A novel two-dimensional reversible data hiding scheme based on high-efficiency histogram shifting for JPEG images

Ben He¹,², Yi Chen²,², Yonghui Zhou¹, Yongrong Wang¹ and Yanli Chen¹

Abstract
In recent years, reversible data hiding technology has been widely used in JPEG images for special purposes such as file management and image authentication. Histogram shifting is one of the most popular techniques for achieving reversible data hiding technology. However, invalid shifting in histogram shifting limits the performance of existing reversible data hiding schemes. Therefore, we propose a two-dimensional histogram shifting-based reversible data hiding scheme in this article to improve the performance of marked JPEG images in terms of visual quality and file size. In the proposed histogram shifting method, only the coefficient pairs containing two non-zero quantized discrete cosine transform coefficients are changed for embedding data. Specifically, the coefficient pairs with at least one quantized discrete cosine transform coefficient valued \( \pm 1 \) are shifted and the rests leave room for embedding data. With our proposed reversible data hiding scheme, the number of invalid shifting pixels is reduced so that it improves the performance of marked JPEG images. The experimental results show that the proposed method achieves a higher peak signal-to-noise ratio and has a lower increase in file size than state-of-art methods.

Keywords
Reversible data hiding, histogram shifting, JPEG, pixel value pairs, embedding distortion

Date received: 5 September 2021; accepted: 9 February 2022

Handling Editor: Peio Lopez Iturri

Introduction
For the last few decades, reversible data hiding (RDH) methods are springing up like mushrooms. The RDH is also called lossless data hiding in many cases. The difference between RDH methods and traditional information hiding methods is that the former can recover the original carrier image without losing any information.¹⁻⁴ And by RDH, not only the hidden data but also the original cover medium can be completely recovered by the authorized users.⁵ There are four main categories for existing RDH methods: (1) lossless compression,⁶⁻¹¹ (2) difference expansion (DE),¹²⁻¹⁴ (3) histogram shifting (HS),¹⁵⁻¹⁹ and (4) integer-to-integer transform-based methods.²⁰⁻²³ Recently, by applying RDH, the embedding distortions can be eliminated when extracting embedded data. RDH is usually used in some sensitive scenarios with distortions, such as medical images, military images, legal forensics,
Copyright protection, and other related fields, in which any distortion is not prohibited.

So far, various RDH algorithms have emerged. Among them, the prediction error expansion (PEE) algorithm is widely used. For example, in Qu and Kim, an improved pixel-based pixel value ordering (PPVO) method was proposed to predict each pixel. Compared with previous pixel value ordering (PVO) methods, it was not constrained by blocks and secret data are embedded in some smoother areas. To further reduce distortions and improve embedding capacity, a new scheme was proposed in Long et al., in which an image is divided into three L-shaped blocks with non-overlapping pixels. After these blocks are scrambled with dromion and stream encryption, the difference is calculated, and the paired prediction errors corresponding to each block are accumulated to form a two-dimensional (2D) prediction error histogram to carry data. Then, a halftone image RDH algorithm based on minimizing the visual distortions of pixel flipping and an RDH scheme based on the image texture to reduce invalid pixels shifting were proposed in Yin et al. and Jia et al. respectively. A method proposed by Li et al. to reduce invalid pixel shifting is based on PVO and PEE, which can not only reduce the number of shiftable pixels, but also improve the image visual quality for using PVO in the PEE scheme.

For RDH schemes, PVO is very important and practical. The literature used block-based ordering (neighboring pixel ordering). Sachnev et al. proposed a popular efficient data hiding using a sorting prediction technique. In Weng et al., a method based on improved dynamic pixel value ordering (IPVO) was proposed, which can flexibly modify the number of pixels in a block by classifying the local complexity into multiple levels.

HS is a very successful method of RDH, which was first proposed by Ni et al. In this method, the histogram is generated and some values in the histogram are extended to carry data. Based on the sensitivity of Human Vision System, Chang et al. introduced discrete cosine transform (DCT) technique and discarded the low-frequency components, and directly modified the continuous zero value coefficients in the mid-frequency coefficients. Xuan et al. proposed that HS technique can be used in JPEG images to extend multiple pairs of bins of DCT coefficient histograms and modify them to carry secret data. Peng et al. proposed a method in which secret data are first encoded into grouped pixels through some specific pixel transformation operations. Then, multi-segment left and right HS based on threshold operation is used to record the pixel change operation. This method is one of the checkerboard prediction techniques and multiple embedding strategies. Gao et al. proposed an RDH method based on prediction error histogram (PEH), which makes the full use of correlations of coded channels, and designed a quadratic time prediction RDH scheme based on 2D PEH. He et al. proposed an improved RDH scheme for JPEG images based on block sorting and frequency selection. According to the distribution of DCT coefficients, the scheme accurately estimates the distortion caused by the embedded data in each block.

The rest of this article is organized as follows. In section “Related works,” the related works are introduced. The proposed scheme is described formally in section “The proposed method.” The experimental results are given in section “Experimental results” and followed by a conclusion in section “Conclusion.”

**Related works**

As presented in section “Introduction,” lots of RDH schemes are based on HS. We review three main RDH schemes proposed in Huang et al., Wedaj et al., and Xiao et al., respectively, in this section.

**Huang et al.’s scheme**

Huang et al. selected quantized alternating current (AC) coefficients values of +1 and −1 in some specific smooth blocks for extended embedding. The smoothness of the block is defined according to the number of AC coefficients with 0 value in each block. That is, the smoothness of the kth block can be defined by the following formula

\[
T_k = \# \{i : z_{k,i} = 0, 0 \leq i \leq 63 \}
\]

where \# means the cardinality of a set. Note that, for a JPEG image of size \( M \times N \), the number of blocks \( k \) ranges from \( [1, M \times N/64] \). Based on the smoothness, the sequence of the quantized blocks \( \{T_{a(1)}, \ldots, T_{a(M \times N/64)}\} \), where \( a : \{1, \ldots, M \times N/64\} \rightarrow \{1, \ldots, M \times N/64\} \) is the unique one-to-one mapping such that \( T_{a(i)} \geq \ldots \geq T_{a(M \times N/64)} \), and \( a(i) \neq a(j) \) if \( T_{a(i)} = T_{a(j)} \) with \( i < j \). Finally, for a given \( p \) (\( p \) indicates the selected blocks number, and is determined as the smallest one that can provide sufficient embedding capacity), the AC coefficients in the first \( p \) blocks are modified to embed data, that is, for each \( k \in \{a(1), \ldots, a(p)\} \) and \( i \in \{1, \ldots, 63\} \), \( z_{k,i} \) is modified to \( Z_{k,i} \) as

\[
Z_{k,i} = \begin{cases} 
    z_{k,i}, & \text{if } z_{k,i} = 0 \\
    z_{k,i} + m, & \text{if } z_{k,i} = 1 \\
    z_{k,i} - m, & \text{if } z_{k,i} = -1 \\
    z_{k,i} + 1, & \text{if } z_{k,i} > 1 \\
    z_{k,i} - 1, & \text{if } z_{k,i} < -1
\end{cases}
\]
where \( m \in \{0, 1\} \) is a to-be-embedded bit. Note that, in the above process, only the first \( p \) smooth blocks are used for data embedding, and the rough blocks with indices \( \{p + 1, \ldots, M \times N / 64\} \) are kept unchanged.

**Wedaj’s scheme**

The embedding method in Wedaj et al.\(^{52}\) is the same as that in Huang et al.\(^{51}\) The difference is that the former optimized the selection of cover blocks. The embedding efficiency of each AC frequency band should be calculated, compared, and descending ranked, and \( p \) blocks with maximum embedding efficiency should be selected to carry data. Then, for each AC band \( i \in \{1, \ldots, 63\} \), the embedding efficiency denoted by \( R_i \) is calculated as

\[
R_i = \sum_{n=1}^{M \times N / 64} \frac{E_{i,n}}{S_{i,n} + \frac{Q_i}{2}} \times Q_i^2
\]

where \( Q_i \) is the quantization table entry at a position at \( i \), and \( E_{i,n} \) and \( S_{i,n} \) are defined as follows

\[
E_{i,n} = \begin{cases} 1, & \text{if } |z_{k,i}| = 1 \\ 0, & \text{if } |z_{k,i}| \neq 1 \end{cases}
\]

\[
S_{i,n} = \begin{cases} 1, & \text{if } |z_{k,i}| > 1 \\ 0, & \text{if } |z_{k,i}| \leq 1 \end{cases}
\]

After the embedding efficiency is calculated, descending order is performed from high to low. Therefore, the embedding can be done for the corresponding AC coefficients that conform to the embedding rules until all embedding payloads are satisfied.

**Li’s scheme**

Similarly, Xiao et al.\(^ {53}\) also calculated the embedding efficiency of all AC coefficient bands. However, different from Huang et al.\(^ {51}\) and Wedaj et al.,\(^ {52}\) they did not embed data in the AC coefficients with value \( \pm 1 \). Instead, they selected a pair of bins \((c_i, d_i)\) with \( c_i < d_i \) for expansion, in which the bins between \( c_i \) and \( d_i \) remain unchanged, and the bins smaller than \( c_i \) or larger than \( d_i \) are shifted. Specifically, for each \( k \in \{1, \ldots, p, \ldots, M \times N / 64\} \) and \( i \in \{1, \ldots, l, \ldots, 63\} \), the AC coefficient \( Z_{k,i} \) is modified to \( Z_{k,i}^* \) as

\[
Z_{k,i}^* = \begin{cases} Z_{k,i}, & \text{if } c_i < Z_{k,i} < d_i \\ Z_{k,i} + m, & \text{if } Z_{k,i} = d_i \\ Z_{k,i} - m, & \text{if } Z_{k,i} = c_i \\ Z_{k,i} + 1, & \text{if } Z_{k,i} > d_i \\ Z_{k,i} - 1, & \text{if } Z_{k,i} < c_i \end{cases}
\]

where \( m \in \{0, 1\} \) is the secret bit.

**The proposed method**

In this section, we will describe the proposed scheme in details.

Figure 1 shows the flowchart of the proposed scheme, in which data hiding is applied in quantized discrete cosine transform (QDCT) coefficients of cover images, and it includes two stages, block selection and data embedding. In block selection, every two adjacent QDCT coefficients are paired, and then embedding efficiency is evaluated and sorted in descending order. According to the ordered coefficient pairs, the minimum number of blocks that can meet the requirement...
of embedding capacity is \( p \), and the secret bits are then embedded.

**Coordinate classification based on QDCT coefficients pairs**

As shown in Figure 2, JPEG encoder includes three main components, namely, DCT, quantitative, and entropy code. Through JPEG compression, the original image is orthogonally transformed. The specific method is used to perform 2D DCT on the non-overlapping \( 8 \times 8 \) blocks and transport the DCT coefficients into the quantizer. The quantized DCT coefficients are arranged in a zig-zag scanning mode, and direct current (DC) coefficients are modulated by differential pulse code on one hand. Then, the run-length encoding (RLE) on the AC coefficients is pre-compressed and encoded by Huffman so that we can get the JPEG file.

What is more, in Figure 2, all coefficients are obtained by zig-zag scanning in the encoder, and then every two adjacent coefficients are paired. The subsequent embedding is based on the embedding of coefficient pairs. For example, \( S = \{s_1, s_2, \ldots, s_{64}\} \) of 64 coefficients in an \( 8 \times 8 \) block is paired as \( (a, b) = \{(s_1, s_2), (s_3, s_4), \ldots, (s_{63}, s_{64})\} \).

Figure 3 shows a quantized \( 8 \times 8 \) DCT coefficient block. Since there are 64 coefficients in a block, thus, there are 32 coefficients pairs which are then mapped into a 2D plane, as shown in Figure 4.

In Figure 4, all the coefficient pairs are divided into three categories: embeddable, immutable, and shiftable pairs. Note that the purple coordinates in Figure 4 (i.e. coefficient pairs) contain at least one 0 QDCT coefficient, and they are immutable coefficient pairs. Since modifying the coefficient 0 will produce more distortions and lead to the degradation of image quality, coefficient pairs containing the coefficient 0 are kept unchanged.

As shown in Figure 4, the red arrows of the solid line indicate that the coefficient pairs are embeddable, and the binary secret message bit is “0” or “1” around these arrows. In addition, the arrows also indicate the corresponding expansion directions of the coefficient pairs. The black dotted arrows indicate the shiftable pairs and the pairs can only be shifted to reserve room. In addition, there are four special circular red arrows, corresponding to four points, \((1, 1), (-1, 1), (-1, -1), \) and \((1, -1)\), when the embedded secret message is “10,” coefficients of these four points will be kept unchanged. What is more, if the embedded secret message is “11,” then the vertical operation is performed, that is, the coordinate value is modified up or down accordingly, as shown in Figure 4.

**Block selection**

In the method proposed in Huang et al., blocks are sorted according to the number of coefficients with 0 values in the blocks, and the experimental results also accurately show that in blocks with more zero AC coefficients, there are more AC coefficients of \(+1\) or \(-1\) generally. Therefore, taking full advantage of this statistical property, selecting the first \( p \) blocks sorted blocks,
pairing the coefficients in these blocks, and embedding the data into coefficient pairs with a value of +1 or −1 can reduce the distortion of the file and the incremental value of storage size.

However, we do not take the above scheme into two accounts here. On one hand, all AC coefficients of each block can be classified into three categories: embeddable, immutable, and shiftable AC coefficients. In some cases, the number of embeddable coefficients at a given spatial frequency is greater than the number of shiftable coefficients. Figure 5 shows the total number of embeddable and shiftable coefficients on each AC coefficient bit of all blocks of the aircraft F-16 image and the boat image, respectively. Figure 5 shows
that the embeddable AC coefficient at position 25 is obviously more suitable for embedding. On the other hand, the modification cost of each location is different, that is, the non-uniform quantization table structure leads to less distortion caused by modifying the AC coefficients near the DC coefficient than those at high frequency. Therefore, we will choose a method to select the position of the embedding coefficient below.

Fortunately, for a JPEG image with size $M \times N$, an embedding efficiency metric is introduced, which was proposed in Wedaj et al., that is, equation (3), including the number of embeddable and shiftable AC coefficients in all non-overlapping $8 \times 8$ block to select a position of the AC coefficients to reduce distortions. In addition, in equation (3), for the non-uniform structure of the quantization table, modifying the AC coefficient which is near the DC coefficient may lead to less distortion than modifying the last one.

So based on the idea of Wedaj et al. and according to equation (3), the embedding efficiency $R_i$ of 31 AC coefficient pairs is calculated (The first coefficient pair contains DC coefficient, and this coefficient pair will be discarded.) Then, the range of $i$ in equations (3)–(5) is changed to $i \in \{1, 2, 3, \ldots, 31\}$. Finally, descending order is performed.

**Data embedding**

Compared with the three schemes mentioned in section “Related works,” our method is different in the following two aspects:

1. In all coefficient pairs, those containing at least one zero element are kept unchanged.
2. Only coefficient pairs including at least one of the AC coefficients of $+1$ and $-1$ are expanded for data embedding.

For convenience, we use $(a_i, b_i)$ and $m \in \{0, 1\}$ to represent the coefficient pairs and the secret bit, respectively.

The embedding is implemented as follows:

Step 1: get the sequence $T_k$ of the block as the method in section “Huang et al.’s scheme.” As described in section “Block selection,” the QDCT coefficient pairs (called coordinates) of each block can be obtained from these blocks. In coordinates, if there is one or two zero elements, acquire a next coordinate. If not, go to Step 2.

Step 2: if the coordinate $(a_i, b_i)$ belongs to $(1, 1), (-1, 1), (-1, -1)$ and $(1, -1)$, enter Step 3, otherwise go to Step 5.

Step 3: next, if $m_i = 0$, the embedding operation is as follows, otherwise go to Step 4

\[
(a_i, b_i) = \begin{cases} 
(a_i + 1, b_i), & \text{if } a_i = 1 \text{ and } b_i = \pm 1 \\
(a_i - 1, b_i), & \text{if } a_i = -1 \text{ and } b_i = \pm 1 
\end{cases}
\]

(7)

Step 4: when $m_i = 1$

\[
(a_i, b_i) = \begin{cases} 
(a_i, b_i + 1), & \text{if } m_{i+1} = 0 \\
(a_i - 1, b_i), & \text{if } m_{i+1} = 1, a_i = \pm 1 \text{ and } b_i = -1 \\
(a_i + 1, b_i), & \text{if } m_{i+1} = 1, a_i = \pm 1 \text{ and } b_i = 1 \\
(a_i, b_i - 1), & \text{if } m_{i+1} = 1, a_i = \pm 1 \text{ and } b_i = -1 
\end{cases}
\]

(8)

Step 5: in this step, all operations are based on $(a_i, b_i)$ in which only one of the values belongs to $+1$ or $-1$, otherwise, skip to Step 6. When $m_i = 0$

\[
(a_i, b_i) = \begin{cases} 
(a_i, b_i + 1), & \text{if } a_i = \pm 1 \text{ and } b_i \geq 2 \\
(a_i + 1, b_i), & \text{if } a_i = \pm 1 \text{ and } b_i \leq -2 \\
(a_i - 1, b_i), & \text{if } a_i \geq 2 \text{ and } b_i = 1 \\
(a_i, b_i - 1), & \text{if } a_i \leq -2 \text{ and } b_i = 1 
\end{cases}
\]

(9)

When $m_i = 1$

\[
(a_i, b_i) = \begin{cases} 
(a_i, b_i + 1), & \text{if } |a_i| \geq 2 \text{ and } b_i = 1 \\
(a_i - 1, b_i + 1), & \text{if } |a_i| \geq 2 \text{ and } b_i = -1 \\
(a_i, b_i - 1), & \text{if } a_i = 1 \text{ and } b_i \geq 2 \\
(a_i - 1, b_i), & \text{if } a_i = -1 \text{ and } b_i \geq 2 \\
(a_i - 1, b_i - 1), & \text{if } a_i = -1 \text{ and } b_i \leq -2 \\
(a_i + 1, b_i - 1), & \text{if } a_i = 1 \text{ and } b_i \leq -2 
\end{cases}
\]

(10)

where $|x|$ represents the absolute value of $x$.

Otherwise, shifting $(a_i, b_i)$ as follows

\[
(a_i, b_i) = \begin{cases} 
(a_i + 1, b_i + 1), & \text{if } a_i \geq 2 \text{ and } b_i \geq 2 \\
(a_i - 1, b_i + 1), & \text{if } a_i \leq -2 \text{ and } b_i \geq 2 \\
(a_i - 1, b_i - 1), & \text{if } a_i = -2 \text{ and } b_i \leq -2 \\
(a_i + 1, b_i - 1), & \text{if } a_i = 2 \text{ and } b_i \leq -2 
\end{cases}
\]

(11)

**Data extraction**

Data extraction is reverse processing of data embedding. For the shiftable coefficient pairs, they are shifted back to their original place, while other parts require reversible extraction. Denote $m_i$ the extracted bit
messages, where $i$ represents the $i$th binary bit. The extraction steps are as follows:

Step 1: this step is to recover all shifted coefficient pairs

$$
(a_i, b_i)' = \begin{cases} 
(a_i - 1, b_i - 1)^*, & \text{if } a_i > 2 \text{ and } b_i > 2 \\
(a_i + 1, b_i - 1)^*, & \text{if } a_i < -2 \text{ and } b_i > 2 \\
(a_i + 1, b_i + 1)^*, & \text{if } a_i < -2 \text{ and } b_i < -2 \\
(a_i - 1, b_i + 1)^*, & \text{if } a_i > 2 \text{ and } b_i < 2 
\end{cases}
$$

(12)

where $(a_i, b_i)^*$ represents the marked QDCT coefficient pairs and $(a_i, b_i)$ represents the recovered QDCT coefficient pair. After that, do inverse quantization and inverse DCT transforms to get the original pixel values.

Step 2: in this step, we do not only need to restore the paired pixel values to their original values

$$
(a_i, b_i)' = \begin{cases} 
(a_i - 1, b_i - 1)^*, & \text{if } a_i > 1 \text{ and } b_i = \pm 1 \\
(a_i, b_i - 1)^*, & \text{if } a_i = \pm 1 \text{ and } b_i > 2 \\
(a_i + 1, b_i)^*, & \text{if } a_i < -1 \text{ and } b_i = \pm 1 \\
(a_i, b_i + 1)^*, & \text{if } a_i = \pm 1 \text{ and } b_i < -2 
\end{cases}
$$

(13)

but also extract the bit of the embedded secret message whose binary bit is “0,” that is, $m_j = 0$.

Step 3: the following formula is used to extract the binary bit “1,” that is, $m'_i = 1$

$$
(a_i, b_i)' = \begin{cases} 
(a_i, b_i - 1)^*, & \text{if } |a_i| \geq 2 \text{ and } b_i = 2 \\
(a_i, b_i + 1)^*, & \text{if } |a_i| \geq 2 \text{ and } b_i = -2 \\
(a_i + 1, b_i - 1)^*, & \text{if } a_i = -2 \text{ and } b_i > 2 \\
(a_i + 1, b_i + 1)^*, & \text{if } a_i = -2 \text{ and } b_i < -2 \\
(a_i - 1, b_i - 1)^*, & \text{if } a_i = 2 \text{ and } b_i > 2 \\
(a_i - 1, b_i + 1)^*, & \text{if } a_i = 2 \text{ and } b_i < -2 
\end{cases}
$$

(14)

where $|x|$ represents the absolute value of $x$.

Step 4: for $(1, 2), (-1, 2), (-1, -2), \text{and} (1, -2)$, two bits of secret information need to be extracted continuously, that is, $m_i m'_{i+1} = 11$, such that

$$
(a_i, b_i)' = \begin{cases} 
(a_i, b_i - 1)^*, & \text{if } a_i = \pm 1 \text{ and } b_i = 2 \\
(a_i, b_i + 1)^*, & \text{if } a_i = \pm 1 \text{ and } b_i = -2 
\end{cases}
$$

(15)

Step 5: finally, for $(1, 1), (1, -1), (-1, -1), \text{and} (-1, 1)$, the coefficient pair will be recovered according to formula (16). At the same time, two consecutive binary bits 10, that is, $m_i m'_{i+1} = 10$ will be extracted.

$$
(a_i, b_i)' = \begin{cases} 
(a_i, b_i)^*, & \text{if } a_i = \pm 1 \text{ and } b_i = \pm 1 
\end{cases}
$$

(16)

If the marked image is transmitted in a lossless channel, $(a_i, b_i)' = (a, b)$, then the cover image can be recovered losslessly.

**Experimental results**

Visual quality and file size increment are two performance evaluation metrics commonly used for evaluating RDH methods. In this section, the proposed method is evaluated against three state-of-the-art methods, that is, Huang et al.’s method,\textsuperscript{51} Wedaj et al.’s method,\textsuperscript{52} and Li et al.’s method.\textsuperscript{53}

To better show the performance, we give simulations for four methods, that is, the above three methods and ours, and compared each other in both the visual quality and the file size increment, where the setting is coherent.

Six color images—that is, Lena, Sailboat, F-16, Baboon, Peppers, and Splash—with the size of $512 \times 512$ are downloaded from the USC-SIPI database for experiments. These color images are first converted into six grayscale images and then into JPEG images with $QF = 70$ and $QF = 80$. Related analysis and comparisons are described in the following.

**Visual quality**

In this section, we take advantage of PSNR (Peak-Signal-to-Noise-Rate) to evaluate the visual quality of marked JPEG images with fixed embedding payloads. The relationship between PSNRs and embedding payloads on the six marked JPEG images is shown in Figure 6, where the horizontal axis represents the embedding capacity and the vertical axis represents the PSNRs. As shown in Figure 6, with the increasing of embedding payload, PSNRs for the four RDH schemes decrease. However, although PSNRs of the four methods decrease, it can still be found that under the same embedding payload condition, PSNRs of the proposed method are higher than those of the other three methods.

It can be clearly concluded as follows:

1. PSNRs in the proposed method are higher than those in other methods with the same embedding capacity.
2. With the increasing of the embedding capacity, PSNRs of the proposed method are closed to those in Xiao et al.\textsuperscript{53} The reason is that the proposed method in this article combines every two coefficients in each block into a coefficient pair and maps them to a 2D histogram, which has less embedding capacity than the method in Xiao et al.\textsuperscript{53} However, the method in this article uses more blocks, but makes fewer changes to the coefficients within each block.
In order to clearly explain the performance, we list PSNRs of six test images with 2500 bits and 4096 bits embedded by the four schemes in Tables 1 and 2. It can be found that the scheme proposed in this article is with high PSNRs. Comparing Tables 1 and 2, it is found that PSNRs of our scheme are higher than those of the other three schemes with less embedded bits. However, as the number of embedded secret bits increases, PSNRs of the method presented in this article become closer to those of the other three schemes, because when more and more secret data are embedded, more blocks are needed; thus, block selection is equivalent to no selection at all. However, the method of coefficient pairing adopted in this article has less embedding capacity compared with the other three methods.

These experiment results show that the proposed scheme goes well, which is profited from the 2D RDH: on one hand, we select relatively smooth regions for embedding through the pairwise method and block selection on the 2D JPEG domain; on the other hand,
we discard all the AC coefficients containing 0. The reason is that whenever a zero AC coefficient is modified to non-zero, an extra symbol is needed to be coded, which will increase the size of the file.

Also, in some images, such as Lena and Aircraft F-16 in Figure 6, PSNRs in Huang et al.51 are larger than those of the other three schemes, which may be due to the texture features. It is worth mentioning that our proposed method is based on 2D HS method, and only the AC coefficient $c_{6,1}$ is selected for embedding, which is different from Xiao et al.,53 so the improvement of PSNRs is not significant in some images.

### File Size Increment

For RDH, the file size increment is also an important comparison index. In general, a better RDH scheme of JPEG images should minimize the increase storage size as much as possible while ensuring the embedding capacity. Table 3 lists the size increasing of each test image.

**Table 1.** Comparison of PSNR (dB) when the embedded bits are 2500 bits.

| Image  | Methods         | Huang et al.51 | Wedaj et al.52 | Xiao et al.53 | Proposed |
|--------|-----------------|----------------|----------------|--------------|----------|
| Lena   |                 | 49.58          | 49.55          | 52.21        | 53.88    |
| Boat   |                 | 46.64          | 46.98          | 47.03        | 47.67    |
| F-16   |                 | 47.85          | 48.06          | 48.30        | 48.75    |
| Baboon |                 | 46.19          | 46.47          | 46.66        | 46.91    |
| Peppers|                 | 46.93          | 47.34          | 47.31        | 47.76    |
| Splash |                 | 47.74          | 48.29          | 48.32        | 48.52    |

PSNR: peak signal-to-noise ratio.

**Table 2.** Comparison of PSNR (dB) when the embedded bits are 4096 bits.

| Image  | Methods         | Huang et al.51 | Wedaj et al.52 | Xiao et al.53 | Proposed |
|--------|-----------------|----------------|----------------|--------------|----------|
| Lena   |                 | 49.58          | 49.55          | 52.21        | 53.88    |
| Boat   |                 | 46.64          | 46.98          | 47.03        | 47.67    |
| F-16   |                 | 47.85          | 48.06          | 48.30        | 48.75    |
| Baboon |                 | 46.19          | 46.47          | 46.66        | 46.91    |
| Peppers|                 | 46.93          | 47.34          | 47.31        | 47.76    |
| Splash |                 | 47.74          | 48.29          | 48.32        | 48.52    |

PSNR: peak signal-to-noise ratio.

**Table 3.** The file size increment (bits) of each test image compared with the original image at different embedding capacities when QF = 70 and QF = 80.

| Image  | Method         | Embedding capacity (bits) with QF = 70 | Embedding capacity (bits) with QF = 80 |
|--------|----------------|---------------------------------------|---------------------------------------|
| Lena   | Huang et al.51 | 6000  7029 10,725 14,012 17,567       | 6000  7088 11,952 14,691 18,715       |
|        | Wedaj et al.52 | 6880 10,576 13,957 17,296             | 7021 11,084 14,264 18,201             |
|        | Xiao et al.53  | 6344 10,284 13,648 16,432             | 6218 10,241 13,971 17,891             |
|        | Proposed       | 6412 10,326 13,064 16,284             | 6318 10,471 13,641 17,611             |
| Boat   | Huang et al.51 | 7256 11,225 15,039 18,742             | 7882 11,754 15,106 19,225             |
|        | Wedaj et al.52 | 7256 11,225 15,039 18,742             | 7882 11,754 15,106 19,225             |
|        | Xiao et al.53  | 6671 10,824 14,552 17,360             | 7605 11,085 14,810 19,051             |
|        | Proposed       | 6640 9975 14,914 17,280                | 7512 10,591 13,759 18,761             |
| F-16   | Huang et al.51 | 7248 10,503 13,776 16,914             | 7583 11,052 14,251 17,782             |
|        | Wedaj et al.52 | 6992 10,208 13,468 16,843             | 7310 10,768 13,894 17,549             |
|        | Xiao et al.53  | 6438 9456 12,741 15,676                | 7105 9768 13,549 17,109               |
|        | Proposed       | 6192 9014 12,891 15,240                | 6571 10,254 13,715 16,861             |
| Baboon | Huang et al.51 | 7201 10,405 14,392 17,802             | 7691 11,092 15,068 18,861             |
|        | Wedaj et al.52 | 6944 10,296 14,352 17,709             | 7529 11,248 14,892 18,267             |
|        | Xiao et al.53  | 6371 9983 13,766 17,183                | 7058 11,369 13,982 17,852             |
|        | Proposed       | 6011 9890 13,474 16,852                | 6746 9969 13,196 16,992                |
| Peppers| Huang et al.51 | 8031 11,729 14,640 17,568             | 9024 13,054 15,743 18,628             |
|        | Wedaj et al.52 | 8643 11,586 14,421 17,423             | 9037 12,473 15,449 18,120             |
|        | Xiao et al.53  | 8518 11,025 14,621 17,005              | 8728 11,476 15,001 17,682             |
|        | Proposed       | 8479 11,056 14,342 17,074              | 8522 10,996 14,862 17,666             |
| Splash | Huang et al.51 | 8906 12,579 15,996 19,106             | 9658 13,568 16,031 19,827             |
|        | Wedaj et al.52 | 8849 12,303 15,743 18,934             | 9315 13,157 15,990 19,821             |
|        | Xiao et al.53  | 8845 12,620 15,397 18,713             | 9018 13,033 15,364 19,305             |
|        | Proposed       | 8831 12,269 15,534 18,529             | 9022 12,694 15,514 18,934             |
| Average| Huang et al.51 | 7894 11,328 14,723 17,962             | 8231 12,261 15,296 18,870             |
|        | Wedaj et al.52 | 7594 11,032 14,497 17,825             | 7999 11,747 14,933 18,531             |
|        | Xiao et al.53  | 7198 10,699 14,121 17,212             | 7655 11,162 14,446 18,148             |
|        | Proposed       | 7094 10,422 14,111 16,877             | 7449 10,829 14,115 17,804             |
image compared with the original image at different embedding capacities when \( QF = 70 \) and \( QF = 80 \). In Table 3, it demonstrates that in the four schemes, all the file sizes increase. It means that embedding secret message necessarily results in increasing the file size of the marked image compared to the original image.

However, although in this article, two values in a coefficient pair may be increased or decreased by 1, respectively, or kept unchanged in each embedding operation, the size of the final file does not increase significantly.

Figure 7 shows the average file size increment for the three reference methods and the proposed method for

Figure 7. Under the condition of \( QF = 70 \), the method proposed in this article is compared with the file size increment corresponding to the three state-of-the-art methods.51–53
different embedding capacities. The horizontal axis represents the size of embedded capacity (in bits), and the vertical axis represents the value of increment of the file.

As shown in Figure 7, the file size increment is compared with that in Huang et al.\textsuperscript{51} and Wedaj et al.\textsuperscript{52} under the same embedded capacity condition. It shows that the file size increment of the method in this article is smaller and better. However, this advantage is not obvious comparing with that in Xiao et al.\textsuperscript{53} Similarly, as shown in Figure 7, the comparison of file size increment between the methods in this article and Xiao et al.\textsuperscript{53} under the same embedding capacity is very close or even slightly insufficient. The reason is that data are embedded in coefficient pairs. Each embedding operation is to modify at least one AC coefficient, and in most cases, both two AC coefficients will be modified.

**Figure 8.** Under the condition of QF = 80, the method proposed in this article is compared with the file size increment corresponding to the three state-of-the-art methods.\textsuperscript{51–53}
Xiao et al.\textsuperscript{53} did not embed data in all the AC coefficients with value $+1$ or $-1$, but adaptively select AC coefficients.

Figure 8 shows the comparison of the file size increment with different payloads when QF = 80. The scheme in this article has smaller file size increment. What is more, combining the results in Figure 7 with Figure 8, it can be found that with different QF conditions, the growth trend of file size increment is same.

**Computational complexity**

Usually, time complexity is used to illustrate the computational complexity. And the time complexity of schemes in this article and the research studies\textsuperscript{51-53} is $O(n^2)$. However, it is generally affected by the performance of computer hardware. All experiments in this article have been implemented in Windows 10 64-bit operating system with Intel(R) Core(TM) i5-9500F CPU @ 3.00 GHz and memory size of 8192 MB RAM. Simulation codes have been run in MATLAB R2018b. As shown in Table 4, it is a comparison of the runtime with 2500 bits and 14,400 bits secret message, respectively. The average runtime of six carrier images in three aspects is compared, such as total runtime, embedding process runtime, and extraction process runtime. It can be seen from Table 4 that the average runtime of the method proposed in this article is lower than that in Wedaj et al.\textsuperscript{52} and Xiao et al.,\textsuperscript{53} and higher than that in Huang et al.\textsuperscript{51} For the schemes in Wedaj et al.\textsuperscript{52} and Xiao et al.\textsuperscript{53} and the scheme in this article all based on the effectiveness of the scheme in Huang et al.\textsuperscript{51}

### Table 4. Average runtime (seconds) of the six test images.

| Embedded bits | Methods          | Total time | Embedded time | Extracted time |
|---------------|------------------|------------|---------------|----------------|
| 2500          | Huang et al.\textsuperscript{51} | 1.37       | 0.75          | 0.68           |
|               | Wedaj et al.\textsuperscript{52} | 6.25       | 3.52          | 3.29           |
|               | Xiao et al.\textsuperscript{53}  | 7.05       | 4.05          | 4.34           |
|               | Proposed         | 4.56       | 2.68          | 2.59           |
| 14,400        | Huang et al.\textsuperscript{51} | 2.06       | 1.11          | 0.98           |
|               | Wedaj et al.\textsuperscript{52} | 8.69       | 4.69          | 4.25           |
|               | Xiao et al.\textsuperscript{53}  | 10.50      | 6.74          | 5.38           |
|               | Proposed         | 5.79       | 2.74          | 2.66           |

### Conclusion

JPEG image format is becoming a research hotspot because of its universality. In this article, a novel RDH method based on 2D HS is proposed. It is based on non-zero AC coefficient pairs. In general, modifications to JPEG images introduce distortions and file size increment. The experimental results have shown that our proposed RDH scheme can reduce the distortions while ensuring the image quality, which is better than some advanced RDH schemes. Also, the bit rate increase of the marked image is lower than state-of-the-art schemes. This proposed method can be applied to the sensitive scenarios with distortions, especially for the embedding distortions. In this article, secret data are only embedded in QDCT coefficients with $+1$ and $-1$ values. However, more embedding locations would be discussed for more payloads and the application of the proposed scheme would be an interesting topic in the future.

### Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

### Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work is supported by the National Key Research and Development Program of China (grant no. 2020YFC0833406), the National Natural Science Foundation of China (NSFC) under the grant nos 62102112, 61901096, and 61902085, and the Basic Research Plan of Guizhou Province (grant no. Qiankehejichu-ZK (2021) Yiban310).

### ORCID iDs

Ben He \(\text{https://orcid.org/0000-0001-6046-4294}\)

Yi Chen \(\text{https://orcid.org/0000-0003-4272-7956}\)

Yanli Chen \(\text{https://orcid.org/0000-0003-4452-5725}\)

### References

1. Ni Z, Shi YQ, Ansari N, et al. Reversible data hiding. *IEEE Trans Circuits Syst Video Technol* 2006; 16: 354–362.

2. Chen X, Zhong H and Bao Z. A GLCM-feature-based approach for reversible image transformation. *CMC: Comput Mater Con* 2019; 59: 239–255.

3. Chen X, Zhong H and Qiu A. Reversible data hiding scheme in multiple encrypted images based on code division multiplexing. *Multimed Tools Appl* 2019; 78: 7499–7516.
4. Xiong L and Dong D. Reversible data hiding in encrypted images with somewhat homomorphic encryption based on sorting block-level prediction-error expansion. *J Inf Secur Appl* 2019; 47: 78–85.

5. Shi YQ, Li XL, Zhang XP, et al. Reversible data hiding: advances in the past two decades. *IEEE Access* 2016; 4: 3210–3237.

6. Celik MU, Sharma G, Tekalp AM, et al. Lossless generalized-LSB data embedding. *IEEE Trans Image Process* 2005; 14(2): 253–266.

7. Zhang W, Hu X, Li X, et al. Recursive histogram modification: establishing equivalency between reversible data hiding and lossless data compression. *IEEE Trans Image Process* 2013; 22(7): 2775–2785.

8. Zhang XP. Separable reversible data hiding in encrypted image. *IEEE Trans Inf Forensic Secur* 2012; 7(2): 826–832.

9. Zhang XP. Reversible data hiding with optimal value transfer. *IEEE Trans Multimedia* 2013; 15(2): 316–325.

10. Sang J, Wang HX, Qian Q, et al. An efficient fingerprint identification algorithm based on minutiae and invariant moment. *Pers Ubiquitous Comput* 2018; 22(1): 71–80.

11. Qian Z and Zhang X. Reversible data hiding in encrypted images with distributed source encoding. *IEEE Circuits Syst Video Technol* 2016; 26(4): 636–646.

12. Wu HZ, Shi YQ, Wang HX, et al. Separable reversible data hiding for encrypted palette images with color partitioning and flipping verification. *IEEE Trans Circuits Syst Video Technol* 2017; 27(8): 1620–1631.

13. Tian J. Reversible data embedding using a difference expansion. *IEEE Trans Circuits Syst Video Technol* 2003; 13(8): 890–896.

14. Li XL, Zhang WM, Gui XL, et al. A novel reversible data hiding scheme based on two-dimensional difference-histogram modification. *IEEE Trans Inf Forensic Secur* 2013; 8(7): 1091–1100.

15. Dragoi C and Coltuc D. Improved rhombus interpolation for reversible watermarking by difference expansion. In: *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, Bucharest, 27–31 August 2012, pp.1688–1692. New York: IEEE.

16. Tsai P, Hu YC and Yeh HL. Reversible image hiding scheme using predictive coding and histogram shifting. *Signal Process* 2009; 89(6): 1129–1143.

17. Wu HT and Huang JW. Reversible image watermarking on prediction errors by efficient histogram modification. *Signal Process* 2012; 92(12): 3000–3009.

18. Li XL, Li B, Yang B, et al. General framework to histogram-shifting-based reversible data hiding. *IEEE Trans Image Process* 2013; 22(6): 2181–2191.

19. Wang JX, Chen X, Ni JQ, et al. Multiple histograms-based reversible data hiding: framework and realization. *IEEE Trans Circuits Syst Video Technol* 2020; 30(8): 2313–2328.

20. Zhang WM, Hu XC, Li XL, et al. Optimal transition probability of reversible data hiding for general distortion metrics and its applications. *IEEE Trans Image Process* 2015; 24(1): 294–304.

21. Coltuc D and Chassery JM. Very fast watermarking by reversible contrast matching. *IEEE Signal Process Lett* 2007; 14(4): 255–258.

22. Peng F, Li XL and Yang B. Adaptive reversible data hiding scheme based on integer transform. *Signal Process* 2012; 92(1): 54–62.

23. Chen Y, Wang HX, Wu HZ, et al. Reversible video data hiding using zero QDCT coefficient-pairs. *Multimed Tools Appl* 2019; 78(16): 23097–23115.

24. Wu HZ and Wang HX. Multibit color-mapping steganography using depth-first search. In: *Proceeding: 2013 international symposium on biometrics and security technologies*, Chengdu, China, 2–5 July 2013, pp.224–229. New York: IEEE.

25. Wu HZ, Wang HX and Shi YQ. Dynamic content selection-and-prediction framework applied to reversible data hiding. In: *Proceeding: IEEE international workshop on information forensics and security (WIFS)*, Abu Dhabi, United Arab Emirates, 4–7 December 2016, pp.1–6. New York: IEEE.

26. Xuan GT, Shi YQ, Teng JZ, et al. Double-threshold reversible data hiding. In: *2010 IEEE international symposium on circuits and systems (ISCAS)*, Paris, 30 May–2 June 2010, pp.1129–1132. New York: IEEE.

27. Gao XB, An LL, Yuan Y, et al. Lossless data embedding using generalized statistical quantity histogram. *IEEE Trans Circuits Syst Video Technol* 2011; 21(8): 1061–1070.

28. Li XL, Yang B and Zeng TY. Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection. *IEEE Trans Image Process* 2011; 20(12): 3524–3533.

29. Coltuc D. Improved embedding for prediction-based reversible watermarking. *IEEE Trans Inf Forensic Secur* 2011; 6(3): 873–882.

30. Yang Y, Xiao XX, Cai X, et al. A secure and privacy-preserving technique based on contrast-enhancement reversible data hiding and plaintext encryption for medical images. *IEEE Signal Process Lett* 2020; 27: 256–260.

31. Qin C, Chang CC, Huang YH, et al. An inpainting-assisted reversible steganographic scheme using histogram shifting mechanism. *IEEE Trans Circuits Syst Video Technol* 2013; 23(7): 1109–1118.

32. Wu HZ, Liu G, Yao YW, et al. Watermarking neural networks with watermarked images. *IEEE Trans Circuits Syst Video Technol* 2021; 31: 2591–2601.

33. Qu XC and Kim HJ. Pixel-based pixel value ordering predictor for high-fidelity reversible data hiding. *Signal Process* 2015; 111: 249–260.

34. Chen YL, Wang HX, Wu HZ, et al. Adaptive video data hiding through cost assignment and STCs. *IEEE Trans Dependable Secur Comput* 2021; 18(3): 1320–1335.

35. Chen YL, Wang HX, Hu Y, et al. Intra-frame error concealment scheme using 3D reversible data hiding in mobile cloud environment. *IEEE Access* 2018; 6: 77004–77013.

36. Long M, Zhao Y, Zhang X, et al. A separable reversible data hiding scheme for encrypted images based on Trojino scrambling and adaptive pixel value ordering. *Signal Process* 2020; 176: 107703.

37. Yin XL, Lu W, Zhang JH, et al. Reversible data hiding in halftone images based on minimizing the visual distortion of pixels flipping. *Signal Process* 2020; 173: 107605.

38. Jia YJ, Yin ZX, Zhang XP, et al. Reversible data hiding based on reducing invalid shifting of pixels in histogram shifting. *Signal Process* 2019; 163: 238–246.
39. Li XL, Li J, Li B, et al. High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion. *Signal Process* 2013; 93(1): 198–205.

40. Peng F, Li XL and Yang B. Improved PVO-based reversible data hiding. *Digit Signal Process* 2014; 25: 255–265.

41. Malik A, Wang HX, Wu HZ, et al. Reversible data hiding with multiple data for multiple users in an encrypted image. *Int J Digit Crime Forensics* 2019; 11(1): 46–61.

42. Wu HZ, Wang HX, Zhao H, et al. Multi-layer assignment steganography using graph-theoretic approach. *Multimed Tools Appl* 2015; 74(18): 8171–8196.

43. Malik A, Wang HX, Chen TL, et al. Reversible data hiding in homomorphically encrypted image using interpolation technique. *J Inf Secur Appl* 2019; 48: 102374.

44. Sachnev V, Kim HJ, Nam J, et al. Reversible watermarking algorithm using sorting and prediction. *IEEE Trans Circuits Syst Video Technol* 2009; 19(7): 989–999.

45. Weng SW, Shi YQ, Hong W, et al. Dynamic improved pixel value ordering reversible data hiding. *Inf Sci* 2019; 489: 136–154.

46. Chang CC, Lin CC, Tseng CS, et al. Reversible hiding in DCT-based compressed images. *Inf Sci* 2007; 177(13): 2768–2786.

47. Xuan GR, Li XL and Shi YQ. Minimum entropy and histogram-pair based JPEG image reversible data hiding. *J Int Secur Appl* 2019; 45: 1–9.

48. Peng F, Zhao Y, Zhang X, et al. Reversible data hiding based on RSBEMD coding and adaptive multi-segment left and right histogram shifting. *Signal Process: Image Commun* 2020; 81: 115715.

49. Gao XY, Pan ZB, Gao ED, et al. Reversible data hiding for high dynamic range images using two-dimensional prediction-error histogram of the second time prediction. *Signal Process* 2020; 173: 107579.

50. He JH, Pan XL, Wu HT, et al. Improved block ordering and frequency selection for reversible data hiding in JPEG images. *Signal Process* 2020; 175: 107647.

51. Huang FJ, Qu XC, Kim HJ, et al. Reversible data hiding in JPEG images. *IEEE Trans Circuits Syst Video Technol* 2016; 26(9): 1610–1621.

52. Wedaj FT, Kim S, Kim HJ, et al. Improved reversible data hiding in JPEG images based on new coefficient selection strategy. *J Image Video Process* 2017; 1(60): 1–11.

53. Xiao MY, Li XL, Ma B, et al. Efficient reversible data hiding for JPEG images with multiple histograms modification. *IEEE Trans Circuits Syst Video Technol* 2021; 31(7): 2535–2546.