plane of the cover image. High fidelity improved pixel value ordering (IPVO)-based pairwise embedding ensures that the embedding performance of the proposed method is improved.

KEYWORDS

Pairwise Embedding, Pixel Value Ordering, Prediction Error Expansion, Robust Reversible Data Hiding, Sorting

INTRODUCTION

Due to advancements in digital communication over the internet, the sensitive information is prone to various security attacks. This has led to the need for proposing methods to address the security-related issues. Data hiding is a popular approach for information and data security where the main goal is to safely conceal or hide the secret data into some cover medium such as images, video, or audio. In some specific fields like telemedicine, biometrics, intrusion detect system, and military applications, it is required that the cover medium be not changed while retrieving the hidden data. It is because these applications are very information sensitive and even a small change in the information content can have menacing effects. To overcome this issue, Reversible image data hiding comes into the picture. Reversible image data hiding assures that the original cover image is recovered without causing any modification after we extract the secretly concealed data from it. Besides data security, this approach has also found its applications in watermarking. Watermarking consists of embedding some secret information in an image to preserve its copyright. This hidden information can be

DOI: 10.4018/IJSDA.20220701.oa6

extracted from the image to prove its ownership. The difference between the usage of reversible data hiding in watermarking and information security lies in the fact that in the former, less amount of data is embedded as compared to the latter. Reversible data hiding (RDH) can be used in order to protect the privacy of patient records. The patient information such as personal details, medical history, reports, etc. are sensitive pieces of information and their protection must be ensured when transmitted over internet. It can be used for covert communication in fields such as military, criminal investigations, or other applications requiring transmission of sensitive information over the internet. Another application is the centralized nature of cloud computing makes sensitive data susceptible to security attacks. In order to protect user-data over cloud, data coloring approach along with cloud watermarking is used. RDH can be employed in transmission of satellite information embedded in the satellite images. This ensures the protection of satellite data from unauthorized access. The major challenges to be addressed in any reversible image data hiding technique are to keep the image distortion low and making the payload carrying capacity high at the same time. Sometimes we are just interested in having a balance between the image distortion and data-carrying capacity. And one main feature of any data hiding method that makes it reversible is obvious i.e. reversibility, which means that after data extraction, the original grayscale cover image must be recoverable and there should not be any loss. There is the various application M. et al. (2020); Panda (2019); Bhardwaj (2020); Pierce et al. (2019); 855 (2018); Shukla et al. (2019) which can be implemented for secure data hiding communication.

Many RDH methods have been introduced that work in the compression domain Fridrich et al. (2001), spatial domain Tian (2003); Li et al. (2013a); Ni et al. (2006); Li et al. (2013b); Kim et al. (2008); Ou et al. (2014); Peng et al. (2014); Sachnev et al. (2009), transform domain Battisti et al. (2010), and encryption domain Huang et al. (2016). In the compression domain, cover-image is first losslessly compressed to make vacant space for the secret message Spatial domain methods directly manipulate the pixel values in a reversible manner hide secret data. In the transform domain, some transformations are applied to the image, and then the embedding is performed on the transformed image. In some fields to protect the original cover image from security compromises, image encryption and data hiding are used together i.e. encryption domain. Among these, spatial domain-based methods are the least complex. The spatial domain methods are based on difference expansion Tian (2003); Kim et al. (2008), histogram shifting Li et al. (2013a); Ni et al. (2006), sorting and prediction Sachnev et al. (2009); Kamstra and Heijmans (2005), prediction error expansion Li et al. (2013b); Ou et al. (2014); Peng et al. (2014); Thodi and Rodriguez (2007), pixel value ordering Li et al. (2013b); Peng et al. (2014) etc.

In 2003, difference expansion (DE) method proposed by Tian (2003) 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. This method was a turning point in the history of modern RDH. However, it lacked capacity control and various improvements have been proposed till date. The method of PEE was first introduced by Thodi and Rodriguez (2007) in which three-pixel neighborhood context of a pixel is used to compute its predicted value. Then the prediction error obtained from the difference of the original pixel and its predicted value is expanded to embed the secret data. The PEE method proved to be better than DE. Ou et al. (2013) presented a pairwise embedding style by considering a pair of prediction errors at the same time, called as pairwise PEE. It exploits the inherent correlation among the prediction errors and simultaneously modified the prediction error pair to embed the secret data. In pairwise PEE, the prediction error pair (0, 0) is expanded to (0, 0), (0, 1) and (1, 0) to accommodate $\log_2 3$ data bits into a single pair, while in the conventional PEE, each pair could only embed 1 bit of data. Moreover, pairwise PEE expands the error pair (1,1) to itself and (2,2) to embed 1-bit of data, while this error pair is simply shifted in conventional PEE. Sachnev et al. (2009) introduced a novel PEE-RDH 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 neighborhood. Ni et al. (2006) proposed a histogram shifting technique in

which the image intensity histogram is modified for data embedding. In this method, the peak bin of the histogram is expanded to carry data while other bins are shifted to ensure reversibility. Li et al. (2013a) extended the HS method to propose a general framework of data embedding. Recently several PVO based methods have been proposed that offer impressive visual quality.

The conventional PVO schemes Li et al. (2013b); Ou et al. (2014); Peng et al. (2014) divide the cover image into uniform blocks. The values of optimal block size and complexity threshold need to be searched exhaustively for providing the best performance. The exhaustive search to determine the best values of block size and threshold is a costly and time-consuming operation. To overcome this exhaustive-search problem, dynamic block partitioning based PVO-RDH schemes are introduced Wang et al. (2015); Di et al. (2019); He et al. (2017). Though the dynamic block size does not have a significant impact on the embedding capacity, however, the image quality gets slightly improved. This strategy for dynamically partitioning the image into blocks. This method, known by the name of multi-stage blocking or n-stage blocking, segregate the host image into root blocks or 1-stage blocks. This has led to a significant increment in the embedding capacity by exploiting multi-pass embedding. This method utilizes different block selection criteria. The unused pixels of each block along with the neighborhood pixels are used to calculate the block complexity so that more comprehensive analysis can be done for smooth and complex block classification. These blocks are called leaf blocks or n-stage blocks. The term for remaining blocks is 'intermediate blocks'. In all the above-discussed methods, each block could only embed two data bits (into smallest and largest pixel), irrespective of the block size. Weng et al. (2016, 2019) introduced the concept of dynamically choosing the number of data bits to be embedded in a block depending on the block complexity. In this method, the complexity level of a block is determined first, and then the decision regarding the number of bits to be embedded is made. The incorporation of pairwise PEE into PVO-based methods for improving the stego-image quality is also done Ou et al. (2016); Weng et al. (2018); Wu et al. (2019); He et al. (2018).

The desirable characteristics of any RDH method are embedding capacity, visual quality, and robustness. However, the above-discussed methods only focus on two aspects i.e. embedding capacity and visual quality and do not address an equally important issue of robustness. In digital transmission, the presence of noise is inevitable, and an intruder may try to harm the secret data. To protect the secret data from these modifications during transmission, there is a need of more robust reversible data hiding techniques. The idea of robustness in RDH schemes was proposed by Ni et al. (2008). From then, various other techniques for robust data hiding have been presented to overcome the problem of limited embedding capacity in Ni et al. (2008) method. Zong et al. (2015) proposed a transform domain-based robust RDH method using Guassian low pass filter and random selection of Gray intensities levels for histogram creation. Rajkumar and Vasuki (2019) extended this idea by hiding the data using ASCII values directly instead of binary data. This method increased the embedding capacity. Wang et al. (2017) proposed a significant bit difference expansion (SBDE) based method in which the cover image is decomposed into two planes and embedding is per- formed in higher significant bits. Recently Kumar and Jung (2020) proposed a novel TLE (two-layer embedding) method for RRDH in which a complementary layer of embedding is designed to overcome the loss of quality. This method is explained in the next section. Inspired from the TLE and concept of high-fidelity achieved by pairwise PVO based methods, the proposed method promises to achieve better embedding performance. The main contribution of the proposed work are as follows:

In this paper, a Robust RDH scheme using IPVO based pairwise embedding is proposed. The proposed scheme is designed to prevent unintentional modifications caused to the secret data by JPEG compression. The cover image is decomposed into two planes namely HSB plane and LSB plane. As JPEG compression most likely modifies the LSBs of the cover image during compression, it is best not to hide the secret data into LSB planes. So, the proposed method utilizes a pairwise embedding to embed secret data into HSB plane of the cover image. High fidelity improved pixel value ordering (IPVO) based pairwise embedding ensures that the embedding performance of the proposed method is improved.

Table 1. Predictors used in Kumar and Jung (2020)

Predictor	#1	#2	#3
$\widehat{p}_1$	$x_{\alpha(1)}$	$\frac{x_{\alpha(1)} + x_{\alpha(2)}}{2}$	$\frac{x_{\alpha(1)} + x_{\alpha(2)} + x_{\alpha(3)}}{2}$
$\widehat{p}_2$	$x_{\alpha(4)}$	$\frac{x_{\alpha(3)} + x_{\alpha(4)}}{2}$	$\frac{x_{\alpha(2)} + x_{\alpha(3)} + x_{\alpha(4)}}{2}$

The rest of paper is organized as follows. The review of TLE proposed by Kumar and Jung (2020) is discussed in section 2. The proposed method is explained in Section 3. Experimental results are demonstrated in Section 4 and finally, Section 5 concludes the paper.

REVIEW OF KUMAR AND JUNG (2020) TWO-LAYER EMBEDDING (TLE)

Kumar and Jung (2020) proposed a robust RDH scheme using two-layer embedding. In this method, the cover image is decomposed into planes – HSB & LSB. Data embedding is done in HSB plane by firstly image scanning and sorting the pixels according to their correlation. The image is scanned in chessboard method starting from left to right and top to bottom for secret data embedding. The gray-colored pixels are first used for embedding followed by white. It is making use of the chessboard pattern as the pattern produces independent cells which allows the hider to sort the embeddable pixels for achieving optimum performance. The sorting process virtually arranges the scanned pixels in a sorted sequence (i.e., ascending order) of their local complexity for efficient capacity-distortion trade-off. After sorting the pixels based on their local variance, prediction error needs to be calculated for data embedding. Each pixel is predicted using its four neighbouring pixels (refer to Fig. 2), but prior to predicting the value of $p_{i,j}$, the pixels x_1, x_2, x_3, x_4 are sorted in increasing order of their intensity values to obtain $x_{\alpha(1)}, x_{\alpha(2)}, x_{\alpha(3)}, x_{\alpha(4)}$ where α is a unique one-to-one mapping such that $x_{\alpha(1)} \leq x_{\alpha(2)} \leq x_{\alpha(3)} \leq x_{\alpha(4)}$. According to this, a predictor pair is defined for every pixel as illustrated in Table 1.

Each of the predictor has three variants i.e. Predictor #1, Predictor #2 and Predictor #3. The choice of the predictor is determined initially according to the cover-image characteristics. Then a two-layer embedding is performed on each pixel using two predictors. The prediction error is calculated as:

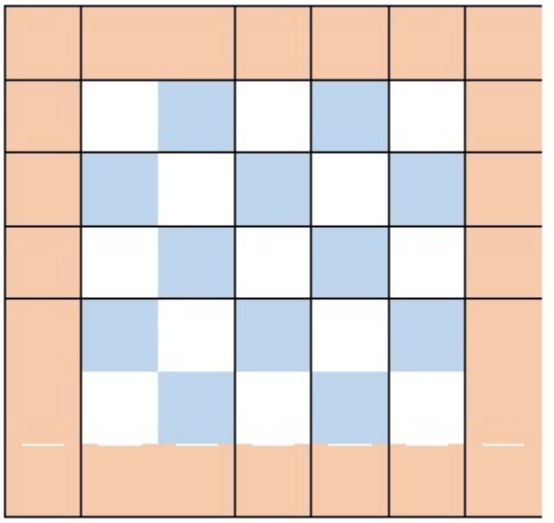
$$e_1 = p_{i,j} - \widehat{p}_1 \tag{1}$$

Then the secret data is embedded. If $s_1 \in \{0,1\}$ is the data bit to hide, then the pixel is modified as follows:

$$p'_{i,j} = \begin{cases} p_{i,j} + s_1 & if \;\; e_1 = 1 \\ p_{i,j} + 1 & if \;\; e_1 > 1 \\ p_{i,j} & otherwise \end{cases} \tag{2}$$

where $p_{i,j}$ is the stego-pixel value after first layer of embedding. The first layer of embedding increases the pixel value by s_1 if the prediction error (e_1) is 1 and by 1 for $e_1 > 1$. Otherwise, the pixel value

Figure 1. Chessboard representation of I-HSB



remains as it is. Thus, one bit can be embedded only if the prediction error is 1, which is the peak bin. For the second layer embedding, the prediction error is calculated as:

$$e_2 = p'_{i,j} - \widehat{p_2} \tag{3}$$

If $s_2 \in \{0,1\}$ is the data bit to hide, then the pixel is modified as follows:

$$p''_{i,j} = \begin{cases} p'_{i,j} - s_2 \ if \ \ e_2 = -1 \\ p'_{i,j} - 1 \ \ if \ \ e_2 < -1 \#(4) \\ p'_{i,j} \qquad otherwise \end{cases} \tag{4}$$

Here, the pixel value is decreased by s_2 if the prediction error (e_2) is -1 and decreased by 1 for $e_2 < -1$. Otherwise, the pixel value remains as it is. Thus, one bit can be embedded only if the prediction error is -1, which is the peak bin.

For blind extraction and image recovery, some of the side/auxiliary information needs to be embedded in the image so that the receiver can start the process instantly. To avoid the problem of overflow/underflow and to lossless extraction of secret message and recovery of the image, the image is pre-processed and a location map (LM) is constructed. This method also maintained the location of last data carrying pixel and predictor number. The index location of the last pixel used for the secret data embedding is safely saved for blind decoding. This method used all the three predictors for secret data embedding. The hider has chosen the predictor which is providing the required performance. So, the selected predictor should be conveyed to the receiver in the auxiliary information. It also maintained the least significant bits of the first and last row and column without alter the border pixels (first and last row and column for embedding the secret data. However, some of the pixels are used for embedding a part of the auxiliary information like predictor number, location of last data carrying block.

Hence a TLE embedding is performed on each pixel, that's why Kumar and Jung (2020) achieves high embedding capacity. Moreover, the second layer would most likely recover the original pixel value, hence enhancing the visual quality. As the embedding is only performed in HSB plane, any unintended attack or modification to the cover image would not affect the hidden data.

In this paper, we take inspiration from the way TLE achieves robustness i.e. embedding the data into HSB image and propose a Robust RDH method based on sorting and pairwise IPVO embedding. As PVO methods are designed to enhance the image quality, and pairwise embedding also boosts the performance, we perform data embedding using these in the HSB images.

PROPOSED METHOD

In this section, the proposed method is introduced. The proposed method uses a pairwise embedding scheme in which a pair of prediction errors is modified simultaneously to embed secret data. The cover image is first decomposed into two planes namely, Higher Significant Bit (HSB) and Least Significant Bit (LSB) planes. As some of the unintended attacks such as JPEG compression affects the LSBs of the image, it is best not to hide the secret data into LSBs to ensure data protection. Therefore, the proposed scheme performs data embedding on the HSB plane. The proposed method is discussed in detail below.

Cover Image Decomposition

A grayscale cover image of size $p \times q$ is decomposed into two planes – HSB and LSB. This decomposition method is derived from the SBDE approach proposed by Wang et al. According to this approach, each pixel p_{ij} of the cover image is represented by two planes as follows:

$$p_{ij} = \sum_{k=n}^{7} b_k \times 2^k + \sum_{k=0}^{n-1} b_k \times 2^k \tag{5}$$

such that $h_{ij} = \sum_{k=n}^{7} b_k \times 2^k$ and $l_{ij} = \sum_{k=0}^{n-1} b_k \times 2^k$ represent pixels of HSB plane and LSB plane respectively, $b_k \in \{0,1\}$ is the bit-value at k^{th} position in the binary representation of cover pixel. Thus, two images I-HSB and I-LSB are obtained.

Data Embedding

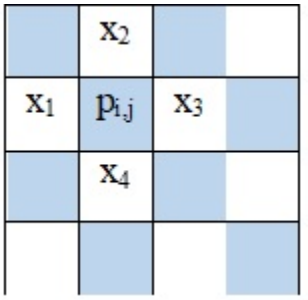
The I-HSB image obtained after cover image decomposition is used for data embedding. The steps to be followed for data embedding into HSB plane are explained in the subsequent subsections.

Image Scanning and Sorting

The image (I-HSB) is traversed in a chessboard-pattern as shown in Fig. 1. This traversal is basically adopted to divide the image into two independent sets (represent as white pixels and blue pixels in the figure). Pixels in a set are sorted according to their local variance which is calculated using four neighboring pixels i.e. rhombus context. For a pixel $p_{i,j}$, its neighboring pixels constituting the rhombus context are illustrated in Fig. 2. The mean value of the neighboring pixels i.e. rhombus context of $p_{i,j}$ is computed as:

$$\mu_{i,j} = \frac{1}{4} \sum_{k=1}^{4} x_k \tag{6}$$

Figure 2. Rhombus context of a pixel pi,j



The local variance of the pixel $p_{i,j}$ is calculated as follows:

$$\left(LC\right)_{i,j} = \frac{1}{4}\sum_{k=1}^{4}\left(\mu_{i,j} - x_k\right)^2 \tag{7}$$

The local complexity as computed in Eq. 7 is exploited for the rearrangement of pixels. The pixels are sorted in increasing order of their local variances because pixels with low complexity are preferable for data embedding as they would more likely generate embeddable prediction errors. Moreover, the identical sorting order is also obtained at the decoder side because two sets of pixels are independent of each other. Due to this, the decoder would also compute the same values of local variance, hence ensuring accurate data extraction. Then the sorted pixels are partitioned into blocks of size 1×3 .

Layer-1 Embedding

Peng et al. (2014) proposed the I-PVO method to hide secret payload and achieve high-fidelity images with reasonable embedding capacity. In this method, the cover-image is segregated into blocks of size $r\times c$ where $r,c\geq 2$. Let a block B_k consists of n pixels: $\left\{p_1,p_2,\ldots,p_n\right\}$. These pixels are sorted to get a sequence $\left\{p_{\delta(1)},p_{\delta(2)},\ldots,p_{\delta(n)}\right\}$ where $\delta:\left\{1\ldots,n\right\}\rightarrow\left\{1\ldots,n\right\}$ is a unique one-to-one mapping that follows the property: $p_{\delta(1)}\leq p_{\delta(2)}\leq\cdots\leq p_{\delta(n-1)}\leq p_{\delta(k)}$ and $\delta\left(\mathrm{i}\right)\leq\delta\left(j\right)$ if $p_{\delta(i)}=p_{\delta(j)}$ and $i<j$. Then two prediction errors, each for the largest-valued pixel and the smallest-valued pixel, are calculated as:

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

$$e_{max} = p_u - p_v \tag{9}$$

Here, $s=\min\left(\delta\left(1\right),\delta\left(2\right)\right)$, $t=\max\left(\delta\left(1\right),\delta\left(2\right)\right)$, $u=\min\left(\delta\left(n\right),\delta\left(n-1\right)\right)$, $v=\max\left(\delta\left(n\right),\delta\left(n-1\right)\right)$. The data embedding is performed if the values of e_{min} and e_{max} are either

0 or 1, because these are generally the peak bins of the prediction error histogram. If $b_1 \in \{0,1\}$ is the secret bit, then the smallest-valued pixel ($p_{\delta(1)}$) is modified as follows:

$$p'_{\delta(1)} = \begin{cases} p_{\delta(1)} - b_1 & if\ e_{min} = 1\ or\ e_{min} = 0 \\ p_{\delta(1)} - 1 & if\ e_{min} > 1\ or\ e_{min} < 0 \end{cases} \tag{10}$$

Here, $p'_{\delta(1)}$ is known as the marked value of the smallest-valued pixel. Thus, the smallest-valued pixel $p_{\delta(1)}$ is either decreased by $b_1 \in \{0,1\}$ or 1. If $b_2 \in \{0,1\}$ is the secret bit, then the largest-valued pixel ($p_{\delta(n)}$) is modified as follows:

$$p'_{\delta(n)} = \begin{cases} p_{\delta(n)} + b_2 & if\ e_{max} = 1\ or\ e_{max} = 0 \\ p_{\delta(n)} + 1 & if\ e_{max} > 1\ or\ e_{max} < 0 \end{cases} \tag{11}$$

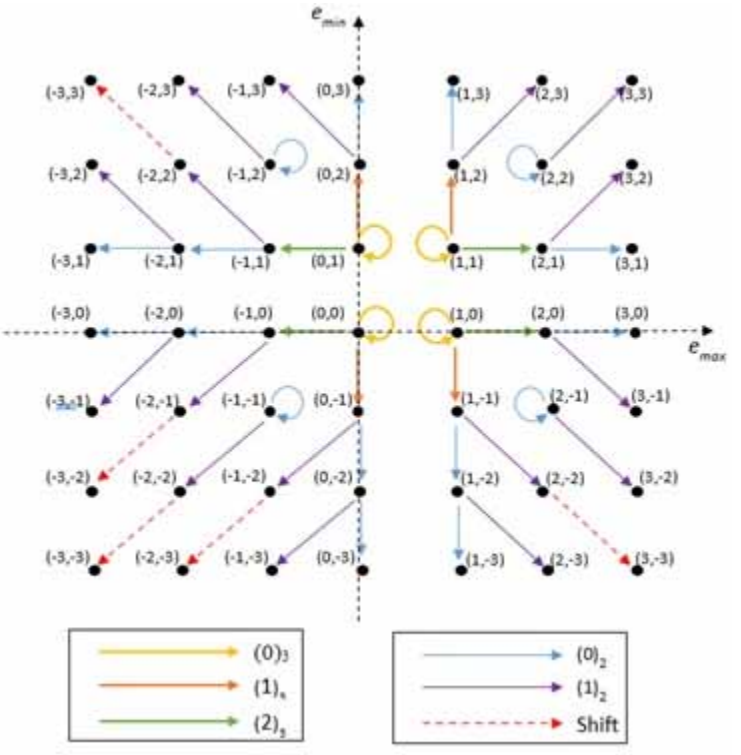
Here, $p'_{\delta(n)}$ is the marked value of the largest-valued pixel. Thus, the largest-valued pixel $p_{\delta(n)}$ is increased by either $b_2 \in \{0,1\}$ or 1.

Taking inspiration from the work of Ou et al. (2016), Wu et al. (2019) brought up the idea of pairwise IPVO to better utilize the block redundancy. Wu et al. (2019) asserted that the correlation between the prediction errors e_{min} and e_{max} should be utilized to obtain better performance of data embedding. They argue that the prediction errors e_{min} and e_{max} are highly correlated, so they should be modified simultaneously using 2D histogram modification, rather than utilizing them independently. In this method, the prediction error pair (1,1) can be expanded to (2,1), (1,2) and itself to accommodate $\log_2 3$ bits into this pair. In the conventional IPVO, the prediction error 2 could not embed data and is vulnerable to shifting while in the Pairwise-IPVO, the prediction error pair (2,2) is expanded to itself and (3,3) to embed 1 data bit. Likewise, the prediction error pairs (0,0), (0,1) and (1,0) also follow the similar embedding procedure as the aforementioned pair (1,1). The complete illustration of the 2D-mappings of the prediction error pairs is shown in Fig. 3, where each point represents an error-pair, the arrows represent the transitioning of prediction-errors as a result of shifting and expansion, and these arrows are labeled with the secret data bits in case of expansions. The data embedding of the proposed method takes advantage of the pairwise-IPVO discussed above with $n = 3$. The rationale behind the use of Pairwise-IPVO in the is its ability to provide better performance as compared to the conventional PEE techniques.

Layer-2 Embedding

The marked blocks obtained from pass-1 are used in pass-2 for an additional layer of data embedding using a rhombus predictor. For each block, let $\{\mu_1, \mu_2, \mu_3\}$ are rhombus mean values of the marked pixels $\{p'_1, p'_2, p'_3\}$ such that $\mu_1 \le \mu_2 \le \mu_3$. Then the first value and the last value in the mean-sorted sequence are predicted using the rhombus mean μ_1 and μ_2 respectively. In contrast to the pass-1 embedding, the first value $\left(p'_1\right)$ is either increased or unchanged, while the last value $\left(p'_3\right)$ is either decreased or unchanged. If the smallest-valued pixel $\left(p_{\delta(1)}\right)$ corresponds to the least value of rhombus mean (μ_1), and the largest-valued pixel $\left(p_{\delta(3)}\right)$ corresponds to the highest value of the rhombus mean

Figure 3. IPVO based Pairwise Data embedding



(μ_3), then the modifications performed in pass-1 embedding are inversed to recover original pixel values of the cover-image during the additional layer of embedding.

This would considerably lower the distortion and help in improving the embedding performance. However, for some cases, if this assumption does not hold true, then the difference between the original and marked value of any pixel would still not exceed by 2. However, the effect of this worst-case modification is balanced by the cases in which the pixels are restored to their original values during embedding. The prediction errors for the first and third pixel are calculated according to Eq. (7) and (8) respectively:

$$PE_1 = p'_1 - \mu_1 \tag{12}$$

$$PE_3 = p'_3 - \mu_3 \tag{13}$$

The secret data is then embedded using 1D prediction histogram modification and generally, the peak bin of the prediction error histogram is 0. So, a secret bit is embedded if the prediction error is 0, otherwise the histogram is shifted. If $b_3 = \{0,1\}$ is the secret bit, then the first pixel is modified as follows:

$$p_1^{''} = \begin{cases} p_1^{'} + b_3 \; if \;\; PE_1 = 0 \\ p_1^{'} + 1 \;\; if \;\; PE_1 > 0 \\ p_1^{'} \qquad otherwise \end{cases} \tag{14}$$

If $b_4 = \{0,1\}$ is the secret bit, then the last pixel is modified as follows:

$$p_3^{''} = \begin{cases} p_3^{'} - b_4 \; if \;\; PE_3 = 0 \\ p_3^{'} - 1 \;\; if \;\; PE_3 < 0 \\ p_3^{'} \qquad otherwise \end{cases} \tag{15}$$

Therefore, the first pixel is either unchanged or increased by (b_3 or 1) and the last pixel is either unchanged or decreased by (b_4 or 1).

Pseudocode is presented in Algorithm 1 for data embedding.

Data Extraction

At the receiver side, the image is decomposed into HSB and LSB planes. Then Algorithm 2 is applied on the HSB image to extract data and image recovery.

EXPERIMENTAL RESULTS AND DISCUSSION

In this section, the experimental results of the proposed method are presented. The proposed method performance is evaluated in terms of embedding capacity and PSNR values. The experimentation is done on eight standard grayscale images of size 512×512 including Lena, Baboon, F16, Peppers, Boat, and Lake taken from the USC-SIPI dataset. These images are shown in Fig. 5. The experiments are implemented in MATLAB and the secret data is a randomly generated sequence of 0s and 1s using inbuilt function of MATLAB. As already mentioned, the two parameters considered for performance evaluation are the embedding capacity and PSNR. The embedding capacity is defined as the number of bits that the method can hide in a particular cover image. On the other side, Peak-

Algorithm 1. Data embedding

Inputs: Cover-block $B_k = \{p_1, p_2, p_3\}$, S : secret data

Output: Marked-block $B_k^{'}$

• For the block $B_k = \{p_1, p_2, p_3\}$, sort the pixels in increasing order of intensity to obtain the sequence $\{p_{\delta(1)}, p_{\delta(2)}, p_{\delta(3)}\}$.

• Calculate the prediction errors using Eq. (8) & (9) and embed the secret data using Pairwise-IPVO method (refer to Fig. 3) to obtain the pixels $\{p_1^{'}, p_2^{'}, p_3^{'}\}$.

• Calculate the prediction errors for the first and last pixel using Eq. (12) & (13) and embed the secret data into the block using rhombus-mean prediction rules.

• Output the marked block $B_k^{'} = \{p_1^{''}, p_2^{''}, p_3^{''}\}$

Algorithm 2. Data extraction and recovery

Inputs: Marked-block $B_k^{'} = \left\{p_1^{''}, p_2^{''}, p_3^{''}\right\}$

Output: Cover-block B_k

• For the block $B_k^{'} = \left\{p_1^{''}, p_2^{''}, p_3^{''}\right\}$, calculate the prediction errors using Eq. (12) & (13) and extract the secret data using the rules illustrated in tables 2&3. Restore the values of $\left\{p_1^{'}, p_2^{'}, p_3^{'}\right\}$.

• Sort the pixels $\left\{p_1^{'}, p_2^{'}, p_3^{'}\right\}$ according to increasing intensity to calculate prediction errors using Eq. (8) & (9).

• Extract the data using Pairwise-IPVO extraction shown in Fig.4.

• Recover the original block $B_k = \left\{p_1, p_2, p_3\right\}$.

Table 2. Extraction and recovery rules for the first pixel

$e_1^{'}$	**<0**	**0**	**1**	**>1**
p_1	$p_1^{'}$	$p_1^{'}$	$p_1^{'}-1$	$p_1^{'}-1$
b	-	0	1	-

Table 3. Extraction and recovery rules for the third pixel

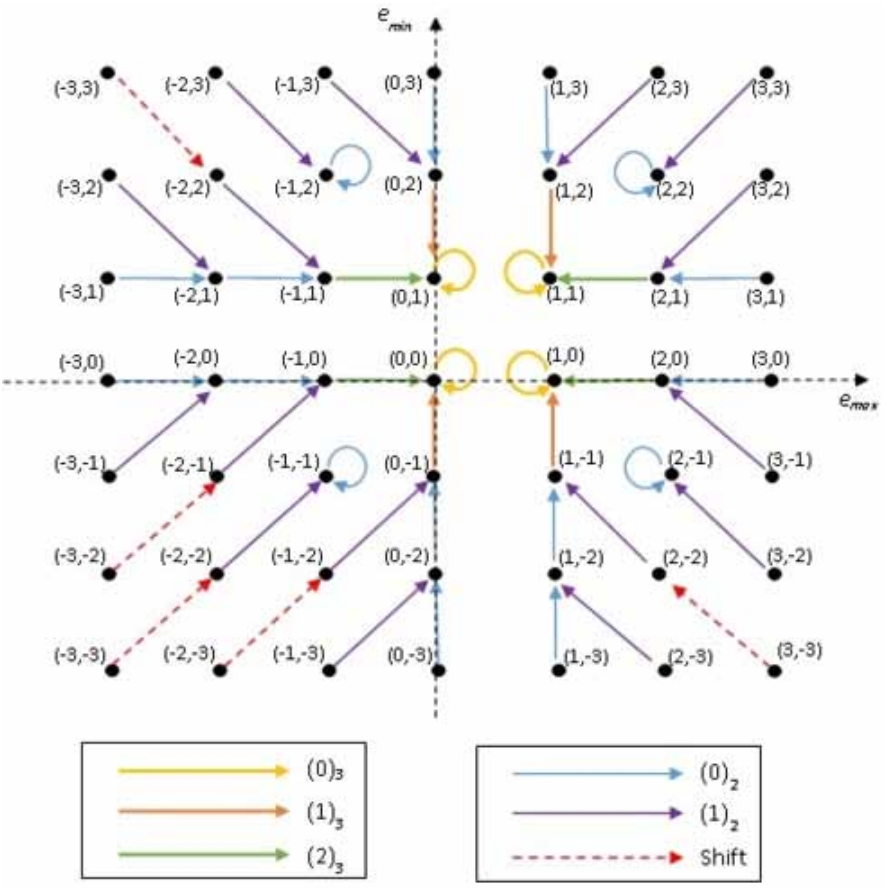
$e_3^{'}$	**<-1**	**-1**	**0**	**>0**
p_3	$p_3^{'}+1$	$p_3^{'}+1$	$p_3^{'}$	$p_3^{'}$
b	-	1	0	-

signal-to-noise Ratio (PSNR) is a quality assessment metric measured in decibels (db). The PSNR computes the peak signal-to-noise ratio between two images i.e., original image and stego-image. This ratio is used as a quality measurement between the original and a stego-image. The higher the PSNR, the better the quality of the stego-image. The calculation of the PSNR is done using Eq. 16 given below:

$$PSNR = 10\log_{10}\left(\frac{255^2}{MSE}\right) \tag{16}$$

Here MSE is 'mean squared error' i.e. the sum of squared differences between the pixel values of original cover image and the stego image. PSNR is measured in decibels (db). Generally, high values of PSNR indicate the high level of similarity between the stego image and the cover image.

Figure 4. IPVO based pairwise Data extraction



The experimental results for the embedding capacity versus image-fidelity tradeoff in a graphical representation are shown in Fig. 6. The proposed method is compared with some of the latest high capacity robust reversible data hiding schemes such as Kumar and Jung (2020), Wang et al. (2017), Rajkumar and Vasuki (2019). Kumar and Jung (2020) and Wang et al. (2017) both offer high embedding capacity as they take advantage of high correlation of pixels in HSB planes. It can be observerd that Kumar and Jung (2020) method performs better than Wang et al. (2017) in terms of visual quality. On the other hand, Rajkumar and Vasuki (2019) method works in transform domain, but has a limited embedding capacity. However, it does provide high PSNR value at low embedding capacities. It is observed and clear from the results that the proposed method improves the existing methods in terms of PSNR at all embedding capacities. The pairwise PVO based embedding incorporated into the first layer of embedding in the proposed scheme is responsible for providing high image quality.

Further, it is clear from the Fig.6 that the experimental results of the proposed method are on superior side in terms of PSNR at all embedding capacities in comparison to existing methods. The main reason behind the superior quality of the proposed scheme is the use of pairwise pixel value ordering based embedding into the first layer of embedding which basically provide high-fidelity images as it provides sharpest prediction error histogram while limiting the modification to one

Figure 5. Cover images

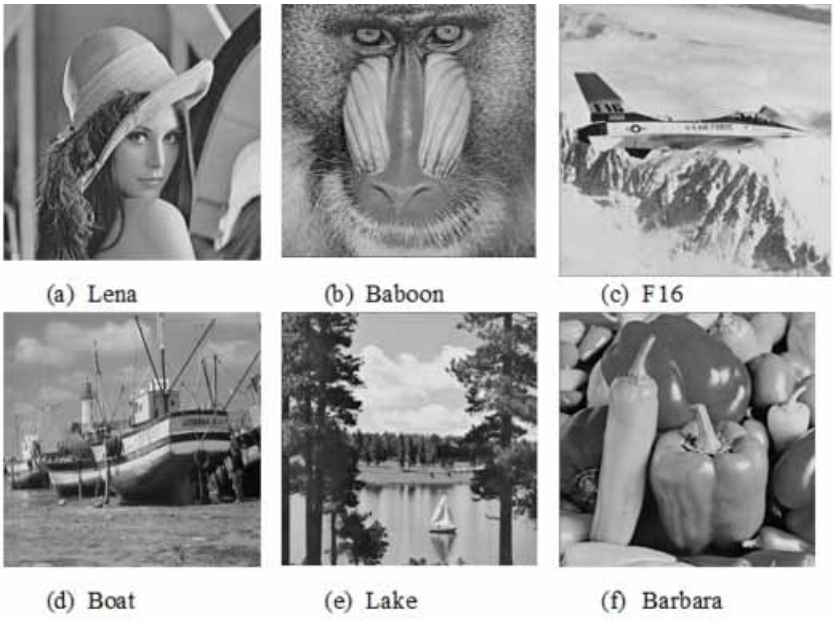
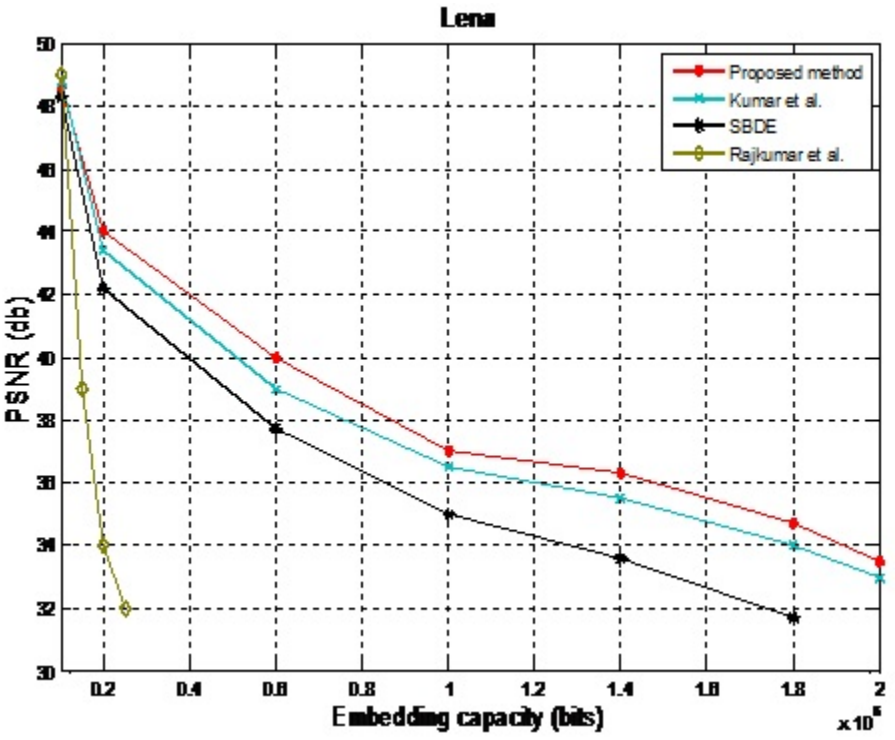


Figure 6. Performance comparison PSNR versus Embedding capacity



per block (instead of two) in case of embedding in most of the cases. Additionally, the layer two embedding helps in increasing the embedding capacity of the proposed scheme. Thus, the proposed scheme provides higher embedding capacity with better quality of stego-image while maintaining the same level of robustness as Kumar and Jung (2020) and Wang et al. (2017).

CONCLUSION

In this paper, a Robust RDH method is proposed using two-layer embedding. Firstly, the cover image is decomposed into two planes i.e. HSB and LSB. As unintended attacks or modifications such as JPEG compression affect the LSB pixels, we ignore the LSB plane. The secret data is hidden into the HSB plane to ensure its safety. The proposed method utilizes rhombus context to calculate local complexity of pixels and then the pixels are arranged according to their local complexity. Data embedding is done using pairwise IPVO into 1×3 size blocks. Then the second layer of embedding is done using rhombus mean of the pixels within the block. The proposed scheme exploits the high fidelity nature of PVO as well as high embedding scheme offered due to sorting of pixels. Sorting allows the highly correlated pixels to come closer, hence the number of embeddable prediction errors is increased. The experimental result proves that the proposed scheme does fairly well as compared to other RRDH based methods.

REFERENCES

Battisti, F., Carli, M., & Neri, A. (2010). *Reversible data hiding in the Fibonacci-Haar transform domain*. doi:10.1117/12.840691

Bhardwaj, A. (2020). Health Insurance Claim Prediction Using Artificial Neural Networks. *International Journal of System Dynamics Applications*, *9*(3), 40–57. doi:10.4018/IJSDA.2020070103

Di, F., Zhang, M., Liao, X., & Liu, J. (2019). High-fidelity reversible data hiding by Quadtree-based pixel value ordering. *Multimedia Tools and Applications*, *78*(6), 7125–7141. doi:10.1007/s11042-018-6469-4

Fridrich, J., Goljan, M., & Du, R. (2001). Invertible authentication. Academic Press.

He, W., Cai, J., Zhou, K., & Xiong, G. (2017). Efficient PVO-based reversible data hiding using multistage blocking and prediction accuracy matrix. *Journal of Visual Communication and Image Representation*, *46*, 58–69. doi:10.1016/j.jvcir.2017.03.010

He, W., Xiong, G., Weng, S., Cai, Z., & Wang, Y. (2018). Reversible data hiding using multi-pass pixel-value-ordering and pairwise prediction-error expansion. *Information Sciences*, *467*, 784–799. doi:10.1016/j.ins.2018.04.088

Huang, F., Huang, J., & Shi, Y. Q. (2016). New Framework for Reversible Data Hiding in Encrypted Domain. *IEEE Transactions on Information Forensics and Security*, *11*(12), 2777–2789. doi:10.1109/TIFS.2016.2598528

Kamstra, L., & Heijmans, H. J. A. M. (2005). Reversible data embedding into images using wavelet techniques and sorting. *IEEE Transactions on Image Processing*, *14*(12), 2082–2090. doi:10.1109/TIP.2005.859373 PMID:16370461

Kim, H. J., Sachnev, V., Shi, Y. Q., Nam, J., & Choo, H.-G. (2008). A Novel Difference Expan- sion Transform for Reversible Data Embedding. *IEEE Transactions on Information Forensics and Security*, *3*(3), 456–465. doi:10.1109/TIFS.2008.924600

Kumar, R., & Jung, K. H. (2020). Robust reversible data hiding scheme based on two-layer embedding strategy. *Information Sciences*, *512*, 96–107. doi:10.1016/j.ins.2019.09.062

Li, X., Li, B., Yang, B., & Zeng, T. (2013a). General Framework to Histogram-Shifting-Based Reversible Data Hiding. *IEEE Transactions on Image Processing*, *22*(6), 2181–2191. doi:10.1109/TIP.2013.2246179 PMID:23399962

Li, X., Li, J., Li, B., & Yang, B. (2013b). High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion. *Signal Processing*, *93*(1), 198–205. doi:10.1016/j.sigpro.2012.07.025

M., A., K., R., & K., K. (2020). Cloud-Based Access Control Framework for Effective Role Provi- sioning in Business Application. *International Journal of System Dynamics Applications*, *9*(1), 63–80.

Nasution, F. B. B., Bazin, N. E. N., Rosalyn, R., & Hasanuddin, H. (2018). Public Policymaking Framework Based on System Dynamics and Big Data. *International Journal of System Dynamics Applications*, *7*(4), 38–53. doi:10.4018/IJSDA.2018100103

Ni, Z., Shi, Y.-Q., Ansari, N., & Su, W. (2006). Reversible data hiding. *IEEE Transactions on Circuits and Systems for Video Technology*, *16*(3), 354–362. doi:10.1109/TCSVT.2006.869964

Ni, Z., Shi, Y. Q., Ansari, N., Su, W., Sun, Q., & Lin, X. (2008). Robust Lossless Image Data Hiding Designed for Semi-Fragile Image Authentication. *IEEE Transactions on Circuits and Systems for Video Technology*, *18*(4), 497–509. doi:10.1109/TCSVT.2008.918761

Ou, B., Li, X., & Wang, J. (2016). High-fidelity reversible data hiding based on pixel-value-ordering and pairwise prediction-error expansion. *Journal of Visual Communication and Image Representation*, *39*, 12–23. doi:10.1016/j.jvcir.2016.05.005

Ou, B., Li, X., Zhao, Y., & Ni, R. (2014). Reversible data hiding using invariant pixel-value-ordering and prediction-error expansion. *Signal Processing Image Communication*, *29*(7), 760–772. doi:10.1016/j.image.2014.05.003

Ou, B., Li, X., Zhao, Y., Ni, R., & Shi, Y.-Q. (2013). Pairwise Prediction-Error Expansion for Efficient Reversible Data Hiding. *IEEE Transactions on Image Processing*, *22*(12), 5010–5021. doi:10.1109/TIP.2013.2281422 PMID:24043388

Panda, M. (2019). Software Defect Prediction Using Hybrid Distribution Base Balance Instance Selec- tion and Radial Basis Function Classifier. *International Journal of System Dynamics Applications*, *8*(3), 53–75. doi:10.4018/IJSDA.2019070103

Peng, F., Li, X., & Yang, B. (2014). Improved PVO-based reversible data hiding. *Digital Signal Processing*, *25*, 255–265. doi:10.1016/j.dsp.2013.11.002

Pierce, D., Shepherd, S., & Johnson, D. (2019). Modelling the Impacts of Inter-City Connectivity on City Specialisation. *International Journal of System Dynamics Applications*, *8*(4), 47–70. doi:10.4018/IJSDA.2019100104

Rajkumar, R., & Vasuki, A. (2019). Reversible and robust image watermarking based on histogram shifting. *Cluster Computing*, *22*(S5), 12313–12323. doi:10.1007/s10586-017-1614-9

Sachnev, V., Kim, H. J., Nam, J., Suresh, S., & Shi, Y. Q. (2009). Reversible Watermarking Algorithm Using Sorting and Prediction. *IEEE Transactions on Circuits and Systems for Video Technology*, *19*(7), 989–999. doi:10.1109/TCSVT.2009.2020257

Shukla, O. J., Jangid, V., Soni, G., & Kumar, R. (2019). Grey Based Decision Making for Evaluating Sustainable Performance of Indian Marble Industries. *International Journal of System Dynamics Applications*, *8*(2), 1–18. doi:10.4018/IJSDA.2019040101

Thodi, D. M., & Rodriguez, J. J. (2007). Expansion Embedding Techniques for Reversible Water- marking. *IEEE Transactions on Image Processing*, *16*(3), 721–730. doi:10.1109/TIP.2006.891046 PMID:17357732

Tian, J. (2003). Reversible data embedding using a difference expansion. *IEEE Transactions on Circuits and Systems for Video Technology*, *13*(8), 890–896. doi:10.1109/TCSVT.2003.815962

Wang, W., Ye, J., Wang, T., & Wang, W. (2017). Reversible data hiding scheme based on significant- bit-difference expansion. *IET Image Processing*, *11*(11), 1002–1014. doi:10.1049/iet-ipr.2017.0151

Wang, X., Ding, J., & Pei, Q. (2015). A novel reversible image data hiding scheme based on pixel value ordering and dynamic pixel block partition. *Information Sciences*, *310*, 16–35. doi:10.1016/j.ins.2015.03.022

Weng, S., Pan, J., & Li, L. (2016). Reversible data hiding based on an adaptive pixel- embedding strategy and two-layer embedding. *Information Sciences*, *369*, 144–159. doi:10.1016/j.ins.2016.05.030

Weng, S., Pan, J. S., Jiehang, D., & Zhou, Z. (2018). Pairwise IPVO-based reversible data hiding. *Multimedia Tools and Applications*, *77*(11), 13419–13444. doi:10.1007/s11042-017-4959-4

Weng, S., Shi, Y., Hong, W., & Yao, Y. (2019). Dynamic improved pixel value ordering reversible data hiding. *Information Sciences*, *489*, 136–154. doi:10.1016/j.ins.2019.03.032

Wu, H., Li, X., Zhao, Y., & Ni, R. (2019). Improved reversible data hiding based on PVO and adaptive pairwise embedding. *Journal of Real-Time Image Processing*, *16*(3), 685–695. doi:10.1007/s11554-019-00867-w

Zong, T., Xiang, Y., Natgunanathan, I., Guo, S., Zhou, W., & Beliakov, G. (2015). Robust Histogram Shape-Based Method for Image Watermarking. *IEEE Transactions on Circuits and Systems for Video Technology*, *25*(5), 717–729. doi:10.1109/TCSVT.2014.2363743

Aruna Malik received her B. Tech. in Computer Science and Engineering from Uttar Pradesh Technical University, Lucknow, India and M. Tech. in Computer Science and Engineering from National Institute of Technology, Jalandhar, Punjab, India. She did her doctoral degree in Computer Science and Engineering from National Institute of Technology, Jalandhar, Punjab, India. Now, she is working as an Assistant Professor in the Department of Computer Science & Engineering at National Institute of Technology, Jalandhar, Punjab, India. Her research areas lie in the area of Data hiding and Image processing.

Rajeev Kumar received his B.Tech. in Information Technology from Uttar Pradesh Technical University, Lucknow, India and M.Tech. in Information Systems from Netaji Subhas Institute of Technology, Delhi University, India. He is completed his doctoral degree is at the Computer Engineering Department, Netaji Subhas Institute of Technology, Delhi University, Delhi, India. He is post-doc fellow at Kyungil University, South Korea. His research areas lie in the area of data hiding and image processing.