Efficient Reversible Data Hiding Based on Multiple Histograms Modification

Xiaolong Li, Weiming Zhang, Xinlu Gui, and Bin Yang

Abstract—Prediction-error expansion (PEE) is the most successful reversible data hiding (RDH) technique, and existing PEE-based RDH methods are mainly based on the modification of one- or two-dimensional prediction-error histogram (PEH). The two-dimensional PEH-based methods perform generally better than those based on one-dimensional PEH; however, their performance is still unsatisfactory since the PEH modification manner is fixed and independent of image content. In this paper, we propose a new RDH method based on PEE for multiple histograms. Unlike the previous methods, we consider in this paper a sequence of histograms and devise a new embedding mechanism based on multiple histograms modification (MHM). A complexity measurement is computed for each pixel according to its context, and the pixels with a given complexity are collected together to generate a PEH. By varying the complexity to cover the whole image, a sequence of histograms can be generated. Then, two expansion bins are selected in each generated histogram and data embedding is realized based on MHM. Here, the expansion bins are adaptively selected considering the image content such that the embedding distortion is minimized. With such selected expansion bins, the proposed MHM-based RDH method works well. Experimental results show that the proposed method outperforms the conventional PEE and its miscellaneous extensions including both one- or two-dimensional PEH-based ones.

Index Terms—Reversible data hiding, prediction-error expansion, multiple histograms modification, adaptive embedding.

I. INTRODUCTION

REVERSIBLE data hiding (RDH) aims to embed secret message into a cover image by slightly modifying its pixels, and more importantly, the original image as well as the embedded message should be completely restored from the marked image [1], [2]. In the last decade, RDH has received much attention from the information hiding community and this technique has also been applied in some applications such as image authentication [3], [4], medical image processing [5]–[7], multimedia archive management [8], image trans-coding [9], and data coloring in the cloud [10], etc. In general, RDH is a fragile technique and the marked image cannot undergo any degradation. In this light, a RDH method is usually evaluated by its capacity-distortion performance, i.e., for a given embedding capacity (EC), one expects to minimize the embedding distortion measured by PSNR of the marked image versus the original one.

Early RDH methods are mainly based on lossless compression [11]–[14]. The idea behind these methods is to losslessly compress a feature set of cover image and utilize the saved space for reversible embedding. In [11], Fridrich *et al.* proposed to compress a proper bit-plane with the minimum redundancy. In [13], Celik *et al.* proposed a generalized least significant bit (LSB) compression method to improve the compression efficiency by using unaltered bit-planes as side information. However, the lossless-compression-based methods cannot yield satisfactory performance, since the correlation within a bit-plane is too weak to provide a high EC. As EC increases, one needs to compress more bit-planes, thus the distortion increases dramatically.

Later on, more efficient RDH methods based on histogram modification and expansion technique have been devised. The histogram-modification-based method is firstly proposed by Ni *et al.* [15]. This method focuses on high visual quality with quite limited EC, in which the peak point of image histogram is utilized for data embedding. In this method, each pixel value is modified at most by 1, and thus the marked image quality is well guaranteed. Ni *et al.*'s method is improved by Lee *et al.* [16] by using the histogram of difference image. The spatial correlation of natural images is exploited in [16] by considering the difference of adjacent pixels. Thus, a regular-shaped histogram is utilized in Lee *et al.*'s method. This histogram is centered at origin and has rapid two-sided decay which is more suitable for RDH. The expansion technique is firstly proposed by Tian [17]. This method is performed on pixel pairs, and one data bit is embedded into each selected pixel pair by expanding its difference. Compared with the lossless-compression-based RDH, Tian's difference expansion (DE) based method can provide a higher EC with an improved PSNR. The DE approach has attracted considerable attention and it makes an important progress in RDH. Afterwards, the expansion technique has been widely investigated and developed, mainly in the aspects of integer-to-integer

transformation [18]–[22], location map reduction [23]–[25], and prediction-error expansion (PEE) [26]–[35]. Besides the histogram modification and the expansion technique, the analysis about theoretical capacity limit subjected to admissible distortion has also been studied in some recent works [36]–[39].

Nowadays, the most effective and extensively exploited RDH technique is the PEE technique which is firstly proposed by Thodi and Rodriguez [26]. Instead of the difference value in DE, the prediction-error is utilized in PEE for expansion embedding. Thus, unlike DE where only the correlation of two adjacent pixels is considered, the local correlation of a larger neighborhood is exploited in PEE. As a result, compared with DE, better performance can be derived by PEE. Following Thodi and Rodriguez's work, many RDH techniques related to PEE have been proposed in recent years, for example, double-layered embedding [31], [40], [41], adaptive embedding [40]–[43], context modification [32], optimal expansion bins selection [34], [44]–[47], and two-dimensional histogram modification [48], [49], etc. On the other hand, some PEE-based methods [50]–[54] exploit advanced prediction techniques to generate a more sharply distributed prediction-error histogram (PEH), and this is also helpful for enhancing the embedding performance.

Notice that, most previous PEE-based methods are based on one- or two-dimensional PEH modification. The two-dimensional PEH based methods perform generally better than those based on one-dimensional PEH, however, their performance is still unsatisfactory since the PEH modification manner is fixed and independent of image content. In this work, we focus on PEE and propose a new RDH method based on PEE for multiple histograms. Unlike the previous methods, we consider here a sequence of histograms and devise a new embedding mechanism based on multiple histograms modification (MHM). By MHM, the embedding performance can be optimized by adaptively selecting expansion bins in each histogram considering the image content. Specifically, for each pixel, its prediction value and complexity measurement are computed according to its context, and multiple histograms are generated for different complexity levels. That is to say, the pixels with a given complexity are collected together to generate a PEH, and by varying the complexity measurement to cover the whole image, a sequence of histograms can be derived. After that, two expansion bins are selected in each generated histogram and data embedding is realized based on MHM. Moreover, based on an estimation of embedding distortion, the expansion bins can be effectively determined such that the distortion is minimized. The proposed method is a generalization of some existing methods and it can well exploit image redundancy to achieve improved embedding performance. Experimental results show that the proposed method outperforms the conventional PEE (C-PEE) and its miscellaneous extensions including both one- or two-dimensional PEH based ones. Our advantages mainly lie in the MHM-based embedding mechanism and the selection of optimal expansion bins.

The rest of the paper is organized as follows. Some previous PEE-based works including C-PEE and its improvements are briefly introduced in Section II. Section III presents the proposed MHM-based RDH method in detail. The experimental results as well as the comparison with the prior arts are shown in Section IV. Finally, Section V concludes this paper.

II. RELATED WORKS

We present in this section some RDH works related to PEE including C-PEE [26]–[29], the PEE with adaptive embedding (A-PEE) [40]–[43], the PEE with optimal expansion bins selection (O-PEE) [34], [44], and the PEE which combines both adaptive embedding and optimal expansion bins selection (AO-PEE) [55], [56].

The first step of these PEE-based methods is the generation of PEH. First of all, in a specific scanning order, the cover pixels are collected into a one-dimensional sequence as $(x_1, \ldots, x_N)$ where N is the total number of collected pixels. Then, a predictor is employed to predict each x_i, and the prediction value denoted by $\widehat{x_i}$ should be rounded off if it is not an integer. Next, the prediction-error is computed by

$$e_i = x_i - \widehat{x_i}. \tag{1}$$

Finally, the prediction-error sequence $(e_1, \ldots, e_N)$ is derived and the corresponding PEH denoted by h can be established as

$$h(e) = \#\{1 \le i \le N : e_i = e\}, \quad \forall e \in \mathbb{Z} \tag{2}$$

where # means the cardinal number of a set. The second step of PEE-based methods is the modification of PEH. We now introduce the PEH modification mechanisms for the above four types of PEE-based methods, respectively.

A. C-PEE

After PEH generation, the C-PEE embedding procedure contains following steps. First, for a prediction-error e_i, it is expanded or shifted as

$$\widetilde{e}_i = \begin{cases} e_i + m, & \text{if } e_i = 0 \\ e_i - m, & \text{if } e_i = -1 \\ e_i + 1, & \text{if } e_i > 0 \\ e_i - 1, & \text{if } e_i < -1 \end{cases} \tag{3}$$

where $m \in \{0, 1\}$ is a to-be-embedded data bit. With (3), the bins -1 and 0 are expanded to embed data, while other bins are shifted to create vacancies to ensure the reversibility. Then, the cover pixel x_i is modified to $\widetilde{x}_i = \widehat{x_i} + \widetilde{e}_i$ to generate the marked pixel. Notice that the above procedure will stop once all data bits are embedded, i.e., only the cover pixels $(x_1, \ldots, x_{N_{\text{end}}})$ need to be processed where $N_{\text{end}} \le N$ is the smallest index such that the payload can be embedded into the first N_{end} cover pixels.

An illustration of histogram modification mechanism for C-PEE is shown in Fig. 1. Particularly, the mapping of bins for C-PEE is shown in Fig. 1(b), in which the red points (bins -1 and 0) are expanded while black points (other bins) are shifted.

The C-PEE extraction and image restoration procedure can be summarized as follows. First, determine the prediction $\widehat{x_i}$ of marked pixel $\widetilde{x}_i$ for each $i \in \{1, \ldots, N_{\text{end}}\}$. The marked
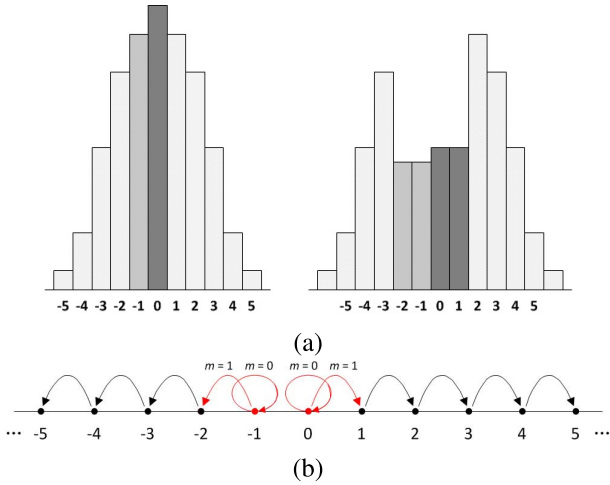
(a)

(b)

Fig. 1. Histogram modification mechanism for C-PEE. (a) PEH before (left) and after (right) C-PEE embedding. (b) Mapping of bins for C-PEE.

prediction-error is thus $\widetilde{e}_i = \widetilde{x}_i - \widehat{x}_i$. Then, for each $\widetilde{e}_i$, the original prediction-error can be recovered as

$$e_i = \begin{cases} \widetilde{e}_i, & \text{if } \widetilde{e}_i \in \{-1, 0\} \\ \widetilde{e}_i - 1, & \text{if } \widetilde{e}_i > 0 \\ \widetilde{e}_i + 1, & \text{if } \widetilde{e}_i < -1. \end{cases} \quad (4)$$

Meanwhile, the embedded data bit can be extracted as $m = 0$ if $\widetilde{e}_i \in \{-1, 0\}$, or $m = 1$ if $\widetilde{e}_i \in \{-2, 1\}$. Finally, restore the cover pixel as $x_i = \widehat{x}_i + e_i$.

A key issue for the reversibility of C-PEE is that the prediction values obtained by decoder should be the same as those of encoder. For example, by using median-edge-detector (MED [28]) or gradient-adjusted-predictor (GAP [42]) which is based on half-enclosing casual pixels for prediction, the decoder can inversely scan and process pixels to get the same prediction values.

B. A-PEE

As an extension to C-PEE, the adaptive embedding strategy has been proposed to better exploit the image redundancy [40]–[43]. Specifically, for A-PEE, a complexity measurement denoted by n_i is computed for each x_i according to its context. Then, only the pixels satisfying $n_i < T$ will be embedded, where $T > 0$ is a pre-selected threshold. This means, for each x_i with $n_i < T$, it will be processed according to the C-PEE embedding procedure. Otherwise, i.e., $n_i \geq T$, x_i is ignored and its value keeps unchanged. Here, the threshold T is an important factor for the embedding performance of A-PEE. To better utilize smooth pixels, T is taken as the smallest positive integer such that the payload can be successfully embedded.

The extraction procedure of A-PEE is almost the same as that of C-PEE. The difference is that, only the pixels with $n_i < T$ need to be processed, while other pixels can be directly restored as themselves since they are unmodified during data embedding.

C. O-PEE

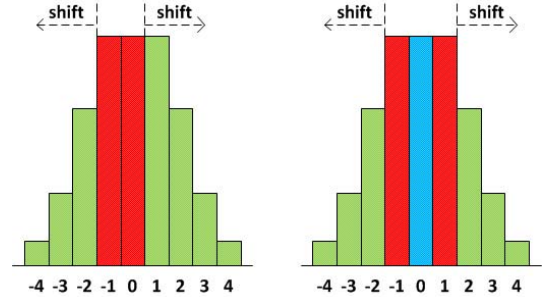We use the example introduced in [44] to illustrate O-PEE. Consider here a specific PEH satisfying

Fig. 2. C-PEE (left) v.s. O-PEE (right).

$h(-1) = h(0) = h(1) = H$ and suppose that EC is $2H$. In this case, for C-PEE (see the left figure of Fig. 2 for an illustration), the expected value of embedding distortion in l^2-norm can be formulated as

$$\mathbf{E}(\|\widetilde{I} - I\|_2^2) = \sum_{i=1}^{N} \mathbf{E}((\widetilde{x}_i - x_i)^2)$$
$$= \frac{1}{2}\Big(h(-1) + h(0)\Big) + \sum_{e \notin \{-1,0\}} h(e) = N - H \quad (5)$$

where I and $\widetilde{I}$ are cover and marked images, respectively. However, as pointed out in [44], the choice of expansion bins as -1 and 0 is not mandatory. For this specific PEH, as shown in the right figure of Fig. 2, one can select the bins -1 and 1 for expansion. In this situation, only the bins larger than 1 or smaller than -1 need to be shifted, while the bin 0 may remain unchanged. As a result, compared with C-PEE, the expected value of embedding distortion is reduced from $N - H$ to $N - 2H$.

Through this example, we see that it is possible to improve C-PEE by suitably selecting expansion bins. For a given EC, one can select two expansion bins $a < b$ to minimize the embedding distortion, and the optimal expansion bins can be determined by repeated embedding for a collection of (a, b). An example for determining (a, b) will be given later at the end of this section. We now describe the O-PEE embedding procedure when a and b are already determined. This is a natural extension of C-PEE embedding. First, for a prediction-error e_i, it is modified as

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

where $m \in \{0, 1\}$ is a to-be-embedded data bit. Then, x_i is modified to $\widetilde{x}_i = \widehat{x}_i + \widetilde{e}_i$ to generate the marked pixel. Clearly, O-PEE turns out to be C-PEE when $(a, b) = (-1, 0)$. An illustration of O-PEE with $(a, b) = (-3, 2)$ is shown in Fig. 3. One can see the difference between C-PEE and O-PEE by comparing this figure with Fig. 1.

The O-PEE extraction is just the inverse of data embedding. The obvious data extraction and image restoration procedure of O-PEE is omitted.
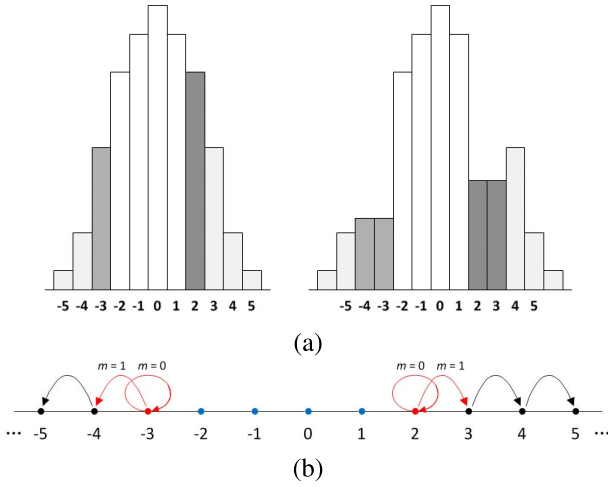
Fig. 3. Histogram modification mechanism for O-PEE with $(a, b) = (-3, 2)$. Here, the bins $\{-2, -1, 0, 1\}$ are unmodified. (a) PEH before (left) and after (right) O-PEE embedding. (b) Mapping of bins for O-PEE.

Fig. 4. Context of a pixel x_i.

D. AO-PEE

If the aforementioned two improvements for C-PEE, adaptive embedding and optimal expansion bins selection, are combined together, better performance can be expected [55], [56]. This combined embedding is called AO-PEE in our work. We now briefly describe the AO-PEE embedding procedure as follows. First, for a cover pixel x_i, if its complexity measurement $n_i < T$, the prediction-error e_i is modified according to (6) to get $\widetilde{e}_i$. Then, x_i is modified to $\widetilde{x}_i = \widehat{x}_i + \widetilde{e}_i$ to get the marked pixel. Here, the same as A-PEE, the pixels with $n_i \geq T$ are ignored and unmodified.

There are three parameters to be determined in AO-PEE: the complexity threshold T and expansion bins (a, b). In [55], a recursive search method is proposed for determining these parameters. Specifically, for given (a, b), T is taken as the smallest positive integer such that the payload can be successfully embedded and the corresponding embedding distortion is recorded as $D_{a,b}$. Then, repeat the above step for a collection of (a, b), and finally obtain the optimal (a, b, T) as the one providing the smallest $D_{a,b}$.

We now give some experimental results of C-PEE, A-PEE, O-PEE and AO-PEE for a better explanation of these methods. The experiment is conducted as follows.

- The MED predictor is employed for prediction, i.e., referring to Fig. 4, the prediction value of x_i is computed as

$$\widehat{x}_i = \begin{cases} \min(u, v), & \text{if } w \geq \max(u, v) \\ \max(u, v), & \text{if } w \leq \min(u, v) \\ u + v - w, & \text{otherwise.} \end{cases} \quad (7)$$

- For A-PEE and AO-PEE, the complexity measurement n_i is computed as the mean of pixel differences in the

context of x_i, i.e.,

$$n_i = \left\lfloor \frac{|u - v| + |v - w| + |w - u|}{3} \right\rfloor \quad (8)$$

where $\lfloor \cdot \rfloor$ is the floor function.

- For O-PEE and AO-PEE, the bins (a, b) satisfying $-7 \leq a < b \leq 7$ are tested to determine the optimal expansion bins.

The embedding results for the standard 512×512 sized gray-scale image Lena are listed in Table I. The corresponding parameters including the complexity threshold T (for A-PEE and AO-PEE) and optimal expansion bins (a, b) (for O-PEE and AO-PEE) are also listed in this table, e.g., for AO-PEE with EC of 20,000 bits, $(a, b, T) = (-2, 2, 3)$. According to this table, compared with C-PEE, better performance is achieved using A-PEE, O-PEE and AO-PEE. Moreover, with the combined improvements, AO-PEE performs the best among these PEE-based methods.

III. PROPOSED METHOD

The proposed method is presented in this section. We first show that, C-PEE can be implemented in a framework of MHM. Based on this viewpoint and by considering a generalized expansion and shifting mechanism, a new RDH method based on MHM is proposed in Section III-A. The MHM-based method relies on a set of parameters and it includes C-PEE, A-PEE, O-PEE and AO-PEE as special cases by taking specific parameters. Besides, to optimize the embedding performance, a method determining the best parameters is presented in Section III-B. Finally, the implementation details of the proposed method are described in Section III-C.

A. General Embedding Framework Based on MHM

Using the notations introduced in Section II, we define a histogram sequence h_n for $n \geq 0$ as

$$h_n(e) = \#\{1 \leq i \leq N : e_i = e, \ n_i = n\} \quad (9)$$

i.e., h_n counts the frequencies of prediction-errors for the pixels whose complexity is a fixed value n. By this definition, the PEH h defined in (2) is divided into a sequence of histograms $\{h_n\}_{n \geq 0}$ since $h(e) = \sum_{n \geq 0} h_n(e)$ holds for each e. In this case, the modification on h can be conducted by modifying each h_n. As a result, C-PEE can be implemented by modifying the histograms $\{h_n\}_{n \geq 0}$. An illustration for this MHM-based embedding mechanism of C-PEE is shown in Fig. 5(a). In this figure, the two-dimensional coordinate of (e, n) is plotted for $e \in \mathbb{Z}$ and $n \geq 0$, in which each horizontal line can be viewed as the coordinate of e for a fixed n. And, C-PEE can be viewed as a mapping of (e, n), where each red point is mapped (expanded) to two points to embed one data bit while each black point is mapped (shifted) to another black point to ensure the reversibility. For example, for a cover pixel with $(e_i, n_i) = (-1, 3)$, it corresponds to the point A in Fig. 5(a). Since e_i is either unchanged or modified to -2 in C-PEE, it is equivalent to expanding A to itself and $(-2, 3)$ in the two-dimensional coordinate. For another

TABLE I

COMPARISON OF PSNR (IN dB) FOR C-PEE, A-PEE, O-PEE AND AO-PEE, FOR THE STANDARD
512×512 SIZED GRAY-SCALE IMAGE LENA WITH DIFFERENT CAPACITIES

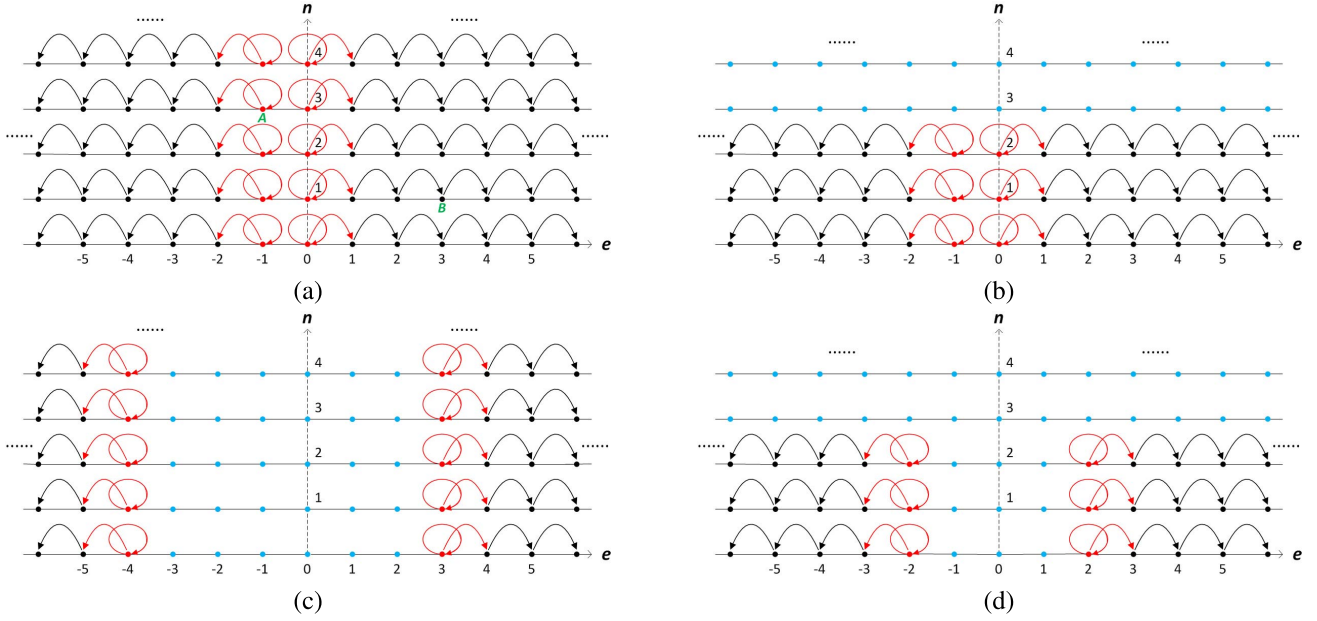| EC (in bits)	C-PEE	A-PEE	O-PEE	AO-PEE
10,000	57.24	57.96 ($T = 2$)	57.37 ($a = 0$, $b = 2$)	59.08 ($a = -4$, $b = 4$, $T = 3$)
20,000	53.21	54.38 ($T = 3$)	53.71 ($a = -4$, $b = 3$)	55.38 ($a = -2$, $b = 2$, $T = 3$)
30,000	51.15	52.43 ($T = 4$)	51.85 ($a = -3$, $b = 3$)	53.00 ($a = -2$, $b = 2$, $T = 5$)
40,000	49.89	50.86 ($T = 5$)	50.44 ($a = -2$, $b = 2$)	51.00 ($a = -2$, $b = 1$, $T = 8$)
50,000	48.87	49.39 ($T = 10$)	49.06 ($a = -2$, $b = 0$)	49.39 ($a = -1$, $b = 0$, $T = 10$)

Fig. 5. MHM-based embedding mechanisms for C-PEE, A-PEE, O-PEE and AO-PEE, in which the blue points are mapped to themselves. For A-PEE, O-PEE and AO-PEE, the parameters are taken as the ones used in Table I with EC of 20,000 bits. (a) C-PEE. (b) A-PEE with $T = 3$. (c) O-PEE with $(a, b) = (-4, 3)$. (d) AO-PEE with $(a, b, T) = (-2, 2, 3)$.

cover pixel with $(e_i, n_i) = (3, 1)$, it corresponds to the point B, and shifting e_i to 4 in C-PEE is equivalent to shifting B to $(4, 1)$ in the two-dimensional coordinate.

With the histogram sequence (9) which is defined considering the complexity measurement, we see that C-PEE is based on MHM and the modification manner for each h_n is the same. However, the identical modification on h_n is not obligatory. By taking different modification mechanisms for different h_n, a general embedding method can be designed. We now describe our idea. For each $n \geq 0$, we first take two parameters $a_n < b_n$. In MHM embedding, for each h_n, the bins a_n and b_n will be expanded, and the bins larger than b_n or smaller than a_n will be shifted, while other bins are unmodified. Specifically, for a cover pixel x_i, take n as its complexity measurement n_i, and modify its prediction-error e_i to get $\widetilde{e}_i$ according to (a_n, b_n) as

$$
\widetilde{e}_i = \begin{cases}
e_i, & \text{if } a_n < e_i < b_n \\
e_i + m, & \text{if } e_i = b_n \\
e_i - m, & \text{if } e_i = a_n \\
e_i + 1, & \text{if } e_i > b_n \\
e_i - 1, & \text{if } e_i < a_n
\end{cases} \tag{10}
$$

where $m \in \{0, 1\}$ is a to-be-embedded data bit. Then, modify x_i to $\widetilde{x}_i = x_i + \widetilde{e}_i$. Notice that, besides integer-valued

parameters, one may take $(a_n, b_n) = (-\infty, \infty)$ as well. In this special case, for x_i with $n_i = n$, the pixel is unchanged whatever its prediction-error e_i is.

The data extraction and image restoration of MHM can be summarized as follows. For a marked pixel $\widetilde{x}_i$, take $n = n_i$, then the original prediction-error can be recovered according to $\widetilde{e}_i$ and (a_n, b_n) as

$$
e_i = \begin{cases}
\widetilde{e}_i & \text{if } a_n \leq \widetilde{e}_i \leq b_n \\
\widetilde{e}_i - 1, & \text{if } \widetilde{e}_i > b_n \\
\widetilde{e}_i + 1, & \text{if } \widetilde{e}_i < a_n.
\end{cases} \tag{11}
$$

Next, the cover pixel can be restored as $x_i = \widehat{x}_i + e_i$. Meanwhile, the embedded data bit can be extracted as $m = 0$ if $\widetilde{e}_i \in \{a_n, b_n\}$, or $m = 1$ if $\widetilde{e}_i \in \{a_n - 1, b_n + 1\}$.

The MHM embedding is implemented by modifying the histograms $\{h_n\}_{n \geq 0}$ according to the parameters $\{(a_n, b_n)\}_{n \geq 0}$. It can be illustrated using a mapping of (e, n), in which

- (a_n, n) is mapped to itself and $(a_n - 1, n)$ to embed one data bit,
- (b_n, n) is mapped to itself and $(b_n + 1, n)$ to embed one data bit,
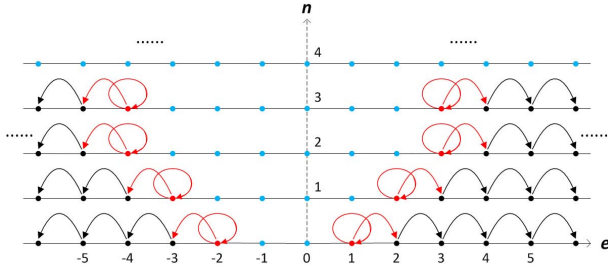- for $e < a_n$, (e, n) is mapped to $(e - 1, n)$ for shifting,

Fig. 6.   Example of MHM-based RDH embedding.

- for $e > b_n$, (e, n) is mapped to $(e + 1, n)$ for shifting,
- for $a_n < e < b_n$, (e, n) is mapped to itself meaning that it is unmodified in data embedding.

By using the mapping, one can verify that MHM includes C-PEE, A-PEE, O-PEE and AO-PEE as special cases for the following selections of $\{(a_n, b_n)\}_{n \geq 0}$:

- C-PEE: $(a_n, b_n) = (-1, 0)$ for each $n \geq 0$,
- A-PEE: $(a_n, b_n) = (-1, 0)$ if $0 \leq n < T$, and $(a_n, b_n) = (-\infty, \infty)$ if $n \geq T$,
- O-PEE: $(a_n, b_n) = (a, b)$ for each $n \geq 0$,
- AO-PEE: $(a_n, b_n) = (a, b)$ if $0 \leq n < T$, and $(a_n, b_n) = (-\infty, \infty)$ if $n \geq T$.

The corresponding mappings are shown in Fig. 5. In this situation, compared with these existing PEE-based works, better performance can be expected if taking suitable $\{(a_n, b_n)\}_{n \geq 0}$ in MHM. We give here an example. Referring to Fig. 6, we take $(a_0, b_0) = (-2, 1)$, $(a_1, b_1) = (-3, 2)$, $(a_2, b_2) = (a_3, b_3) = (-4, 3)$, and $(a_n, b_n) = (-\infty, \infty)$ for $n > 3$. With this selection and by using (7) and (8) for prediction and complexity computation, one can get a PSNR of 55.64 dB for the image Lena with a capacity of 20,000 bits. This result is better than those of C-PEE, A-PEE, O-PEE and AO-PEE listed in Table I. Through this example, one can see that a suitable choice of parameters may provide better performance in MHM. Then, the problem is, how to determine the optimal parameters in MHM to minimize the embedding distortion? We give a solution to this problem in the following subsection.

B. Determination of Optimal Parameters

As we have mentioned in Section II, the PEE-based embedding is conducted in a pixel-wise manner according to the scanning order, and the embedding procedure will stop once all data bits are embedded. The situation is the same for MHM, i.e., only the cover pixels $(x_1, \ldots, x_{N_{\text{end}}})$ need to be processed where $N_{\text{end}} \leq N$ is the smallest index such that the payload can be embedded into the first N_{end} cover pixels. Then, to estimate the embedding distortion, we first define g_n as

$$g_n(e) = \#\{1 \leq i \leq N_{\text{end}} : e_i = e, \ n_i = n\}. \quad (12)$$

Unlike h_n, g_n counts the frequencies of prediction-errors only for the pixels which will be processed in data embedding. By this definition, the expected value of embedding distortion

can be estimated as

$$ED \triangleq \mathbf{E}(\|\widetilde{I} - I\|_2^2) = \sum_{i=1}^{N_{\text{end}}} \mathbf{E}((\widetilde{x}_i - x_i)^2) = \sum_{n \geq 0} ED_n \quad (13)$$

where

$$
\begin{aligned}
ED_n &= \sum_{1 \leq i \leq N_{\text{end}}, \ n_i = n} \mathbf{E}((\widetilde{x}_i - x_i)^2) \\
&= \frac{g_n(a_n) + g_n(b_n)}{2} + \sum_{e < a_n} g_n(e) + \sum_{e > b_n} g_n(e). \quad (14)
\end{aligned}
$$

Since g_n counts the frequencies of prediction-errors only for the first N_{end} pixels, a reasonable assumption is that, for each (e, n)

$$g_n(e) \approx \frac{N_{\text{end}}}{N} h_n(e). \quad (15)$$

It yields

$$ED_n \approx \frac{N_{\text{end}}}{N} \left(\frac{h_n(a_n) + h_n(b_n)}{2} + \sum_{e < a_n} h_n(e) + \sum_{e > b_n} h_n(e) \right). \quad (16)$$

Moreover, the embedded payload size denoted PS is just

$$PS = \sum_{n \geq 0} \left(g_n(a_n) + g_n(b_n) \right) \approx \frac{N_{\text{end}}}{N} \sum_{n \geq 0} \left(h_n(a_n) + h_n(b_n) \right). \quad (17)$$

Then, according to (13), (16) and (17), one can derive that

$$ED \approx PS \left(\frac{\sum_{n \geq 0}(\sum_{e < a_n} h_n(e) + \sum_{e > b_n} h_n(e))}{\sum_{n \geq 0}(h_n(a_n) + h_n(b_n))} + \frac{1}{2} \right). \quad (18)$$

By (18), since PS is a constant which is the sum of EC and the length of auxiliary information (the auxiliary information will be clarified later in Section III-C). The optimal parameters determination is actually equivalent to the following minimization problem

$$
\begin{cases}
\text{minimize} \ \dfrac{\sum_{n \geq 0}(\sum_{e < a_n} h_n(e) + \sum_{e > b_n} h_n(e))}{\sum_{n \geq 0}(h_n(a_n) + h_n(b_n))}, \\
\text{subject to} \ \sum_{n \geq 0}(h_n(a_n) + h_n(b_n)) \geq PS.
\end{cases} \quad (19)
$$

In summary, for a given EC, after establishing h_n for each $n \geq 0$, the optimal parameters $\{(a_n, b_n)\}_{n \geq 0}$ can be determined according to (19).

C. Implementation Details

1) Rhombus Prediction and Double-Layered Embedding: Previous RDH works have shown that the full-enclosing-based prediction is better in accuracy than half-enclosing-based prediction. Thus, we adopt the rhombus prediction and double-layered embedding in our implementation. Specifically, the equilateral parallelogram pattern in Sachnev *et al.*'s work [40] is employed in our method for prediction (see Fig. 7 for an illustration), in which the cover image is divided into two sets denoted as "shadow" and "blank". One half of the secret message will be embedded into shadow pixels and the rest half will be embedded into
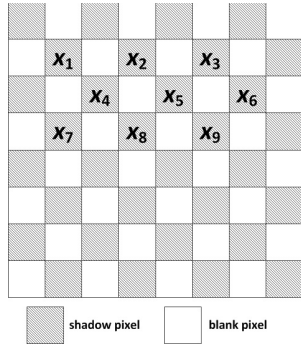
Fig. 7.   Shadow and blank pixels partition. The scanning order for shadow (blank) pixels is from left to right and top to bottom.
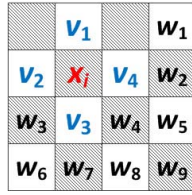

Fig. 8.   The four pixels $\{v_1, v_2, v_3, v_4\}$ are used in rhombus prediction and the twelve pixels $\{v_2, v_3, v_4, w_1, \ldots, w_9\}$ are used as a context of x_i to compute the complexity measurement.

blank pixels. In this case, twice embedding need to be processed to cover the whole image, and the prediction of blank pixels is processed only after the embedding of shadow pixels is completed. And, in the decoding phase, we first extract the embedded message and realize image recovery for blank pixels, and then, extract the embedded message and realize image recovery for shadow pixels. Since the two layers' embedding are processed similarly, we only take the shadow layer for illustration.

Referring to Fig. 7, except the pixels located in borders, the shadow pixels are scanned from left to right and top to bottom to derive the cover sequence $(x_1, \ldots, x_N)$. Here, to avoid the overflow and underflow, the pixels valued 0 will be changed to 1, and the pixels valued 255 will be changed to 254. Meanwhile, a location map will be established to record these problematic locations. The location map is a binary sequence sized N and it will be losslessly compressed to reduce its size. In our implementation, arithmetic coding is used for lossless compression. The compressed location map and its size are denoted as CLM and S_{CLM}, respectively. As a result, each x_i is ranged from 1 to 254, and it can be freely increased or decreased by 1 without overflow or underflow. Then, each x_i is predicted using its four blank neighbors (see Fig. 8 for an illustration) to determine $\widehat{x_i}$ as

$$\widehat{x_i} = \left\lfloor \frac{v_1 + v_2 + v_3 + v_4}{4} \right\rfloor. \tag{20}$$

2) *Complexity Measurement:* For the complexity measurement n_i, it is computed based on a pixel context as shown in Fig. 8. Specifically, it is defined as the sum of both vertical and horizontal absolute differences of every two consecutive pixels in the context of x_i consisting of 12 pixels
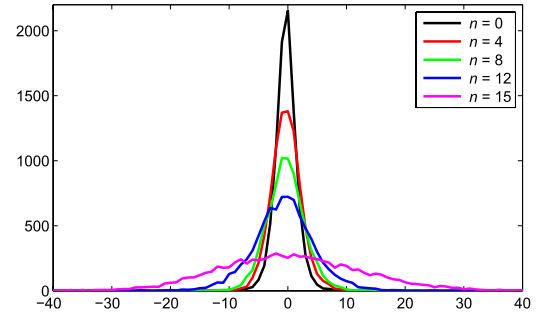

Fig. 9.   Histograms h_n for $n \in \{0, 4, 8, 12, 15\}$, for the image Lena.

$\{v_2, v_3, v_4, w_1, \ldots, w_9\}$, i.e., n_i is computed as

$$\begin{aligned}
n_i = \ & |v_2 - w_3| + |w_3 - w_6| + |v_3 - w_7| + |v_4 - w_4| \\
& + |w_4 - w_8| + |w_1 - w_2| + |w_2 - w_5| + |w_5 - w_9| \\
& + |v_4 - w_2| + |w_3 - v_3| + |v_3 - w_4| + |w_4 - w_5| \\
& + |w_6 - w_7| + |w_7 - w_8| + |w_8 - w_9|. \tag{21}
\end{aligned}$$

Moreover, to reduce the number of histograms and simplify the minimization problem (19), n_i will be scaled to M values for a relatively small M. That is, we first select $M-1$ thresholds $s_0, \ldots, s_{M-2}$ as

$$\begin{aligned}
s_j = \ & \arg\min_n \left\{ \frac{\#\{1 \le i \le N : n_i \le n\}}{N} \ge \frac{j+1}{M} \right\}, \\
& \forall \, j \in \{0, \ldots, M-2\}. \tag{22}
\end{aligned}$$

Then, we get M intervals $V_0 = [0, s_0]$, $V_1 = [s_0 + 1, s_1]$, …, $V_{M-2} = [s_{M-3} + 1, s_{M-2}]$, and $V_{M-1} = [s_{M-2} + 1, \infty)$, which form a partition for the range of n_i. Finally, n_i is updated as j if its original value is contained in V_j. In this way, after scaling, n_i takes values from 0 to $M-1$, and only M histograms $\{h_0, \ldots, h_{M-1}\}$ are established using (9). The complexity partition here is to guarantee that the number of pixels for each complexity level is approximately the same and every h_n counts sufficient pixels. In our implementation, M is empirically taken as 16. As an example, the histograms h_n for $n \in \{0, 4, 8, 12, 15\}$, for the image Lena, are shown in Fig. 9. For this figure, the shadow pixels and rhombus prediction are utilized to generate the histograms. One can see from that, as expected, the histogram with a smaller complexity level has a higher peak with more rapid two-sided decay.

Finally, we remark that, the pixel scanning order of our extraction procedure is inverse to that of embedding. Thus, when processing a pixel in data extraction, the pixels in its context have already been recovered. In this way, the same complexity measurement can be obtained by decoder.

3) *Simplified Parameters Determination:* For (19), we utilize exhaustive search to determine the $2M$ parameters $\{(a_n, b_n)\}_{0 \le n \le M-1}$. To reduce the time cost, the following conditions are imposed on our implementation,

- for each $n \in \{0, \ldots, M-1\}$, $a_n = -b_n - 1$,
- for each $n \in \{0, \ldots, M-1\}$, $b_n \in \{0, 1, 2, 3, 4, 5, 6, 7, \infty\}$,
- $b_0 \le b_1 \le \cdots \le b_{M-1}$.

The first condition is heuristic such that the expansion bins are symmetrically selected in each h_n. The second condition is

based on the fact that h_n has rapid two-sided decay especially for small n. Thus only some informative bins are exploited for expansion. For the third condition, our idea is that, we try to embed more data into h_n with smaller n. By this condition, the data will be priorly embedded into smooth pixels. Particularly, when $b_n = \infty$, we have $b_{n'} = \infty$ for $n' > n$, which means that, every pixel with $n_i \geq n$ is considered as a noisy one and ignored in data embedding. With these restrictions, the parameters $\{(a_n, b_n)\}_{0 \leq n \leq M-1}$ can be efficiently determined and the time cost is less than one second for a single layered embedding of a 512×512 sized image. Here, our method is implemented by C++ and run on a PC.

4) Auxiliary Information: In RDH, some auxiliary information should be embedded into the cover image as a part of payload for blind data extraction and image restoration. For each layer embedding of our method, the necessary auxiliary information includes

- the parameters b_n for $0 \leq n \leq M-1$ ($4M$ bits),
- the thresholds s_n for $0 \leq n \leq M-2$ ($10(M-1)$ bits),
- the index N_{end} ($\lceil \log_2 N \rceil$ bits),
- the parameter S_{CLM} ($\lceil \log_2 N \rceil$ bits),
- the compressed location map CLM (S_{CLM} bits).

Here, $\lceil \cdot \rceil$ is the ceiling function. Thus, the auxiliary information size is

$$S_{\text{aux}} \triangleq 14M + 2\lceil \log_2 N \rceil + S_{\text{CLM}} - 10. \quad (23)$$

For example, for a 512×512 sized image with $M = 16$, we have $S_{\text{aux}} = 248 + S_{\text{CLM}}$. The auxiliary information only occupies a very small portion of the payload especially for the large capacity cases and those without overflow and underflow.

5) Data Embedding: We now introduce the data embedding procedure for the cover sequence $(x_1, \ldots, x_N)$. We will successively embed the secret message and the auxiliary information into it. First, determine $(e_1, \ldots, e_N)$, $(n_1, \ldots, n_N)$, $\{h_n\}_{0 \leq n \leq M-1}$, $\{(a_n, b_n)\}_{0 \leq n \leq M-1}$, and the auxiliary information. Next, using MHM embedding, embed the secret message into the first N' pixels to derive $(\tilde{x}_1, \ldots, \tilde{x}_{N'}, x_{N'+1}, \ldots, x_N)$. Here, $x_{N'}$ is the last processed pixel for the secret message embedding. Then, record LSBs of the first S_{aux} pixels to obtain a binary sequence S_{LSB}, and embed this sequence into the unprocessed pixels to derive $(\tilde{x}_1, \ldots, \tilde{x}_{N_{\text{end}}}, x_{N_{\text{end}}+1}, \ldots, x_N)$. More specifically, also using MHM embedding, embed S_{LSB} into $(x_{N'+1}, \ldots, x_{N_{\text{end}}})$ where $x_{N_{\text{end}}}$ is the last processed pixel. Finally, by using LSB replacement, embed the auxiliary information into the first S_{aux} pixels to generate the marked image.

6) Data Extraction and Image Restoration: As opposite to embedding, we first extract the auxiliary information, then extract the secret message. Suppose that the marked sequence is $(\tilde{x}_1, \ldots, \tilde{x}_N)$. First, determine the auxiliary information by reading LSBs of the first S_{aux} pixels. Next, in the reverse scanning order, extract the sequence S_{LSB} from $(\tilde{x}_{N'+1}, \ldots, \tilde{x}_{N_{\text{end}}})$ according to MHM extraction method, and meanwhile realize restoration for these pixels. Then, replace LSBs of the first S_{aux} pixels by S_{LSB}, and, also in the reverse scanning order, extract the embedded message from the first N' pixels and meanwhile realize restoration for these pixels. Notice that, the pixels with index $i > N_{\text{end}}$ can be restored
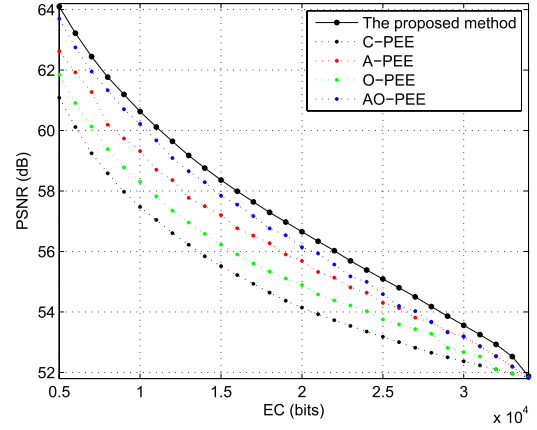

Fig. 10. Performance comparison among the proposed method, C-PEE, A-PEE, O-PEE and AO-PEE, for the shadow pixels of image Lena. For all these methods, (20) is used for prediction, and the complexity measurement is computed according to the procedure described in Section III-C2.

as themselves since they are unmodified in data embedding. Now, the embedded message is extracted and the sequence $(x_1, \ldots, x_N)$ is recovered. After that, determine the overflow and underflow locations by decompressing CLM. Finally, for each overflow or underflow pixel x_i, update its value as 255 if $x_i = 254$, or 0 if $x_i = 1$.

IV. EXPERIMENTAL RESULTS

In this section, several experiments are conducted to demonstrate the performance of the proposed method. Six standard 512×512 sized gray-scale images including Lena, Baboon, Airplane, Elaine, Lake and Boat are used in our experiments. All these images are downloaded from the USC-SIPI database.[1]

First, in Fig. 10, by using the same prediction and complexity measurement described in Section III-C1 and III-C2 for all tested methods, the embedding results of the proposed method, C-PEE, A-PEE, O-PEE and AO-PEE are plotted, for the shadow pixels of image Lena (i.e., only the first layer embedding is implemented). According to this figure, one can see that better performance is achieved comparing our method with the previous PEE-based techniques. Since the same prediction and complexity are employed, we believe that our advantage mainly lies in the MHM-based embedding strategy. Moreover, for a better illustration, the parameters b_n for $0 \leq n \leq M-1$ of our method are listed in Table II. Here, only the parameters b_n are listed since a_n are fixed as $-(b_n + 1)$ in our implementation.

Then, the proposed method is implemented according to Section III-C and it is compared with seven state-of-the-art methods of Hong *et al.* [29], Wang *et al.* [44], Sachnev *et al.* [40], Hwang *et al.* [55], Li *et al.* [48], Ou *et al.* [49] and Coatrieux *et al.* [35]. The performance comparison is shown in Fig. 11. For our method, we vary EC from 5,000 bits to its maximum with a step size of 1,000 bits. Moreover, for a fixed capacity of 10,000 bits, the resulting PSNRs of these methods are listed in Table III. In addition, the parameters b_n ($0 \leq n \leq M-1$), s_n ($0 \leq n \leq M-2$)

[1] http://sipi.usc.edu/database/database.php?volume=misc

TABLE II

PARAMETERS b_n FOR $0 \le n \le M - 1$ OF OUR METHOD WITH $M = 16$, FOR THE SHADOW PIXELS OF IMAGE LENA WITH DIFFERENT CAPACITIES

EC (in bits)	b_0	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}	b_{11}	b_{12}	b_{13}	b_{14}	b_{15}
5,000	1	2	3	4	7	7	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
10,000	1	2	2	3	3	3	3	4	7	7	∞	∞	∞	∞	∞	∞
20,000	0	1	1	1	1	2	2	3	3	3	5	7	∞	∞	∞	∞
30,000	0	0	0	0	0	0	0	1	1	1	2	3	4	4	∞	∞
34,000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3
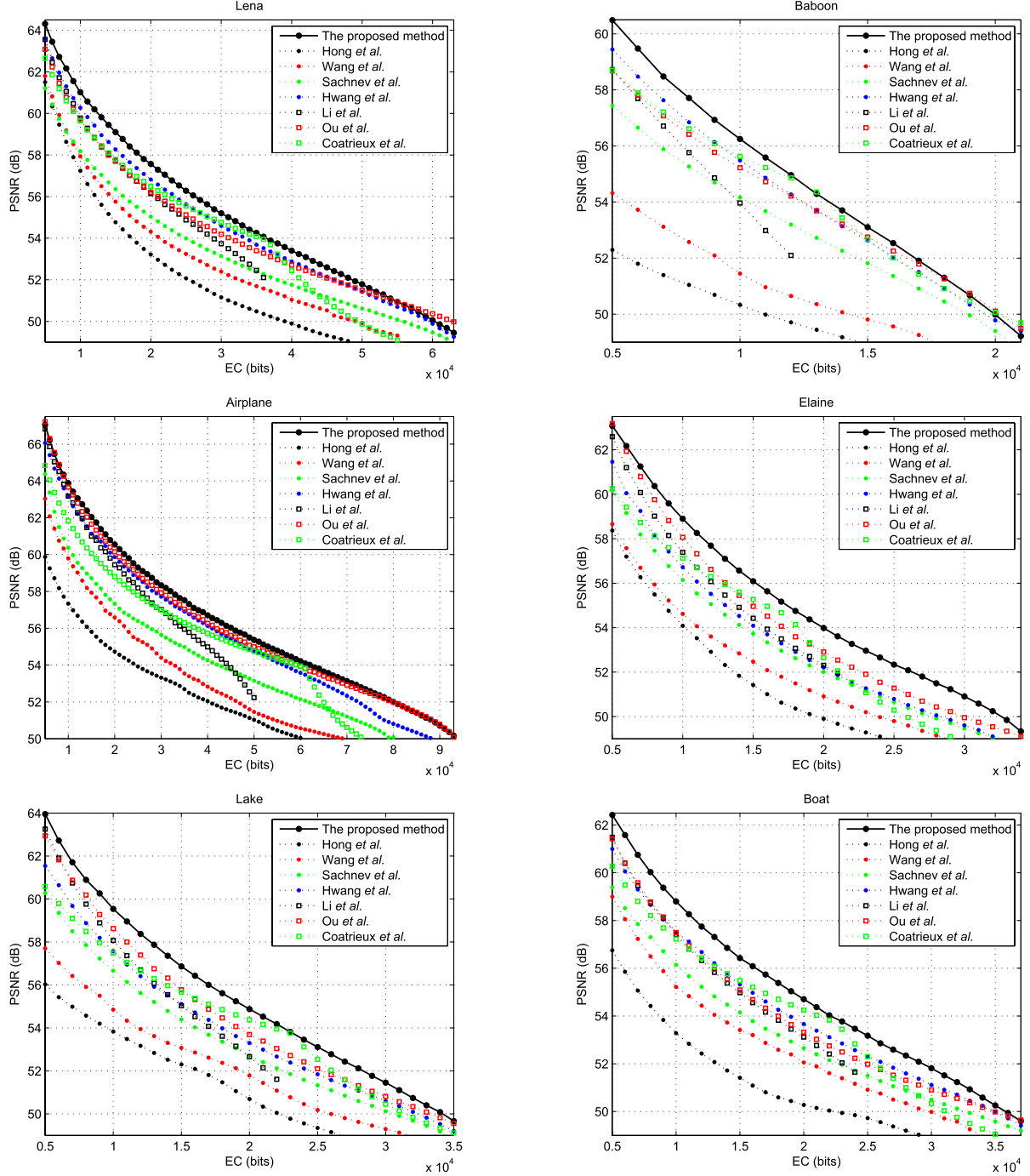
Fig. 11. Performance comparison between the proposed method and seven methods of Hong *et al.* [29], Wang *et al.* [44], Sachnev *et al.* [40], Hwang *et al.* [55], Li *et al.* [48], Ou *et al.* [49] and Coatrieux *et al.* [35].

and N_{end} of our method, for the shadow (first layer embedding) and blank (second layer embedding) pixels of image Lena with three different capacities (low, medium and high), are listed in Table IV for a better illustration. Notice that, the pixel values of image Lena range in [25, 245], and thus there is no overflow or underflow in this image.

TABLE III
COMPARISON OF PSNR (IN dB) BETWEEN THE PROPOSED METHOD AND THE METHODS OF HONG *et al.* [29], WANG *et al.* [44], SACHNEV *et al.* [40], HWANG *et al.* [55], LI *et al.* [48], OU *et al.* [49] AND COATRIEUX *et al.* [35], FOR A CAPACITY OF 10,000 BITS

Image	[29]	[44]	[40]	[55]	[48]	[49]	[35]	Proposed
Lena	57.24	57.94	58.19	60.28	59.78	59.75	59.64	61.02
Baboon	50.33	51.44	54.16	55.49	53.96	55.21	55.63	56.25
Airplane	57.34	59.79	60.38	63.19	63.18	63.76	61.83	63.88
Elaine	54.09	54.62	56.14	56.72	57.39	58.06	57.13	58.91
Lake	53.84	54.85	56.66	57.57	58.08	58.72	57.48	59.55
Boat	53.28	55.22	56.15	57.49	57.42	57.55	57.23	58.81
Average	**54.35**	**55.64**	**56.95**	**58.46**	**58.30**	**58.84**	**58.16**	**59.74**

TABLE IV
PARAMETERS b_n ($0 \leq n \leq M-1$), s_n ($0 \leq n \leq M-2$) AND N_{end} OF OUR METHOD WITH $M = 16$, FOR THE SHADOW (FIRST LAYER EMBEDDING) AND BLANK (SECOND LAYER EMBEDDING) PIXELS OF IMAGE LENA WITH THREE DIFFERENT CAPACITIES

	EC = 5,000 bits shadow	blank	EC = 30,000 bits shadow	blank	EC = 63,000 bits shadow	blank
b_0	2	2	0	1	0	0
b_1	3	3	1	1	0	0
b_2	3	4	2	1	0	0
b_3	7	5	2	2	0	0
b_4	7	5	2	2	0	0
b_5	∞	∞	2	3	0	0
b_6	∞	∞	3	3	0	0
b_7	∞	∞	4	3	0	0
b_8	∞	∞	5	4	1	0
b_9	∞	∞	5	4	1	0
b_{10}	∞	∞	6	6	1	0
b_{11}	∞	∞	∞	7	2	0
b_{12}	∞	∞	∞	∞	3	0
b_{13}	∞	∞	∞	∞	4	1
b_{14}	∞	∞	∞	∞	7	1
b_{15}	∞	∞	∞	∞	∞	2
s_0	29	30	29	35	29	39
s_1	35	37	35	41	35	45
s_2	40	41	40	45	40	50
s_3	45	46	45	49	45	55
s_4	49	50	49	54	49	59
s_5	54	55	54	59	54	64
s_6	60	60	60	64	60	69
s_7	66	66	66	70	66	75
s_8	74	74	74	77	74	83
s_9	84	84	84	87	84	92
s_{10}	99	99	99	101	99	106
s_{11}	119	119	119	120	119	126
s_{12}	150	150	150	151	150	156
s_{13}	202	203	202	203	202	208
s_{14}	292	293	292	293	292	296
N_{end}	122055	120814	128112	127971	129161	129530

For the typical C-PEE based method [29] and O-PEE based method [44], MED and GAP are used for prediction in [29] and [44], respectively. In general, our superiority over the two methods is significant. According to Table III, our method can improve them by increasing PSNR by 3-6 dB.

Sachnev *et al.*'s method [40] is based on A-PEE. The same as our method, rhombus prediction and double-layered embedding are employed in [40]. Moreover, a sorting technique is used in [40] to record prediction-errors based on the local variance, and a pixel will be priorly embedded if its local variance is small. Hwang *et al.*'s method [55] is an extension of [40], in which the expansion bins are dynamically selected to optimize the embedding performance. In this case, [55] can be viewed as an example of AO-PEE. Referring to Fig. 11, compared with [40] and [55], our method can provide a larger PSNR whatever the image or EC is. According to Table III, our method improves [40] and [55] by increasing PSNR by 2.79 and 1.28 dB in average, respectively. We then conclude that our method outperforms A-PEE [40] and AO-PEE [55].

Li *et al.*'s method [48] and Ou *et al.*'s method [49] are based on two-dimensional PEH modification. For [48], by considering each pixel pair and its context, a sequence consisting of pairs of difference values is computed, and data embedding is realized by modifying the resulting two-dimensional difference-histogram. For [49], every two adjacent prediction-errors are counted to generate a two-dimensional PEH, and a pairwise modification mechanism is utilized to embed data. In general, the two-dimensional PEH based methods perform better than those based on one-dimensional PEH. However, for [48] and [49], the histogram modification manner is fixed and independent of image content, thus the heuristic modification mechanism may limit the embedding performance. Unlike these two methods, instead of two-dimensional PEH, our method considers a sequence of histograms and embeds data by selecting optimal expansion bins depending on image content. The experimental results show that our method is better than [48] and [49] in most cases. Only for some images with large capacities (e.g., Lena with EC larger than 55,000 bits, and Baboon with EC larger than 19,000 bits), [49] is better than ours. According to Table III, for a capacity of 10,000 bits, our method outperforms [48] and [49] with an average increase of PSNR by 1.44 and 0.90 dB, respectively.

Coatrieux *et al.*'s method [35] is also based on PEE with a dynamic histogram shifting strategy, in which the rhombus prediction and multiple layered embedding are utilized. In this method, for a given pixel, the prediction-errors of its eight neighbors are averaged as a benchmark, and this pixel will be adaptively modified according to this benchmark. Instead of directly modifying a histogram, a distinguishing feature of this method is the pixel-wise prediction-error modification mechanism. According to the experimental verification, our method can derive a better PSNR in most cases and generally outperform this well designed method. Referring to Table III, for a capacity of 10,000 bits, our method outperforms [35] with an average increase of PSNR by 1.58 dB.

In conclusion, compared with the state-of-the-art works [29], [35], [40], [44], [48], [49], [55], the superiority of the propose method is experimentally verified. It demonstrates the effectiveness of the proposed MHM-based embedding strategy.

V. CONCLUSION

In this paper, an efficient RDH method is proposed based on PEE of multiple histograms. For each pixel, its prediction value and complexity measurement are first computed according to its context, then multiple histograms are generated by counting the prediction-errors for different complexity levels. Finally, data embedding is implemented according to the proposed embedding strategy based on multiple histograms modification. Moreover, to optimize the embedding performance, the expansion bins are adaptively selected in each generated histogram such that the distortion is minimized. Experimental results have shown that the proposed method outperforms the previous PEE-based techniques and some state-of-the-art methods by improving the marked image quality. However, one drawback of our method is the limited embedding capacity. In the future, we plan to extend the technique of multiple histograms modification to high capacity RDH to further enhance its practicability. In addition, incorporating advanced predictor into our method is also a topic worthy of investigation in the future study.

REFERENCES

[1] Y. Q. Shi, "Reversible data hiding," in *Proc. IWDW*, vol. 3304. 2004, pp. 1–12.

[2] R. Caldelli, F. Filippini, and R. Becarelli, "Reversible watermarking techniques: An overview and a classification," *EURASIP J. Inf. Security*, vol. 2010, Jun. 2010, Art. ID 134546.

[3] J. M. Barton, "Method and apparatus for embedding authentication information within digital data," U.S. Patent 5 646 997, Jul. 8, 1997.

[4] C. W. Honsinger, P. W. Jones, M. Rabbani, and J. C. Stoffel, "Lossless recovery of an original image containing embedded data," U.S. Patent 6 278 791, Aug. 21, 2001.

[5] G. Coatrieux, C. Le Guillou, J.-M. Cauvin, and C. Roux, "Reversible watermarking for knowledge digest embedding and reliability control in medical images," *IEEE Trans. Inf. Technol. Biomed.*, vol. 13, no. 2, pp. 158–165, Mar. 2009.

[6] M. Fontani *et al.*, "Reversible watermarking for image integrity verification in hierarchical PACS," in *Proc. ACM MM&SEC*, 2010, pp. 161–168.

[7] F. Battisti, M. Carli, and A. Neri, "Secure annotation for medical images based on reversible watermarking in the Integer Fibonacci–Haar transform domain," *Proc. SPIE*, vol. 7870, p. 78700G, Feb. 2011.

[8] S. Lee, C. D. Yoo, and T. Kalker, "Reversible image watermarking based on integer-to-integer wavelet transform," *IEEE Trans. Inf. Forensics Security*, vol. 2, no. 3, pp. 321–330, Sep. 2007.

[9] R. Y. M. Li, O. C. Au, C. K. M. Yuk, S.-K. Yip, and T.-W. Chan, "Enhanced image trans-coding using reversible data hiding," in *Proc. IEEE ISCAS*, May 2007, pp. 1273–1276.

[10] K. Hwang and D. Li, "Trusted cloud computing with secure resources and data coloring," *IEEE Internet Comput.*, vol. 14, no. 5, pp. 14–22, Sep./Oct. 2010.

[11] J. Fridrich, M. Goljan, and R. Du, "Invertible authentication," *Proc. SPIE*, vol. 4314, pp. 197–208, Aug. 2001.

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

[13] 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.

[14] M. U. Celik, G. Sharma, and A. M. Tekalp, "Lossless watermarking for image authentication: A new framework and an implementation," *IEEE Trans. Image Process.*, vol. 15, no. 4, pp. 1042–1049, Apr. 2006.

[15] 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.

[16] S.-K. Lee, Y.-H. Suh, and Y.-S. Ho, "Reversible image authentication based on watermarking," in *Proc. IEEE ICME*, Jul. 2006, pp. 1321–1324.

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

[18] 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.

[19] D. Coltuc and J.-M. Chassery, "Very fast watermarking by reversible contrast mapping," *IEEE Signal Process. Lett.*, vol. 14, no. 4, pp. 255–258, Apr. 2007.

[20] X. Wang, X. Li, B. Yang, and Z. Guo, "Efficient generalized integer transform for reversible watermarking," *IEEE Signal Process. Lett.*, vol. 17, no. 6, pp. 567–570, Jun. 2010.

[21] F. Peng, X. Li, and B. Yang, "Adaptive reversible data hiding scheme based on integer transform," *Signal Process.*, vol. 92, no. 1, pp. 54–62, Jan. 2012.

[22] D. Coltuc, "Low distortion transform for reversible watermarking," *IEEE Trans. Image Process.*, vol. 21, no. 1, pp. 412–417, Jan. 2012.

[23] L. Kamstra and H. J. A. M. Heijmans, "Reversible data embedding into images using wavelet techniques and sorting," *IEEE Trans. Image Process.*, vol. 14, no. 12, pp. 2082–2090, Dec. 2005.

[24] S. Weng, Y. Zhao, J.-S. Pan, and R. Ni, "Reversible watermarking based on invariability and adjustment on pixel pairs," *IEEE Signal Process. Lett.*, vol. 15, pp. 721–724, 2008.

[25] 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. 3, no. 3, pp. 456–465, Sep. 2008.

[26] 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.

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

[28] 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.

[29] 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, Nov. 2009.

[30] 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.

[31] L. Luo, Z. Chen, M. Chen, X. Zeng, and Z. Xiong, "Reversible image watermarking using interpolation technique," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 1, pp. 187–193, Mar. 2010.

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

[33] X. Gao, L. An, Y. Yuan, D. Tao, and X. Li, "Lossless data embedding using generalized statistical quantity histogram," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 8, pp. 1061–1070, Aug. 2011.

[34] H.-T. Wu and J. Huang, "Reversible image watermarking on prediction errors by efficient histogram modification," *Signal Process.*, vol. 92, no. 12, pp. 3000–3009, Dec. 2012.

[35] G. Coatrieux, W. Pan, N. Cuppens-Boulahia, F. Cuppens, and C. Roux, "Reversible watermarking based on invariant image classification and dynamic histogram shifting," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 1, pp. 111–120, Jan. 2013.

[36] S.-J. Lin and W.-H. Chung, "The scalar scheme for reversible information-embedding in gray-scale signals: Capacity evaluation and code constructions," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 4, pp. 1155–1167, Aug. 2012.

[37] W. Zhang, B. Chen, and N. Yu, "Improving various reversible data hiding schemes via optimal codes for binary covers," *IEEE Trans. Image Process.*, vol. 21, no. 6, pp. 2991–3003, Jun. 2012.

[38] W. Zhang, X. Hu, X. Li, and N. Yu, "Recursive histogram modification: Establishing equivalency between reversible data hiding and lossless data compression," *IEEE Trans. Image Process.*, vol. 22, no. 7, pp. 2775–2785, Jul. 2013.

[39] X. Zhang, "Reversible data hiding with optimal value transfer," *IEEE Trans. Multimedia*, vol. 15, no. 2, pp. 316–325, Feb. 2013.

[40] V. Sachnev, H. J. Kim, J. Nam, S. Suresh, and Y. Q. Shi, "Reversible watermarking algorithm using sorting and prediction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 7, pp. 989–999, Jul. 2009.

[41] W. Hong, "An efficient prediction-and-shifting embedding technique for high quality reversible data hiding," *EURASIP J. Appl. Signal Process.*, vol. 2010, May 2010, Art. ID 104835.

[42] 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.

[43] W. Hong, "Adaptive reversible data hiding method based on error energy control and histogram shifting," *Opt. Commun.*, vol. 285, no. 2, pp. 101–108, Jan. 2012.

[44] C. Wang, X. Li, and B. Yang, "Efficient reversible image watermarking by using dynamical prediction-error expansion," in *Proc. IEEE ICIP*, Sep. 2010, pp. 3673–3676.

[45] J. Wang and J. Ni, "A GA optimization approach to HS based multiple reversible data hiding," in *Proc. IEEE WIFS*, Nov. 2013, pp. 203–208.

[46] I. Caciula and D. Coltuc, "Improved control for low bit-rate reversible watermarking," in *Proc. IEEE ICASSP*, May 2014, pp. 7425–7429.

[47] L. Dong, J. Zhou, Y. Y. Tang, and X. Liu, "Estimation of capacity parameters for dynamic histogram shifting (DHS)-based reversible image watermarking," in *Proc. IEEE ICME*, Jul. 2014, pp. 1–6.

[48] 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.

[49] B. Ou, X. Li, Y. Zhao, R. Ni, and Y.-Q. Shi, "Pairwise prediction-error expansion for efficient reversible data hiding," *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 5010–5021, Dec. 2013.

[50] C. Qin, C.-C. Chang, Y.-H. Huang, and L.-T. Liao, "An inpainting-assisted reversible steganographic scheme using a histogram shifting mechanism," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 7, pp. 1109–1118, Jul. 2013.

[51] B. Ou, X. Li, Y. Zhao, and R. Ni, "Reversible data hiding based on PDE predictor," *J. Syst. Softw.*, vol. 86, no. 10, pp. 2700–2709, Oct. 2013.

[52] T.-C. Lu, C.-Y. Tseng, and K.-M. Deng, "Reversible data hiding using local edge sensing prediction methods and adaptive thresholds," *Signal Process.*, vol. 104, pp. 152–166, Nov. 2014.

[53] I.-C. Dragoi and D. Coltuc, "Local-prediction-based difference expansion reversible watermarking," *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1779–1790, Apr. 2014.

[54] I.-C. Dragoi, D. Coltuc, and I. Caciula, "Gradient based prediction for reversible watermarking by difference expansio," in *Proc. ACM IH&MMSec*, 2014, pp. 35–40.

[55] H. J. Hwang, H. J. Kim, V. Sachnev, and S. H. Joo, "Reversible watermarking method using optimal histogram pair shifting based on prediction and sorting," *KSII Trans. Internet Inf. Syst.*, vol. 4, no. 4, pp. 655–670, Aug. 2010.

[56] G. Xuan, X. Tong, J. Teng, X. Zhang, and Y. Q. Shi, "Optimal histogram-pair and prediction-error based image reversible data hiding," in *Proc. IWDW*, vol. 7809. 2012, pp. 368–383.
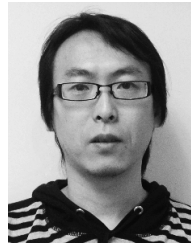
Xiaolong Li received the B.S. degree from Peking University, Beijing, China, in 1999, the M.S. degree from Ecole Polytechnique, Palaiseau, France, in 2002, and the Ph.D. degree in mathematics from the ENS de Cachan, Cachan, France, in 2006. Before joining Peking University as a Researcher, he worked as a postdoctoral fellow at Peking University from 2007 to 2009. His research interests are image processing and information hiding.

Weiming Zhang received the M.S. and Ph.D. degrees from the Zhengzhou Information Science and Technology Institute, Zhengzhou, China, in 2002 and 2005, respectively. He is currently an Associate Professor with the School of Information Science and Technology, University of Science and Technology of China. His research interests include multimedia security, information hiding, and privacy protection.
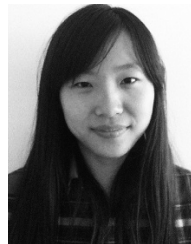
Xinlu Gui received the B.S. degree from Peking University, Beijing, China, in 2012, where she is currently pursuing the master's degree in computer science. Her research interests are image processing and information hiding.

Bin Yang received the B.S. and M.S. degrees in computer science from Peking University, Beijing, China, in 1991 and 1994, respectively. He is currently a Professor with the Institute of Computer Science and Technology, Peking University. His research interests are image processing and information hiding.