A Novel Reversible Data Hiding Scheme Based on Two-Dimensional Difference-Histogram Modification

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

***Abstract*—In this paper, based on two-dimensional difference-histogram modification, a novel reversible data hiding (RDH) scheme is proposed by using difference-pair-mapping (DPM). First, by considering each pixel-pair and its context, a sequence consisting of pairs of difference values is computed. Then, a two-dimensional difference-histogram is generated by counting the frequency of the resulting difference-pairs. Finally, reversible data embedding is implemented according to a specifically designed DPM. Here, the DPM is an injective mapping defined on difference-pairs. It is a natural extension of expansion embedding and shifting techniques used in current histogram-based RDH methods. By the proposed approach, compared with the conventional one-dimensional difference-histogram and one-dimensional prediction-error-histogram-based RDH methods, the image redundancy can be better exploited and an improved embedding performance is achieved. Moreover, a pixel-pair-selection strategy is also adopted to priorly use the pixel-pairs located in smooth image regions to embed data. This can further enhance the embedding performance. Experimental results demonstrate that the proposed scheme outperforms some state-of-the-art RDH works.**

***Index Terms*—Difference-pair-mapping (DPM), histogram modification, reversible data hiding (RDH), two-dimensional difference-histogram.**

I. Introduction

REVERSIBLE data hiding (RDH) aims to embed secret message into a cover image by slightly modifying its pixel values, and, unlike conventional data hiding, the embedded message as well as the cover image should be completely recovered from the marked content [1]–[3]. RDH is a special type of information hiding and its feasibility is mainly due to the lossless compressibility of natural images. The reversibility in RDH is quite desirable and helpful in some practical applications such as medical image processing [4], [5], multimedia archive management [6], image trans-coding [7], and video error-concealment coding [8], etc. Generally, the performance of a RDH scheme is evaluated by the capacity-distortion behavior. For a required embedding capacity (EC), to obtain a good marked image quality, one expects to reduce the embedding distortion as much as possible.

Many RDH methods have been proposed so far, e.g., the methods based on lossless compression [9]–[11], difference expansion [12]–[15], histogram modification [16]–[19], prediction-error expansion [20]–[25], and integer transform [26]–[30], etc. Among them, the histogram-based ones have attracted much attention. The histogram-based methods modify the histogram in such a way that certain bins are shifted to create vacant space while some other bins are utilized to carry data by filling the vacant space. This type of methods can well control the embedding distortion and provide a sufficient EC.

The first histogram-based RDH method is the one proposed by Ni *et al.* in [16]. This method uses peak and minimum points of the pixel-intensity-histogram to embed data. It changes each pixel value at most by 1, and thus a good marked image quality can be obtained. However, its EC is quite low and this method does not work well if the cover image has a flat histogram. To facilitate it, Lee *et al.* [17] proposed to utilize the difference-histogram instead. This novel method exploits the correlation among neighboring pixels and can embed larger payload with reduced distortion compared with Ni *et al.*'s. Moreover, we will see later (see Section III-A) that Lee *et al.*'s method can be in fact implemented, in an equivalent way, by modifying the two-dimensional pixel-intensity-histogram according to a pixel-pair-mapping (PPM) which is an injective mapping defined on pixel-pairs. In this light, the superiority of Lee *et al.*'s method over Ni *et al.*'s is explained in another viewpoint. Afterwards, Fallahpour [18] introduced a method by modifying the histogram of prediction-error. Like difference-histogram, the prediction-error-histogram is also Laplacian-like and sharply distributed which guarantees an excellent embedding performance. Instead of only using the correlation of two adjacent pixels in Lee *et al.*'s method, Fallahpour's method can exploit the local correlation of a larger neighborhood, and thus can provide relatively better performance. Besides the aforementioned methods, many other works are also based on histogram by incorporating some strategies such as double-layered embedding [31]–[33], embedding-position-selection [24], [31], [34], adaptive embedding [24], context-modification [22], and optimal-bins-selection [25], [35], etc.

Manuscript received November 23, 2012; revised February 27, 2013 and April 24, 2013; accepted April 24, 2013. Date of publication May 01, 2013; date of current version May 29, 2013. The work of W. Zhang was supported by the funding of Science and Technology on Information Assurance Laboratory under Grant KJ-13-02. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. C.-C. Jay Kuo.

X. Li, X. Gui, and B. Yang are with the Institute of Computer Science and Technology, Peking University, Beijing 100871, China (e-mail: lixiaolong@pku.edu.cn; guixinlu@pku.edu.cn; yang_bin@pku.edu.cn).

W. Zhang is with the School of Information Science and Technology, University of Science and Technology of China, Hefei 230026, China (e-mail: zwmshu@gmail.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2013.2261062

We remark that, the histogram-based RDH methods generally contain two basic steps:

- (Histogram generation) First, each local image region consisting of several pixels (e.g., a pixel-pair consisting of two adjacent pixels) is projected to a one-dimensional space (e.g., difference value of a pixel-pair) to get a scalar sequence. Then, a one-dimensional histogram (e.g., difference-histogram) is generated by counting the frequency of the resulting sequence.
- (Histogram modification) Finally, embed data into the cover image by modifying the histogram. In most cases, the histogram bins with high frequencies are expanded to carry data while some others are shifted to ensure the reversibility.

In the first step, the complex local image correlation is simplified to a one-dimensional statistic. Clearly, by this simplification, the image redundancy cannot be fully exploited and it only contributes to the second step since a one-dimensional histogram is easy to deal with. Based on this consideration, instead of one-dimensional histogram used in current RDH methods and to better exploit the image redundancy, we propose in this paper a novel RDH scheme by using a two-dimensional difference-histogram.

For the proposed method, by considering a pixel-pair and its context, a local image region is projected to a two-dimensional space to obtain a sequence consisting of difference-pairs. Then, a two-dimensional difference-histogram is generated by counting the difference-pairs. Finally, reversible data embedding is implemented according to a specifically designed difference-pair-mapping (DPM). Here, the DPM is an injective mapping defined on difference-pairs, and it is a natural extension of expansion embedding and shifting techniques used in current histogram-based methods. By using the two-dimensional difference-histogram and this specific DPM, compared with the conventional one-dimensional histogram based methods, more pixels are used for carrying data while the number of shifted pixels is reduced as well, and thus an improved embedding performance is achieved. In addition, inspired by the embedding-position-selection techniques introduced in previous works [24], [31], [34], a pixel-pair-selection strategy is adopted in our method to priorly use the pixel-pairs located in smooth image regions to embed data. This may further enhance the embedding performance. Experimental results demonstrate that the proposed method outperforms some state-of-the-art works.

The rest of the paper is organized as follows. The related works are briefly introduced in Section II. Section III presents the proposed RDH scheme in details. The comparisons with the prior arts are shown in Section IV. Section V concludes this paper.

II. Related Works

To well exploit the redundancy of natural images, the difference-histogram-based RDH methods utilize pixel-pairs with small differences for expansion embedding and other pairs for shifting. These methods may control the maximum modification to each pixel value and thus the marked image quality can be well guaranteed. There are several related works providing valuable thoughts.

In 2006, Lee *et al.* [17] proposed a difference-histogram-based RDH method. The method modifies the pixel-pairs with differences 1 or -1 to carry data. Specifically, for data embedding, the difference image D is first computed for a gray-scale cover image I as

$$D(i,j) = I(i,2j+1) - I(i,2j) \tag{1}$$

where $(I(i,2j), I(i,2j+1))$ is a pixel-pair consisting of two consecutive pixels. Then the marked image I^m can be obtained as

$$I^m(i,2j+1) = \begin{cases} I(i,2j+1), & \text{if } D(i,j)=0 \\ I(i,2j+1)+b, & \text{if } D(i,j)=1 \\ I(i,2j+1)-b, & \text{if } D(i,j)=-1 \\ I(i,2j+1)+1, & \text{if} D(i,j)\geq 2 \\ I(i,2j+1)-1, & \text{if } D(i,j)\leq -2 \end{cases} \tag{2}$$

where $b \in \{0,1\}$ is a data bit to be embedded. Notice that the first pixel $I(i,2j)$ in the pair keeps unchanged in this embedding procedure, i.e., one simply takes $I^m(i,2j) = I(i,2j)$.

Accordingly, from a marked image I^m, the embedded data bit b can be extracted as

$$b = \begin{cases} 0, & \text{if } D^m(i,j) = \pm 1 \\ 1, & \text{if } D^m(i,j) = \pm 2 \end{cases} \tag{3}$$

and the original pixel value can be recovered as

$$I(i,2j+1) = \begin{cases} I^m(i,2j+1), & \text{if } D^m(i,j) \in \{0,\pm 1\} \\ I^m(i,2j+1)-1, & \text{if } D^m(i,j)\geq 2 \\ I^m(i,2j+1)+1, & \text{if } D^m(i,j)\leq -2 \end{cases} \tag{4}$$

where $D^m(i,j) = I^m(i,2j+1) - I^m(i,2j)$ is the difference value computed from the marked image.

In this method, the bins 1 and -1 are utilized for expansion embedding and other bins (except bin 0) for shifting. It outperforms some classical RDH methods such as [12] and [16].

In 2009, another difference-histogram-based RDH method was proposed by Tai *et al.* [15]. This method could markedly extend the embedding space by utilizing a binary tree structure (BTS). Particularly, unlike Lee *et al.*'s, the pixel-pairs for difference calculation in this method can be overlapped since in the data extraction procedure, the first pixel of the pair is already recovered when processing this pixel-pair. In this overlapped way and by utilizing BTS, Tai *et al.*'s method performs well and it can provide a higher EC than that of Lee *et al.*'s.

Recently, Tai *et al.*'s method is improved by Hong [34] by utilizing a dual binary tree (DBT), a better pixel predictor and an error energy estimator. Compared with Tai *et al.*'s, Hong's DBT-based method can increase EC at the same level of distortion. Moreover, in addition to DBT, median-edge-detection (MED, see also [36] and [19]) and error energy estimator are used to obtain a more concentrate histogram and to locate the embeddable pixels. With the error energy estimator, only pixels with error energy less than a threshold will be expanded or shifted, leading to a marked decrease of distortion. Experimental results demonstrate that Hong's method can

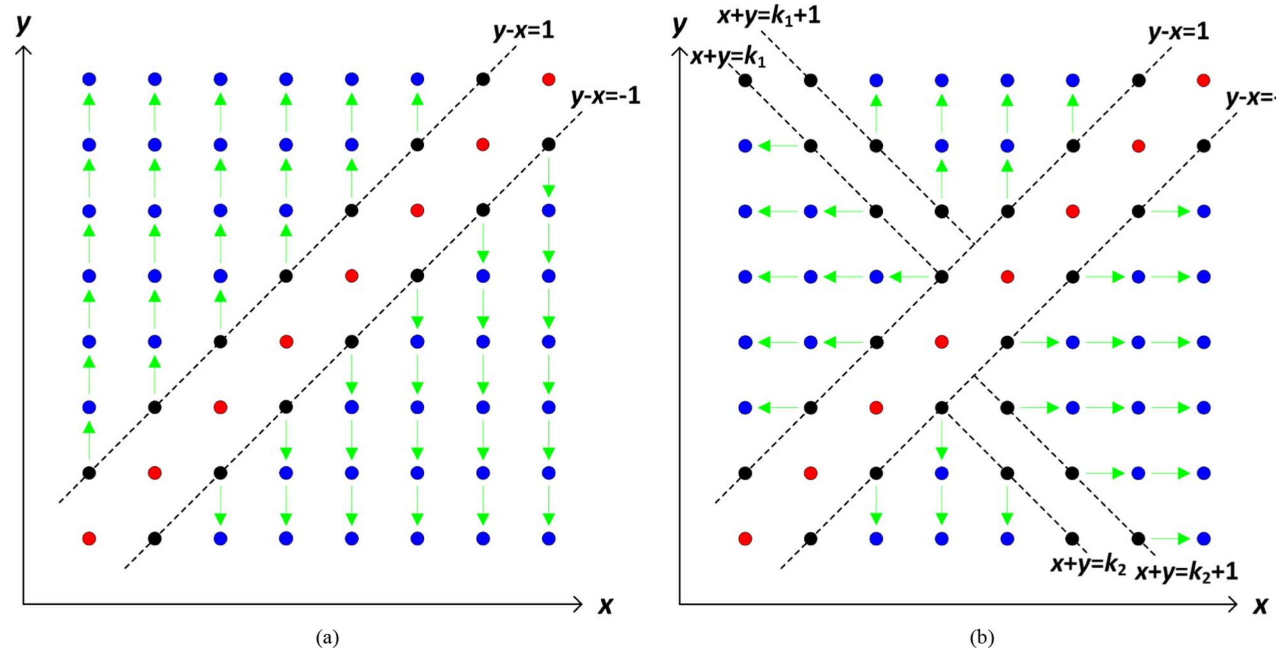


Fig. 1. PPM for illustrating the data embedding procedure for (a) Lee *et al.*'s method and (b) its improvement.

significantly improve the image quality and EC of Tai *et al.*'s and some state-of-the-art works such as [37] and [38].

It is worth mentioning that some recent works [18]–[21] can also be viewed as Tai *et al.*'s improvement. In [18]–[21], the difference value is replaced by the prediction-error computed using a larger local image region so that the redundancy can be better exploited.

Our method is motivated by the aforementioned works and will be described in details in next section.

III. Proposed RDH Scheme

Basically, the proposed RDH scheme is based on modification of two-dimensional difference-histogram by constructing a DPM which is an injective mapping defined on difference-pairs.

This Section is organized as follows. First, to better explain our idea, we introduce Lee *et al.*'s method from a PPM viewpoint in Section III-A. We show that this difference-histogram-based method can be implemented by modifying the two-dimensional pixel-intensity-histogram according to a PPM which is an injective mapping defined on pixel-pairs. Then in Section III-B, we extend the idea of PPM to DPM and present our DPM-based scheme. In this way, by extending difference-histogram (or, in an equivalent way, two-dimensional pixel-intensity-histogram) to two-dimensional difference-histogram, the proposed scheme extends current histogram-based RDH methods. The image redundancy can be better exploited by our approach and the advantage of DPM is demonstrated in Section III-C. Finally, the detailed embedding and extraction procedures of our scheme are summarized in Section III-D.

A. PPM for RDH

We point out that, in an equivalent way, Lee *et al.*'s embedding procedure can be demonstrated by a PPM shown in Fig. 1(a), in which a subset of $\mathbb{Z}^2$ is divided into two disjointed parts as black points and blue points, each black point is mapped to a blue one (indicated by a green arrow) and each blue point is mapped to another blue point. Here, each point represents the value of a pixel-pair, and the black points are used for expansion embedding while the blue ones for shifting. According to this PPM, for a cover pixel-pair (x, y), its marked value can be determined in the following way:

1) if $y - x = 0$ (i.e., (x, y) is a red point), the marked pixel-pair is taken as (x, y) itself.
2) if $y - x = 1$ or $y - x = -1$ (i.e., (x, y) is a black point)
   a) if the to-be-embedded data bit $b = 0$, the marked pixel-pair is taken as (x, y) itself.
   b) if the to-be-embedded data bit $b = 1$, the marked pixel-pair is taken as its associate blue point.
3) if $y - x > 1$ or $y - x < -1$ (i.e., (x, y) is a blue point), the marked pixel-pair is taken as its associate blue point.

The corresponding data extraction and image restoration process can also be demonstrated according to the PPM since it is an injection, i.e., each point has at most one inverse. The trivial description is omitted.

From the PPM viewpoint, Lee *et al.*'s difference-histogram-based method is actually implemented by modifying the two-dimensional pixel-intensity-histogram.

Lee *et al.*'s method only modifies the second pixel of the pair. Thus two modification directions, up and down, are allowed in data embedding. This is to say, in PPM, a point (x, y) can be either mapped to its upper neighbor $(x, y+1)$ or lower neighbor $(x, y-1)$. Actually, one can also modify the first pixel without introducing additional distortion resulting in modification directions left and right. In this way, the associate mapped point of (x, y) has four choices: $(x-1, y)$, $(x+1, y)$, $(x, y-1)$ or $(x, y+1)$ (see Fig. 2(a)). Based on these four modification directions, Lee *et al.*'s method can be improved by designing a new PPM shown in Fig. 1(b). According to this figure, one can see

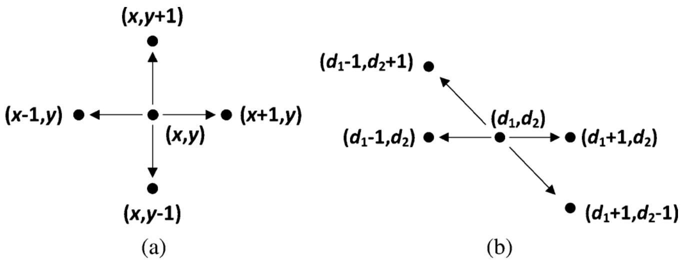


Fig. 2. (a) By modifying either x or y by 1, (x, y) has four modification directions. (b) The corresponding difference-pair (d_1, d_2) also has four modification directions, where $d_1 = x - y$, $d_2 = y - z$, and z is a prediction of y.

that more pixel-pairs (black points) are utilized for expansion embedding, and the number of shifted pixel-pairs (blue ones) is reduced as well. Here in Fig. 1(b), the parameters k_1 and k_2 can be adaptively selected by maximizing EC.

By this new PPM, more data bits can be embedded without degrading the marked image quality. Taking the 512×512 sized gray-scale Lena image for an example, compared with Lee *et al.*'s original method, EC is increased from 24,000 bits to 25,000 bits and PSNR is slightly improved from 52.12 dB to 52.14 dB. The advantage of the new PPM lies in the exploitation of more modification directions.

B. DPM for RDH

For a pixel-pair (x, y), we propose to compute two difference values $d_1 = x - y$ and $d_2 = y - z$ to form a two-dimensional difference-histogram of (d_1, d_2), where z is a prediction of y which will be clarified later. Inspired by the aforementioned new PPM, we will modify either x or y by 1. In this situation, since (x, y) has four modification directions, the difference-pair (d_1, d_2) also has four modification directions: $(d_1 - 1, d_2)$, $(d_1 + 1, d_2)$, $(d_1 + 1, d_2 - 1)$ or $(d_1 - 1, d_2 + 1)$ (see Fig. 2(b)). For example, by modifying y to $y + 1$, the modification direction to (x, y) is "up" and the corresponding modification direction to (d_1, d_2) is "upper-left", since d_1 changes to $d_1 - 1$ and d_2 changes to $d_2 + 1$. Based on these four modification directions, we will introduce a new RDH scheme by designing a DPM.

The ideas of related works play an important role in our scheme. We extend the idea of two-dimensional pixel-intensity-histogram (or, in an equivalent sense, one-dimensional difference-histogram) of Lee *et al.* to two-dimensional difference-histogram. Besides, for each (x, y), we compute the predication of y based on the context of (x, y) for an accurate estimation. Here, the gradient-adjusted-prediction (GAP) will be used in our scheme. Moreover, to further improve the marked image quality, we adopt a strategy to select smooth pixel-pairs for data embedding. The main idea of this strategy is similar to those of pixel selection of our previous work [24] and error energy estimation of Hong [34]. By pixel-pair-selection, a noisy-level is computed for each pixel-pair, and only the pixel-pairs with relatively small noisy-levels will be embedded.

We now briefly introduce our embedding procedure. First, divide the cover image into nonoverlapped pixel-pairs. For each pixel-pair (x, y), compute the prediction of y to get z using GAP predictor:

	j	$j+1$	$j+2$	$j+3$
i	x	y	v_1	v_2
$i+1$	v_3	v_4	v_5	v_6
$i+2$	v_7	v_8	v_9	v_{10}

Fig. 3. Context of (x, y), where the location of pixel x is (i, j). The blue pixels are used to compute the GAP predictor for y, and all ten pixels $\{v_1, \ldots, v_{10}\}$ are used to compute the noisy-level.

$$z = \begin{cases} v_1, & \text{if } d_v - d_h > 80 \\ \frac{(v_1+u)}{2}, & \text{if } d_v - d_h \in (32, 80] \\ \frac{(v_1+3u)}{4}, & \text{if } d_v - d_h \in (8, 32] \\ u, & \text{if } d_v - d_h \in [-8, 8] \\ \frac{(v_4+3u)}{4}, & \text{if } d_v - d_h \in [-32, 8) \\ \frac{(v_4+u)}{2}, & \text{if } d_v - d_h \in [-80, -32) \\ v_4, & \text{if } d_v - d_h < -80 \end{cases} \tag{5}$$

where $\{v_1, \ldots, v_5, v_7, v_8\}$ are neighboring pixels of (x, y) (see Fig. 3), $d_v = |v_1 - v_5| + |v_3 - v_7| + |v_4 - v_8|$ and $d_h = |v_1 - v_2| + |v_3 - v_4| + |v_4 - v_5|$ represent the vertical and horizontal gradients, and $u = (v_1 + v_4)/2 + (v_3 - v_5)/4$. Notice that z should be rounded to its nearest integer if it is not an integer. Then, compute the noisy-level of (x, y) denoted as $NL(x, y)$ using its ten neighboring pixels $\{v_1, \ldots, v_{10}\}$ (see also Fig. 3) as

$$NL(x, y) = \int_{(i', j') \in V} |\nabla I(i', j')| \tag{6}$$

where V represents the context of (x, y) containing the ten pixels and ∇ stands for the gradient operator. Here, for discrete image, the noisy-level is computed by summing both vertical and horizontal differences of every two consecutive pixels in V, and it is less than or equal to $13 \times 255 = 3315$. Clearly, a pixel-pair located in smooth regions may have a small noisy-level. Finally, for each pixel-pair with noisy-level less than a threshold T, compute the difference-pair (d_1, d_2) and implement data embedding according to the DPM shown in Fig. 4.

Here, as the same as the case of PPM, each black point in this DPM is mapped to a blue one and it will be used for carrying one data bit, while each blue point is mapped to another blue one and it is simply shifted to ensue the reversibility. Also, like the PPMs in Fig. 1, this DPM is an injection.

It should be mentioned that there are many ways to design DPM to derive different RDH schemes with different performance. Any DPM can give a RDH scheme if it is an injection. As for the DPM shown in Fig. 4, our idea is to use as much as possible the points with high frequency to carry data. So we take some points (d_1, d_2) with either d_1 or d_2 small as black to carry data, and meanwhile we shift the other points according to the four allowable modification directions. The superiority of this

TABLE I
MARKED VALUE OF COVER PIXEL-PAIR (x, y) ACCORDING TO THE DPM SHOWN IN FIG. 4 AND DIFFERENT CASES OF DIFFERENCE-PAIR (d_1, d_2), WHERE $b \in \{0, 1\}$ IS A TO-BE-EMBEDDED DATA BIT

Conditions on (d_1, d_2)	Operation in data embedding	Modification direction to the difference-pair	Modification direction to the pixel-pair	Marked value
$d_1 = 1$ and $d_2 > 0$	expansion embedding	right	right	$(x + b, y)$
$d_1 = -1$ and $d_2 < 0$	expansion embedding	left	left	$(x - b, y)$
$d_1 = 0$ and $d_2 \geq 0$	expansion embedding	upper-left	up	$(x, y + b)$
$d_1 < 0$ and $d_2 = 0$				
$d_1 = 0$ and $d_2 < 0$	expansion embedding	lower-right	down	$(x, y - b)$
$d_1 > 0$ and $d_2 = 0$				
$d_1 = 1$ and $d_2 = -1$				
$d_1 > 1$ and $d_2 > 0$	shifting	right	right	$(x + 1, y)$
$d_1 < -1$ and $d_2 < 0$	shifting	left	left	$(x - 1, y)$
$d_1 < 0$ and $d_2 > 0$	shifting	upper-left	up	$(x, y + 1)$
$d_1 > 1$ and $d_2 < 0$	shifting	lower-right	down	$(x, y - 1)$
$d_1 = 1$ and $d_2 < -1$				

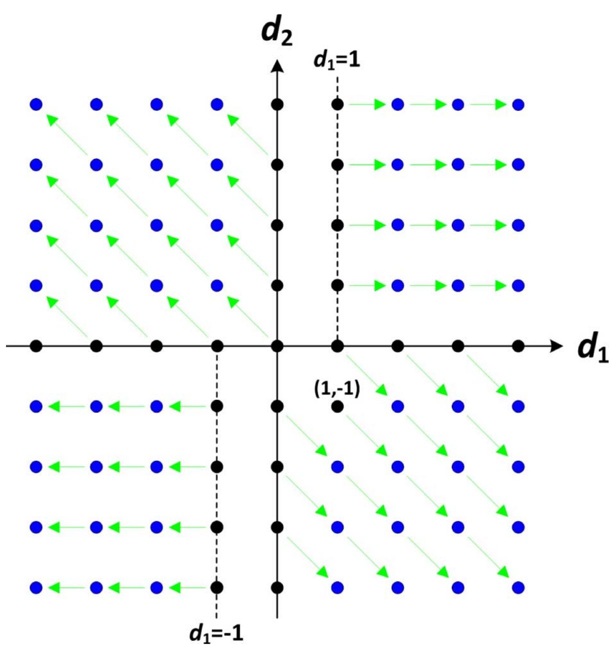


Fig. 4. DPM for illustrating the proposed data embedding procedure.

specific DPM will be experimentally demonstrated in the next subsection.

We give some explanations for the final step of our method to clarify how DPM works. For example, when $(x, y) = (132{,}132)$ and $z = 131$, the corresponding difference-pair is $(d_1, d_2) = (0, 1)$ which is a black point in Fig. 4. In this case, the pixel-pair will be expanded to carry one data bit:

1) if the to-be-embedded data bit $b = 0$, the marked pixel-pair is taken as $(x{,}y)$ itself.
2) if the to-be-embedded data bit $b = 1$, the marked difference-pair is then taken as its associate point $(-1{,}2)$ where the modification direction is "upper-left", and thus the marked pixel-pair is taken as the upper neighbor of $(x{,}y)$, i.e., $(x{,}y + 1) = (132{,}133)$.

For another example, $(x{,}y) = (132{,}131)$ and $z = 133$, the difference-pair is $(d_1, d_2) = (1, -2)$ which is blue. In this situation, the pixel-pair will be shifted. Since the modification direction to this difference-pair is "lower-right", the marked pixel-pair is then taken as the lower neighbor of $(x{,}y)$, i.e.,

TABLE II
EXTRACTED DATA BIT AND RECOVERED VALUE FROM A MARKED PIXEL-PAIR (x^m, y^m) ACCORDING TO THE DPM SHOWN IN FIG. 4 AND DIFFERENT CASES OF DIFFERENCE-PAIR (d_1^m, d_2^m)

Conditions on (d_1^m, d_2^m)	Extracted data bit b	Recovered value
$d_1^m \in \{1, 2\}$ and $d_2^m > 0$	$d_1^m - 1$	$(x^m - b, y^m)$
$d_1^m \in \{-1, -2\}$ and $d_2^m < 0$	$-1 - d_1^m$	$(x^m + b, y^m)$
($d_1^m = 0$ and $d_2^m \geq 0$) or ($d_1^m = -1$ and $d_2^m \geq 1$)	$-d_1^m$	$(x^m, y^m - b)$
($d_1^m < 0$ and $d_2^m = 0$) or ($d_1^m < -1$ and $d_2^m = 1$)	d_2^m	
($d_1^m = 0$ and $d_2^m < 0$) or ($d_1^m = 1$ and $d_2^m < -1$)	d_1^m	$(x^m, y^m + b)$
($d_1^m > 0$ and $d_2^m = 0$) or ($d_1^m > 1$ and $d_2^m = -1$)	$-d_2^m$	
($d_1^m = 1$ and $d_2^m = -1$) or ($d_1^m = 2$ and $d_2^m = -2$)	$d_1^m - 1$	
$d_1^m > 2$ and $d_2^m > 0$	no embedded data bit	$(x^m - 1, y^m)$
$d_1^m < -2$ and $d_2^m < 0$	no embedded data bit	$(x^m + 1, y^m)$
$d_1^m < -1$ and $d_2^m > 1$	no embedded data bit	$(x^m, y^m - 1)$
$d_1^m > 2$ and $d_2^m < -1$	no embedded data bit	$(x^m, y^m + 1)$
$d_1^m = 2$ and $d_2^m < -2$		

$(x{,}y - 1) = (132{,}130)$. For general cases, as a detailed description of the DPM shown in Fig. 4, the marked value is listed in Table I.

Notice that the pixel-pair scanning order in our data extraction procedure is inverse to that of embedding. By this means, when processing a pixel-pair in data extraction, its context has already been recovered. Thus the same prediction and noisy-level used by encoder can be obtained by decoder. This issue and the injectivity of DPM guarantee the reversibility of our scheme. The corresponding data extraction and pixel-pair recovery process are listed in Table II. In the table, $d_1^m = x^m - y^m$ and $d_2^m = y^m - z$ are difference values computed from the marked pixel-pair (x^m, y^m), where z is the same prediction used in data embedding by encoder.

C. Evaluation for DPM

As we have known (see Section II-C of [24] for details), for the histogram-based RDH, if the maximum modification to pixel values is 1 in data embedding, the expected value of the modification (in l^2-norm) to cover image is $N_{\text{exp}}/2 + N_{\text{shift}}$, where N_{exp} and N_{shift} are numbers of expanded and shifted pixels, respectively. Thus the ratio of expanded pixels

$$\frac{N_{\text{exp}}}{N_{\text{exp}} + N_{\text{shift}}} \tag{7}$$

is a measurement of the embedding performance. The larger the ratio is, the less modification to cover image and better performance is. We then use (7) to demonstrate the superiority of

TABLE III
COMPARISONS FOR THE RATION OF EXPANDED PIXELS BETWEEN THE PROPOSED SCHEME AND THE METHODS OF LEE *et al.* [17] (INCLUDING ITS IMPROVEMENT PRESENTED IN SECTION III-A), HONG *et al.* [19], AND FALLAHPOUR [18], FOR SIX STANDARD 512 × 512 SIZED GRAY-SCALE IMAGES

Image	Proposed scheme (for different pixel-pair-selection threshold T)							[17]	[17] (improved)	[19] (MED)	[18] (GAP)
	$T=20$	$T=40$	$T=60$	$T=80$	$T=100$	$T=200$	maximum				
Lena	0.528	0.395	0.353	0.333	0.321	0.295	0.283	0.207	0.215	0.205	0.226
Baboon	0.263	0.230	0.206	0.189	0.172	0.133	0.090	0.059	0.070	0.067	0.073
Barbara	0.452	0.407	0.377	0.359	0.344	0.300	0.237	0.176	0.184	0.164	0.181
Airplane (F-16)	0.677	0.544	0.500	0.474	0.455	0.417	0.388	0.283	0.300	0.306	0.337
Peppers	0.362	0.288	0.248	0.228	0.216	0.201	0.195	0.143	0.153	0.126	0.150
Fishing boat	0.476	0.348	0.276	0.240	0.223	0.197	0.186	0.131	0.150	0.127	0.138

the proposed DPM-based scheme. Clearly, for our method, according to Fig. 4, the ratio of expanded pixels can be formulated as

$$\frac{\sum\limits_{d_1=0 \text{ or } d_2=0} h(d_1,d_2)+\sum\limits_{d_2>0} h(1,d_2)+\sum\limits_{d_2<0} h(-1,d_2)+h(1,-1)}{\sum\limits_{(d_1,d_2)\in\mathbb{Z}^2} h(d_1,d_2)} \tag{8}$$

where $h(d_1, d_2)$ means the frequency of (d_1, d_2).

Table III shows the comparisons of the ration of expanded pixels for: our method with different pixel-pair-selection thresholds, Lee *et al.*'s method [17] and its improved one presented in Section III-A, Hong *et al.*'s MED-based method [19], and Fallahpour's GAP-based method [18]. Except Barbara, all the images used in our experiments are downloaded from the USC-SIPI database[1]. For our method, when the pixel-pair-selection threshold T is taken as its maximum, i.e., the sum of maximum noisy-level of all pixel-pairs and 1

$$1 + \max_{(x,y)} NL(x,y) \tag{9}$$

it means that the pixel-pair-selection strategy is disabled and all pixel-pairs are used for data embedding. According to this Table, one can see that our method is better than those prior arts since it always has a larger ratio. The superiority of the two-dimensional difference-histogram and this specifically designed DPM is thus verified. On the other hand, for our method, the ration becomes larger with smaller T. This demonstrates the advantage of pixel-pair-selection strategy. Finally, as a complement to Table III, Fig. 5 shows performance comparisons with the aforementioned works. Our scheme performs well and it significantly outperforms the prior arts.

The detailed data embedding and extraction procedures including the treatment of overflow/underflow will be given in the next subsection. More experimental results of comparisons with state-of-the-art works will be reported in Section IV.

D. Data Embedding and Extraction Procedures

The proposed data embedding procedure contains several basic steps. First, divide the cover image I into nonoverlapping pixel-pairs. Then, embed the secret message into a part of cover image (noted as I'). Next, record the least significant bits (LSB) of some pixels of I' (noted as I'') to get a binary sequence, and embed this sequence into the rest part of I, i.e., $I - I'$. Finally, by using LSB replacement, embed the auxiliary information and the compressed location map into I''. The detailed data embedding procedure is described as bellow step-by-step.

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

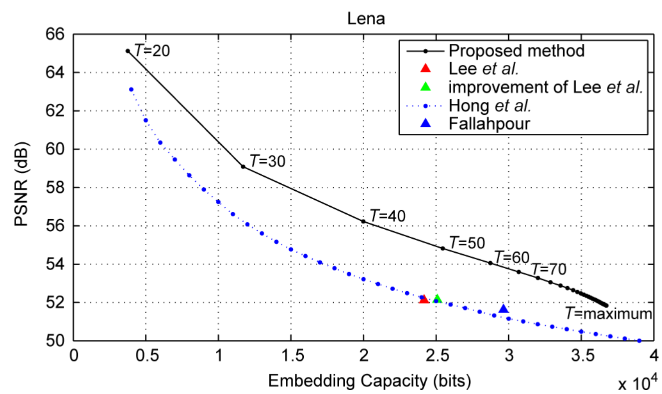


Fig. 5. Performance comparisons between the proposed method, Lee *et al.*'s method [17] and its improved one presented in Section III-A, Hong *et al.*'s MED-based method [19], and Fallahpour's GAP-based method [18], for image Lena.

Step-1: Except the last two columns and last two rows, from left to right and top to bottom, divide the cover image into k nonoverlapped pixel-pairs. Then we define a location map noted L to solve the overflow/underflow problem. Notice that the maximum modification to each pixel is 1 in our method, so we only need to deal with the pixels with extreme values 0 or 255. Specifically, for the i-th pixel-pair (x,y), if $x \in \{0,255\}$ or $y \in \{0,255\}$, we define $L(i) = 1$. Otherwise, we take $L(i) = 0$. Consequently, L is a binary sequence of length k. Then losslessly compress the location map using arithmetic coding. Denote the length of the compressed location map as l. For each pixel-pair (x, y) with $L(i) = 0$, compute the prediction of y and the noisy-level $NL(x, y)$ according to (5) and(6).

Step-2: The secret message will be embedded in this step according to the location map and a pixel-pair-selection threshold T, i.e., successively, for each $i \in \{1, \ldots, k\}$, consider the i-th pixel-pair:

1) if $L(i) = 1$, the overflow/underflow would occur and we do nothing with it.
2) if $L(i) = 0$ and its noisy-level is no less than T, the pixel-pair is considered as a rough one and we also do nothing with it.
3) otherwise, it will be shifted or expanded to carry one data bit according to Table I or the DPM shown in Fig. 4.

This step will stop if all message bits have been embedded, and we denote k^* as the index of the last data-carrying pixel-pair.

Fig. 6. Six test images. From left to right: Lena, Baboon, Barbara, Airplane (F-16), Peppers, and Fishing boat.

Step-3: The auxiliary information and the compressed location map will be embedded in this step for blind decoding. This step contains several parts:

1) First, record LSB of the first $(12 + 3\lceil\log_2 k\rceil + l)$ image pixels to obtain a binary sequence S, where $\lceil\cdot\rceil$ is the ceiling function.
2) Then, embed S into the remaining pixel-pairs (i.e., the pixel-pairs with index k^*+1 to k) using the same method in *Step-2*. After embedded, we denote $k_{\rm end} \in \{k^*+1, \ldots, k\}$ as the index of the last data-carrying pixel-pair.
3) Finally, replace LSB of the first $(12+3\lceil\log_2 k\rceil+l)$ pixels by the following auxiliary information and the compressed location map defined in *Step-1* to generate the marked image.
   a) pixel-pair-selection threshold: T (12 bits),
   b) index of the last embedded pixel-pair in *Step-2*: k^* ($\lceil\log_2 k\rceil$ bits),
   c) length of the compressed location map: l ($\lceil\log_2 k\rceil$ bits),
   d) index of the last embedded pixel-pair: $k_{\rm end}$ ($\lceil\log_2 k\rceil$ bits).

We remark that, in the above embedding procedure, to priorly use smooth pixel-pairs, the threshold T is taken as the smallest positive integer such that the message can be embedded. This parameter can be determined iteratively.

The corresponding data extraction and image restoration process are summarized as follows.

Step-1: Read LSB of the first $(12 + 3\lceil\log_2 k\rceil)$ pixels of marked image to determine the values of T, k^*, l and $k_{\rm end}$. Then, read the next l LSB to determine the compressed location map. Finally, generate the location map L by decompressing the compressed location map.

Step-2: In this step, we extract the LSB sequence S defined in 1) of Step-3 of data embedding. First, as the same as data embedding, except the last two columns and last two rows, divide the marked image into k nonoverlapped pixel-pairs. Then, in a reverse order, from the $k_{\rm end}$-th to (k^*+1)-th pixel-pair, extract the sequence S and restore the cover image as follows. Suppose the index of the to-be-processed pixel-pair is i:

1) First, compute the prediction and noisy-level according to (5) and(6). Notice that the pixel-pair processing order here is inverse to that of embedding, so every pixel-pair with an index larger than i has already been restored when processing the i-th pixel-pair. In this light, the decoder can obtain the same prediction and noisy-level as used by encoder.
2) Then, if $L(i) = 0$ and the noisy-level is less than T, the data extraction and pixel-pair restoration can be realized according to Table II; otherwise, there is no data embedded and the pixel-pair is just recovered as itself.

It should be mentioned that for the j-th pixel-pair with $j > k_{\rm end}$, it is unchanged in embedding procedure, and thus the pixel-pair can be recovered as itself.

Step-3: Replace LSB of the first $(12 + 3\lceil\log_2 k\rceil + l)$ image pixels by the sequence S extracted in the step above. Then use the same method as Step-2 to extract the embedded message from the first k^* pixel-pairs, and meanwhile to realize restoration for these pixel-pairs. Finally, the embedded message is extracted and the original image is recovered.

IV. Experimental Results

Six 512×512 sized gray-scale images including Lena, Baboon, Barbara, Airplane (F-16), Peppers, and Fishing boat (see Fig. 6) are used in our experiment. Fig. 7 shows the performance comparisons between the proposed method and following state-of-the-art works: Lee *et al.* [17] (including its improved one presented in Section III-A), Fallahpour [18], Hong *et al.* [19], Sachnev *et al.* [31], Li *et al.* [24], Hong [34], and Hong [33]. For our method, we vary EC from 5,000 bits to its maximum with a step of 1,000 bits. Here, 5,000 bits of EC means an embedding rate (ER) of 0.019 bits per pixel (bpp).

According to this figure, compared with Lee *et al.*'s method and its improvement, one can see that our superiority is significant. It experimentally demonstrates that the DPM-based scheme can provide a much better performance than PPM.

Moreover, compared with the methods of Fallahpour and Hong *et al.*, our superiority is also significant. The two methods are based on the prediction-error-histogram with different predictors. Although these methods may exploit the spatial redundancy for a larger pixel context and perform better than Lee *et al.*'s in most cases, our scheme can improve them by increasing PSNR by 1–6 dB. Our advantage lies in the utilization of two-dimensional difference-histogram and pixel-pair-selection strategy.

Sachnev *et al.*'s method is based on the prediction-error-histogram incorporating with an embedding-position-selection strategy similar to ours. A sorting technique is used in this method to record prediction-errors based on the magnitude of local variance, and a pixel will be priorly embedded if it has a small local variance. This method performs well and it is superior to some typical RDH schemes such as [6], [14] and [20]. Referring to Fig. 7, one can see that in most cases, our scheme is better than this well performed method. But, for some images, when EC approaches its maximum (e.g., for Baboon when EC is larger than 9,000 bits, for Airplane when EC is larger than 45,000 bits), Sachnev *et al.*'s method can

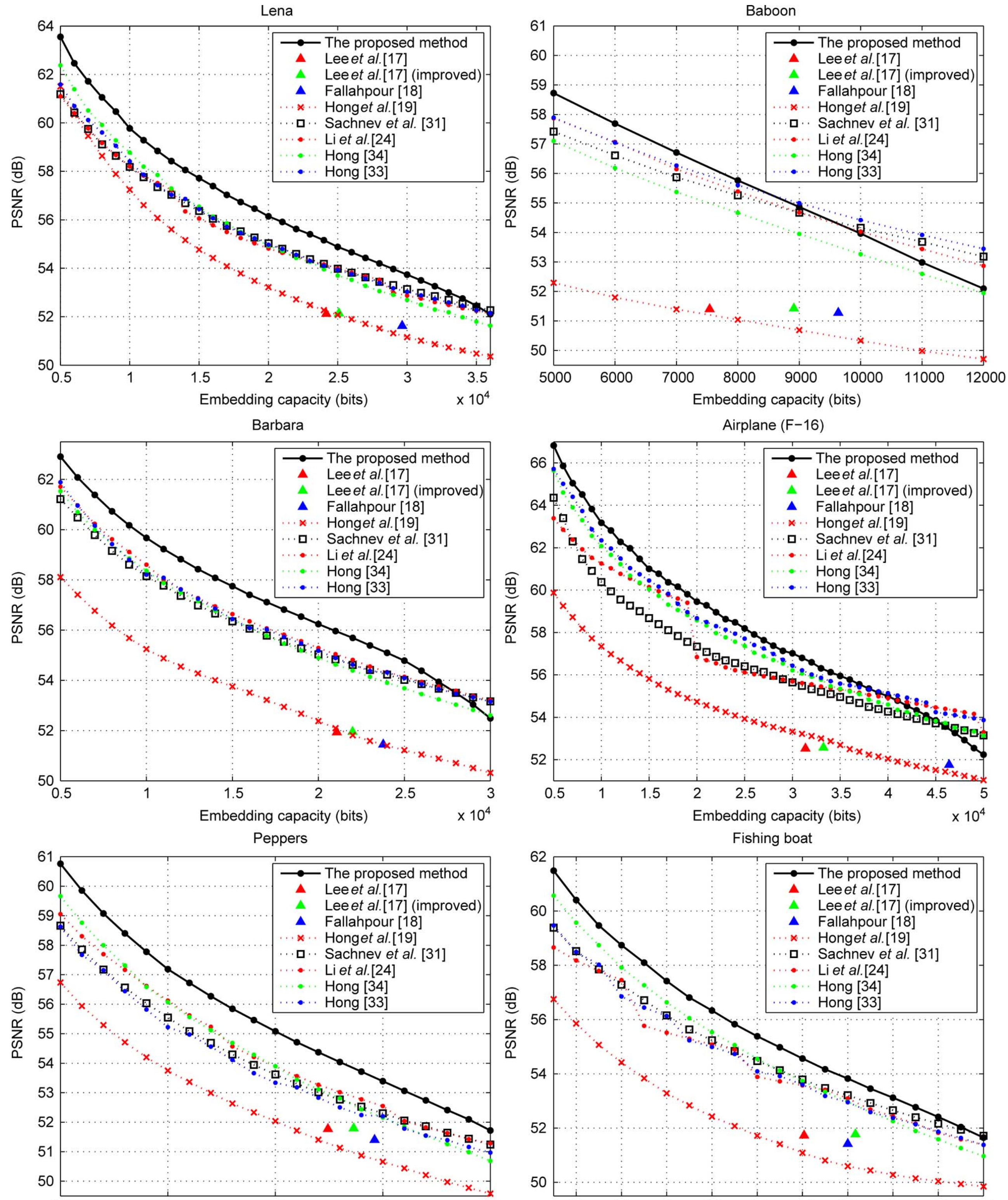


Fig. 7. Performance comparisons between our scheme and following seven methods: Lee *et al.* [17] (including its improvement presented in Section III-A), Fallahpour [18], Hong *et al.* [19], Sachnev *et al.* [31], Li *et al.* [24], Hong [34], and Hong [33].

achieve a larger PSNR. The reason is that in such cases, since EC is high, the smooth pixel-pairs are insufficient and our method should necessarily use noisy pixel-pairs to embed data which is unfavorable to the performance. On the other hand, since Sachnev *et al.*'s method modifies each pixel to embed data (recall that in our method, we modify only one pixel in a pixel-pair), its sorting technique still works well while our pixel-pair-selection strategy is no longer effective when our EC approaches its maximum. In this light, Sachnev *et al.*'s method may perform better than ours in a few cases. However, referring

TABLE IV
COMPARISONS OF PSNR (IN dB) BETWEEN OUR METHOD AND FOUR METHODS OF SACHNEV *et al.* [31], LI *et al.* [24], HONG [34], AND HONG [33], FOR AN EC OF 10,000 BITS (ER = 0.038 BPP)

Image	[31]	[24]	[34]	[33]	Proposed scheme
Lena	58.18	58.20	58.78	58.41	59.78
Baboon	54.15	54.03	53.26	54.42	53.96
Barbara	58.15	58.61	58.36	58.21	59.67
Airplane (F-16)	60.38	61.26	62.08	62.35	63.18
Peppers	55.55	56.12	56.07	55.22	57.19
Fishing boat	56.15	55.52	56.64	56.13	57.42
Average	**57.09**	**57.29**	**57.53**	**57.46**	**58.53**

TABLE V
COMPARISONS OF PSNR (IN dB) BETWEEN OUR METHOD AND FOUR METHODS OF SACHNEV *et al.* [31], LI *et al.* [24], HONG [34], AND HONG [33], FOR AN EC OF 20,000 BITS (ER = 0.076 BPP)

Image	[31]	[24]	[34]	[33]	Proposed scheme
Lena	55.03	54.82	54.92	54.97	56.15
Barbara	55.04	55.29	54.89	55.18	56.24
Airplane (F-16)	57.34	56.84	58.58	58.67	59.45
Peppers	52.30	52.55	52.16	52.20	53.39
Fishing boat	52.65	52.43	52.26	52.37	53.12
Average	**54.47**	**54.39**	**54.56**	**54.68**	**55.67**

to Tables IV and V, our method generally achieves a larger PSNR than Sachnev *et al.*'s. Our average gains are 1.44 dB for an EC of 10,000 bits, and 1.2 dB for an EC of 20,000 bits. Notice that the result for Baboon is not presented in Table V since our method cannot embed such a payload into this image.

Li *et al.* and Hong's methods [24], [33], [34] are also based on the prediction-error-histogram incorporating with an embedding-position-selection strategy. According to Fig. 7, as expected, these methods perform similarly to Sachnev *et al.*'s and they are better than ours only for some images when our EC approaches its maximum. In most cases, our method can achieve a larger PSNR. Referring to Tables IV and V, one can see that our method improves Li *et al.* and Hong's by increasing PSNR by at least 1 dB in average.

We remark that, comparing our method with [24], [31], [33], [34], our main advantage is the utilization of two-dimensional difference-histogram.

We now discuss the size of the compressed location map. Notice that in our method, the maximum modification to each image pixel is at most 1 in value, so the overflow/underflow problem may only occur for the pixel-pair (x,y) where $x \in \{0,255\}$ or $y \in \{0,255\}$. For the six test images Lena, Baboon, Barbara, Airplane (F-16), Peppers, and Fishing boat, except the last two columns and last two rows which are not used in data embedding, the amount of pixel-pairs (x,y) satisfying $x \in \{0,255\}$ or $y \in \{0,255\}$ is 0, 0, 0, 0, 2, and 9, respectively. So, in the test cases, there are only a few "1" in the location map (actually, there is no overflow/underflow in most cases). Thus the size of the compressed location map is rather small.

Finally, it should be mentioned that some tested methods here can provide a much higher maximum EC (about 1.0 bpp) than the proposed one (e.g., only 0.14 bpp, for the image Lena). The maximum EC in bits (ER in bpp) for our method is 36,690 (0.14) for Lena, 12,830 (0.049) for Baboon, 30,790 (0.117) for Barbara, 50,399 (0.192) for Airplane (F-16), 25,309 (0.097) for Peppers, 24,137 (0.092) for Fishing boat, respectively. This is a drawback of our method. However, this level of EC is sufficient for many practical applications, e.g., Coatrieux *et al.* [4] pointed out that an EC of 3,500 bits (about 0.014 bpp for a 512×512 sized image) is enough for the application of RDH in medical image sharing. Improving EC is beyond the scope of this paper, and we will investigate this issue in our future work.

V. CONCLUSION

In this paper, we presented a novel RDH scheme by using a two-dimensional difference-histogram according to a specifically designed DPM. In addition, a pixel-pair-selection strategy is also proposed to further enhance the embedding performance.

This work is the first attempt to employ higher dimensional histogram to design RDH. Compared with the previously introduced one-dimensional histogram based methods, our approach can exploit the image redundancy better and achieve an improved performance. However, since only one pixel of a pixel-pair is allowed to be modified by 1 in value, our EC is low. This issue should be investigated in the future. Moreover, utilizing more suitable two-dimensional histogram and designing more meaningful DPM (e.g., in an image dependent way) to achieve the best embedding performance is also a valuable problem.

REFERENCES

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

[2] Y. Q. Shi, Z. Ni, D. Zou, C. Liang, and G. Xuan, "Lossless data hiding: fundamentals, algorithms and applications," in *Proc. IEEE ISCAS*, 2004, vol. 2, pp. 33–36.

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

[4] G. Coatrieux, C. L. 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.

[5] M. Fontani, A. D. Rosa, R. Caldelli, F. Filippini, A. Piva, and M. Consalvo, "Reversible watermarking for image integrity verification in hierarchical pacs," in *Proc. 12th ACM Workshop on Multimedia and Security*, 2010, pp. 161–168.

[6] 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.

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

[8] K.-L. Chung, Y.-H. Huang, P.-C. Chang, and H.-Y. Liao, "Reversible data hiding-based approach for intra-frame error concealment in H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 11, pp. 1643–1647, Nov. 2010.

[9] 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.

[10] 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.

[11] 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.

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

[13] 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.

[14] 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.

[15] 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.

[16] 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.

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

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

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

[20] 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.

[21] 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.

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

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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, no. , pp. 721–724, 2008.

[28] 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.

[29] 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.

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

[31] 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.

[32] 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.

[33] W. Hong, "An efficient prediction-and-shifting embedding technique for high quality reversible data hiding," *EURASIP J. Adv. Signal Process.*, vol. 2010, 2010.

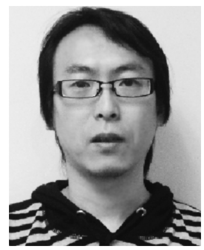
[34] W. Hong, "Adaptive reversible data hiding method based on error energy control and histogram shifting," *Opt. Commun.*, vol. 285, no. 2, pp. 101–108, 2012.

[35] C. Wang, X. Li, and B. Yang, "High capacity reversible image watermarking based on integer transform," in *Proc. IEEE ICIP*, 2010, pp. 217–220.

[36] M. J. Weinberger, G. Seroussi, and G. Sapiro, "The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-ls," *IEEE Trans. Image Process.*, vol. 9, no. 8, pp. 1309–1324, Aug. 2000.

[37] P. Tsai, Y. C. Hu, and H. L. Yeh, "Reversible image hiding scheme using predictive coding and histogram shifting," *Signal Process.*, vol. 89, no. 6, pp. 1129–1143, Jun. 2009.

[38] W. Hong and T. S. Chen, "A local variance-controlled reversible data hiding method using prediction and histogram-shifting," *J. Syst. Software*, vol. 83, no. 12, pp. 2653–2663, Dec. 2010.

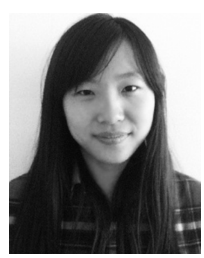


Xiaolong Li received the B.S. degree from Peking University, Beijing, China, the M.S. degree from Ecole Polytechnique, Palaiseau, France, and the Ph.D. degree in mathematics from ENS de Cachan, Cachan, France, in 1999, 2002, and 2006, respectively.

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 in 2002 and 2005, respectively, from the Zhengzhou Information Science and Technology Institute, Zhengzhou, China.

Currently, he is 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. She is currently working toward the Master degree in computer science at Peking University, Beijing, China. 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.

Currently, he is a Professor with the Institute of Computer Science and Technology, Peking University. His research interests are image processing and information hiding.