IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received August 27, 2019, accepted September 11, 2019, date of publication September 19, 2019, date of current version October 2, 2019.

*Digital Object Identifier 10.1109/ACCESS.2019.2942449*

# Novel Scheme of Reversible Watermarking With a Complementary Embedding Strategy

**XIANGGUANG XIONG**
School of Big Data and Computer Science, Guizhou Normal University, Guiyang 550001, China
e-mail: xxg0851@163.com

This work was supported in part by the National Natural Science Foundation of China under Grant 61762021, in part by the Natural Science Foundation of Educational Commission of Guizhou Province under Grant Qian-Xue-Wei-He-Zi ZDXK[2016]8, in part by the Major Research Program of Creative Groups of Educational Commission of Guizhou Province under Grant Qian-Jiao-He KY Zi [2016]027, and in part by the Construction Project of Scientific and Technological Innovation Talents Group of Guizhou Province under Grant [2016] 5629.

**ABSTRACT** In this paper, a new scheme of reversible watermarking is proposed using a complementary embedding strategy in the spatial domain. The proposed scheme consists of two stages: horizontal direction embedding and vertical direction embedding. A complementary embedding strategy is designed to increase the embedding capacity and decrease the distortion of the watermarked image in the vertical direction embedding. Specifically, in the horizontal direction, the proposed scheme embeds one secret data bit by increasing values of pixels in even rows and decreasing values of pixels in odd rows by one. In the vertical direction, it embeds another secret data bit by decreasing values of pixels in even rows and increasing values of pixels in odd rows by one. In addition, a histogram shrinkage technique is adopted to prevent overflow and underflow problems. Experimental results demonstrate that the proposed reversible watermarking scheme outperforms state-of-the-art methods in terms of both embedding capacity and watermarked image quality.

**INDEX TERMS** Complementary embedding strategy, prediction error, reversible watermarking.

## I. INTRODUCTION

Reversible watermarking (also called reversible information hiding or reversible data hiding) aims to hide secret data in digital multimedia and can restore the original signal completely: the embedded secret data can be correctly extracted from the watermarked digital multimedia [1]. In recent years, many reversible watermarking algorithms have been presented. For digital images, three major approaches have been proposed: lossless compression [2]–[3], difference expansion (DE) [4]–[12], and histogram shifting (HS) [13]–[29].

For compression-based reversible watermarking schemes [2]–[3], the original cover image is first compressed by a lossless compression technique (such as arithmetic coding or run-length coding), and then the compressed image is encoded to hide secret data. Generally, the compression ratio is poor. Hence, the embedding capacity of the schemes based on lossless compression is rather low.

Later, more efficient schemes based on DE have been presented [4]–[12]. In 2003, Tian [4] first proposed a novel reversible watermarking algorithm based on DE. The basic idea of this algorithm is to use the difference of pair pixels instead of the pixel itself to hide secret data. Tian's scheme offers a higher embedding capacity while keeping a lower distortion compared with the compression-based reversible watermarking schemes [2]–[3]. However, this scheme needs a location map to distinguish which pixel pair is expanded. The generated location map has to be compressed first, and then the compressed location map is embedded into the cover image. Hence, the pure embedding capacity of this scheme is reduced significantly because of the large size of the compressed location map. Afterwards, the DE technique has been improved by researchers. Alattar [5] generalized Tian's scheme using a vector instead of a pixel pair. In this scheme, $n-1$ secret data bits can be hidden in a vector with $n$ cover pixels. Studies [6]–[9] extended the DE method by reducing the size of the location map. Thodi and Rodriguez [10] improved DE by a prediction error expansion (PEE). Their scheme embedded secret data using each pixel's prediction value instead of the pixel difference. Subsequently, this scheme was improved in various ways: for example, using generalized prediction-error expansion and adaptive embedding [11]–[12] and context embedding [13].

Besides the DE technique, the HS-based reversible watermarking scheme was first proposed by Ni *et al.* [14] in 2006, where the peak point of the cover pixel histogram was used

The associate editor coordinating the review of this manuscript and approving it for publication was Senthil Kumar.

to embed the secret data. In this scheme, each pixel value of the original cover image is changed by one at most. Hence, watermarked images have an exceptionally good visual quality. However, the embedding capacity of this scheme is limited. Several schemes have been proposed to improve performance, such as using prediction error [15]–[17], pixel difference [18]–[19], interpolation technique [20]–[22], interpolation error [23], generalized HS [24]–[25], two-dimensional HS [26], pixel-value ordering [27]–[28], multi-predictor sorting [29], combinatorial strategy [30], human visual system characteristics [31], and pixel location [32].

Unlike DE-based and HS-based schemes, Kieu and Chang [33] proposed a novel high-quality reversible watermarking scheme based on multiple embedding strategies in 2009. It includes two phases: horizontal embedding phase and vertical embedding phase. The basic idea of this scheme is to use the pixel odd-even value instead of the pixel difference to embed secret data; here, horizontal and vertical embedding phases are complementary. For the horizontal embedding phase, if the cover pixel is an odd value and the to-be-embedded bit is 0, then this pixel is modified by subtracting one. If the cover pixel is an even value or the to-be-embedded bit is 1, then this pixel is unchanged. For the vertical embedding phase, if the cover pixel is an even value and the to-be-embedded bit is 1, then this pixel is modified by adding one. If the cover pixel is an odd value or the to-be-embedded bit is 0, then this pixel is untouched. Later on, this scheme was improved by Chang and Kieu [34] and Jia *et al.* [35]. All three schemes have a much larger embedding capacity while keeping a low distortion (compared with most previous schemes). However, their pure embedding capacity is rather limited because the size of the location map is very large (even if the location map has been compressed by a lossless compression technique). Hence, this paper presents a new high-fidelity reversible watermarking scheme based on a complementary embedding strategy. In the proposed scheme, the data embedding procedure consists of horizontal direction embedding and vertical direction embedding. Specifically, for the horizontal direction embedding, the prediction value of each pixel is first obtained by computing the average value of its two neighboring pixels, and then the to-be-embedded data bit is embedded into the cover pixel by increasing or decreasing the pixel value by one when the prediction error of the current pixel is 0. Similarly, for the vertical direction embedding, the secret data bit is embedded by decreasing or increasing the pixel value by one. The horizontal and vertical direction embeddings are complementary. This property enables the proposed scheme to obtain a particularly good visual quality of the watermarked image. Moreover, the histogram shrinkage technique is used to prevent overflow and underflow problems. Experimental results demonstrate that the proposed scheme outperforms similar reversible watermarking schemes. The main contributions of this paper are summarized as the following three aspects:

1. We point out that the Kieu et al.'s scheme [33], Chang et al.' scheme [34] and Jia et al.'s scheme [35] cannot embed the secret data into a cover image if the compressed location map needs to be embedded. The reason is that for natural image the number of "1" almost equals to the number of "0" in location map, and lossless compression cannot significantly reduce the size of the original location map.
2. We propose a complementary embedding strategy to embed watermarking signal. The horizontal and vertical direction embedding are complementary. This property enables the proposed scheme to obtain a better visual quality of the watermarked image.
3. We verify its effectiveness by experiments and comparisons with the state of the art.

The rest of this paper is organized as follows. The reversible watermarking scheme based on a complementary embedding method is briefly reviewed in Section 2. The details of the proposed scheme are introduced in Section 3. Experimental results are shown in Section 4. Finally, conclusions are given in Section 5.

## II. RELATED WORK

In 2009, Kieu *et al.* [33] proposed a novel high-quality reversible watermarking scheme using multiple embedding strategies. Their scheme can horizontally and vertically embed one secret data bit into one pixel pair of a cover image. Experimental results demonstrate that this scheme achieves a good image quality and ideal embedding capacity. Later, Kieu et al.'s work has been improved by Chang and Kieu [34] and Jia *et al.* [35]. Jia et al.'s scheme also horizontally and vertically embeds one secret bit into one pixel pair of a cover image. Unlike Kieu et al.'s scheme and Jia et al.'s scheme, Chang et al.'s scheme horizontally and vertically embeds one secret bit into one pixel of a cover image. However, in essence, the embedding procedure used by Jia et al. and Chang et al. is the same as in Kieu et al. Therefore, this section mainly reviews the Kieu et al.'s method. More details about the schemes by Jia et al. and Chang et al. can be found in their papers.

The embedding steps of Kieu et al.'s scheme are described as follows.

*Step 1:* Horizontally scan the original cover image $I$ by a raster scan to group two neighboring pixels $I(i,j)$ and $I(i,j+1)$ into a pixel pair, where $1 \leq i \leq M$, $1 \leq j \leq N-1$. If $I(i,j+1)$ is an odd value, then this pixel pair is defined as a horizontal embeddable pixel pair. The first pixel of each pixel pair always remains unchanged.

*Step 2:* Create a location map $LM_1$ sized $M \times (N/2)$ to record the positions of the horizontal embeddable pixel pairs and identify the value of $LE_1$, where $LE_1$ is the number of the horizontal embeddable pixel pairs. Initially, all the elements of $LM_1$ are set to 0. If the pixel pair is the horizontal embeddable pixel pair, then the corresponding element of $LM_1$ is set to be 1. Note that location map $LM_1$ is a binary sequence.

**TABLE 1.** Results of Kieu et al.'s scheme [33], Chang et al.'s scheme [34], and Jia et al.'s scheme [35].

| Schemes | Images | Horizontal direction embedding | | | Vertical direction embedding | | |
|---|---|---|---|---|---|---|---|
| | | $L_e$ | $L_m$ | $L_c$ | $L_e$ | $L_m$ | $L_c$ |
| [33] | Aerial | 65403 | 131072 | 131120 | 82118 | 131072 | 123928 |
| | Airplane | 65374 | 131072 | 131104 | 82115 | 131072 | 123928 |
| | Elaine | 66843 | 131072 | 131080 | 81045 | 131072 | 124560 |
| | Lena | 65732 | 131072 | 131112 | 81648 | 131072 | 124532 |
| [34] | Aerial | 130518 | 262144 | 262184 | 196746 | 262144 | 212496 |
| | Airplane | 130490 | 262144 | 262168 | 197119 | 262144 | 211904 |
| | Elaine | 133498 | 262144 | 262128 | 195329 | 262144 | 214720 |
| | Lena | 131522 | 262144 | 262176 | 196507 | 262144 | 212880 |
| [35] | Aerial | 65669 | 131072 | 131120 | 98328 | 131072 | 106344 |
| | Airplane | 65698 | 131072 | 131104 | 98200 | 131072 | 106544 |
| | Elaine | 64229 | 131072 | 131080 | 99008 | 131072 | 105248 |
| | Lena | 65340 | 131072 | 131112 | 98494 | 131072 | 106080 |

*Step 3:* Losslessly compress $LM_1$ using arithmetic coding to obtain a compressed location map $C_1$ whose length is $LC_1$.

*Step 4:* Let $B_1$ be the first $LB_1$ bits of $B$, where $LB_1 = LE_1 - LC_1$.

*Step 5:* For the first $LC_1$ pixels in $I$, when each pixel has been embedded using (1), its least significant bit (LSB) is taken as auxiliary information bit of $A_1$ and needs to be appended to the end of $B_1$. This step will be repeated until all secret bits in $B_1$ are completely embedded and image $X$ can be obtained.

$$X(i,j+1)=\begin{cases} I(i,j+1), & \text{if } b=1 \text{ and } LM_1=1 \\ I(i,j+1)-1, & \text{if } b=0 \text{ and } LM_1=1 \end{cases} \quad (1)$$

*Step 6:* Replace LSB of the first $LC_1$ image pixels in $X$ by $C_1$ to obtain image $H$.

*Step 7:* As in Steps 1–3, obtain $LE_2$ and compressed location map $C_2$ whose length is $LC_2$, where $LE_2$ is the number of the vertical embeddable pixel pairs.

*Step 8:* Let $B_2$ be the remaining $LC_2$ bits of $B$, where $LB_2 = LE_2 - LC_2$.

*Step 9:* As in Step 5, for the first $LC_2$ pixels in $H$, when each pixel has been embedded using (2), its LSB is taken as auxiliary information bit of $A_2$ and needs to be appended to the end of $B_2$. This step will be repeated until all secret bits in $B_2$ are completely embedded and image $Y$ can be obtained.

$$Y(i,j+1)=\begin{cases} H(i,j+1), & \text{if } b=0 \text{ and } LM_2=1 \\ H(i,j+1)+1, & \text{if } b=1 \text{ and } LM_2=1 \end{cases} \quad (2)$$

*Step 10:* Replace LSB of the first $LC_2$ pixels in $Y$ by $C_2$ to obtain the final watermarked image $V$.

This data embedding procedure has a big problem: although it achieves a high visual quality and hiding capacity in the experiments, this scheme almost cannot embed the secret data into a cover image if the compressed location map needs to be embedded. In fact, for natural images, the number of pixel values with odd values and even values are roughly the same. That is, the number of "1" almost equals to the number of "0" in location maps $LM_1$ or $LM_2$. Consequently, lossless compression cannot significantly reduce the size of the original location map. For schemes by Kieu and Chang [33], Chang and Kieu [34], and Jia *et al.* [35], location maps $LM_1$ and $LM_2$ first need to be compressed using arithmetic coding [36] and then are embedded into the cover image. The performance comparison results of the schemes by Kieu et al., Jia et al., and Chang et al. for four standard test images (see Fig. 7) are shown in Table 1. In Table 1, $L_e$, $L_m$, and $L_c$ represent the number of embeddable secret data bits, length of the original location map, and length of the compressed location map, respectively.

For schemes by Kieu et al., Jia et al., and Chang et al., Table 1 shows that, if location maps $LM_1$ and $LM_2$ are directly sent to the expected receivers, these three schemes can embed ideal secret data. However, if compressed location maps $LM_1$ and $LM_2$ need to be embedded into the cover image with the to-be-embedded secret data, these three schemes cannot embed any secret data because $L_c$ is much larger than $L_e$ for all test images.

Taking "Aerial" image as an example, consider the horizontal direction embedding in the Kieu et al.'s scheme. Here, the number of embeddable secret data bits, length of the original location map, and length of the compressed location map are 65403, 131072, and 131120, respectively. It can be clearly seen that the number of embeddable secret data bits is much less than the length of the compressed location map. That is, in the horizontal direction, their scheme cannot really embed any secret data into "Aerial" image. Similarly, in the vertical direction, their scheme also cannot really embed any secret data.

## III. PROPOSED SCHEME

This section describes image preprocessing, data embedding, and data extraction in the proposed scheme.

### A. PREVENT OVERFLOW AND UNDERFLOW PROBLEMS

For the proposed scheme, the pixel values of the original cover image may be increased or decreased by one at most. Hence, for an eight-bit gray-scale cover image, if the pixel value is equal to 0 or 255, then the pixel value of the watermarked image may be outside of [0, 255]. That is, these pixels may cause an overflow or underflow problem after the secret data bit has been embedded. To solve this problem, the histogram shrinkage technique is used. The original cover image is processed as follows.

*Step 1:* The original cover image $I$ is scanned from left to right and from top to bottom. If the pixel value is equal to 0 or 1, then it is changed to 1 and generates one bit, 1 or 0. That is, if the pixel value is 0, then 1 is added to $L$. Otherwise, if the pixel value is 1, then 0 is added to $L$.

*Step 2:* As in Step 1, if a pixel value is equal to 254 or 255, then it is changed to 254 and generates one bit, 0 or 1. That is, if the pixel value is 254, then 0 is added to $R$. Otherwise, if the value is 255, then 1 is added to $R$.

Note that $L$ and $R$ are one-dimensional binary sequences. For most test images, the length of $L$ and $R$ is exceedingly small. Hence, $L$ and $R$ do not require a lossless compression.

Similarly, the original image is restored as follows.

*Step 1:* Obtain $L$ and $R$ from the extraction bit stream.

*Step 2:* Recovery image $I$ is scanned from left to right and from top to bottom. If the pixel value of the recovery image is 1 and the extracted auxiliary data bit is 0, then it is restored to 1. If the pixel value is 1 and the extracted auxiliary data bit is 1, then it is modified to 0.

*Step 3:* As in Step 2, if the pixel value of the recovery image is 254 and the extracted auxiliary data bit is 0, then it is recovered to 254. If the pixel value is 254 and the extracted auxiliary data bit is 1, then it is changed to 255.

For example, in the data embedding procedure, if a pixel $I(i,j)$ equals to 0, then this pixel is changed to 1 and is encoded by "1" in $L$. Assume that the current prediction error $d(i,j) = 0$, which means that this pixel can embed a secret data bit. If the to-be-embedded secret data bit is above auxiliary information bit $b = 1$, the watermarked pixel $H(i,j) = I(i,j) + 1 = 2$ according to the data embedding method based on histogram shifting. In data extraction and image recovery procedure, the embedded auxiliary information bit $b = 1$ can be extracted correctly, and the recovery pixel is $I(i,j) = H(i,j) - 1 = 1$. Finally, original cover pixel can be restored by $I(i,j) = I(i,j) - 1 = 0$.

### B. DATA EMBEDDING PROCEDURE

Assume that $I$ and $V$ with size $M \times N$ represent an eight-bit original gray-scale cover image and its watermarked image, respectively. $I(i,j)$ and $\overline{I}(i,j)$ represent cover pixel value and its predicted value, respectively. The to-be-embedded secret data bits $B = \{b | b \in \{0,1\}\}$ are a binary sequence.

The data embedding procedure of the proposed scheme is composed of two stages: the horizontal direction embedding and vertical direction embedding. When a secret data bit has to be embedded, the cover image is processed from left to right and from top to bottom, as shown in Fig. 1.

Fig. 2 presents a relationship between cover pixel $I(i,j)$ and its neighboring pixels. In horizontal direction embedding, for each pixel $I(i,j)$, its predicted value $\overline{I}(i,j)$ can be calculated as follows.

$$\overline{I}(i,j) = \begin{cases} \lfloor (H(i,j-1) + I(i,j+1))/2 \rfloor, & \text{if } 1 \le i \le M \text{ and } 1 < j < N \\ I(i,j+1), & \text{if } 1 \le i \le M \text{ and } j = 1 \\ H(i,j-1), & \text{if } 1 \le i \le M \text{ and } j = N \end{cases} \quad (3)$$

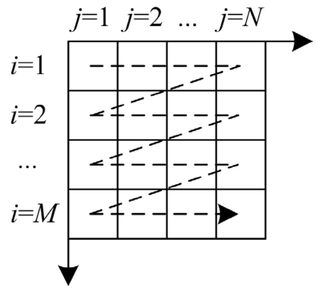



**FIGURE 1.** Order of the data embedding procedure.

| | | |
|---|---|---|
| $I(i\text{-}1, j\text{-}1)$ | $I(i\text{-}1, j)$ | $I(i\text{-}1, j\text{+}1)$ |
| $I(i, j\text{-}1)$ | $I(i, j)$ | $I(i, j\text{+}1)$ |
| $I(i\text{+}1, j\text{-}1)$ | $I(i\text{+}1, j)$ | $I(i\text{+}1, j\text{+}1)$ |

**FIGURE 2.** Cover pixel $I(i,j)$ and its neighboring pixels.

where $\lfloor . \rfloor$ is the floor function and $H(i,j)$ is an embedded pixel.

After predicted value $\overline{I}(i,j)$ is obtained, prediction error $d_1(i,j)$ between current pixel $I(i,j)$ and its predicted value $\overline{I}(i,j)$ is computed using (4).

$$d_1(i,j) = I(i,j) - \overline{I}(i,j) \quad (4)$$

In the horizontal embedding, the pixel is shifted by (5) and a secret data bit $b \in \{0,1\}$ is embedded by (6), respectively.

$$H(i,j) = \begin{cases} I(i,j) + 1, & \text{if } d_1(i,j) > 0 \text{ and } \bmod(i,2) = 1 \\ I(i,j) - 1, & \text{if } d_1(i,j) < 0 \text{ and } \bmod(i,2) = 0 \end{cases} \quad (5)$$

$$H(i,j) = \begin{cases} I(i,j) + b, & \text{if } d_1(i,j) = 0 \text{ and } \bmod(i,2) = 1 \\ I(i,j) - b, & \text{if } d_1(i,j) = 0 \text{ and } \bmod(i,2) = 0 \end{cases} \quad (6)$$

In the vertical embedding, for each pixel $H(i,j)$, its predicted value $\overline{H}(i,j)$ can be calculated as follows.

$$\overline{H}(i,j) = \begin{cases} \lfloor (V(i-1,j) + H(i+1,j))/2 \rfloor, & \text{if } 1 \le j \le N \text{ and } 1 < i < M \\ H(i+1,j), & \text{if } 1 \le j \le N \text{ and } i = 1 \\ V(i-1,j), & \text{if } 1 \le j \le N \text{ and } i = M \end{cases} \quad (7)$$

where $V(i,j)$ is a new embedded pixel.

After predicted value $\overline{H}(i,j)$ is obtained, prediction error $d_2(i,j)$ between current $H(i,j)$ and its predicted value $\overline{H}(i,j)$ is computed using (8).

$$d_2(i,j) = H(i,j) - \overline{H}(i,j) \quad (8)$$

In the vertical embedding, shift the pixel by (9) and embed another secret data bit $b \in \{0,1\}$ by (10), respectively.

$$V(i,j) = \begin{cases} H(i,j) - 1, & \text{if } d_2(i,j) < 0 \text{ and } \bmod(i,2) = 1 \\ H(i,j) + 1, & \text{if } d_2(i,j) > 0 \text{ and } \bmod(i,2) = 0 \end{cases} \quad (9)$$

$$V(i,j) = \begin{cases} H(i,j) - b, & \text{if } d_2(i,j) = 0 \text{ and } \bmod (i,2) = 1 \\ H(i,j) + b, & \text{if } d_2(i,j) = 0 \text{ and } \bmod (i,2) = 0 \end{cases} \tag{10}$$

The flowchart of the data embedding procedure of the proposed scheme is shown in Fig. 3. Next, the details are described step by step.

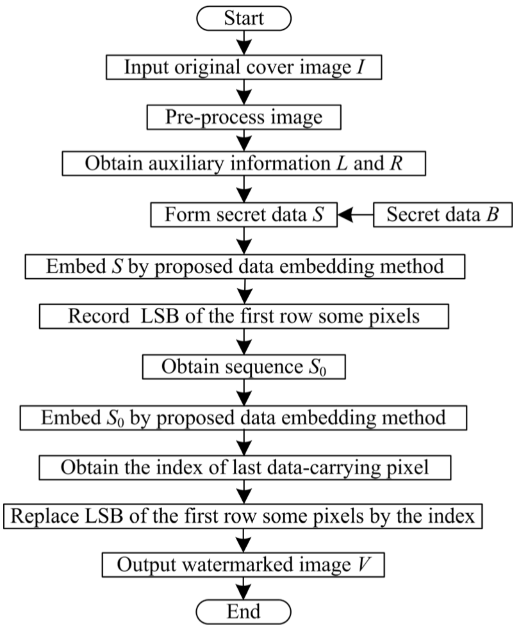



**FIGURE 3. Flowchart of data embedding procedure.**

*Step 1:* Preprocess the original cover image $I$ by a raster scan to obtain the auxiliary information $A_1 = L||R$, where the notation "||" represents the concatenation operation. The size of $L$ and $R$ is $L_1$ and $L_2$, respectively. If the original cover image has no pixels with values 0 or 255, then it does not require image preprocessing and the size of auxiliary information $A_1$ is 0. Hence, we have to use one bit to record whether auxiliary information $A_1$ exists. That is, the length of auxiliary information $A_1$ is at least 1. After image preprocessing, embed the secret data bits into the processed image without overflow and underflow problems.

*Step 2:* Final to-be-embedded secret data bits $S$ are created by $S = F||Len_1||Len_2\,||L||R||B$, where $F$ (1 bit) is a flag that is used to record whether auxiliary information exists; $Len_1$ ($\lceil \log_2(L_1 + 1) \rceil$ bits) and $Len_2$ ($\lceil \log_2(L_2 + 1) \rceil$ bits) represent the size of auxiliary information $L$ and $R$, and $B$ denotes real embeddable data.

*Step 3:* Embed the secret data bits into the cover image as follows.

- In the horizontal direction embedding, calculate prediction error $d_1(i,j)$ by (4). The pixel is modified, and the secret data bit is embedded by (5) and (6), respectively.
- In the vertical direction embedding, calculate prediction error $d_2(i,j)$ by (8). The pixel is modified, and the secret data bit is embedded by (9) and (10), respectively.

*Step 4:* Repeat Step 3 until all secret data bits in $S$ have been embedded.

*Step 5:* Take the LSBs of the first $\lceil \log_2(M + 1) \rceil + \lceil \log_2(N+1) \rceil$ watermarked image pixels in the first row to obtain a binary sequence $S_0$, where $\lceil \cdot \rceil$ is a ceiling function.

*Step 6:* As in Step 3, embed binary sequence $S_0$ defined in Step 5 into the remaining pixels of the cover image, and denote $i_{\text{end}}$ and $j_{\text{end}}$ as the index of the last data-carrying pixel.

*Step 7:* Replace the LSB of the first image pixels by auxiliary information $A_2$ to obtain the final watermarked image, where $A_2$ is the index of end position $i_{\text{end}}$ ($\lceil \log_2(M + 1) \rceil$ bits) and $j_{\text{end}}$ ($\lceil \log_2(N + 1) \rceil$ bits).

Here, the procedures of horizontal and vertical embedding are inverse. That is, in the horizontal direction embedding, the proposed scheme increases or decreases pixel values by one. In the vertical direction embedding, it decreases or increases pixel values by one to balance out the embedded secret bits that were produced in the horizontal embedding. Thanks to this inverse property, the proposed scheme has a ideal embedding capacity and a good image quality.

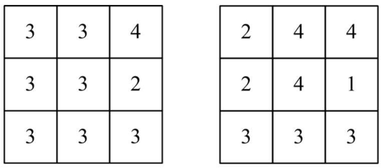
Fig. 4 shows a simple example of the data embedding procedure. Note that auxiliary information is not embedded into the cover image. Assume that the to-be-embedded secret data $B$ is 01101010110. Fig. 4(a) is an original cover image sized $M \times N$, where $M = N = 3$. The pixel $I(1, 1) = 3$ is first processed in the horizontal direction according to the data embedding rule, and its predicted value is calculated as $\bar{I}(1, 1) = H(1, 2) = 3$. Prediction error $d_1(1, 1)$ is obtained by $d_1(1, 1) = I(1, 1) - \bar{I}(1, 1) = 0$, which means that this pixel can embed a secret data bit. Because the first bit of $B$ is 0 and mod $(1, 2) = 1$, this pixel is modified by $H(1, 1) = I(1, 1) + 0 = 3$. In vertical direction, its predicted value is calculated as $\overline{H}(1, 1) = I(2, 1) = 3$. Prediction error $d_2(1, 1)$ is 0. Because the second bit of $B$ is 1, the final watermarked pixel is obtained by $V(1, 1) = H(1, 1) - 1 = 2$. For the next pixel $I(1, 2) = 3$, in the horizontal direction, its predicted value is computed as $\bar{I}(1, 2) = \lfloor (V(1, 1) + I(1, 3))/2 \rfloor = 3$ and the prediction error $d_1(1, 2)$ is 0. Hence, this pixel is modified by $H(1, 2) = I(1, 2) + 1 = 4$ because the to-be-embedded secret data bit is 1 and mod $(1, 2) = 1$. In the vertical direction, its predicted value is calculated as $\overline{H}(1, 2) = I(2, 2) = 3$, and prediction error $d_2(1, 2) = 1$, which means that this pixel cannot embed a secret data bit and remains unchanged. Hence, final watermarked pixel $V(1, 2)$ is 4. A similar operation can be performed for the remaining pixels of this image. When all secret data bits in $B$ have been embedded completely, watermarked image $V$ is obtained, as shown in Fig. 4(b). Table 2 presents the detailed data embedding procedure.

### C. DATA EXTRACTION PROCEDURE

The data extraction procedure is actually the reverse process of the data embedding procedure. The data extraction procedure of the proposed scheme also includes two stages: the vertical direction extraction and horizontal direction extraction.

**TABLE 2.** Results of the data embedding procedure.

| Horizontal direction embedding | | | | | Vertical direction embedding | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Location | $I(i,j)$ | $d_1$ | $b$ | $H(i,j)$ | Location | $H(i,j)$ | $d_2$ | $b$ | $V(i,j)$ |
| (1, 1) | 3 | 0 | 0 | 3 | (1, 1) | 3 | 0 | 1 | 2 |
| (1, 2) | 3 | 0 | 1 | 4 | (1, 2) | 4 | 1 | - | 4 |
| (1, 3) | 4 | 0 | 0 | 4 | (1, 3) | 4 | 2 | - | 4 |
| (2, 1) | 3 | 0 | 1 | 2 | (2, 1) | 2 | 0 | 0 | 2 |
| (2, 2) | 3 | 1 | - | 3 | (2, 2) | 3 | 0 | 1 | 4 |
| (2, 3) | 2 | -2 | - | 1 | (2, 3) | 1 | -2 | - | 1 |
| (3, 1) | 3 | 0 | 0 | 3 | (3, 1) | 3 | 1 | - | 3 |
| (3, 2) | 3 | 0 | 1 | 4 | (3, 2) | 4 | 0 | 1 | 3 |
| (3, 3) | 3 | 0 | 0 | 3 | (3, 3) | 3 | 2 | - | 3 |




(a) Cover image $I$ (b) Watermarked image $V$

**FIGURE 4.** Simple example for the data embedding procedure.

When a secret data bit is to be extracted, the watermarked image is processed from right to left and from bottom to top, as shown in Fig. 5.

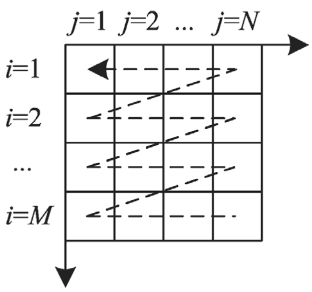



**FIGURE 5.** Order of the data extraction procedure.

For the vertical direction extraction, the predicted value $\overline{V}(i,j)$ can be calculated as follows.

$$\overline{V}(i,j) = \begin{cases} \lfloor (V(i-1,j) + H(i+1,j))/2 \rfloor, & \\ \quad \text{if } 1 \le j \le N \text{ and } 1 < i < M \\ H(i+1,j), & \text{if } 1 \le j \le N \text{ and } i = 1 \\ V(i-1,j), & \text{if } 1 \le j \le N \text{ and } i = M \end{cases} \tag{11}$$

where $H(i,j)$ is a restored pixel.

After predicted value $\overline{V}(i,j)$ is obtained, prediction error $d_2(i,j)$ between pixel $V(i,j)$ and predicted pixel $\overline{V}(i,j)$ is computed by (12) as below.

$$d_2(i,j) = V(i,j) - \overline{V}(i,j) \tag{12}$$

In the vertical direction extraction, restore the pixel value by (13) and extract a secret data bit by (14), respectively.

$$H(i,j) = \begin{cases} V(i,j)-1, & \text{if } \bmod(i,2)=0 \text{ and } d_2(i,j) \ge 1 \\ V(i,j)+1, & \text{if } \bmod(i,2)=1 \text{ and } d_2(i,j) \le -1 \end{cases} \tag{13}$$

$$b = \begin{cases} 0, & \text{if } d_2(i,j) = 0 \\ 1, & \text{if } (\bmod(i,2) = 0 \text{ and } d_2(i,j) = 1) \\ & \text{or } (\bmod(i,2) = 1 \text{ and } d_2(i,j) = -1) \end{cases} \tag{14}$$

Similarly, for the horizontal direction extraction, predicted value $\overline{H}(i,j)$ can be calculated as follows.

$$\overline{H}(i,j) = \begin{cases} \lfloor (H(i,j-1) + I(i,j+1))/2 \rfloor, & \\ \quad \text{if } 1 \le i \le M \text{ and } 1 < j < N \\ I(i,j+1), & \text{if } 1 \le i \le M \text{ and } j = 1 \\ H(i,j-1), & \text{if } 1 \le i \le M \text{ and } j = N \end{cases} \tag{15}$$

where $I(i,j)$ is a restored pixel.

After predicted value $\overline{H}(i,j)$ is obtained, prediction error $d_1(i,j)$ between cover pixel value $H(i,j)$ and predicted value $\overline{H}(i,j)$ is computed by (16).

$$d_1(i,j) = H(i,j) - \overline{H}(i,j) \tag{16}$$

In horizontal direction extraction, restore the pixel value by (17) and extract a secret data bit by (18), respectively.

$$I(i,j) = \begin{cases} H(i,j)-1, & \text{if } \bmod(i,2) = 1 \text{ and } d_1(i,j) \ge 1 \\ H(i,j)+1, & \text{if } \bmod(i,2) = 0 \text{ and } d_1(i,j) \le -1 \end{cases} \tag{17}$$

$$b = \begin{cases} 0, & \text{if } d_1(i,j) = 0 \\ 1, & \text{if } (\bmod(i,2) = 1 \text{ and } d_1(i,j) = 1) \\ & \text{or } (\bmod(i,2) = 0 \text{ and } d_1(i,j) = -1) \end{cases} \tag{18}$$

The flowchart of the data extraction procedure of the proposed scheme is shown in Fig 6. The extraction steps are described as below.

*Input:* Watermarked image $V$.

*Output:* Recovered image $I$ and embedded secret data bits $B$.

*Step 1:* Read LSB of the first $\lceil \log_2(M+1) \rceil + \lceil \log_2(N+1) \rceil$ pixels of watermarked image $V$ to obtain the auxiliary information $A_2$ including the values of $i_{\text{end}}$ and $j_{\text{end}}$.

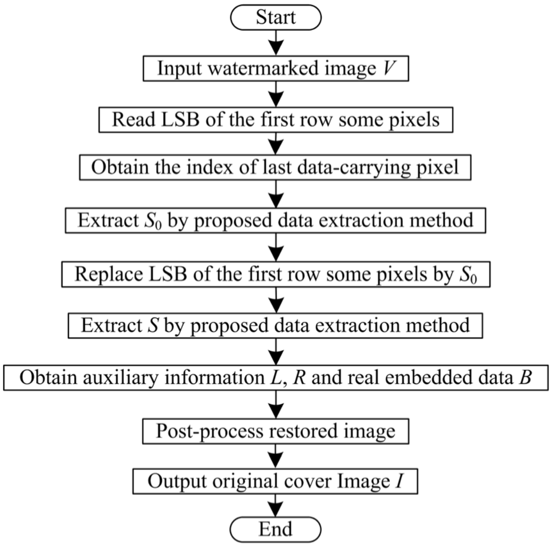



**FIGURE 6.** Flowchart of the data extraction procedure.

*Step 2:* Based on indexes $i_{end}$ and $j_{end}$, extract binary sequence $S_0$ defined in Step 5 of the data embedding procedure. Meanwhile, do the original cover image restoration. For each pixel $V(i, j)$, the data extraction and image restoration procedures are defined as follows.

- In the vertical direction extraction, calculate its prediction error $d_2(i, j)$ by (12). Next, extract the embedded secret data bits and recover the image using (13) and (14), respectively.
- In the horizontal direction extraction, calculate its prediction error $d_1(i, j)$ by (16). Next, extract the embedded secret data bits and recover the image using (17) and (18), respectively.

*Step 3:* Repeat Step 2 until sequence $S_0$ has been extracted.

*Step 4:* Replace the LSB of the first $\lceil (\log_2(M + 1)) \rceil + \lceil (\log_2(N + 1)) \rceil$ image pixels by $S_0$ extracted in Step 2.

*Step 5:* As in Step 2, extract the embedded secret data bits from the remaining pixels. Meanwhile, restore those pixels. Finally, embedded secret data $S$ are extracted correctly and a secondary image can be obtained.

*Step 6:* Extracted secret data $S$ are separated into auxiliary information $A_1$ and secret data $B$. If the first bit of $A_1$ is 1, the original cover image can be recovered exactly from the secondary image without any distortion according to auxiliary information $A_1$. If the first bit is 0, it means that the overflow and underflow problems did not occur in the data embedding, and the secondary image is the final recovered original cover image.

The following example illustrates the data extraction and image restoration procedure. Suppose that the input watermarked image is shown in Fig. 4(b). Note that the extraction procedure should be processed first in vertical direction, and then in horizontal direction. For the vertical extraction, pixel $V(3, 3) = 3$ is first processed to predict its predicted value, which is computed as $\overline{V}(3, 3) = V(2, 3) = 1$. Prediction error $d_2(3, 3)$ is obtained by $d_2(3, 3) = V(3, 3) - \overline{V}(3, 3) = 2$, which means this pixel does not contain the secret data bit and remains unchanged because $d_2(3, 3)$ is larger than 1 and mod(3, 2) = 1. For horizontal direction, its predicted value is obtained by $\overline{H}(3, 3) = V(3, 2) = 3$. Prediction error $d_1(3, 3) = H(3, 3) - \overline{H}(3, 3) = 0$, which means this pixel contains the secret data bit. According to the data extraction and image restoration methods, the embedded secret bit 0 can be extracted and this pixel is unchanged. For the next pixel, $V(3, 2) = 3$, in vertical direction, its predicted value is obtained as $\overline{V}(3, 2) = V(2, 2) = 4$ and the prediction error $d_2(3, 2)$ is $-1$, which means this pixel contains the secret data bit. According to the proposed method, the bit 1 is extracted, and this pixel will be modified to 4. In horizontal direction, its predicted value is obtained as $\overline{H}(3, 2) = \lfloor (V(3, 1) + I(3, 3))/2 \rfloor = 3$ and the prediction error $d_2(3, 2)$ is 1. The bit 1 is extracted and this pixel will be changed to 3. Similar method can be used for other pixels. Finally, the embedded secret data bits can be extracted correctly and the original cover image $I$ can be restored perfectly, as shown in Fig. 4(a). Table 3 presents the data extraction and image restoration procedure.

## IV. EXPERIMENTAL RESULTS

In this section, in order to validate the performance of the proposed scheme in terms of embedding capacity and visual quality of the embedded image, the Matlab language is used to perform experiments with the proposed scheme and six baseline schemes. The experiments are performed in Windows 7, SP1, using Matlab 2010a software running on the AMD Sempron X2 180 Processor, 2.4 GHz CPU, and 2.0 GB RAM. Four 8-bit standard test gray-scale images sized 512 × 512 (namely, "Aerial", "Airplane", "Elaine" and "Lena") are selected as original cover images, as shown in Fig. 7. All images are obtained from USC SIPI image dataset [37]. The to-be-embedded secret data bits are generated using the random function.

In the experiments, the visual quality of the watermarked image is measured using the peak signal-to-noise ratio (PSNR), which is defined as follows.

$$PSNR = 10 \times \log_{10}(\frac{255^2}{MSE}) \tag{19}$$

where the mean square error (MSE) is defined as below.

$$MSE = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} (I(i, j) - V(i, j))^2 \tag{20}$$

where $I(i, j)$ and $V(i, j)$ are the pixel values of the original cover image and watermarked image. $M$ and $N$ are the width and height of original cover image, respectively.

In addition, the embedding capacity is estimated using the number of secret data bits that can really be embedded into a cover image. That is, the embedding capacity is a pure payload. For the proposed scheme, to ensure the embedded pixel value within the allowable range [0, 255], the auxiliary

**TABLE 3.** An example of the data extraction and image restoration procedure.

| Vertical direction extraction | | | | | Horizontal direction extraction | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Location | $V(i,j)$ | \$d_2$ | $b$ | $H(i,j)$ | Location | $H(i,j)$ | $d_1$ | $b$ | $I(i,j)$ |
| (3, 3) | 3 | 2 | - | 3 | (3, 3) | 3 | 0 | 0 | 3 |
| (3, 2) | 3 | -1 | 1 | 4 | (3, 2) | 4 | 1 | 1 | 3 |
| (3, 1) | 3 | 1 | - | 3 | (3, 1) | 3 | 0 | 0 | 3 |
| (2, 3) | 1 | -2 | - | 1 | (2, 3) | 1 | -3 | - | 2 |
| (2, 2) | 4 | 1 | 1 | 3 | (2, 2) | 3 | 1 | - | 3 |
| (2, 1) | 2 | 0 | 0 | 2 | (2, 1) | 2 | -1 | 1 | 3 |
| (1, 3) | 4 | 2 | - | 4 | (1, 3) | 4 | 0 | 0 | 4 |
| (1, 2) | 4 | 1 | - | 4 | (1, 2) | 4 | 1 | 1 | 3 |
| (1, 1) | 2 | -1 | 1 | 3 | (1, 1) | 3 | 0 | 0 | 3 |

(a) Aerial (b) Airplane (c) Elaine (d) Lena

**FIGURE 7.** Four standard gray-scale test images.

**TABLE 4.** Performance results of the proposed scheme and other schemes.

| Scheme | Metric | Aerial | Airplane | Elaine | Lena | **Average** |
|---|---|---|---|---|---|---|
| Propose scheme | Payload | 58298 | 82738 | 33774 | 58038 | **58212** |
| | PSNR | 49.705 | 49.7011 | 49.2102 | 49.5042 | **49.5301** |
| Non-complementary | Payload | 55948 | 79710 | 32658 | 56372 | **56172** |
| | PSNR | 45.7419 | 45.716 | 45.5758 | 45.5744 | **45.6520** |
| [9] | Payload | 31797 | 50033 | 18379 | 33566 | **33444** |
| | PSNR | 48.4249 | 48.6026 | 48.3099 | 48.4544 | **48.4480** |
| [15] | Payload | 44935 | 66158 | 26291 | 46797 | **46045** |
| | PSNR | 48.5332 | 48.7332 | 48.3719 | 48.5546 | **48.5482** |
| [16] | Payload | 26974 | 47599 | 16502 | 29469 | **30136** |
| | PSNR | 51.1867 | 51.2398 | 51.2416 | 51.3257 | **51.2485** |
| [17] | Payload | 62909 | 84231 | 36196 | 55000 | **59584** |
| | PSNR | 48.6982 | 48.9207 | 48.4688 | 48.6395 | **48.6818** |
| [18] | Payload | 23219 | 39331 | 16039 | 22045 | **25159** |
| | PSNR | 48.3404 | 48.487 | 48.2832 | 48.3345 | **48.3613** |
| [19] | Payload | 43394 | 71140 | 30066 | 42844 | **46861** |
| | PSNR | 48.5109 | 48.7804 | 48.4057 | 48.5195 | **48.5541** |

information is needed. Thus, in the following experiments, the auxiliary information bits are excluded from computing embedding capacity.

For a comparison with similar reversible watermarking schemes, the maximum embeddable secret data bits of each scheme are embedded into each test image using schemes by Tai *et al.* [18], Hong *et al.* [15], Lee and Chen [9], Fallahpour [16], Yang and Tsai [17], Li *et al.* [19], and non-complementary scheme. For Tai et al.'s scheme [18], the difference of two adjacent pixels is used to embed the secret data. For Hong et al.'s scheme [15], the difference between the original pixel and its predicted value obtained by median edge detector is used to embed the secret data. For Lee et al.'s scheme [9], the difference between the original pixel and the average of its neighboring pixels is used to embed the secret data. For Fallahpour's scheme [16], the difference between the original pixel and its predicted value calculated by gradient adjusted predictor is used to embed the secret data. For Li et al.'s scheme [19], the difference of two adjacent pixels is computed first and then two peak points are selected to embed the secret data. For Yang et al.'s scheme [17], the interleaving predictor is used to predict the pixel value. For non-complementary scheme, the embedding procedure is similar to the proposed scheme, but the embedding rule in vertical and horizontal embedding direction is the same. The performance comparison results are shown in Table 4.

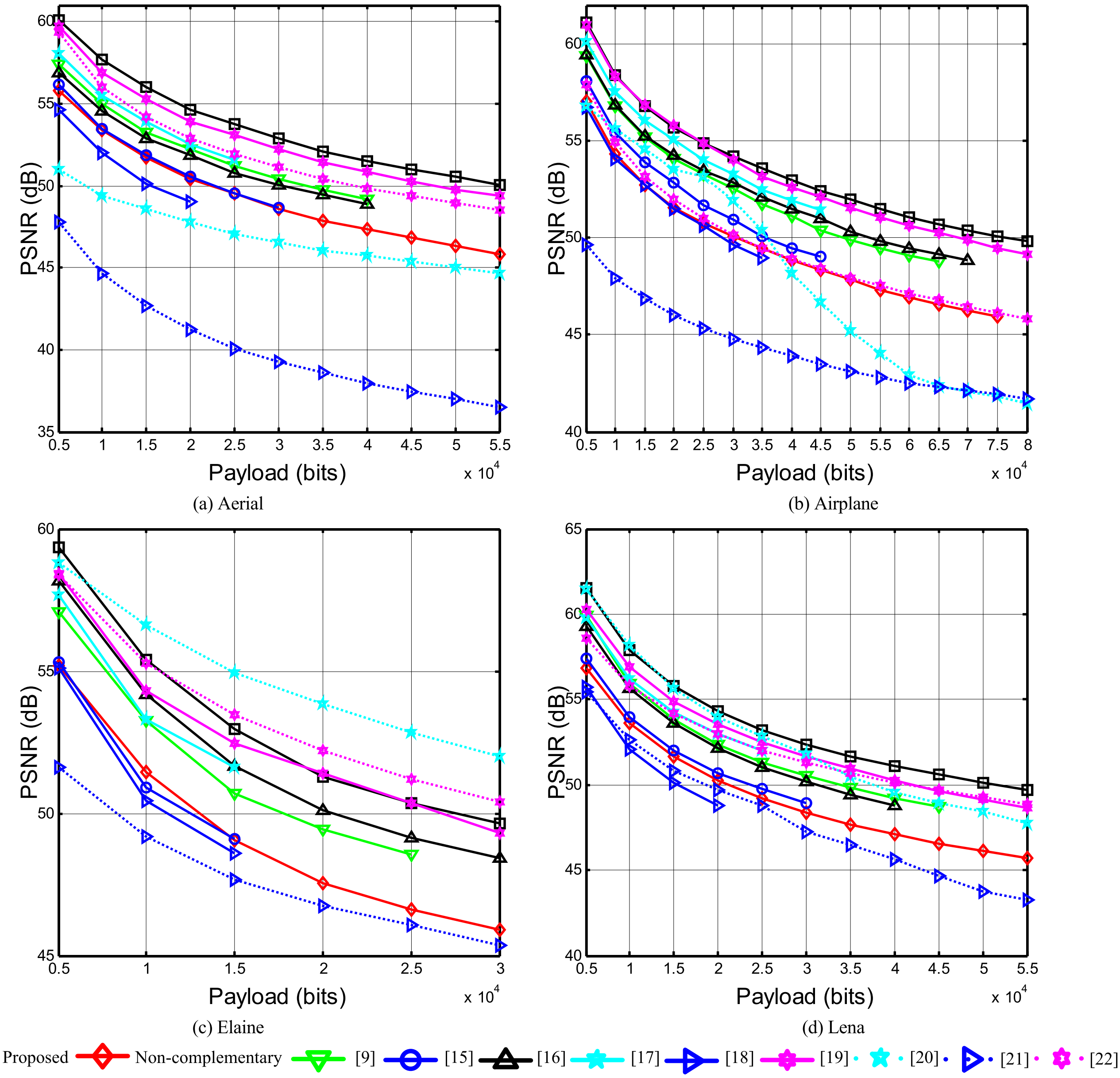



**FIGURE 8.** Performance comparison of the proposed scheme and other schemes.

From Table 4, it can be clearly seen that the proposed scheme has a better performance in terms of the average PSNR value and pure embedding capacity compared with similar schemes [9, 15, 18–19] and non-complementary scheme. The average pure embedding capacity of the proposed scheme is 58212 bits, which is 28076 bits, 12167 bits, 33054 bits, 11351 bits, 24768 bits, and 2040 bits larger than that of Fallahpour's scheme [16], Hong et al.'s scheme [15], Tai et al.'s scheme [18], Li et al.'s scheme [19], Lee et al.'s scheme [9], and non-complementary scheme. The average PSNR value of the proposed scheme is 49.5301 dB, which is 0.8483 dB, 0.9819 dB, 1.1689 dB, 0.9760 dB, 1.0822 dB, and 3.8781 dB larger than that of Yang et al.'s scheme [17], Hong et al.'s scheme [15], Tai et al.'s scheme [18], Li et al.'s scheme [19], Lee et al.'s scheme [9] and non-complementary scheme.

It is noted that although the average PSNR value of Fallahpour's scheme is 51.2485 dB, which is larger than that of the proposed scheme, its average payload is 28076 bits less than that of the proposed scheme. For Yang et al.'s scheme, the embedding capacity is 59584 bits larger than that of the proposed scheme, but its average PSNR value is 48.6818 dB less than that of the proposed scheme.

Fig. 8 shows the performance comparison results of the proposed scheme and other similar schemes which are Yang et al.'s scheme [17], Tai et al.'s scheme [18], Hong et al.'s scheme [15], Lee et al.'s scheme [9], Fallahpour's scheme [16], Li et al.'s scheme [19], Yang et al.'s scheme [20], Hu et al.'s scheme [21], Wahed et al.'s scheme [22] and non-complementary scheme. For these schemes, we vary the embedding capacity from 5000 bits to

**TABLE 5. Performance comparison results of Kodak image data set in terms of PSNR value for [9], [15]–[22] and the proposed scheme with the pure embedding capacity of 20000 bits.**

| Images | Proposed | [9] | [15] | [16] | [17] | [18] | [19] | [20] | [21] | [22] |
|---|---|---|---|---|---|---|---|---|---|---|
| kodim01 | 53.8745 | 49.7996 | 51.3780 | 52.7862 | 53.1809 | 50.1608 | 51.2721 | 50.3858 | 42.3703 | 56.0153 |
| kodim02 | 56.1054 | 52.3535 | 53.7939 | 54.5318 | 55.6194 | 52.3783 | 54.0755 | 52.9948 | 46.9836 | 58.4495 |
| kodim03 | 59.1136 | 56.1139 | 57.1410 | 58.5497 | 59.6009 | 56.7946 | 58.8186 | 50.7278 | 44.7300 | 56.5595 |
| kodim04 | 57.1918 | 52.7738 | 54.5147 | 55.1972 | 56.0767 | 51.8361 | 53.9365 | 50.5180 | 41.9879 | 53.9905 |
| kodim05 | 56.7201 | 52.0596 | 54.1800 | 54.0324 | 55.3086 | 51.2003 | 53.8940 | 47.8853 | 41.5865 | 57.3390 |
| kodim06 | 59.9354 | 57.7963 | 58.0369 | 60.2150 | 60.5976 | 58.4603 | 59.8299 | 46.0502 | 38.4505 | 54.5520 |
| kodim07 | 58.7837 | 55.3215 | 56.6866 | 57.3934 | 59.1594 | 55.6547 | 58.0540 | 53.1371 | 47.4874 | 56.8902 |
| kodim08 | 53.0701 | 48.3823 | 51.2092 | 51.8771 | 51.6272 | 48.7515 | 50.5958 | 48.4247 | 38.9502 | 54.5835 |
| kodim09 | 57.1872 | 53.7254 | 55.5593 | 56.0798 | 57.0522 | 54.3308 | 56.8126 | 56.6413 | 50.7580 | 57.2703 |
| kodim10 | 56.7879 | 53.1316 | 55.2754 | 55.6615 | 56.1891 | 53.3393 | 55.6305 | 50.7584 | 44.9601 | 57.4946 |
| kodim11 | 59.3271 | 54.4605 | 57.3745 | 57.1513 | 58.1366 | 57.0398 | 59.3782 | 51.8820 | 46.8093 | 58.8445 |
| kodim12 | 59.0262 | 54.4931 | 56.5024 | 56.4143 | 58.0098 | 55.3119 | 58.2252 | 49.3984 | 39.9662 | 54.0508 |
| kodim13 | 53.6196 | 48.3108 | 50.0920 | — | 51.3629 | 48.2744 | 50.7455 | 48.4523 | 39.9364 | 54.1864 |
| kodim14 | 56.0184 | 50.4995 | 53.5336 | 52.8428 | 54.8729 | 50.8001 | 53.8578 | 50.5830 | 44.0418 | 57.8767 |
| kodim15 | 55.9349 | 52.9663 | 54.4864 | 55.4667 | 55.9894 | 53.4353 | 55.0440 | 50.3288 | 42.7892 | 56.1292 |
| kodim16 | 59.3387 | 56.3963 | 57.4990 | 58.3836 | 60.0066 | 57.2386 | 59.2912 | 52.0742 | 48.6720 | 58.1636 |
| kodim17 | 58.3758 | 54.3621 | 56.2129 | 56.8209 | 57.8973 | 54.9086 | 57.7328 | 48.4081 | 43.4113 | 58.5436 |
| kodim18 | 54.5317 | 50.6140 | 52.4419 | 52.7028 | 53.4713 | 49.5976 | 52.3916 | 52.8637 | 46.2570 | 59.1639 |
| kodim19 | 57.7977 | 54.6272 | 56.0811 | 56.5372 | 57.9099 | 54.3470 | 56.8598 | 59.4410 | 57.1470 | 57.0437 |
| kodim20 | 55.3322 | 57.7547 | 56.4468 | 57.8644 | 53.1994 | 59.4436 | 52.6949 | 44.7204 | 34.9949 | 51.2037 |
| kodim21 | 57.3555 | 54.2413 | 55.8976 | 56.5723 | 57.5698 | 54.7025 | 57.2256 | 54.3289 | 60.3961 | 58.4927 |
| kodim22 | 56.1571 | 51.5090 | 54.1739 | 53.7564 | 54.2871 | 50.9320 | 55.1306 | 53.8160 | 49.9046 | 53.8000 |
| kodim23 | 58.0532 | 55.3902 | 56.5170 | 57.5548 | 58.5169 | 55.0934 | 57.3709 | 60.7231 | 56.6181 | 57.0741 |
| kodim24 | 55.9227 | 53.4533 | 53.7130 | 56.0425 | 53.8893 | 54.0016 | 53.5242 | 47.0231 | 40.1938 | 51.6310 |
| **Average** | **56.8983** | **53.3557** | **54.9478** | **55.8450** | **56.2305** | **53.6680** | **55.3484** | **51.3153** | **45.3918** | **56.2229** |

their maximum with a step size of 5000 bits. From Fig. 8, it is clearly found that the proposed scheme outperforms these schemes [9], [15]–[22] in terms of the PSNR value under the same embedding capacity for most cases. For example, the proposed scheme can improve by 5.6517 dB compared with Tai et al.'s scheme [18], 2.5280 dB compared with Hong et al.'s scheme [15], 4.1258 dB compared with Lee et al.'s scheme [9], 2.8979 dB compared with Li et al.'s scheme [19], 0.6742 dB compared with Li et al.'s scheme [17], 2.1370 dB compared with Fallahpour's scheme [16], 4.2387 dB compared with non-complementary scheme, 6.6340 dB compared with Yang et al.'s scheme [20], 13.3603 dB compared with Hu et al.'s scheme [21], and 1.6010 dB compared with Wahed et al.'s scheme [22] for the image "Aerial."

In addition, although the embedding procedure of the non-complementary scheme is similar to the proposed scheme, the proposed scheme has better performance with a same embedding capacity. The reason can be explained as follows. In the horizontal embedding procedure, the proposed scheme embeds the secret data bits by increasing or decreasing the pixel values by one, while in the vertical embedding procedure, the proposed scheme embeds the secret data bits by decreasing or increasing the pixel values by one. Hence, according to the vertical embedding procedure, among the pixel values modified by the horizontal embedding procedure, many have a chance to return to their original pixels value. This means that the horizontal and vertical embedding procedures of the proposed scheme are complementary. This property enables the proposed scheme to obtain a particularly good visual quality of the watermarked image.

The Kodak image data set [38] is selected as test image to further compare the proposed scheme with these nine schemes, which contains 24 color images sized 512 × 768 or 768 × 512. In experiments, all color images are first transformed into gray-scale image, and then embed with 20000 and 30000 secret data bits into each gray-scale image, respectively. Because the size of auxiliary information $A_1$ is very large for some images (e.g., for image kodim20, the length of $A_1$ is 90027, for image kodim24, the length of $A_1$ is 20768, etc.) in Kodak image set, the auxiliary information $A_1$ is first compressed by arithmetic coding [34] and then is embedded into the cover image with secret data $B$. Tables 5 and 6 show the performance comparison results. Note that the embedding capacity is pure capacity. The average PSNR value of the proposed scheme is 3.5427 dB, 1.9506 dB, 1.0534 dB, 0.6679 dB, 3.2303 dB, 1.5499 dB, 5.5831 dB, 11.5066 dB, and 0.6755 dB larger than that of Lee et al.'s scheme [9], Hong et al.'s scheme [15], Fallahpour's scheme [16], Yang et al.'s scheme [17], Tai et al.'s scheme [18], Li et al.'s scheme [19], Yang et al.'s scheme [20], Hu et al.'s scheme [21], and Wahed et al.'s scheme [22], respectively, when the pure embedding capacity is 20000 bits. The average PSNR value of the proposed scheme is 2.8689 dB, 1.9861 dB, 0.7885 dB, 0.6558 dB, 2.8616 dB, 1.3654 dB, 6.1747 dB, 11.0579 dB and 0.6749 dB larger than that of Lee et al.'s scheme [9], Hong et al.'s scheme [15], Fallahpour's scheme [16], Yang et al.'s scheme [17], Tai et al.'s scheme [18], Li et al.'s scheme [19], Yang et al.'s scheme [20], Hu et al.'s scheme [21], and Wahed et al.'s scheme [22], respectively, when the payload is 30000 bits. These results show that the proposed scheme outperforms nine

**TABLE 6. Performance comparison results of Kodak image data set in terms of PSNR value for [9], [15]–[22] and the proposed scheme with the pure embedding capacity of 30000 bits.**

| Images | Proposed | [9] | [15] | [16] | [17] | [18] | [19] | [20] | [21] | [22] |
|---|---|---|---|---|---|---|---|---|---|---|
| kodim01 | 52.2959 | — | 50.4266 | 51.2073 | 51.6321 | 48.5875 | 50.3778 | 47.3665 | 40.5227 | 53.3899 |
| kodim02 | 54.3170 | 50.8657 | 52.3653 | 53.1956 | 54.1495 | 50.8720 | 52.7678 | 51.2495 | 45.2383 | 56.7655 |
| kodim03 | 57.2181 | 54.1298 | 55.1091 | 56.6645 | 57.5689 | 54.6394 | 56.8184 | 48.1877 | 43.4133 | 54.8446 |
| kodim04 | 54.9052 | 51.0863 | 52.6107 | 53.5588 | 54.2580 | 50.3465 | 52.3079 | 48.6195 | 40.9479 | 52.2864 |
| kodim05 | 54.3859 | 49.9140 | 52.0451 | 51.9686 | 53.4338 | 49.1237 | 51.7606 | 45.6954 | 39.9903 | 55.5589 |
| kodim06 | 58.2658 | 55.9930 | 56.2324 | 58.5470 | 57.4635 | 56.7478 | 58.2100 | 43.9294 | 38.1768 | 52.5960 |
| kodim07 | 56.8830 | 53.2429 | 54.7846 | 55.2287 | 57.2452 | 53.4903 | 55.9154 | 50.7966 | 45.4427 | 54.7184 |
| kodim08 | 51.9491 | — | 49.9693 | 51.2096 | 49.9143 | 48.3205 | 49.3666 | 46.7856 | 37.9303 | 52.9210 |
| kodim09 | 55.3468 | 51.8038 | 53.6817 | 54.3293 | 55.0885 | 52.1903 | 54.9489 | 52.0286 | 50.3558 | 54.7022 |
| kodim10 | 55.0519 | 51.4149 | 53.4783 | 53.9502 | 54.4459 | 51.5444 | 53.9102 | 48.9295 | 43.2751 | 55.7438 |
| kodim11 | 57.0155 | 52.4497 | 54.3926 | 55.0963 | 55.8792 | 53.5448 | 56.7322 | 48.6685 | 45.0744 | 57.0435 |
| kodim12 | 56.7206 | 51.5439 | 53.8004 | 53.9889 | 55.5933 | 52.3606 | 55.7170 | 47.0276 | 39.7519 | 52.1662 |
| kodim13 | 50.7106 | — | 48.3305 | — | 49.5127 | — | 48.5430 | 46.6910 | 39.1366 | 52.4139 |
| kodim14 | 53.7185 | 49.1863 | 51.1071 | 51.5217 | 53.0315 | 49.2509 | 51.3486 | 49.2799 | 42.0084 | 55.9105 |
| kodim15 | 54.6280 | 51.7366 | 53.1060 | 54.3064 | 54.8403 | 52.0906 | 53.7393 | 48.9915 | 41.5032 | 54.3856 |
| kodim16 | 57.5072 | 54.1787 | 55.4802 | 56.2210 | 57.7042 | 55.1016 | 57.4098 | 50.4266 | 47.3475 | 56.3419 |
| kodim17 | 56.0587 | 52.1193 | 53.9419 | 54.1182 | 55.5784 | 52.2809 | 55.3358 | 46.0879 | 43.0160 | 56.7109 |
| kodim18 | 52.7248 | 48.9868 | 50.5623 | — | 51.7264 | — | 50.4577 | 50.9331 | 44.2985 | 57.4373 |
| kodim19 | 55.9403 | 52.5382 | 54.2026 | 54.2740 | 55.7769 | 52.3299 | 55.0635 | 52.6711 | 55.0883 | 55.7460 |
| kodim20 | 55.1350 | 57.7668 | 55.4947 | 57.9826 | 52.9519 | 58.6097 | 52.3295 | 43.9391 | 34.6423 | 49.3229 |
| kodim21 | 55.5335 | 52.2829 | 54.0467 | 54.4882 | 55.6224 | 52.6592 | 55.3512 | 51.8352 | 58.3449 | 56.7221 |
| kodim22 | 53.8981 | 49.7114 | 51.5627 | 52.2460 | 52.6712 | 49.2655 | 52.2160 | 50.3458 | 47.4917 | 52.0387 |
| kodim23 | 56.3887 | 53.4745 | 54.7149 | 55.6291 | 56.5624 | 53.0857 | 55.5898 | 57.1935 | 54.5841 | 55.2145 |
| kodim24 | 54.6610 | 51.4304 | 52.1461 | 54.0749 | 52.8682 | 51.7569 | 52.2730 | 45.3868 | 38.2884 | 50.0820 |
| **Average** | **55.0525** | **52.1836** | **53.0663** | **54.2640** | **54.3966** | **52.1909** | **53.6871** | **48.8777** | **43.9946** | **54.3776** |

similar state-of-the-art schemes with the same embedding capacity.

## V. CONCLUSION

In this paper, a new, simple reversible watermarking scheme based on complementary embedding strategy is proposed for high-fidelity images in spatial domain. To achieve high image quality, the complementary embedding method is used in the vertical direction embedding. The distortion which produced in the horizontal direction embedding can be balanced out in vertical direction embedding. This property enables the proposed scheme to obtain an exceptionally good quality of watermarked image. Experimental results show that the watermarked image quality of the proposed scheme is superior to some similar reversible watermarking schemes under the same embedding capacity. However, at the current stage, the performance of the proposed scheme is not very well. In our future work, we will try to further improve the embedding capacity and visual quality of watermarked image by designing more accurate predictor and the proposed complementary embedding strategy.

## ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their helpful comments.

## REFERENCES

[1] Y.-Q. Shi, X. Li, X. Zhang, H.-T. Wu, and B. Ma, "Reversible data hiding: Advances in the past two decades," *IEEE Access*, vol. 4, pp. 3210–3237, 2016.

[2] J. Fridrich, M. Goljan, and R. Du, "Lossless data embedding—New paradigm in digital watermarking," *EURASIP J. Appl. Signal Process.*, vol. 2, no. 2, pp. 185–196, Dec. 2002.

[3] M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, "Lossless generalized-LSB data embedding," *IEEE Trans. Image Process.*, vol. 14, no. 2, pp. 253–266, Feb. 2005.

[4] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 8, pp. 890–896, Aug. 2003.

[5] 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.

[6] H. J. Kim, V. Sachnev, Y. Q. Shi, J. Nam, and H. G. Choo, "A novel difference expansion transform for reversible data embedding," *IEEE Trans. Inf. Forensics Security*, vol. 4, no. 3, pp. 456–465, Sep. 2008.

[7] M. Liu, H. S. Seah, C. Zhu, W. Lin, and F. Tian, "Reducing location map in prediction-based difference expansion for reversible image data embedding," *Signal Process.*, vol. 92, no. 3, pp. 819–828, Mar. 2012.

[8] Y. Hu, H.-K. Lee, and J. Li, "DE-based reversible data hiding with improved overflow location map," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 2, pp. 250–260, Feb. 2009.

[9] C. F. Lee and H. L. Chen, "Adjustable prediction-based reversible data hiding," *Digit. Signal Process.*, vol. 22, pp. 941–953, Dec. 2012.

[10] D. M. Thodi and J. J. Rodriguez, "Expansion embedding techniques for reversible watermarking," *IEEE Trans. Image Process.*, vol. 16, no. 3, pp. 721–730, Mar. 2007.

[11] X. Gui, X. Li, and B. Yang, "A high capacity reversible data hiding scheme based on generalized prediction-error expansion and adaptive embedding," *Signal Process.*, vol. 98, pp. 370–380, May 2014.

[12] X. Li, B. Yang, and T. Zeng, "Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection," *IEEE Trans. Image Process.*, vol. 20, no. 12, pp. 3524–3533, Dec. 2011.

[13] D. Coltuc, "Improved embedding for prediction-based reversible watermarking," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 3, pp. 873–882, Sep. 2011.

[14] Z. Ni, Y.-Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 3, pp. 354–362, Mar. 2006.

[15] W. Hong, T.-S. Chen, and C.-W. Shiu, "Reversible data hiding for high quality images using modification of prediction errors," *J. Syst. Softw.*, vol. 82, no. 11, pp. 1833–1842, 2009.

[16] M. Fallahpour, "Reversible image data hiding based on gradient adjusted prediction," *IEICE Electron. Express*, vol. 5, no. 20, pp. 870–876, 2008.

[17] C.-H. Yang and M.-H. Tsai, "Improving histogram-based reversible data hiding by interleaving predictions," *IET Image Process.*, vol. 4, no. 4, pp. 223–234, 2010.

[18] W.-L. Tai, C.-M. Yeh, and C.-C. 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.

[19] Y.C. Li, C.M. Yeh, C.C. Chang, "Data hiding based on the similarity between neighboring pixels with reversibility," *Digit. Signal Process.*, vol. 20, pp. 1116–1128, Jul. 2010.

[20] C.-N. Yanga, S.-C. Hsua, and C. Kim, "Improving stego image quality in image interpolation based data hiding," *Comput. Standards Interfaces*, vol. 50, pp. 209–215, Feb. 2017.

[21] J. Hu and T. R. Li, "Reversible steganography using extended image interpolation technique," *Comput. Elect. Eng.*, vol. 46, no. 8, pp. 447–455, Aug. 2015.

[22] M. A. Wahed and H. Nyeem, "High capacity reversible data hiding with interpolation and adaptive embedding," *PLoS ONE*, vol. 14, no. 3, Mar. 2019, Art. no. e0212093.

[23] X.-T. Wang, C.-C. Chang, T.-S. Nguyen, and M. C. Li, "Reversible data hiding for high quality images exploiting interpolation and direction order mechanism," *Digit. Signal Process.*, vol. 23, no. 2, pp. 569–577, Mar. 2013.

[24] Q. Ying, Z. Qian, X. Zhang, and D. Ye, "Reversible data hiding with image enhancement using histogram shifting," *IEEE Access*, vol. 7, pp. 46506–46521, 2019.

[25] X. Li, B. Li, B. Yang, and T. Zeng, "General framework to histogram-shifting-based reversible data hiding," *IEEE Trans. Image Process.*, vol. 22, no. 6, pp. 2181–2191, Jun. 2013.

[26] X. Li, W. Zhang, X. Gui, and B. Yang, "A novel reversible data hiding scheme based on two-dimensional difference-histogram modification," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 7, pp. 1091–1100, Jul. 2013.

[27] 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 Process.*, vol. 93, no. 1, pp. 198–205, Jan. 2013.

[28] X. Qu and H. J. Kim, "Pixel-based pixel value ordering predictor for high-fidelity reversible data hiding," *Signal Process*, vol. 111, no. 1, pp. 249–260, 2015.

[29] X. Ma, Z. Pan, S. Hu, and L. Wang, "High-fidelity reversible data hiding scheme based on multi-predictor sorting and selecting mechanism," *J. Vis. Commun. Image Represent.*, vol. 28, pp. 71–82, Apr. 2015.

[30] S. Chand, R. Kumar, and S. Singh, "A reversible high capacity data hiding scheme using combinatorial strategyy," *Int. J. Multimedia Intell. Secur.*, vol. 3, no. 2, pp. 146–161, Nov. 2018.

[31] R. Kumar, S. Chand, and S. Singh, "An improved histogram-shifting-imitated reversible data hiding based on HVS characteristics," *Multimedia Tools Appl.*, vol. 77, no. 11, pp. 13445–13457, Jun. 2018.

[32] R. Kumar, S. Chand, and S. Singh, "A reversible data hiding scheme using pixel location," *Int. Arab J. Inf. Technol.*, vol. 15, no. 4, pp. 763–768, Jul. 2018.

[33] T. D. Kieu and C.-C. Chang, "A high stego-image quality steganographic scheme with reversibility and high payload using multiple embedding strategy," *The J. Syst. Softw.*, vol. 82, pp. 1743–1752, Oct. 2009.

[34] C. C. Chang and T. D. Kieu, "A reversible data hiding scheme using complementary embedding strategy," *Inf. Sci.*, vol. 180, pp. 3045–3058, Aug. 2010.

[35] L. Jia, S.-H. Shin, and -Y. Yoo, "A reversible data hiding scheme using inverse embedding methods in double-embedding strategies," in *Proc. 7th Int. Conf. Informat. Syst.*, Cairo, Egypt, Mar. 2010, pp. 1–7.

[36] *Arithmetic Coding and Huffman Coding in MATLAB*. Accessed: Mar. 18, 2019. [Online]. Available: http://www.ux.uis.no/ karlsk/proj99/index.html

[37] *The USC-SIPI Image Database*. Accessed: Mar. 18, 2019. [Online]. Available: http://sipi.usc.edu/database/

[38] *Kodak Lossless True Color Image Suite*. Accessed: Mar. 18, 2019. [Online]. Available: http://r0k.us/graphics/kodak/


**XIANGGUANG XIONG** was born in 1984. He received the M.S. degree from the College of Computer Science, South-Central Minzu University, Wuhan, China. He is currently an Associate Professor with Guizhou Normal University. His research interests include multimedia information security and image processing.