New Reversible Data Hiding Algorithm Based on Edge Detection and PVO Mechanisms

Hoang-Nam Tram, Thai-Son Nguyen

***Abstract*—Reversible data hiding technique is able to restore the cover data exactly after data extraction. This paper presents a new reversible data hiding scheme based on the combination of edge detection and pixel value ordering mechanism is proposed. In the proposed scheme, an edge detection algorithm is first applied to produce edge pixels to classify the pixels into smooth or rough areas. Then, to increase the embedding capacity, the pixel prediction algorithm is also modified. Additionally, to minimize distortion during embedding process, in smooth areas, the noise level of block is calculated and used to determine the suitable pixels for carrying the secret bit. Then, based on the high redundancy of neighboring pixels, prediction errors are calculated and used to embed data by PVO mechanism. The experimental results demonstrated that our proposed scheme achieves better visual quality than those of state-of-the-art schemes when the average PSNR is greater than 58.6dB. Moreover, the proposed scheme maintains the reversibility.**

***Index Terms*— data hiding, prediction-error, pixel value ordering, edge detection, reversibility.**

I. INTRODUCTION

Reversible data hiding (RDH) technique is able to restore the cover data exactly after data extraction. For this reason, RDH is widely applied in some sensitive scenarios, i.e., military, medical, and remote sensing where distortions are forbidden and the accurate recovery of the original cover data are required. Until now, most RDH schemes have been investigated for image in spatial domain. In general, RDH schemes can be classified into three categories, i.e., difference expansion (DE) [1]–[7], histogram shifting (HS) [8]–[18] and integer transformation (IT) [19]–[21]. In the DE category, J. Tian [1] has used the difference value of a pair of pixels, which is expanded to embed secret data. Their method obtained high embedding capacity (EC) and low image distortion. However, a location map is required to guarantee reversibility. Next, to further improve the performance of Tian's scheme, Alattar [2] embedded several bits in the difference expansion of vectors of adjacent pixels. Alattar's scheme achieved high embedding capacity. However, their scheme obtained the unsatisfied image quality when the PSNR is always smaller than 44dB. It is noted that, in the data hiding field, the improvement of the image quality of stego images is required as one of main criterions. In 2008, Kim et al. [4] proposed a DE-based RDH scheme by using Laplace distribution. Their scheme ensured the small size of location map. In the HS category, the first RDH scheme is introduced by Ni et al. [8]. In Ni et al.'s scheme, a histogram is constructed by statistics of the occurrence frequency of pixel values. Then, the bin with the highest frequency in the histogram is called peak point and chosen for embedding the secret data. Because of the reversible data hiding characteristic, all bins between the peak point and the zero point are shifted by 1 unit, toward to the right or the left for embedding data. Therefore, the height of peak point in the histogram significantly effects on the embedding capacity of HS-based RDH schemes. In 2013, Li et al. [12] proposed a new HS-based RDH scheme. In this scheme, the cover image is divided into non-overlapping blocks. Then, pixels in each block are sorted in ascending order. Therefore, this scheme is also called as pixel-value-order (PVO) scheme. In Li et al.'s scheme, the difference value of the largest pixel and the second largest pixel in each block is calculated for concealing the secret data. The embedding capacity of their scheme is further improved. However, it is still limited when most of the difference value is larger than 1, therefore this leads to the low PSNR. To enhance the embedding capacity, instead of using the difference value between the maximum value and the second largest value of the block as was done in Li et al.'s method, Peng at el. [13] proposed an Improved PVO-based reversible data hiding scheme (IPVO) by computing new differences. Then, the new difference is defined by using the pixel locations of the maximum and second largest values. Moreover, the blocks where the maximum equals to the second largest value can be exploited for hiding the secret data. Later on, to increase the image quality, Qu et al. [14] proposed a new RDH scheme call as Pixel-based pixel value ordering (PPVO) by predicting the value of a current pixel according to its neighboring pixels in each block. Then, the difference value of the current pixel and the prediction value is modified to embed the secret data. The image quality of Qu et al.'s scheme is significantly improved when the redundancy of the neighboring pixels is exploited. However, this scheme obtained low embedding capacity when only one secret bit is embedded per block in the maximum. In [15], Wang et al. proposed new RDH scheme by dividing various-sized blocks. In particular, smooth areas are partitioned into smaller blocks for embedding data to increase embedding

This work was fully supported by Tra Vinh University under Grant contract number 222/HĐ-HĐKH&ĐT.ĐHTV.

Hoang-Nam Tram is with the Department of Information Technology, Tra Vinh University, Vietnam (e-mail: tramhoangnam@tvu.edu.vn).

Thai-Son Nguyen is with the School of Engineering and Technology, Tra Vinh University (e-mail: thaison@tvu.edu.vn).

Digital Object Identifier: 10.14209/jcis.2022.8.

capacity. Later on, a novel RDH scheme is proposed by Weng et al. [16]. In Weng et al.'s scheme, the smooth blocks are partitioned into sub-blocks of arbitrary size according to their local complexity to maintain the minimum modification during the embedding process. To further improve the embedding performance of prior works, in 2019, Di et al. [18] proposed new RDH scheme by quadtree-based pixel value ordering (QPVO). In each block four pixels, their scheme embeds the secret bits selectively with a quadtree structure. To better reduce embedding distortion, the cover image is partitioned into blocks in various sizes by the dynamic adaptive quadtree decomposition based on the image complexity. Di et al.'s scheme obtained the better image quality of mark images while guaranteeing the high embedding capacity.

In this paper, to further improve the image quality of marked images, we proposed a new RDH scheme based on edge detection and pixel value ordering (EPVO). Instead of dividing the image into blocks in various sizes for embedding data as was done in [18], in the proposed scheme, the Canny edge detection algorithm is first applied to produce edge pixels to classify the pixels into smooth or rough areas. Additionally, to minimize distortion during embedding process, in smooth area, based on the high redundancy of neighboring pixels, prediction errors are calculated and used to embed data. The experimental results showed that our proposed scheme achieved the better image quality of mark images while maintaining high embedding capacity.

The rest of the paper is organized as follows: the review of Canny edge detector [22] and some previous RDH scheme [12, 14] are presented in Section II. The proposed scheme is described in Section III. Section IV presents the experimental results. Finally, conclusions of the paper are drawn in Section V.

II. Related Works

A. Canny edge detector

Canny edge detector CED [22] has been widely used because it possesses superior performance on optimality due to satisfying three criterions. i.e., (i) High accuracy: The detected edge should be the real edge and the false detection ratio should be rather low. (ii) High precision: The distance between the actual edge and the extracted position of the edge should be minimal. (iii) A single response to an edge: A given edge in the image should be marked only once and the potentially existing noise in the image should not create false edge.

The steps used in CED implementation are: Using Gaussian filter to smooth the image; taking the gradient by using the Sobel operator; execute the non-maximum suppression to eliminate spurious response to edge detection; perform the double thresholds technique to determine potential edges. The detail of CED processing is described as following steps:

Step 1: Noise reduction

Using Gaussian filter to smooth the image and remove noise. Because the noise can affect to the edge detection, it should be removed to achieve the best detection. First, each block of image is convolved with a Gaussian kernel where the block size of image is equal to the kernel size. The kernel size can be 3×3, 5×5, 7×7 depends on the expected blurring effect. Normally, the size 5×5 is suitable for most cases.

Step 2: Gradient calculation

Calculate the magnitude and directions of the intensity gradient of the smoothened image in step 1 is by using Sobel kernel in both horizontal and vertical direction. It can be implemented by convolving the image with Sobel kernels G_x and G_y, where they are defined as follows:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \; G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Step 3: Non-Maximum suppression

This step will find the pixels that are most likely to be edges by removing the non-maximum values in the gradient calculation step above. If the magnitude of a pixel is greater than the magnitude of its neighborhood, it will be marked as an edge pixel; otherwise, the magnitude of the current pixel is set to zero (the pixel is non-edge pixel).

Step 4: Double threshold

Double threshold is implemented to determine strong, weak, and non-relevant edge pixel in the image. There are two thresholds: high threshold T_H and low threshold T_L is used to identify the strong, weak or the non-relevant pixels. The edge pixel is marked as strong edge pixel if its gradient value is higher than T_H. If the edge pixel gradient value is larger than T_L and smaller than T_H, it is marked as weak edge pixel. Otherwise, it will be suppressed. The two threshold values are empirically defined and they can be adjusted when applied to different images.

B. High fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion

In Li et al. scheme [12], the original image with the size of $H \times W$ will be divided into non-overlapping pixel blocks with the size of $h \times w$. Then, pixels in each block, $X_i = (x_1, x_2, \dots, x_{h\times w})$, are sorted to get $X_\partial = (x_{\partial(1)}, x_{\partial(2)}, \dots, x_{\partial(h\times w)})$, where $\partial: \{1,2,\dots, h \times w\} \to \{1,2,\dots, h \times w\}$ is unique one to one mapping such that $x_{\partial(1)} \le x_{\partial(2)} \le \dots \le x_{\partial(h\times w)}, \partial(i) < \partial(j)$ if $x_{\partial(i)} = x_{\partial(j)}$ and $i < j$. Then, the maximum pixel was predicted by the second largest pixel. Similarly, the minimum pixel was predicted by the second smallest pixel. The corresponding prediction error can be obtained by using (1):

$$\begin{cases} PE_{max} = x_{\partial(h\times w)} - x_{\partial(h\times w-1)} \\ PE_{min} = x_{\partial(1)} - x_{\partial(2)} \end{cases}. \tag{1}$$

Then, the histogram of prediction errors was generated as shown in Fig. 1. The secret bit can be embedded into the prediction error when the value of such prediction error is equal to 1 or -1, whereas the prediction error with the value of 0 is kept unchanged. If the prediction errors which are larger than 1 or smaller than -1 are shifted by 1 unit toward the right or the left to ensure reversibility. To embed data, in Li et al.'s method, the prediction error PE_{max} and PE_{min} are modified by using (2) and (3)

$$PE'_{max} = \begin{cases} PE_{max} & if\ PE_{max} = 0 \\ PE_{max} + s & if\ PE_{max} = 1, \\ PE_{max} + 1 & if\ PE_{max} > 1 \end{cases} \tag{2}$$

$$PE'_{min} = \begin{cases} PE_{min} & if\ PE_{min} = 0 \\ PE_{min} - s & if\ PE_{min} = -1\,, \\ PE_{min} - 1 & if\ PE_{min} < -1 \end{cases} \quad (3)$$

where $s \in \{0,1\}$ denotes as the secret message bit to be embedded.

Then, the mark image is constructed by using (4). It is noted that for each block, only the maximum pixel $x_{\partial(h\times w)}$ and the minimum pixel $x_{\partial(1)}$ are modified in Li et al.'s scheme.

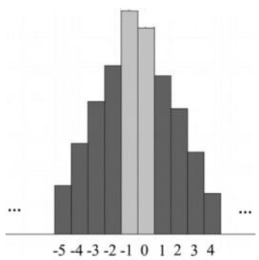

Fig. 1. Histogram of prediction errors.

$$\begin{cases} x'_{\partial(h\times w)} = x_{\partial(h\times w-1)} + PE'_{max} \\ x'_{\partial(1)} = x_{\partial(2)} + PE'_{min} \end{cases}. \quad (4)$$

C. Pixel-based pixel value ordering predictor for high-fidelity reversible data hiding scheme

In [12], the EC of PVO-based scheme is limited; the main reason is that only two pixels in the block are utilized for embedding data. To solve this problem, Qu et al. [14] proposed a novel pixel-based PVO method where each pixel is predicted by its sorted context pixels. In their scheme, some pixels which are neighboring of the current pixel of block are defined as context pixels *C*. Before prediction, sorting and ordering the context pixels must be executed. The prediction value is made by *max(C)* or *min(C)*, the maximum or minimum of context pixel vector *C*. The description of this predictor is expressed by using (5):

$$\hat{x} = \begin{cases} max(C) & max(C) \neq min(C), & x \geq max(C) \\ min(C) & max(C) \neq min(C), & x \leq min(C) \\ 254 & max(C) = min(C) = 254, & x = 254 \\ VC & max(C) = min(C) = VC, & x \leq VC, VC < 254 \\ skip & otherwise & \end{cases}. \quad (5)$$

where VC is a constant number and $VC = max(C) = min(C)$.

Then, the stego pixels are calculated by the Equation (6)

$$\tilde{x} = \begin{cases} x+s & max(C) \neq min(C), & x = max(C) \\ x+1 & max(C) \neq min(C), & x > max\ (C) \\ x-s & max(C) \neq min(C) & x = min\ (C) \\ x-1 & max(C) \neq min(C) & x < min\ (C) \\ x+s & max(C) = min(C) = 254, & x = 254 \\ x-s & max(C) = min(C) = VC, & VC < 254, x = VC \\ x-1 & max(C) = min(C) = VC, & VC < 254, x < VC \end{cases}. \quad (6)$$

where s is the secret bit to be embedded.

In Li et al.'s scheme [12], the cover image is partitioned into non-overlap blocks. Then prediction errors are calculated in a block-by-block manner. Thus, the maximum embedding capacity for each block is therefore at most two bits. In [14], although the cover image is divided into overlap blocks for better embedding capacity, however, the block with pixels in the edge is also considered during embedding process. That caused more distortion of the stego image. As a result, the embedding capacity of the scheme [14] can be improved further. However, their image quality is unsatisfied. In addition, it can be seen that in Table I, the average percentage of pixels located in the edge areas is more than 10%. Therefore, if the edge areas in the cover image are prevented during the embedding process, the smaller distortion of the image is achieved. In order to increase the embedding capacity while maintaining the good quality of stego images, we combined of CED edge detection and PVO mechanism. The main steps of our proposed scheme are described in the following section.

TABLE I
THE PERCENTAGE OF PIXELS LOCATED IN EDGE AREAS IN THE IMAGE

Images (512×512)	Number of Pixel located in the edge areas	Percentage
Lena	22,550	8.6%
Baboon	47,673	18.2%
Barbara	26,894	10.3%
Airplane	21,317	8.1%
Tiffany	26,302	11.1%
House	28,983	10.0%
Average	27,286	10.4%

III. PROPOSED SCHEME

In this section, we present our proposed scheme. In the proposed scheme, the 3 least significant bits (*LSBs*) of each pixel in the cover image are first removed to obtain image I_{MSB}. Next, CED detector is applied on this image I_{MSB} to get edge matrix I_{edge}. Then, according to the matrix I_{edge}, the cover image *I* is divided into smooth and rough areas. In our scheme, CED edge detection is used to achieve a fairly good balance in noise restriction and edge detection. In the smooth areas, all blocks with the size of 2×2 are scanned in raster scan order. Pixels of the smooth block are predicted and used to embed data by using PVO algorithm. Fig. 2 shows the framework of the proposed data embedding process.

A. Pixel prediction

Inspired of the prediction manner in the Qu et al.'s scheme [14], to increase the embedding capacity, the pixel prediction algorithm is modified. In addition, before pixel prediction, the noise level *NL* of block is calculated by using (7). If $NL \leq T$, it means the current pixel of block can be used to carry the secret bit. Otherwise, the block will be skipped to ensure low distortion, where *T* is a given threshold and it is used to control the distortion of the stego images. In our scheme, T is in the range [0, 5]. Noticed that when the larger value of NL is used, the larger distortion of the stego image is obtained, therefore, the value of *T* should be limited as small as possible for good image quality.

$$NL = max(P) - min(P). \quad (7)$$

In each block, some neighbor pixels of the current pixel x are denoted as a reference vector $P = \{P_1, P_2, \ldots, P_{CN}\}$ where CN is the number of pixels. Then, the current pixel can be predicted by $max(P)$ (the maximum pixel) or $min(P)$ (the minimum pixel) of the reference vector. There are only two cases of the prediction, i.e., $max(P) \neq min(P)$ and $max(P) = min(P)$.

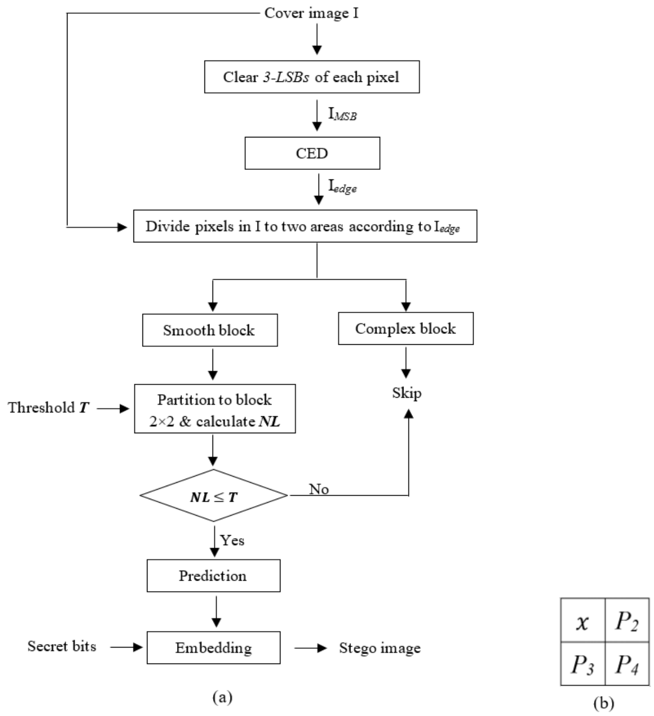

Fig. 2. (a) Framework of the proposed data embedding process; (b) example of the block 2×2.

In the first case, if the current pixel value $x \geq \max(P)$, then its value is predicted by $max(P)$. If the current pixel value $x \leq min(P)$, then its value is predicted by $min(P)$.

In the second case, all pixels in the vector P have the same value, denoted as VP. Then, the current pixel x is predicted in two scenarios: If the current pixel $x \geq VP$, it will be predicted by VP. Otherwise, the prediction error is $VP-1$ when the current pixel $x \leq VP-1$. In addition, when the current pixel value x is in the middle of the values of $max(P)$ and $min(P)$, the current pixel has to be skipped without prediction. The current pixel x in each block can be predicted by the context pixels P_2, P_3, P_4 by using (8):

$$\hat{x} = \begin{cases} max(P) & max(P) \neq min(P), & x \geq \max(P) \\ min(P) & max(P) \neq min(P), & x \leq \min(P) \\ VP & max(P) = min(P) = VP, & x \geq VP \\ VP-1 & max(P) = min(P) = VP, & x \leq VP-1 \\ skip & otherwise & \end{cases}. \quad (8)$$

Moreover, to avoid the underflow and overflow problems, the pixels with value of 0 will be modified to 1 and the pixels with value of 255 will be modified to 254. These modifications are recorded by a location map to guarantee the reversibility.

B. Embedding procedure

\+ **Input:** The original image I with the size of $m\times n$, the secret bits S.

\+ **Output:** The marked image I'.

Embedding algorithm

Determine the smooth area by applying CED to I
for each block P in smooth area **do**
  Compute NL by Equation (7)
  if $NL <= T$ **then**
    Calculate the prediction $\hat{x}$ by Equation (8)
    Calculate the prediction error e by Equation (9)
    if $\min(P) \neq \max(P)$ **then**
      Modify e to e' by Equation (10)
    else
      Modify e to e' by Equation (11)
    Update the original pixel x to x' by Equation (12)
    end if
  else skip the current block
  end if
end for
Embedding auxiliary information

The detail of above mentioned pseudocode of embedding algorithm as followings:

Step 1: Remove 3 *LSBs* of the cover image and apply CED to generate the edge matrix I_{edge}.

Step 2: According to the matrix I_{edge}, divide I into overlap blocks P with the size of 2×2. For each block, compute the NL value by using (7) and predict for current pixel x or skip this pixel depending on NL.

Step 3: Calculate the prediction error e between the current pixel value and prediction pixel value by using (9):

$$e = x - \hat{x}. \quad (9)$$

Step 4: If $max(P) \neq min\ (P)$, the modification of prediction error is implemented by using (10) where $s = \{0,1\}$ is data embedding.

$$e' = \begin{cases} e+s & if\ e=0\ and\ x = max(P) \\ e+1 & if\ e>0 \\ e-s & if\ e=0\ and\ x = min(P) \\ e-1 & if\ e<0 \end{cases}. \quad (10)$$

When $max(P) = min(P) = VP$ (all context pixels have the same value), the initial prediction error e is modified to be e' by using (11)

$$e' = \begin{cases} e+s & if\ e=0\ and\ x = VP \\ e+1 & if\ e>0 \\ e-s & if\ e=0\ and\ x = VP-1 \\ e-1 & if\ e<0 \end{cases}. \quad (11)$$

After that, the current pixel will be updated to achieve marked pixel by using (12)

$$x' = \hat{x} + e'. \quad (12)$$

Continue the steps until all secret bits are embedded and the location of last embedded pixel P_{end} is recorded

Step 5: Embedding auxiliary information: Record the least significant bits (*LSB*) of first $log_2(n\times m)+21$ image pixels to obtain a binary sequence S_{SLB}. Then, replace *LSB* of those pixels by the auxiliary information, i.e., the compressed location map ($log_2(n\times m)$ bits), the threshold T (3 bits), the last embedded pixel P_{end} (18 bits).

Finally, the sequence S_{SLB} is also embedded into the original image from the pixel with the position P_{end+1} by the same manner in Step 4 to generate the marked image I'. Fig. 3 shows an example of embedding process in two cases.

C. Extracting procedure

+ **Input:** The marked image I'.
+ **Output:** The original image I, the secret bits S.

Extracting algorithm

Determine the smooth area by applying CED to I'
Extracting auxiliary information
for each block P in smooth area **do**
    Compute NL by equation (7)
    if $NL <= T$ **then**
        Calculate the prediction $\widehat{x'}$ by equation (8)
        Calculate the prediction error e' by equation (13)
        Extract secret bit by the equation (14)
        Restore e' to e by equation (15)
        Update the marked pixel x' to x by equation (16)
    else skip this block
    end if
end for

The detail of above mentioned pseudocode of extracting algorithm as followings:

Step 1: Remove 3 *LSBs* bits of the marked image. Then, the CED is applied to generate the matrix I'_{edge} that is used to determine whether the marked pixels belong to the smooth or rough areas.

Step 2: Extract the auxiliary information from *LSB* of first $log_2(n \times m) + 21$ marked pixels, i.e., the compressed location map, the threshold *T* and the last embedded pixel P_{end}.

Step 3: According to auxiliary information, extract the secret bits from the last embedded pixel in reverse order with embedding procedure. The smooth area of the marked image is divided into overlap blocks P with the size of *2×2*. The current pixel x' of each block can be predicted by Equation (8) to obtain the prediction value $\widehat{x'}$.

Step 4: Calculate the prediction error by using (13):

$$e' = x' - \widehat{x'}. \quad (13)$$

Step 5: When the prediction error is equal to -1, 0 or 1, the secret bit s can be extracted according to Equation (14). Otherwise, there is no bit is extracted.

$$s = \begin{cases} 0 & if\ e' = 0 \\ 1 & if\ e' = 1\ or\ e' = -1 \end{cases}. \quad (14)$$

Then, the prediction error is modified by using (15):

$$e = \begin{cases} e' & if\ e' = 0 \\ e' - s & if\ e' = 1 \\ e' + s & if\ e' = -1. \\ e' - 1 & if\ e' > 1 \\ e' + 1 & if\ e' < -1 \end{cases} \quad (15)$$

Last, original value pixel will be recovered by using (16):

$$x'' = x = \widehat{x'} + e. \quad (16)$$

Repeat from step 3 to step 5 until all secret bits are extracted completely. Then, extract the sequence S_{SLB} from the pixel with the position P_{end+1} to recover the *LSB* of first $log_2(n \times m) + 21$ pixels of image I' by *LSB* substitution. Later, the restored image I'' is generated. Next, according to the location map, the pixels with the value of 1 or 254 in the image I'' will be modified to 0 or 255 to reconstruct the original image I.

IV. EXPERIMENTAL RESULTS

In this section, the proposed scheme was tested on publicly available, eleven standard images in Fig. 5. All test images with the size of *512×512* are downloaded from the USC-SIPI image database. In proposed scheme, we use three types of secret bit for testing: 1) the binary TVU logo; 2) the random bit stream by using random function in MATLAB and 3) the binary image Rice with the size of *256×256*. After that, some tables are presented for comparing the experimental results of proposed scheme with [12, 13, 14, 18] schemes. Our computations were implemented on a laptop with an Intel(R) Core (TM) i5-8250U CPU (6M Cache, 1.60 GHz), 8 GB of RAM. The experiments results are performed on Windows 10 Pro 64-bit and by MATLAB R2014a.

A. Reversibility

To demonstrate the reversibility of the proposed method, we present an example of embedding process as Fig. 4 in two cases of secret bit, $s = 0$ and $s = 1$. It can be seen in Fig. 4, the secret bit and the original image can be restored exactly in two cases.

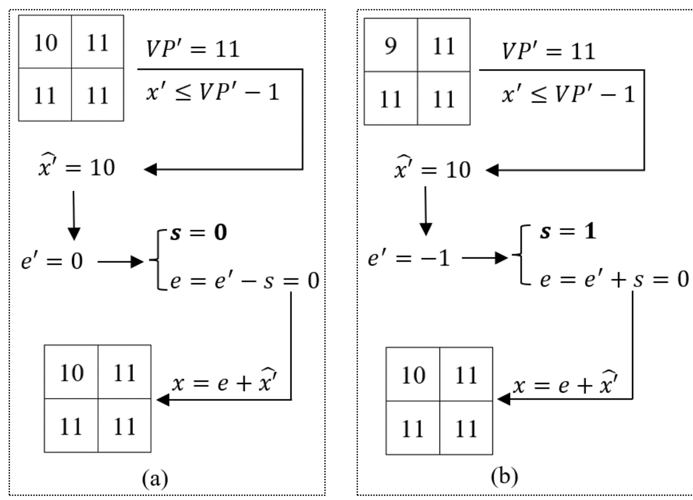

Fig. 4. An example of extracted process in two cases

B. Image quality

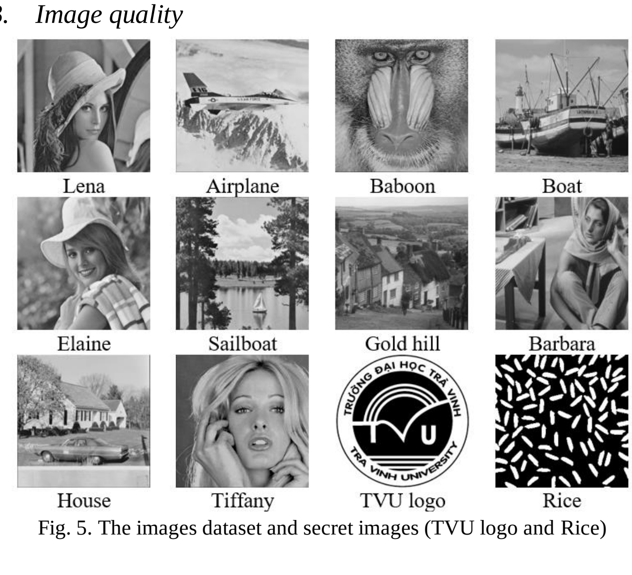

Fig. 5. The images dataset and secret images (TVU logo and Rice)

To demonstrate the superiority of the proposed scheme, we compared the proposed scheme to PVO, IPVO, PPVO and QPVO schemes [12, 13, 14, 18] as shown in Fig. 6. For fair comparison, in the experiment results, the parameter *T* is set as *T=0.5R*, to maintain the balance between the embedding capacity and the quality image of the QPVO scheme. Here, *R* is block complexity for the whole image. The embedding capacity is varied from 5,000 bits to its maximum with a step size of 1,000 bits. Although the proposed scheme and four other schemes [12, 13, 14, 18] are based PVO, however, the better performance was obtained by the proposed method than by four other schemes [12, 13, 14, 18] in most cases.

TABLE II
COMPARISON OF THE PSNR (DB) VALUES BETWEEN PROPOSED SCHEME AND FOUR PREVIOUS SCHEMES FOR EC OF 10,000 BITS FOR TVU LOGO

Images	PVO [12]	IPVO [13]	PPVO [14]	QPVO [18]	Proposed
Lena	56.48	57.82	58.12	57.86	59.51
F16	57.92	58.35	58.18	58.43	58.43
Barbara	55.79	56.93	56.27	57.30	57.91
Boat	58.80	59.14	59.01	59.20	59.07
Tiffany	56.46	57.24	56.85	57.68	57.88
House	59.29	60.85	61.14	61.79	61.79
Sailboat	55.32	56.19	56.30	58.11	58.02
Elaine	54.35	55.57	55.99	56.07	56.16
Gold hill	57.82	57.92	58.13	57.74	57.98
Peppers	55.46	56.31	56.04	56.56	56.43
Baboon	51.48	52.30	52.25	53.96	54.12
Average	56.29	57.15	57.12	57.70	57.94

Fig. 6 shows the embedding performance of our proposed scheme and four other schemes for different test images. In most of test images, the top curve is our proposed scheme. With the same of PSNR value, the embedding capacity of our proposed scheme is higher than those of the other four schemes. Fig. 6 shows that the proposed scheme achieved the larger embedding capacity and the less image distortion than the other four schemes for all of the images (except for Baboon image) that were tested. This is because the proposed scheme skipped all the edge areas so that the better exploitation of smooth areas in the cover image and the minimum modification can be. In almost all test images, the PSNR of our scheme is always better than other schemes [12, 13, 14, 18] in the same embedding capacity.

TABLE III
COMPARISON OF THE PSNR (DB) VALUES BETWEEN PROPOSED SCHEME AND FOUR PREVIOUS SCHEMES FOR EC OF 20,000 BITS FOR TVU LOGO

Images	PVO [12]	IPVO [13]	PPVO [14]	QPVO [18]	Proposed
Lena	52.97	54.29	54.14	54.75	55.89
F16	55.35	55.77	55.53	55.92	55.72
Barbara	51.38	53.11	53.02	53.93	54.62
Boat	54.99	55.78	55.66	55.95	55.95
Tiffany	53.31	54.41	54.11	54.87	55.12
House	53.02	55.92	58.07	57.23	58.87
Gold hill	52.50	53.00	53.13	53.62	54.53
Peppers	52.30	53.17	52.93	53.34	53.44
Average	53.23	54.43	54.57	54.95	55.52

Tables II and III show the comparison results for EC of 10,000 bits and 20,000 bits when the embedded image is logo TVU, respectively. From Tables II and III, it is obvious that the proposed method shows better image quality than other existing schemes, i.e., PVO, IPVO, PPVO and QPVO. It is demonstrated that the proposed scheme achieves better visual quality whenever its embedding capacity is the same number of bits as its related other works. Specifically, our average PSNR reached above 57.9 dB and 55.5 dB when the EC is 10,000 bits and 20,000 bits, respectively.

TABLE IV
COMPARISON OF THE PSNR (DB) VALUES BETWEEN PROPOSED SCHEME AND FOUR PREVIOUS SCHEMES FOR EC OF 10,000 BITS FOR RANDOM BITSTREAMS

Images	PVO [12]	IPVO [13]	PPVO [14]	QPVO [18]	Proposed
Lena	58.69	60.08	60.52	60.16	60.66
F16	58.68	59.20	58.99	59.30	59.30
Barbara	56.24	57.52	56.78	57.95	58.67
Boat	59.75	60.18	60.02	60.26	60.10
Tiffany	56.99	57.89	57.43	58.40	58.64
House	60.37	62.50	62.93	63.95	63.95
Sailboat	55.72	56.68	56.81	58.91	58.81
Elaine	54.67	56.00	56.46	56.55	56.64
Gold hill	58.56	58.69	58.93	58.47	58.75
Peppers	55.88	56.83	56.52	57.11	56.95
Baboon	51.64	52.49	52.44	54.25	54.01
Average	57.02	58.00	57.98	58.66	58.77

When the embedded image is the random bitstreams, the proposed scheme also obtained the higher image quality than [12, 13, 14, 18] in the average. Tables IV, V show the comparison results when the embedded image is random bitstream with the EC of 10,000 bits and 20,000 bits, respectively.

TABLE V
COMPARISON OF THE PSNR (DB) VALUES BETWEEN PROPOSED SCHEME AND FOUR PREVIOUS SCHEMES FOR EC OF 20,000 BITS FOR RANDOM BITSTREAMS

Images	PVO [12]	IPVO [13]	PPVO [14]	QPVO [18]	Proposed
Lena	52.97	54.29	54.14	54.75	55.89
F16	55.35	55.77	55.53	55.92	55.72
Barbara	51.38	53.11	53.02	53.93	54.62
Boat	54.99	55.78	55.66	55.95	55.95
Tiffany	53.31	54.41	54.11	54.87	55.12
House	53.02	55.92	58.07	57.23	58.87
Gold hill	52.50	53.00	53.13	53.62	54.53
Peppers	52.30	53.17	52.93	53.34	53.44
Average	53.23	54.43	54.57	54.95	55.52

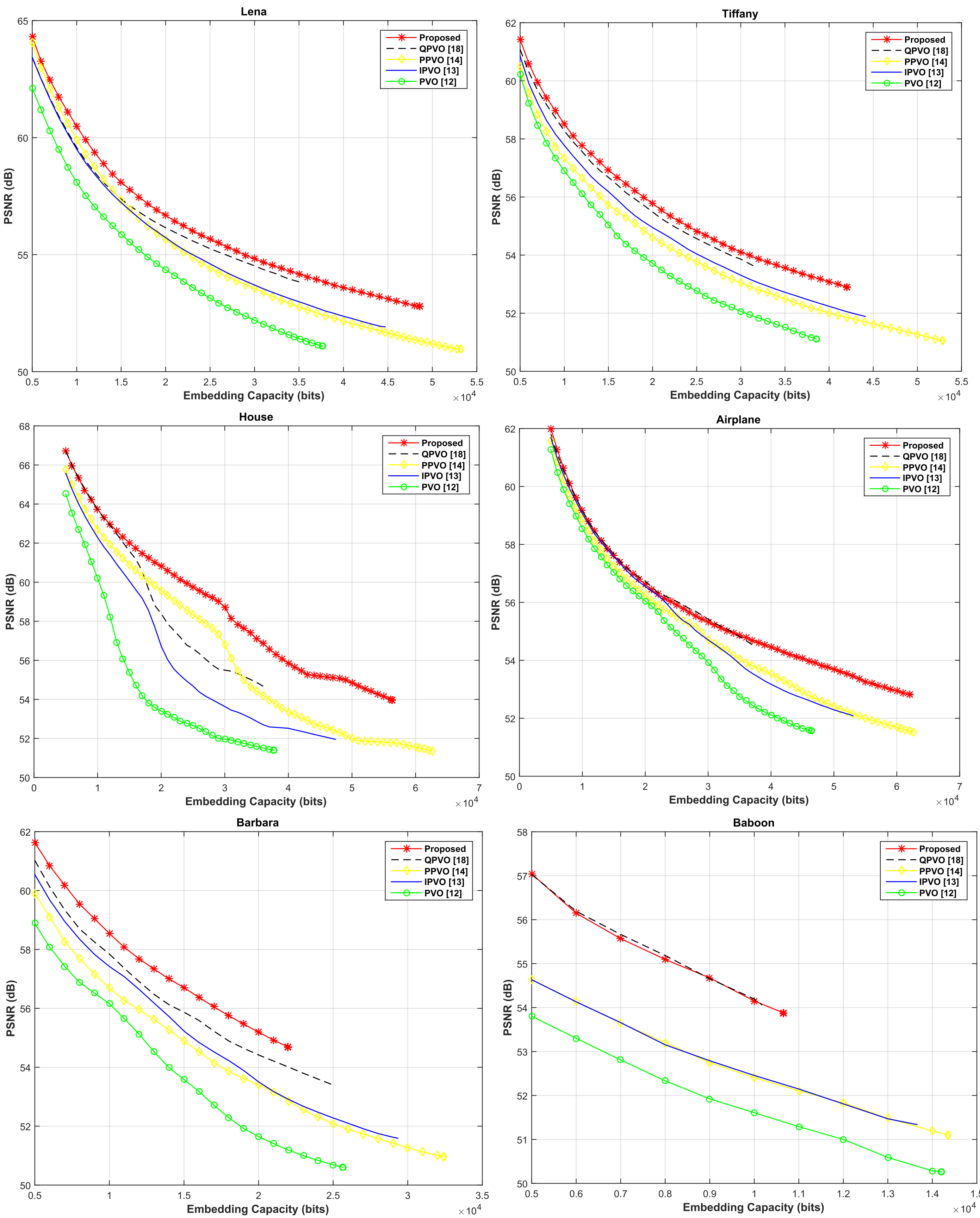

Fig. 6. The performance comparison between the proposed scheme and four previous schemes [12, 13, 14, 18] for the test images of SIPI image data set.

TABLE VI
COMPARISON OF THE PSNR (DB) VALUES BETWEEN PROPOSED SCHEME AND FOUR PREVIOUS SCHEMES FOR EMBEDDING CAPACITY OF 10,000 BITS

Images	PVO [12]	IPVO [13]	PPVO [14]	QPVO [18]	Proposed
Lena	58.08	59.50	59.92	59.57	60.48
F16	58.55	59.06	58.86	59.16	59.17
Barbara	56.16	57.42	56.69	57.84	58.54
Boat	59.60	60.01	59.86	60.10	59.94
Tiffany	56.90	57.78	57.33	58.28	58.51
House	60.20	62.28	62.71	63.72	63.72
Sailboat	55.65	56.60	56.72	58.78	58.75
Elaine	54.62	55.92	56.38	56.47	56.67
Gold hill	58.44	58.56	58.80	58.35	58.63
Peppers	55.80	56.74	56.44	57.01	56.86
Baboon	51.61	52.46	52.41	54.19	54.15
Average	56.87	57.85	57.83	58.50	58.67

Table VI and VII showed the comparison of the average PSNR of the proposed scheme with four other schemes when the all three secret images are embedded. The EC is also 10,000 bits and 20,000 bits, respectively. According to Tables VI, our scheme gets the better visual quality than four other schemes [12, 13, 14, 18] in most cases, except for Baboon, Boat and Peppers images. Our scheme gives slightly lower PSNR than QPVO [18] because in those images, the number of pixels in the edge area is very large. Once the pixels of a block are located in the edge area, the prediction value is often different from the current pixel x. It means the current pixel x can not carry data, but it is still shifted for reversibility. Therefore, the PSNR of the marked image is slightly reduced when the more pixels of the cover image are in the edge. However, the proposed scheme obtained the higher image quality than that of three other schemes, PVO, IPVO and PPVO [12, 13, 14]. In addition, the average image quality of the proposed scheme is superior among five schemes when the larger amount of the secret bits is used. It can be seen in Table VII, the proposed method outperforms all others schemes for all test images. Our average PSNR is also the highest one among five schemes.

Fig. 7 compares the average *PSNR* results of five schemes for the embedding capacities of 5,000, 10,000, 15,000, 20,000 and 25,000 bits. It is demonstrated that on average, our scheme achieves the higher *PSNR* than that of the existing works by at least 0.3 dB

TABLE VII
COMPARISON OF THE PSNR (DB) VALUES BETWEEN PROPOSED SCHEME AND FOUR PREVIOUS SCHEMES FOR EMBEDDING CAPACITY OF 20,000 BITS

Images	PVO [12]	IPVO [13]	PPVO [14]	QPVO [18]	Proposed
Lena	54.35	55.73	55.65	56.15	56.67
F16	56.04	56.55	56.26	56.73	56.62
Barbara	51.64	53.51	53.41	54.42	55.19
Boat	55.62	56.56	56.41	56.76	56.76
Tiffany	53.72	54.95	54.62	55.49	55.78
House	53.41	56.73	59.56	58.40	60.80
Gold hill	52.84	53.38	53.52	54.06	55.01
Peppers	52.62	53.57	53.31	53.75	53.84
Average	53.78	55.12	55.34	55.72	56.33

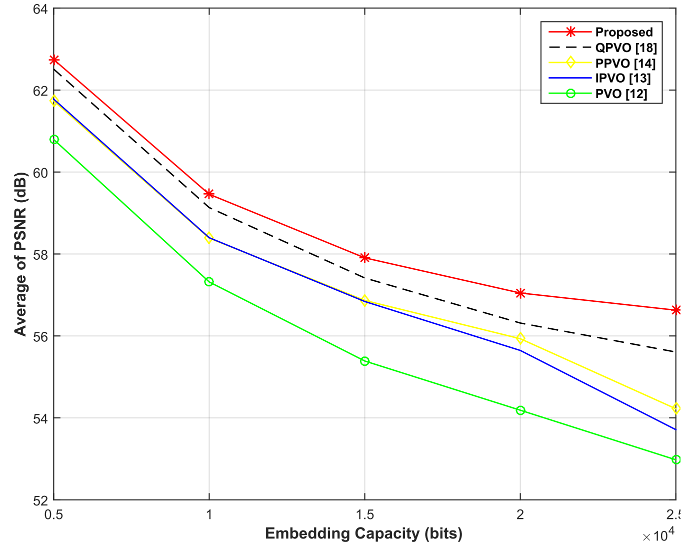

Fig. 7. Comparison of the average *PSNR* values between the proposed scheme and four previous schemes [12, 13, 14, 18] for various embedding capacities of 5,000, 10,000, 15,000, 20,000, 25,000 bits.

V. CONCLUSION

In this paper, a new RDH scheme based on edge detection and pixel value ordering is proposed. In the proposed scheme, the cover image is divided into smooth or rough areas by Canny edge detection algorithm. Then, pixels that belong to the smooth areas are used to carry the secret data by using pixel value ordering algorithm. By excluding all pixel located in the edge areas during embedding process, the proposed scheme achieves the better image quality. When compared with the previous arts, the proposed scheme obtains the higher embedding capacity under the same PSNR. Moreover, the experimental results demonstrated that our proposed scheme is superior to other four schemes in terms of the embedding capacity and the image quality while maintaining the reversibility. In the future, to improve further the performance in complex images, genetic algorithm and fuzzy logic can be used in the proposed scheme.

REFERENCES

[1] Jun Tian, "Reversible data embedding using a difference expansion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 8, pp. 890–896, Aug. 2003, doi: 10.1109/TCSVT.2003.815962.
[2] A. M. Alattar, "Reversible watermark using the difference expansion of a generalized integer transform," *IEEE Trans Image Process*, vol. 13, no. 8, pp. 1147–1156, Aug. 2004, doi: 10.1109/tip.2004.828418.
[3] Wei-Liang Tai, Chia-Ming Yeh, and Chin-Chen Chang, "Reversible Data Hiding Based on Histogram Modification of Pixel Differences," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 6, pp. 906–910, Jun. 2009, doi: 10.1109/TCSVT.2009.2017409.
[4] Hyoung Joong Kim, V. Sachnev, Yun Qing Shi, Jeho Nam, and Hyon-Gon Choo, "A Novel Difference Expansion Transform for Reversible Data Embedding," *IEEE Trans.Inform.Forensic Secur.*, vol. 3, no. 3, pp. 456–465, Sep. 2008, doi: 10.1109/TIFS.2008.924600.
[5] W. Hong, X. Zhou, T.-S. Chen, and C.-H. Hsieh, "An efficient reversible authentication scheme for demosaiced images with improved detectability," *Signal Processing: Image Communication*, vol. 80, p. 115666, Feb. 2020, doi: 10.1016/j.image.2019.115666.
[6] Y. Peng, X. Niu, L. Fu, and Z. Yin, "Image authentication scheme based on reversible fragile watermarking with two images," *Journal of Information Security and Applications*, vol. 40, pp. 236–246, Jun. 2018, doi: 10.1016/j.jisa.2018.04.007.

[7] T.-S. Nguyen and P.-H. Vo, "Reversible image authentication scheme based on prediction error expansion," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 21, no. 1, Art. no. 1, Jan. 2021, doi: 10.11591/ijeecs.v21.i1.pp253-262.

[8] Zhicheng Ni, Y. Q. Shi, N. Ansari, and Wei Su, "Reversible data hiding," in *Proceedings of the 2003 International Symposium on Circuits and Systems, 2003. ISCAS '03.*, Bangkok, Thailand, 2003, vol. 2, p. II-912-II–915. doi: 10.1109/ISCAS.2003.1206123.

[9] S. Lee, Y. Suh, and Y. Ho, "Reversible Image Authentication Based on Watermarking," in *2006 IEEE International Conference on Multimedia and Expo*, Toronto, ON, Canada, Jul. 2006, pp. 1321–1324. doi: 10.1109/ICME.2006.262782.

[10] X. Gao, L. An, X. Li, and D. Tao, "Reversibility improved lossless data hiding," *Signal Processing*, vol. 89, no. 10, pp. 2053–2065, Oct. 2009, doi: 10.1016/j.sigpro.2009.04.015.

[11] V. Sachnev, Hyoung Joong Kim, Jeho Nam, S. Suresh, and Yun Qing Shi, "Reversible Watermarking Algorithm Using Sorting and Prediction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 7, pp. 989–999, Jul. 2009, doi: 10.1109/TCSVT.2009.2020257.

[12] X. Li, J. Li, B. Li, and B. Yang, "High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion," *Signal Processing*, vol. 93, no. 1, pp. 198–205, Jan. 2013, doi: 10.1016/j.sigpro.2012.07.025.

[13] F. Peng, X. Li, and B. Yang, "Improved PVO-based reversible data hiding," *Digital Signal Processing*, vol. 25, pp. 255–265, Feb. 2014, doi: 10.1016/j.dsp.2013.11.002.

[14] X. Qu and H. J. Kim, "Pixel-based pixel value ordering predictor for high-fidelity reversible data hiding," *Signal Processing*, vol. 111, pp. 249–260, Jun. 2015, doi: 10.1016/j.sigpro.2015.01.002.

[15] X. Wang, J. Ding, and Q. Pei, "A novel reversible image data hiding scheme based on pixel value ordering and dynamic pixel block partition," *Information Sciences*, vol. 310, pp. 16–35, Jul. 2015, doi: 10.1016/j.ins.2015.03.022.

[16] S. Weng, Y. Liu, J.-S. Pan, and N. Cai, "Reversible data hiding based on flexible block-partition and adaptive block-modification strategy," *Journal of Visual Communication and Image Representation*, vol. 41, pp. 185–199, Nov. 2016, doi: 10.1016/j.jvcir.2016.09.016.

[17] P.-H. Vo, T.-S. Nguyen, V.-T. Huynh, and T.-N. Do, "A novel reversible data hiding scheme with two-dimensional histogram shifting mechanism," *Multimed Tools Appl*, vol. 77, no. 21, pp. 28777–28797, Nov. 2018, doi: 10.1007/s11042-018-5991-8.

[18] F. Di, M. Zhang, X. Liao, and J. Liu, "High-fidelity reversible data hiding by Quadtree-based pixel value ordering," *Multimed Tools Appl*, vol. 78, no. 6, pp. 7125–7141, Mar. 2019, doi: 10.1007/s11042-018-6469-4.

[19] S. Lee, C. D. Yoo, and T. Kalker, "Reversible Image Watermarking Based on Integer-to-Integer Wavelet Transform," *IEEE Trans.Inform.Forensic Secur.*, vol. 2, no. 3, pp. 321–330, Sep. 2007, doi: 10.1109/TIFS.2007.905146.

[20] Xiang Wang, Xiaolong Li, Bin Yang, and Zongming Guo, "Efficient Generalized Integer Transform for Reversible Watermarking," *IEEE Signal Process. Lett.*, vol. 17, no. 6, pp. 567–570, Jun. 2010, doi: 10.1109/LSP.2010.2046930.

[21] D. Coltuc, "Low distortion transform for reversible watermarking," *IEEE Trans. on Image Process.*, vol. 21, no. 1, pp. 412–417, Jan. 2012, doi: 10.1109/TIP.2011.2162424.

[22] J. Canny, "A Computational Approach to Edge Detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986, doi: 10.1109/TPAMI.1986.4767851.

Hoang-Nam Tram was born in Tra Vinh, Viet Nam in 1988. He was received the bachelor's degree Mathematics and Informatics from Can Tho University in 2009 and the master's degree in Computer Science from the University of Information Technology – Vietnam National University Ho Chi Minh City in 2015. He has worked as a lecturer in the School of Engineering and Technology, Tra Vinh University since 2009. His areas of interest include data hiding, information security, image and signal processing.

Thai-Son Nguyen received the M.S. and Ph.D. degrees in computer science from Feng Chia University, Taichung, Taiwan, in 2011 and 2015, respectively. He served as a lecturer in Tra Vinh University, Vietnam, from 2006. From 2019, he was Dean of the school of Engineering and Technology, Tra Vinh University. He is currently an Associate professor. His research interests include image processing, information hiding, image recognition, information security, and artificial intelligence applications. Dr. Son is currently one of the Area Editors of Journal of Research and Development on Information and Communication Technology and member of Editorial board for Scientific Journal of Tra Vinh University and Journal of Science, Engineering and Technology.