Reversible data hiding based on structural similarity block selection

WANG, Kehao; CHEN, Guohua; AI, Qingsong; CAO, Hui; ZHOU, Pan; WU, Dapeng

Published in:
IEEE Access

Published: 01/01/2020

Document Version:
Final Published version, also known as Publisher’s PDF, Publisher’s Final version or Version of Record

License:
CC BY

Publication record in CityU Scholars:
Go to record

Published version (DOI):
10.1109/ACCESS.2020.2966515

Publication details:
https://doi.org/10.1109/ACCESS.2020.2966515

Citing this paper
Please note that where the full-text provided on CityU Scholars is the Post-print version (also known as Accepted Author Manuscript, Peer-reviewed or Author Final version), it may differ from the Final Published version. When citing, ensure that you check and use the publisher's definitive version for pagination and other details.

General rights
Copyright for the publications made accessible via the CityU Scholars portal is retained by the author(s) and/or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights. Users may not further distribute the material or use it for any profit-making activity or commercial gain.

Publisher permission
Permission for previously published items are in accordance with publisher's copyright policies sourced from the SHERPA RoMEO database. Links to full text versions (either Published or Post-print) are only available if corresponding publishers allow open access.

Take down policy
Contact lbscholars@cityu.edu.hk if you believe that this document breaches copyright and provide us with details. We will remove access to the work immediately and investigate your claim.

Download date: 24/12/2023
Reversible Data Hiding Based on Structural Similarity Block Selection

KEHAO WANG, GUOHUA CHEN, QINGSONG AI, HUI CAO, PAN ZHOU, (Member, IEEE), AND DAPENG WU, (Fellow, IEEE)

1School of Information Engineering, Wuhan University of Technology, Wuhan 430070, China
2School of Electrical and Information Communications, Huazhong University of Science and Technology, Wuhan 430074, China
3Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611, USA

Corresponding author: Kehao Wang (kehao.wang@whut.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61672395, Grant 61972448, and Grant 61911540481.

ABSTRACT
Reversible data hiding (RDH) methods are widely used in many privacy-sensitive real-time applications for digital images. As an efficient RDH method, prediction-error histogram (PEH) shifting technique has found wide application for its high efficiency in increasing embedding capacity. Nowadays, the performance of most PEH-based methods is evaluated only by Peak Signal to Noise Ratio (PSNR) or Structural Similarity (SSIM) Index. However, more and more applications require high quality in both PSNR and SSIM. Therefore, in this paper, we propose a new RDH method to concentrate on both PSNR and SSIM performance, which involves two key techniques: 1) an SSIM-based block selection technique, 2) a PEH-based optimal expansion bins selection technique. Through block selection, the original image is divided into two parts: smooth blocks and rough blocks. We select pixels of smooth block for double embedding and leave pixels of rough block unchanged. According to the optimal expansion bins, reasonable pixels are chosen for embedding such that the embedding distortion can be reduced. With these improvements, our proposed method has higher SSIM and PSNR after embedding compared to other PEH-based methods. The experimental results demonstrate its superiority over some state-of-the-art counterpart and other conventional PEH-based works.

INDEX TERMS
Reversible data hiding, prediction-error histogram shifting, SSIM, block selection, double embedding.

I. INTRODUCTION
With the fast development and wide application of Internet technology, the amount of information is increasing dramatically. Leakage of information and violation of image ownership are becoming more common. To guarantee the security of information, the research on data hiding, cryptography, access control has been the hot topics. Among them, reversible data hiding (RDH), which enables receiver to recover original image losslessly and extract hidden secret data completely, has found wide applications in military, secret data transmission, image authentication [1], medical image processing [2] and so on.

In early work, most RDH methods are based on lossless compression [3], [4]. The typical principle is to losslessly compress certain features of the original image to save space for reversible data embedding. These initial methods usually have the disadvantage of low embedding capacity and may lead to severe degradation of image quality. In this light, more efficient RDH methods have been proposed to achieve better performance, such as difference expansion (DE) [5]–[10] and histogram shifting (HS) [11]–[15].

For DE-based method, it was first proposed by Tian [5]. Tian et al. calculated the difference of two adjacent pixels and expanded it for carrying one bit of embedding data. The main idea of DE-based methods is that there exists strong correlation within adjacent pixels. To further exploit the correlation, DE developed into prediction-error expansion (PEE) by incorporating some prediction methods. PEE-based methods [16]–[20] utilized the correlation of larger neighborhood pixels instead of the correlation of only two adjacent pixels in DE.

In addition to DE-based methods, the efficient RDH method based on histogram shifting (HS) was initially proposed by Ni et al. [11] in which a pair of peak (highest frequency) and zero (zero frequency) bins is chosen such that
the pixels at peak bin are used for embedding one bit data and those bins between two bins are shifted one step towards zero bin to create vacancies. According to such principle, the performance of HS-based schemes heavily depends on the determination of the peak and zero bins as well as the sharpness of the generated histogram.

To obtain a sharper histogram and enhance performance, Thodi and Rodriguez [21] proposed the algorithm to combine PEE with HS, thus the prediction-error histogram (PEH) is usually a Laplacian-like distribution centered at 0 which is suitable for expansion embedding. In this way, a large increase in embedding capacity can be realized without causing more distortion. In the light of high efficiency of PEH-based method, many RDH technologies based on PEH have been proposed in recent years, for example, double-layer embedding [22], adaptive embedding [23], [24], context modification [25], multi histogram modification [26]–[29], optimal expansion bins selection [30]–[33]. Furthermore, based on the principle of PEH-methods, other RDH methods are proposed, such as pixel-value-ordering-based methods [34]–[36]. In general, RDH methods strive to achieve better capacity-distortion performance, i.e., maintaining certain embedding capacity while minimizing the distortion.

For capacity-distortion performance of RDH methods mentioned above, they are usually evaluated only by PSNR of the marked image versus the original one because PSNR is the simplest and most widely used quality measurement index which has error sensitivity in image quality. However, with wide applications of processed image, such as the images used in medicine, machine learning and so on, PSNR is not very suitable for detecting fluctuations in image visual quality. So recently, the Structure Similarity (SSIM) [37]–[41] as structure distortion measure in RDH has attracted a great deal of attention. Thus it is highly sensitive to structural distortion and visual quality. In [39], they hid some information in the original image with taking the Human Visual System (HVS) into account. In [40], the author proposed an optimal RDH algorithm under structural similarity constraint. But these methods mainly paid attention to reduce structure distortion or improve visual quality without considering PSNR and embedding capacity.

Notice that most former RDH methods usually only use SSIM or PSNR as a quality measurement index. In this paper, we concentrate on enhancing SSIM and PSNR performance simultaneously and further employ double embedding to increase embedding capacity. The mechanism of our proposed method can be described as follows. We integrated SSIM-based block selection into the procedure of the optimal expansion bins selection. Specifically, based on block selection threshold, the original image is divided into non-overlapping blocks. For each block, if the structural similarity index is greater than a given threshold, it is marked as a smooth block; otherwise, marked as a rough block. In this way, we exclude those blocks with low structural similarity and keep the pixels in these blocks unchanged in order to avoid distortion. For the smooth blocks, we use optimal expansion bins to carry data and double embedding to increase the embedding capacity. The experiment results show our method performs better than the former methods in both SSIM and PSNR.

The rest of the paper is organized as follows. The fundamental of the Structural Similarity and PEH-based methods are briefly introduced in Section II. Section III describes the proposed the SSIM-based block selection and double embedding scheme in detail. In Section IV, the experimental results as well as the comparison with former block selection method and prior arts. Finally Section V concluded this paper.

II. RELATED WORK

In this section, we explain the concept of structural similarity which was used as a measurement index in the former methods, and then review the conventional prediction-error histogram expansion embedding.

A. STRUCTURAL SIMILARITY

The SSIM index was first put forward by the Laboratory for Image and Video Engineering at the University of Texas at Austin [37]. It is to compare the structures of the reference and the distorted signals and has gained widespread popularity for its reliable evaluation in image structure [38]–[41].

Suppose that \( x \) stands for the set of all pixel points of each block in the reference image, and \( y \) stands for the set of all pixel points of each block in the distorted image in the same position as \( x \). SSIM is based on three comparisons of similarity measurement–luminance, contrast, and structure–between \( x \) and \( y \) as follows

\[
\begin{align*}
\mu(x, y) &= \frac{2\mu_x \mu_y + c_1}{\mu^2_x + \mu^2_y + c_1} \\
\beta(x, y) &= \frac{2\sigma_x \sigma_y + c_2}{\sigma^2_x + \sigma^2_y + c_2} \\
\gamma(x, y) &= \frac{\sigma_{xy} + c_3}{\sigma_x \sigma_y + c_3}
\end{align*}
\]  

(1)

where \( \mu_x, \mu_y, \sigma_x \) and \( \sigma_y \) represent the mean and standard deviation of \( x \) and \( y \), respectively, and \( \sigma_{xy} \) represents the covariance between \( x \) and \( y \). \( c_1, c_2, \) and \( c_3 \) are constants to avoid system errors caused by \( \mu^2_x + \mu^2_y, \sigma^2_x + \sigma^2_y \) and \( \sigma_{x,y} \) being zero.

Then combining the three comparisons of luminance, contrast and structure, the SSIM index function \( \mu(\cdot) \) is denoted as:

\[
\mu(x, y) = [l(x, y)^\alpha \cdot c(x, y)^\beta \cdot s(x, y)^\gamma]
\]  

(2)

To calculate the SSIM between two images, both images are divided into non-overlapping blocks of \( n \times n \) size. Then a window of \( n \times n \) size is slid block-by-block to calculate SSIM value \( \mu \) of each block. Finally taking the average as the SSIM of whole image, the M-SSIM index function \( \tilde{\mu}(X, Y) \) is denoted as:

\[
\tilde{\mu}(X, Y) = \frac{1}{M} \sum_{i=1}^{M} \mu(X, Y_i)
\]  

(3)

where \( M \) is the total number of blocks.
According to (3), we have $|\mu(x, y)| \leq 1$ where the equality happens only when $x = y$. Moreover, $\mu(x, y) = \mu(y, x)$.

**B. PREDICTION-ERROR HISTOGRAM SHIFTING**

The procedure of data embedding of the conventional PEH-based RDH scheme is as follows.

An image is scanned in the raster-scan order from top to bottom, left to right. The pixels are collected into a one-dimensional sequence as $(x_1, \ldots, x_N)$, where $N$ is the total number of collected pixels.

Each pixel $x_i$ is predicted by some prediction algorithms among which the two most popular prediction techniques are median edge detector (MED) [7] and gradient-adjusted predictor (GAP) [16].

Then a predicted value $\tilde{x}_i$ is genetaed where $\tilde{x}_i$ should be rounded to an integer. Thus we obtain the prediction-error of pixel $x_i$ as:

$$e_i = x_i - \tilde{x}_i$$  \hspace{1cm} (4)

and then an one-dimensional prediction-error sequence $(e_1, \ldots, e_N)$.

The corresponding PEH denoted by $h(\cdot)$ can be obtained as:

$$h(e) = \# \{1 \leq i \leq N : e_i = e\}$$  \hspace{1cm} (5)

where $\#$ is the cardinal number.

One bit secret data $m \in \{0, 1\}$ is embedded into $e_i$ through expanding and shifting as:

$$\tilde{e}_i = \begin{cases} e_i + m, & \text{if } e_i = a \\ e_i - m, & \text{if } e_i = b \\ e_i + 1, & \text{if } e_i > b \\ e_i - 1, & \text{if } e_i < b - 1 \end{cases}$$  \hspace{1cm} (6)

where $b$ is the chosen expansion bin.

Then after data embedding, the marked pixel is

$$\hat{x}_i = x_i + \tilde{e}_i$$  \hspace{1cm} (7)

In the early work of PEH-based methods, bins $(b-1, b)$ are usually set as $(-1, 0)$ since the bins $(-1, 0)$ are usually two highest frequency bins. But with the development of PEH-based schemes, it is possible to improve the former modification by properly selecting the expansion bins to minimize the distortion for a given embedding capacity.

After selecting the optimal bins, the modification to prediction-error is adjusted into:

$$\tilde{e}_i = \begin{cases} e_i, & \text{if } a < e_i < b \\ e_i + m, & \text{if } e_i = b \\ e_i - m, & \text{if } e_i = a \\ e_i + 1, & \text{if } e_i > b \\ e_i - 1, & \text{if } e_i < a \end{cases}$$  \hspace{1cm} (8)

where $a < 0 \leq b, a, b \in \mathbb{Z}$.

We use an example introduced in [30] to illustrate the optimal bins selection. First, the prediction-error histogram is created, assume that $h(-1) = h(0) = h(1) = H$ and $EC = 2H$, so according to the modification shown in (6), the expected value of embedding distortion (the $l^2$-norm) is calculated as:

$$\mathbb{E}\left(\|\hat{I} - I\|_2^2\right) = \sum_{i=1}^{N} \mathbb{E}\left((\tilde{x}_i - x_i)^2\right)$$

$$= \frac{1}{2}\left(h(-1) + h(0)\right) + \sum_{e \in [-1, 0]} h(e)$$

$$= N - H$$  \hspace{1cm} (9)

where $\mathbb{E}(\cdot)$ is the expectation operator, $I$ is the original image and $\hat{I}$ is the marked image, $N$ stands for the total number of image pixels.

With the improvement of (8), the bins $(-1, 0)$ are changed into bins $(-1, 1)$. In this case, the bin 0 keeps unchanged, the bins larger than 1 and smaller than $-1$ are shifted, which means the prediction-error equalling 0 does not cause any distortion since its value remains unchanged in the embedding process. The expected value of embedding distortion is reduced from $N - H$ to $N - 2H$.

In order to obtain the optimal expansion bins, Wang et al. [30] pre-calculated the embedding distortion for each possible choice, and determined the choice with minimum distortion. By this way, the embedding distortion can be reduced compared with former methods.

**III. THE PROPOSED METHOD**

In this section, our proposed method is described in Fig. 1. First, we introduce the embedding process and its mechanism. Second, we give an algorithm to determine block selection threshold and optimal expansion bins. Third, we explain the block selection, double embedding and extracting in detail. Finally, we simple point out auxiliary information which is necessary in our scheme.

**A. EMBEDDING PROCESS**

1) PREDICTION

Assume the original image $I$ of $N \times N$ size. By scanning the original image in the raster-scan order, we obtain a two-dimensional original matrix $X$:

$$X = \begin{bmatrix} x_{1,1} & \cdots & x_{1,N} \\ \vdots & \ddots & \vdots \\ x_{N,1} & \cdots & x_{N,N} \end{bmatrix}$$  \hspace{1cm} (10)

where $x_{i,j}$ represents the pixel value of location $(i, j)$.

MED [7] is employed to predict each original pixel. The algorithm of MED is low-complexity with an inherent edge-detection mechanism operating on a three-neighbor context. The pixels $u, v, w$ at the right, bottom and bottom-right of the current pixel $x_{i,j}$ shown in Fig. 2 are defined as the context of $x_{i,j}$.

Therefore, each predicted pixel $\tilde{x}_{i,j}$ is calculated as:

$$\tilde{x}_{i,j} = \begin{cases} \min(u, v), & \text{if } w \geq \max(u, v) \\ \max(u, v), & \text{if } w \leq \min(u, v) \\ u + v - w, & \text{otherwise} \end{cases}$$  \hspace{1cm} (11)
We collect each predicted pixel to generate the corresponding two-dimensional predicted image matrix \( \tilde{X} \):

\[
\tilde{X} = \begin{bmatrix}
\tilde{x}_{1,1} & \cdots & \tilde{x}_{1,N} \\
\vdots & \ddots & \vdots \\
\tilde{x}_{N,1} & \cdots & \tilde{x}_{N,N}
\end{bmatrix}
\]  

(12)

Then we calculate the two-dimensional prediction-error matrix \( E = X - \tilde{X} \), which is denoted as:

\[
E = \begin{bmatrix}
e_{1,1} & \cdots & e_{1,N} \\
\vdots & \ddots & \vdots \\
e_{N,1} & \cdots & e_{N,N}
\end{bmatrix}
\]  

(13)

where \( e_{i,j} \) represents the prediction-error of location \((i, j)\) and can be calculated as:

\[
e_{i,j} = x_{i,j} - \tilde{x}_{i,j}
\]  

(14)

In order to calculate SSIM value, \( X \) and \( \tilde{X} \) are both divided into \([N/n] \times [N/n]\) non-overlapping blocks, where \( \lfloor \cdot \rfloor \) is the floor function and each block size is \( n \times n \). Correspondingly, the prediction-error matrix \( E \) is also divided into \([N/n] \times [N/n]\) non-overlapping blocks. The block representation of \( E \) is denoted as \( \hat{E} \).

We can know that in (1) and (2), \( \alpha > 0, \beta > 0 \) and \( \gamma > 0 \) are parameters used to adjust the relative importance of the three components. In practice to simplify the expression and calculation, we set \( \alpha, \beta, \gamma \) equal to 1, and \( c_3 = c_2/2 \) for each block. Then we calculate SSIM value of each block as:

\[
\hat{\mu}(x, y) = \frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1} \times \frac{2\sigma_x\sigma_y + c_2}{\sigma_x^2 + \sigma_y^2 + c_2} \times \frac{\sigma_x\sigma_y + c_3}{\sigma_x\sigma_y + c_3}
\]  

(15)

Each SSIM value is collected into a two-dimensional matrix \( \hat{\mu} \) as:

\[
\hat{\mu} = \begin{bmatrix}
\hat{\mu}_{1,1} & \cdots & \hat{\mu}_{1,[N/n]} \\
\vdots & \ddots & \vdots \\
\hat{\mu}_{[N/n],1} & \cdots & \hat{\mu}_{[N/n],[N/n]}
\end{bmatrix}
\]  

(16)

Therefore, each block \( \tilde{E}_{i,j} \) has its corresponding SSIM value \( \hat{\mu}_{i,j} \).

2) BLOCK SELECTION

In this part, an algorithm to determine the block selection threshold and optimal expansion bins \((a, b)\) is proposed.

The pixels in the original image need to be processed before data embedding, while such an operation including further data embedding may severely damage the structure and visual quality of some regions of the image. To reduce embedding distortion as soon as possible, we propose an SSIM-based block selection method.

Referring to (16), we use the same method to obtain the matrix \( \hat{\mu} \) because the SSIM value indicates the change in structure and visual quality before and after prediction.

Giving a block selection threshold \( \tau \) and the matrix \( \hat{\mu} \), we have that each block \((i, j)\) is a

- smooth block if \( \hat{\mu}_{i,j} \geq \tau \), or
- rough block if \( \hat{\mu}_{i,j} < \tau \).

Due to such selection, we are able to use the pixels of smooth blocks for data embedding, and keep the pixels of rough blocks unchanged. This SSIM-based block selection ensures that the local structural similarity of the image remains stable after prediction and also prevents more damage on structure and visual quality caused by further data embedding.

To guarantee the reversibility, we establish a binary block map matrix \( M \) to record the locations of smooth and rough blocks as a part of auxiliary information:

\[
M = \begin{bmatrix}
M_{1,1} & \cdots & M_{1,[N/n]} \\
\vdots & \ddots & \vdots \\
M_{[N/n],1} & \cdots & M_{[N/n],[N/n]}
\end{bmatrix}
\]  

(17)

where each \( M_{i,j} = 1 \) if the block \((i, j)\) is smooth; otherwise, \( M_{i,j} = 0 \).

3) DETERMINATION OF OPTIMAL BLOCK SELECTION THRESHOLD

According to \( \tilde{E} \), \( \hat{\mu} \) and the SSIM-based block selection threshold \( \tau \), we collect the prediction-error of block \( \tilde{E} \) as follows:

\[
\tilde{b}(\tilde{E}, \tau) = \{ e_{p,q} \mid e_{p,q} \in \tilde{E}_{i,j}, \hat{\mu}_{i,j} \geq \tau, 1 \leq i, j \leq [N/n] \}
\]  

(18)
From (18), we can know that the higher the \( \tau \), the stricter the selection, which can lead to a better SSIM performance. According to \( \tilde{B}(E, \tau) \), we can obtain the prediction-error histogram denoted as \( h(\cdot) \):

\[
h(E, \tau) = \# \{ 1 \leq p, q \leq N : e_{p,q} = e, e_{p,q} \in \tilde{B}(e, \tau) \}
\]  

(19)

According to the corresponding prediction-error histogram, the value of embedding distortion in our proposed method can be estimated as:

\[
D(a, b, \tau) = \sum_{p,q} \left( \tilde{x}_{p,q} - x_{p,q} \right)^2
\]

\[
= R + K
\]  

(20)

where \( R = \frac{|h(a, \tau)| + |h(b, \tau)|}{2} \) and \( K = \sum_{e < a} |h(e, \tau)| + \sum_{e > b} |h(e, \tau)| \).

Based on (19) and (20), we can see that the block selection result directly affects the number of pixels used for embedding, so the generation of prediction-error histogram heavily depends on the block selection result.

Different \( h(e, \tau) \) lead to different results of \( D(a, b, \tau) \). Therefore, \( D(a, b, \tau) \) is highly related to the determination of \( \tau \).

According to the aforementioned analysis, for a given embedding capacity, we need to maximize \( \tau \) to determine the minimum embedding distortion with its corresponding expansion bins \((a, b)\).

When the embedding capacity \( C \) is given, the problem of how to determine the optimal block selection threshold is formulated as follows:

\[
\max \tau \\
\text{s.t.} \quad C1 : \{ b, \tau \} = \arg \min_{b', \tau'} D(-b' - 1, b', \tau') \\
C2 : |h(-b' - 1, \tau')| + |h(b', \tau')| \geq C \\
C3 : \tau \in [0, 1] \\
C4 : b \geq 0
\]  

(21)

where \( C \) is the sum of embedding capacity (EC) and auxiliary information. \( C1 \) implies that under the constraint of a determined \( \tau \), we calculate embedding distortion for each possible choice of \( b \), and determine one with minimum distortion. \( C2 \) implies that we need to ensure there are enough space for data embedding.

4) DOUBLE EMBEDDING

After obtaining the block selection threshold, we can divide the original image into smooth blocks and rough blocks, and then embed data into the pixels in smooth blocks.

Therefore, the block representation of \( X \) can be rewritten as follows:

\[
B = \begin{bmatrix}
B_{1,1} & \cdots & B_{1,[N/n]} \\
\vdots & \ddots & \vdots \\
B_{n/[N/n],1} & \cdots & B_{n/[N/n],[N/n]}
\end{bmatrix}
\]  

(22)

where \( B_{i,j} \) contains \( n \times n \) original pixels and each block \( B_{i,j} \) has its corresponding classification index \( M_{i,j} \).

The pixels in smooth blocks are preprocessed to avoid overflow/underflow. The maximum modification to each pixel in our proposed method is 1. The pixels which are equal to 0 or 255 may lead to overflow/underflow. The pixels with value 0 are changed into 1 while the pixels with value 255 are changed into 254. In this case, a binary location map \( L_1 \) is generated to record all the locations of these pixels.

After the preprocessing, scanning the block classification matrix \( M \) in the raster-scan order, we skip and keep them unchanged for rough blocks. For smooth blocks, beginning from the pixel \( x_{p1,q1} \), we scan each pixel of each block in the raster-scan order. Each pixel \( x_{p,q} \) is predicted to obtain the predicted value \( \tilde{x}_{p,q} \) as (11). Each prediction-error \( e_{p,q} \) is generated as (14) and then modified as

\[
\tilde{e}_{p,q} = \begin{cases}
 e_{p,q} - 1, & \text{if } a < e_{p,q} < b \\
 e_{p,q} + m, & \text{if } e_{p,q} = b \\
 e_{p,q} - m, & \text{if } e_{p,q} = a \\
 e_{p,q} + 1, & \text{if } e_{p,q} > b \\
 e_{p,q} - 1, & \text{if } e_{p,q} < a
\end{cases}
\]

(23)

where \( m \in \{0, 1\} \) is one bit to-be-embedded data, \( a \) and \( b \) are the determined optimal expansion bins.

Remark 1: The prediction-errors which are equal to \( a \) or \( b \) are expanded one step to carry one bit data. To ensure blind extraction and reversibility, for the prediction-error \( e_{p,q} > b \) and \( e_{p,q} < a \), they are shifted one step right and one step left respectively. Doing so creates vacancies for the expanded prediction-errors and ensures that when blind extracting, the prediction-errors which are equal to \( b \), \( b + 1 \), \( a - 1 \) and \( a \) are all and only used to data embedding.

The shifted pixels do not carry any data and are only used to ensure reversibility and blind extraction. For the prediction-error histogram, there usually exists zero frequency points as shown in Fig. 9, and assume that their corresponding prediction-error are \( \tilde{e} \) (\( \tilde{e} > b \)) and \( \xi \) (\( \xi < a \)) respectively. Because the zero frequency points \( \tilde{e} \) and \( \xi \) create extra vacancies to store the shifted prediction-error, we only need to shift the prediction-error \( \tilde{e} > e_{p,q} > b \) one step right and \( \xi < e_{p,q} < a \) one step left. For the prediction-error \( e_{p,q} > \tilde{e} \), \( e_{p,q} < \xi \), they keep unchanged since they do not affect the reversibility and blind extraction.

By this way, the image distortion from shifting prediction errors can be reduced. Therefore, we have the following
embedding and shifting approach

\[
\tilde{e}_{p,q} = \begin{cases} 
    e_{p,q}, & \text{if } e_{p,q} < a < e_{p,q} < b, e > e_{p,q} \\
    e_{p,q} + m, & \text{if } e_{p,q} = b \\
    e_{p,q} - m, & \text{if } e_{p,q} = a \\
    e_{p,q} + 1, & \text{if } b < e_{p,q} < \tilde{e} \\
    e_{p,q} - 1, & \text{if } e < e_{p,q} < a
\end{cases}
\]  

(24)

After modification to the prediction-errors, the marked pixel \( \hat{x}_{p,q} \) is obtained

\[
\hat{x}_{p,q} = x_{p,q} + \tilde{e}_{p,q}
\]  

(25)

So far, we have completed the first round embedding. To increase embedding capacity, we employ double embedding method similar to [23] rather than those approaches based on block selection.

To implement double embedding, we need to provide a mechanism to inform the decoder unambiguously which pixels are double-embedded. Meanwhile, it reflects the local complexity of the marked pixel \( \hat{x}_{p,q} \) after first embedding. Referring to Fig. 4, we use the mean square error (MSE) of \( \hat{x}_{p,q} \) as the parameter to decide whether or not the double embedding operation is required.

The MSE index function \( \sigma \) for \( \hat{x}_{p,q} \) is denoted as

\[
\sigma_{p,q} = \frac{1}{3} \left[ (u - x_{p,q})^2 + (v - x_{p,q})^2 + (w - x_{p,q})^2 \right]
\]  

(26)

where \( x_{p,q} = \frac{1}{3} (u + v + w) \).

Therefore, we can introduce an MSE threshold \( T_\sigma \) to control which pixels can be utilized to embed the second bit data. The smaller the \( T_\sigma \) is, the less the number of available pixels is.

For each marked pixel \( \hat{x}_{p,q} \), we have

- if \( \sigma_{p,q} < T_\sigma \), the pixel \( \hat{x}_{p,q} \) is used for double embedding.
- otherwise, the pixel \( \hat{x}_{p,q} \) keeps unchanged.

The method in the second embedding is similar with that in the first embedding, i.e., employing (23). Notice that, we need to make sure that the marked pixel value \( \bar{x}_{p,q} \in [1, 254] \) after the first embedding to avoid overflow/underflow. Therefore, we need another location map \( L_2 \) to record all the overflow/underflow pixels location.

After embedding the required data into the pixels from \( x_{p_1,q_1} \) to \( x_{p'_2,q_2} \) in smooth blocks, where \( x_{p_1,q_1} \) is the first available embedding pixel and \( x_{p'_2,q_2} \) is the last pixel for embedding data, we need to embed the auxiliary information as a part of payload. The size of auxiliary information is denoted by \( S_{aux} \).

- First, scanning each pixel of the embedded image in raster-scan order, we record LSBs of the first \( S_{aux} \) pixels to obtain a binary sequence \( s \).
- Next, using the proposed embedding method, \( s \) is embedded into the pixels from \( x_{p_n,q_n} \) to \( x_{p_{end},q_{end}} \) in smooth blocks, where \( x_{p_n,q_n} \) is next available pixel of \( x_{p',q'} \) and \( x_{p_{end},q_{end}} \) is the last embedding pixel in smooth blocks.
- Finally, by using LSB replacement, we embed the auxiliary information into the first \( S_{aux} \) pixels to generate the final marked image.

Finally, after embedding the required data and auxiliary information into the pixels of the original image matrix \( X \), the marked image matrix \( \hat{X} \) is generated.

**B. Extracting Process**

Extracting process is completely opposite to embedding. Through collecting the LSBs of the first \( S_{aux} \) pixels, we obtain the auxiliary information. Then through auxiliary information, we extract the sequence \( s \) first. The extraction procedure is explained as follow.

Referring to (22), correspondingly, the marked image matrix \( \hat{X} \) is divided into \( [N/n] \times [N/n] \) non-overlapping blocks. The representation of \( \hat{X} \) is rewritten as \( \hat{B} \).

According to \( M \) and the matrix \( \hat{B} \), in the revers scanning order to scan each pixels from \( \hat{x}_{p_{end},q_{end}} \) to \( \hat{x}_{p_n,q_n} \) in smooth blocks, we obtain each marked pixel \( \hat{x}_{p,q} \). For each \( \hat{x}_{p,q} \), we calculate \( \sigma_{p,q} \) first to determine whether the pixel \( \hat{x}_{p,q} \) is double embedded or not.

- if \( \sigma_{p,q} \leq T_\sigma \), we extract two bits and recover the original pixel.
- otherwise, we extract one bit data and recover the original pixel.

For each time of extraction, we obtain the predicted value \( \hat{x}_{p,q} \) the same as (11) for each \( \hat{x}_{p,q} \). The expanded or shifted prediction-error is obtained as:

\[
\tilde{e}_{p,q} = \hat{x}_{p,q} - \tilde{x}_{p,q}
\]  

(27)

Then the original prediction-error is calculated as:

- If \( \tilde{e}_{p,q} \in [b, b + 1] \), the data is extracted as \( m = \tilde{e}_{p,q} - b \) and the original prediction-error value \( e_{p,q} \) is recovered as \( \tilde{e}_{p,q} - m \)  
- If \( \tilde{e}_{p,q} \in [a - 1, a] \), the data is extracted as \( m = a - \tilde{e}_{p,q} \) and the original prediction-error value \( e_{p,q} \) is recovered as \( \tilde{e}_{p,q} + m \)  
- If \( \tilde{e} + 1 > \tilde{e}_{p,q} > b + 1 \), no data need to be extracted and the original prediction-error value \( e_{p,q} \) is recovered as \( \tilde{e}_{p,q} - 1 \)  
- If \( \tilde{e} - 1 < \tilde{e}_{p,q} < a - 1 \), no data need to be extracted and the original prediction-error value \( e_{p,q} \) is recovered as \( \tilde{e}_{p,q} + 1 \)  
- Otherwise, no data need to be extracted and the original prediction-error value \( e_i \) is \( \tilde{e}_i \) itself

Then the original pixel value \( x_{p,q} \) is recovered as:

\[
x_{p,q} = \hat{x}_{p,q} + e_{p,q}
\]  

(28)

For those double-embedded pixels, the method of two extractions is completely same. Notice that, after the first
extraction, according to $L_2$, we need to update the overflow/underflow pixel value. Update its value as 255 if $x_i$ is 254, or 0 if $x_i$ is 1.

After the sequence $s$ is extracted, we use $s$ to replace the LSBs of the first $S_{aux}$ pixels to obtain the embedded image.

In the reverse scanning order to scan each pixel from $\tilde{x}_{p_1,q_1}$ to $\tilde{x}_{p,q}$ in smooth blocks, we extract the embedding data according to the above procedure.

Finally, according to $L_1$, we update the overflow/underflow pixel value, i.e., updating its value as 255 if $x_i$ is 254, or 0 if $x_i$ is 1.

C. AUXILIARY INFORMATION

In our proposed method, we introduce two binary variables $\varphi_1$, $\varphi_2$ to record whether there exists a location map in the first and second embedding, respectively. In addition, we utilize $M_i$, $L_1$, $L_2$ to record the block location, overflow location in the first embedding and overflow location in the second embedding. Moreover, $L_1$, $L_2$ and $M$ are all losslessly compressed into $CL_1$, $CL_2$ and $CM$. Their compressed size information is also a part of auxiliary information, denoted as $S_{CL_1}$, $S_{CL_2}$, $S_{CM}$, respectively.

Then for our method, the necessary auxiliary information includes:

- $\tilde{e}$, $\bar{e}$
- $b$, $n$, $T_\sigma$, $p_{end}$, $q_{end}$
- $\varphi_1$, $\varphi_2$
- $S_{aux}$, $S_{CM}$ and $S_{CL_1}$, $S_{CL_2}$
- $CM$, $CL_1$, $CL_2$

D. EMBEDDING AND EXTRACTING ALGORITHM

In this part, the embedding and extracting algorithms are given.

IV. EXPERIMENTAL RESULT

In this section, some experimental simulations are conducted to demonstrate the performance of the proposed method and three benchmark methods. Specifically, we use six $512 \times 512$ sized standard gray-scale images including ‘Lena’, ‘Barbara’, ‘Airplane’, ‘Peppers’, ‘Boat’, ‘Elaine’ as test images, shown in Fig. 5.

In particular, our proposed embedding algorithm is compared with Cai and Gui [36], Sachnev et al. [22], and Wang et al. [30]. Among them, Cai and Gui [36] proposed a method based on block selection, and assigned the median of each block as the reference pixel such that those pixels with the same reference value as the smooth level to select the block. Both Sachnev et al.’s method [22] and Wang et al.’s method [30] are two typical PEH-based ones. Sachnev et al. proposed a sorting technique to select prediction-error based on local variance, while Wang et al. proposed to pre-calculate the minimum embedding distortion to obtain optimal expansion bins.

For the test image ‘Lena’, the performance in different sub-block sizes is shown in Fig. 6, and we can observe that the best performance happens in the case of $n = 8$. Similarly, for other test images, we can obtain the similar result through a larger number of experiments. Hence, the size of sub-block is set to 8, i.e., $n = 8$, in our experiment.

To show the intuitive effect of smooth and rough blocks, we provide a block structure of ‘Lena’ in Fig. 7 with the embedding capacity 10000, 30000 and 50000 bits. The white
Algorithm 2 Extracting Algorithm

Input: \( \hat{X} \)
Output: \( X, m \)
1: Obtain auxiliary information
2: for \( l = l_{\text{end}} \) to \( l_{\text{end}}' \) do
3: \hspace{2em} for \( j = j_{\text{end}} \) to \( j_{\text{end}}' \) do
4: \hspace{4em} if \( M_{l,j} = 1 \) then
5: \hspace{6em} pixel from \( x_{p_{l,j},q_{l,j}} \) to \( x_{p_{l,j}',q_{l,j}'} \) is scanned in the reverse scanning order.
6: \hspace{4em} Extract and recover by (27) to (28)
7: \hspace{4em} if \( \sigma \leq T_\sigma \) then
8: \hspace{6em} Repeat step. 7
9: \hspace{4em} end if
10: \hspace{3em} else
11: \hspace{5em} continue
12: \hspace{3em} end if
13: \hspace{2em} end for
14: \hspace{1em} end for
15: \hspace{1em} Obtain the first embedded image
16: for \( l = l' \) to \( l_{\text{end}} \) do
17: \hspace{2em} for \( j = j' \) to \( j_{\text{end}} \) do
18: \hspace{4em} if \( M_{l,j} = 1 \) then
19: \hspace{6em} pixel from \( x'_{p_{l,j},q_{l,j}} \) to \( x_{p_{l,j}',q_{l,j}'} \) is scanned in the reverse scanning order.
20: \hspace{6em} Repeat step. 7
21: \hspace{6em} if \( \sigma \leq T_\sigma \) then
22: \hspace{8em} Repeat step. 7
23: \hspace{6em} end if
24: \hspace{4em} end if
25: \hspace{3em} end for
26: \hspace{2em} end for
27: \hspace{1em} end for
28: \hspace{1em} Return \( X, m \)

FIGURE 6. Performance comparison with different block size for ‘Lena’. part is the set of smooth blocks, while the black part is the set of rough blocks. We can observe that as the increase of the embedding capacity, the number of smooth blocks (white part, corresponding to the available embedding pixels) begins to increase.

From Fig. 8, we can observe that as the increase of \( T_\sigma \), the number of available embedding pixels begins to increase in the second embedding process.

From Fig. 9, we can observe that as the increase of \( \tau \) the prediction error histogram becomes more sharp, which indicates that the embedding distortion would be less in case of a larger block selection threshold \( \tau \).

As shown in Fig. 10, we can observe that for the four test images ‘Lena’, ‘Peppers’, ‘Boat’, and ‘Elaine’, the performance of our proposed method is much better than other three ones. Our curve is on average 2dB higher in PSNR.
FIGURE 10. PSNR comparison of between our method and three methods in [22], [30], [36].

than that of other three methods. Moreover, the SSIM of ‘Lena’, ‘Peppers’, ‘Boat’ is relatively high which means the structure of the predicted image is highly similar to the original one, and then we are able to eliminate the blocks which are not suitable for further adjustment during the block selection. Notice that, for ‘Elaine’, although the SSIM is low, it has enough sub-blocks with high SSIM to guarantee the embedding capacity and performance.

For the test images ‘Airplane’, ‘Barbara’, the performance of our proposed method is not stable. The SSIM of sub-block is pretty close to each other, and there are a small number of sub-blocks with relatively low SSIM. So when the embedding capacity is not high enough (for ‘Airplane’ about 17500 bits, for ‘Barbara’ about 11000 bits), the embedding algorithm cannot show the advantage of the block selection mechanism, but the influence of auxiliary information in ‘Airplane’ and ‘Barbara’ is more apparent. For ‘Airplane’, [36] has better PSNR when the embedding capacity is less than 17500 bits. For ‘Barbara’, the proposed method is pretty close to [36] and [22] when the embedding capacity is less than 11000 bits. With the increase of the embedding capacity, more pixels are utilized to embed data, and our proposed embedding algorithm outperforms other methods.

For the six test images, when a relatively small embedding capacity, the SSIM difference is not apparent between the proposed method and others. Through the PSNR experimental result, we select a relative high embedding capacity for six test images to make SSIM more intuitive. As shown in Tab. 1,
TABLE 1. SSIM comparison between our method and Cai and Gui [36], Sachnev et al. [22], Wang et al. [30].

<table>
<thead>
<tr>
<th>Image(EC(in bits))</th>
<th>[36]</th>
<th>[22]</th>
<th>[30]</th>
<th>proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena(60000)</td>
<td>0.9951</td>
<td>0.9974</td>
<td>0.9960</td>
<td>0.9974</td>
</tr>
<tr>
<td>Peppers(60000)</td>
<td>0.9948</td>
<td>0.9975</td>
<td>0.9955</td>
<td>0.9978</td>
</tr>
<tr>
<td>Boat(35000)</td>
<td>0.9958</td>
<td>0.9985</td>
<td>0.9962</td>
<td>0.9989</td>
</tr>
<tr>
<td>Elaine(40000)</td>
<td>0.9997</td>
<td>0.9992</td>
<td>0.9993</td>
<td>0.9998</td>
</tr>
<tr>
<td>Airplane(50000)</td>
<td>0.9975</td>
<td>0.9980</td>
<td>0.9977</td>
<td>0.9984</td>
</tr>
<tr>
<td>Barbara(25000)</td>
<td>0.9969</td>
<td>0.9982</td>
<td>0.9991</td>
<td>0.9992</td>
</tr>
<tr>
<td>Average</td>
<td>0.9966</td>
<td>0.9981</td>
<td>0.9973</td>
<td>0.9985</td>
</tr>
</tbody>
</table>

our proposed method performs better than other methods, i.e., having high SSIM and high PSNR.

V. CONCLUSION

In this paper, an efficient reversible data hiding scheme based on SSIM-based block selection and optimal expansion bins selection is proposed. Experimental results show the proposed algorithm outperforms other three methods in both SSIM and PSNR. In the future, an image-adaptive algorithm should be designed to further enhance its practicability and improve the accuracy of threshold determination. In addition, designing an excellent predictor is a worthy topic to investigate in the future.

REFERENCES


KEHAO WANG received the B.S. degree in electrical engineering and the M.S. degree in communication and information system from the Wuhan University of Technology, Wuhan, China, in 2003 and 2006, respectively, and the Ph.D. degree from the Department of Computer Science, University of Paris-Sud XI, Orsay, France, in 2012. He is currently an Associate Professor with the Department of Information Engineering, Wuhan University of Technology. His research interests are cognitive radio networks, wireless network resource management, and data hiding.

GUOHUA CHEN received the B.S. degree in electrical engineering from the Wuhan University of Technology, Wuhan, China, in 2018. He is currently pursuing the M.S. degree with the Department of Information Engineering, Wuhan University of Technology. His research interests include data hiding, machine learning, transfer reinforcement learning, and deep reinforcement learning.

QINGSONG AI received the B.S. degree in electrical engineering, and the M.S. and Ph.D. degrees in communication and information system from the Wuhan University of Technology, Wuhan, China, in 2003, 2006, and 2010, respectively. He is currently a Professor with the Department of Information. His research interests are digital watermark, data hiding, and information security.

HUI CAO received the Ph.D. degree in electronic engineering from the City University of Hong Kong, in October 2017. He is currently a Lecturer with the School of Information Engineering, Wuhan University of Technology. His research interests include statistical signal processing, parameter estimation, and source localization.

PAN ZHOU (Member, IEEE) received the B.S. degree in the Advanced Class of HUST, in 2006, and the Ph.D. degree from the School of Electrical and Computer Engineering, Georgia Institute of Technology (Georgia Tech), Atlanta, USA, in 2011. He was a Senior Technical Member with Oracle Inc., Boston, MA, USA, from 2011 to 2013. He is currently an Associate Professor with the School of Electronic Information and Communications, Huazhong University of Science and Technology (HUST), Wuhan, China. His current research interests include network security, machine learning and big data analytics, and information networks.

DAPENG WU (Fellow, IEEE) received the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, USA, in 2003. He is currently a Professor with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL, USA. His research interests are in the areas of networking, communications, signal processing, computer vision, machine learning, smart grid, data mining, and information and network security.