Digital Signal Processing 25 (2014) 255–265

Contents lists available at ScienceDirect

Digital Signal Processing

www.elsevier.com/locate/dsp

Improved PVO-based reversible data hiding

Fei Peng, Xiaolong Li, Bin Yang [*]

Institute of Computer Science and Technology, Peking University, Beijing 100871, China

ARTICLE INFO

Article history:
Available online 20 November 2013

Keywords:
Reversible data hiding
Pixel-value-ordering
Prediction-error expansion
Block selection
Histogram modification

ABSTRACT

This work extends a recently proposed reversible data hiding (RDH) scheme of Li et al. which is based on pixel-value-ordering and prediction-error expansion. In Li et al.'s method, the maximum and minimum of a pixel block are predicted and modified to embed data. The pixel value order of each block is unchanged after data embedding and the property guarantees the reversibility. In this work, instead of the difference between the maximum and second largest value of a block (or, the minimum and second smallest value of a block) considered in Li et al.'s method, new differences are computed and new histogram-modification-strategy is utilized. Take the maximum for example, the new difference is defined considering the pixel locations of the maximum and second largest value. In this way, the blocks where the maximum equals to the second largest value can be exploited to embed data while these blocks suitable for RDH are not utilized in Li et al.'s work. This can better exploit image redundancy and achieve a superior embedding performance. Extensive experiments verify that the proposed method outperforms Li et al.'s and some other state-of-the-art works.

1. Introduction

Image data hiding is a technique to embed secret data into digital images. The fundamental requirement of this technique is that, for decoder, the embedded data should be exactly extracted from the marked image. Moreover, the visual appearance of the marked image and the cover should be similar. In reversible data hiding (RDH), the decoder requires not only the hidden data but also the original image itself, meaning that the marked image should be inverted to its cover after data extraction [1]. The reversibility is essential in some applications such as military and medical image processing. Usually, the performance of an RDH scheme is evaluated by its capacity-distortion behavior. That is, for a given embedding capacity (EC), one expects to reduce the embedding distortion as much as possible.

The early RDH algorithms are mainly based on lossless compression [2–4]. These methods usually provide a low EC and may lead to severe degradation in image quality. Afterwards, Tian proposed the difference-expansion (DE) technique [5], a significant spatial domain algorithm which performs on pixel pairs. In DE, the secret data is embedded in a reversible way by expanding pixel differences. The correlation between adjacent pixels are exploited by DE and a good performance is achieved. Later on, DE is developed by Thodi and Rodriguez [6], in which the pixel difference is replaced by the prediction-error for expansion embedding. Compared with DE, prediction-error expansion (PEE) may take advantage of image correlation of a larger local image region. In [7], Thodi and Rodriguez's method is extended by Hu et al. by constructing a payload dependent location map, where the compressibility of location map is further improved. Recently, as an extension of PEE, adaptive embedding based on a pixel selection strategy is proposed by Li et al. [8]. It guarantees that more data is embedded into a smoother pixel according to a local complexity measurement. So far, DE and PEE are extensively investigated in many valuable algorithms [9–17]. Moreover, some integer-transform-based methods have also been presented [18–21], which experimentally show superior performance over DE. The integer-transform-based methods group several pixels into a unit and embed data unit by unit. It advantages in reducing the impact of location map on the embedding performance.

Ni et al.'s histogram-shifting (HS) based method [22] is also a benchmarking work of RDH, in which the peak and minimum points of image histogram are modified to embed data. However, its EC is low and the method does not work well if the cover image has a flat histogram. To facilitate it, Lee et al. [23] proposed to utilize the difference-histogram. Lee et al.'s method exploits the correlation of neighboring pixels and can embed larger payload with reduced distortion compared with Ni et al.'s. Recently, the HS technique is further investigated by Li et al. [24] by introducing a general framework to construct HS-based RDH.

Sorting is another important technique of RDH. In [25], Kamstra and Heijmans improved Tian's DE method by sorting pixel pairs according to the local variance before embedding data. Notice that in DE, the integer average of two pixels in a pair is invariant and

[*] Corresponding author. Fax: +86 10 82529207.
E-mail addresses: aniki@pku.edu.cn (F. Peng), lixiaolong@pku.edu.cn (X. Li), yang_bin@pku.edu.cn (B. Yang).

1051-2004/$ – see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.dsp.2013.11.002

only the difference value is altered, and thus these embedding-invariants can be utilized by both encoder and decoder to compute the local variance to sort pixel pairs. If the local variance of a pixel pair is small, the pair is located in a flat image region and it is more probably to be expandable with small difference. Therefore, by sorting, only smooth pixel pairs are selected for data embedding and the location map can be compressed remarkably. The experiments reported in [25] indicated significant improvement over the original DE method. In recent works [9,8,21,26,16], the authors illustrated that combining sorting (or, pixel selection) with other reversible techniques such as integer transform or PEE, can dramatically improve the embedding performance.

Some recently proposed algorithms focus on very high visual quality [22,23,27–33]. These algorithms are implemented by modifying a certain histogram, in which some histogram bins are expanded to carry hidden data while some others are shifted to ensure the reversibility. Since each image pixel is modified at most 1 in value in these methods, the visual quality is well guaranteed (PSNR $\geqslant$ 48.13 dB). Among these works, the one based on pixel-value-ordering (PVO) proposed by Li et al. [33] performs rather well and it is better than some prior arts. The values of a pixel block are first ordered in Li et al.'s method, then the second largest value (second smallest value, resp.) is used to predict the maximum (minimum, resp.). Finally, data embedding is realized through modifying the maximum and minimum.

More specifically, in Li et al.'s method, for a block containing n pixels, its value $(x_1, \dots, x_n)$ is first sorted in ascending order to obtain $(x_{\sigma(1)}, \dots, x_{\sigma(n)})$, where $\sigma : \{1, \dots, n\} \to \{1, \dots, n\}$ is the unique one-to-one mapping. Then the prediction-errors for the maximum and minimum are computed by $PE_{\max} = x_{\sigma(n)} - x_{\sigma(n-1)}$ and $PE_{\min} = x_{\sigma(1)} - x_{\sigma(2)}$. Finally, data embedding is implemented by modifying the histograms of $PE_{\max}$ and $PE_{\min}$. Take $PE_{\max}$ for an example, its histogram is defined on $[0, +\infty)$ since it is always positive, and the histogram peak, the bin 1, is expanded to carry hidden data while the bins larger than 1 are shifted. According to the comparison results reported in Table 1 of [33], the PVO-based predictor is proved more suitable for RDH than MED (median-edge-detector [34]), GAP (gradient-adjusted-prediction [35]), and the mean-value-predictor [9].

Though Li et al.'s method works well, there is still room for improvement. Notice that in this method, a block-selection technique is adopted to priorly select the smooth blocks for data embedding while ignoring the rough ones. By block selection, a more concentrate histogram is obtained and it is more favorable to the embedding performance. However, in Li et al.'s method, the bin 0 (e.g., take the maximum for an example, the block of bin 0 means that its maximum equals to the second largest pixel value) is not used. Therefore, the smooth blocks are not fully exploited in Li et al.'s method since the blocks of bin 0 are usually smooth ones. Based on this consideration, instead of $PE_{\max}$ utilized in Li et al.'s method, we propose to consider a new difference $d_{\max}$, where $d_{\max} = x_u - x_v$, $u = \min(\sigma(n), \sigma(n-1))$ and $v = \max(\sigma(n), \sigma(n-1))$. We will see later that, the two differences $PE_{\max}$ and $d_{\max}$ are closely related, and the corresponding two histograms have the same occurrence at bin 0. However, taking $d_{\max}$ may introduce a Laplacian-like distribution defined on $(-\infty, +\infty)$ centered at 0, and meanwhile the bin 0 can be utilized for expansion embedding. In this way, the blocks with $x_{\sigma(n)} = x_{\sigma(n-1)}$ can be exploited by our method to embed data while they are not utilized in Li et al.'s work.

Based on the new differences and histogram-modification-strategy, an improvement for Li et al.'s PVO-based RDH method is proposed in this work. By the proposed improvement, Li et al.'s maximum EC can be increased. And, more importantly, for a given EC, the improved method has an opportunity to utilize larger sized blocks and it can better exploit the image redundancy to achieve a superior embedding performance. Also, for a given EC and block size, more smooth blocks can be used by the improved method, and this is helpful to the embedding performance as well. In summary, this work improves Li et al.'s PVO-based method in terms of both EC and marked image quality.

The rest of this paper is organized as follows. In Section 2, some related works including Li et al.'s PVO-based method [33] and Lee et al.'s method [23] are briefly reviewed. Our motivation for improving Li et al.'s method is also presented in this section. Then in Section 3, the improvement for Li et al.'s method is first described, and the detailed embedding and extracting procedures of our proposed algorithm are presented at the end of this section. The experimental results comparing our method with the state-of-the-art works are reported in Section 4. Finally, we conclude our work in the last section.

2. Related works

In this section, as a review, we first introduce Li et al.'s PVO-based method [33]. Then, we present Lee et al.'s method [23]. Moreover, an extension for Lee et al.'s method by increasing EC is also presented. Actually, our proposed method is motivated by this extension.

2.1. PVO-based RDH scheme

Li et al.'s work [33] presents a PVO-based RDH scheme using the PEE technique. By modifying the maximum and minimum of a pixel block, this method can embed sufficient payload into a cover image with high marked image quality, and it outperforms several state-of-the-art works such as [7,9,10,8]. We now briefly introduce this method. For clarity of presentation, only the maximum-modification-based embedding stage is presented, and the minimum-modification-based one is omitted here.

First, the cover image is divided into non-overlapped equal-sized blocks. For a given block X containing n pixels, sort its value $(x_1, \dots, x_n)$ in ascending order to obtain $(x_{\sigma(1)}, \dots, x_{\sigma(n)})$, where $\sigma : \{1, \dots, n\} \to \{1, \dots, n\}$ is the unique one-to-one mapping such that: $x_{\sigma(1)} \leqslant \dots \leqslant x_{\sigma(n)}$, $\sigma(i) < \sigma(j)$ if $x_{\sigma(i)} = x_{\sigma(j)}$ and $i < j$. Then, the second largest value, $x_{\sigma(n-1)}$, is used to predict the maximum $x_{\sigma(n)}$. The corresponding prediction-error is

$$PE_{\max} = x_{\sigma(n)} - x_{\sigma(n-1)}. \tag{1}$$

Finally, generate the histogram of $PE_{\max}$. Notice that $PE_{\max}$ is always positive, this histogram is defined on the interval $[0, +\infty)$.

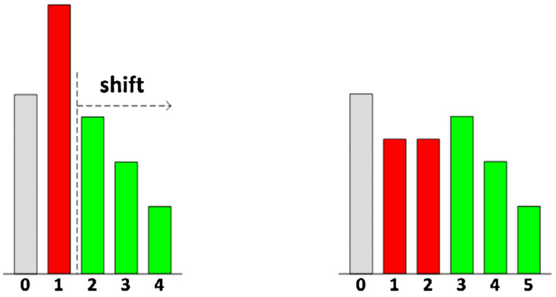
Since the bin 1 (i.e., the bin with $PE_{\max} = 1$) is usually the histogram peak, Li et al.'s method simply takes this bin as inner region for expansion embedding, whereas the bins larger than 1 are taken as outer region and shifted to ensure the reversibility (see Fig. 1 for illustration). In this situation, to embed data via PEE, the prediction-error $PE_{\max}$ is modified to

$$\widetilde{PE}_{\max} = \begin{cases} PE_{\max}, & \text{if } PE_{\max} = 0, \\ PE_{\max} + b, & \text{if } PE_{\max} = 1, \\ PE_{\max} + 1, & \text{if } PE_{\max} > 1 \end{cases} \tag{2}$$

where $b \in \{0, 1\}$ is a data bit to be embedded. Accordingly, the maximum $x_{\sigma(n)}$ is modified to

$$\tilde{x} = x_{\sigma(n-1)} + \widetilde{PE}_{\max} = \begin{cases} x_{\sigma(n)}, & \text{if } PE_{\max} = 0, \\ x_{\sigma(n)} + b, & \text{if } PE_{\max} = 1, \\ x_{\sigma(n)} + 1, & \text{if } PE_{\max} > 1 \end{cases} \tag{3}$$

whereas other values $x_{\sigma(1)}, \dots, x_{\sigma(n-1)}$ keep unchanged. Hence, the marked value of X is $(y_1, \dots, y_n)$, where $y_{\sigma(n)} = \tilde{x}$ and $y_i = x_i$ for every $i \neq \sigma(n)$.



(a) Before data embedding.

(b) After data embedding.

Fig. 1. Histogram of $PE_{\max}$, before and after data embedding, in which the bin 1 (the red one in (a)) is used for expansion embedding and the bins larger than 1 (the green ones in (a)) are shifted. The bin 0 is not involved in Li et al.'s data embedding. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

In the above procedure, since the maximum $x_{\sigma(n)}$ is either unchanged or increased, the pixel value order (i.e., the mapping σ) remains unchanged after data embedding. This property guarantees the exact data extraction and lossless image restoration. Moreover, notice that the modification to each pixel is at most 1, and thus a high visual quality of marked image can be achieved.

2.2. Lee et al.'s RDH scheme and its extension

To better present our motivation improving [33], we now briefly introduce another RDH scheme proposed by Lee et al. [23]. This method modifies the pixel differences to embed data.

In Lee et al.'s method, the cover image is first divided into non-overlapped pixel pairs such that each pair contains two adjacent pixels. Then, for each pair (x, y), compute the difference value

$$d = y - x \tag{4}$$

to obtain the difference-histogram. Here, the first pixel x serves as a prediction for the second pixel y. Finally, data embedding is implemented by modifying the difference-histogram as follows to get the marked difference value $\tilde{d}$ (see Fig. 2(a) for illustration)

$$\tilde{d} = \begin{cases} d, & \text{if } d = 0, \\ d + b, & \text{if } d = 1, \\ d + 1, & \text{if } d > 1, \\ d - b, & \text{if } d = -1, \\ d - 1, & \text{if } d < -1 \end{cases} \tag{5}$$

where $b \in \{0, 1\}$ is a data bit to be embedded. Accordingly, the marked pixel pair is determined by

$$(\tilde{x}, \tilde{y}) = \begin{cases} (x, y), & \text{if } d = 0, \\ (x, y + b), & \text{if } d = 1, \\ (x, y + 1), & \text{if } d > 1, \\ (x, y - b), & \text{if } d = -1, \\ (x, y - 1), & \text{if } d < -1. \end{cases} \tag{6}$$

In the above procedure, only the second pixel y is modified while the first pixel x keeps unchanged.

We now introduce an extension of Lee et al.'s method. Notice that in Lee et al.'s method, the bins 1 and -1 are expanded, the bins larger than 1 or smaller than -1 are shifted, and the bin 0 is not involved in data embedding. However, it is unreasonable to ignore the bin 0 as it is usually the histogram peak (notice that the difference-histogram is a Laplacian-like distribution centered at 0). Based on this consideration, instead of taking bins 1 and -1 for expansion embedding, we propose to utilize bins 1 and 0. Specifically, we propose to modify the difference-histogram as follows to get the marked difference value $\tilde{d}$ (see Fig. 2(b) for illustration)

$$\tilde{d} = \begin{cases} d + b, & \text{if } d = 1, \\ d + 1, & \text{if } d > 1, \\ d - b, & \text{if } d = 0, \\ d - 1, & \text{if } d < 0 \end{cases} \tag{7}$$

where $b \in \{0, 1\}$ is a data bit to be embedded. Then, the marked pixel pair can be determined by

$$(\tilde{x}, \tilde{y}) = \begin{cases} (x, y + b), & \text{if } d = 1, \\ (x, y + 1), & \text{if } d > 1, \\ (x, y - b), & \text{if } d = 0, \\ (x, y - 1), & \text{if } d < 0. \end{cases} \tag{8}$$

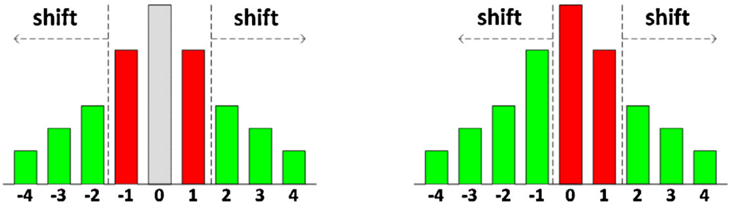
In this way, Lee et al.'s method can be improved by increasing EC with only a slight loss of PSNR. For example, for the standard 512 × 512 sized gray-scale image Airplane, the EC can be increased from 31 000 bits to 36 000 bits (about 16% of increase) and the PSNR is slightly changed from 52.59 dB to 51.81 dB.

3. Proposed method

In this section, we first introduce our improvement for Li et al.'s maximum-modification-based embedding scheme presented in Section 2.1. Some experimental results are also reported to validate our improvement. Then, a similar method for improving Li et al.'s minimum-modification-based embedding scheme is presented. Finally, based on these improvements, we introduce our proposed RDH method by describing the detailed data embedding and extraction procedures.

3.1. Improved maximum-modification-based data embedding

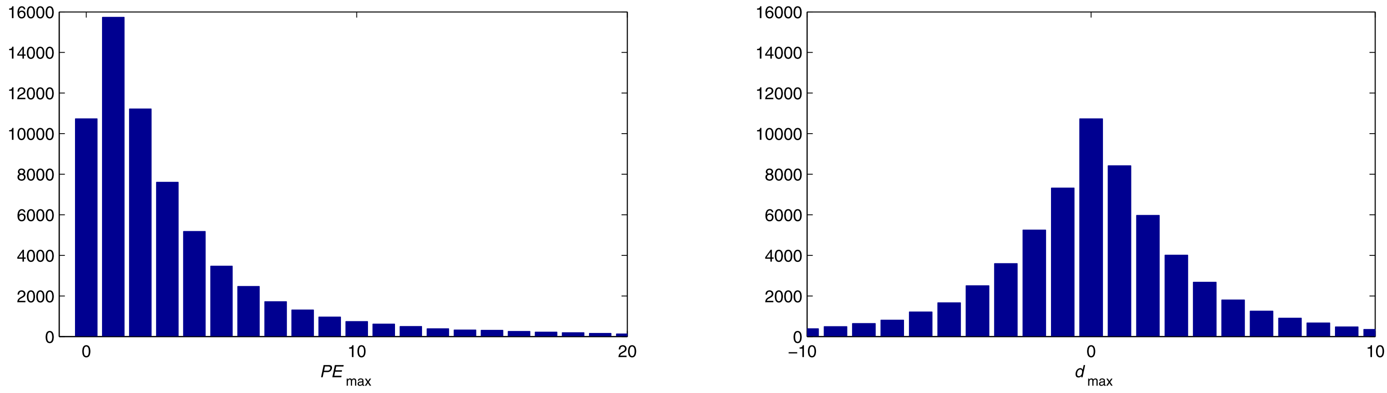
We use the same notations defined in Section 2.1. Reviewing (2), like the case of Lee et al., the bin 0 is not involved in Li et al.'s maximum-modification-based data embedding. In this situation, the extension of Lee et al.'s method presented in Section 2.2 can be applied to Li et al.'s as well. By this means, Li et al.'s method can be improved by taking advantage of bin 0. We will see later that, by the improved method, Li et al.'s EC can be increased. On the other hand, for a given EC, the improved method may utilize larger sized blocks for data embedding, and thus can better exploit the image redundancy which is helpful for the embedding performance.



(a) Lee *et al.*'s method.

(b) Extension of Lee *et al.*'s method.

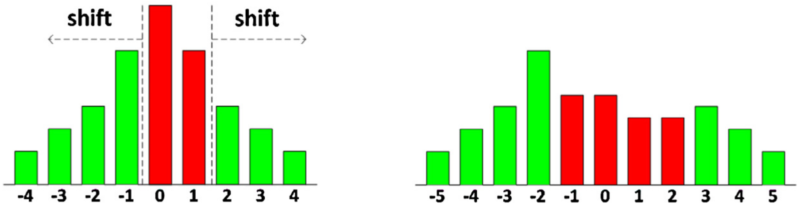
Fig. 2. Illustration of difference-histogram for (a) Lee et al.'s method and (b) its extension. The red bins are expanded while the green ones are shifted. In particular, for (a), the bin 0 is not involved in data embedding. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a)

(b)

Fig. 3. Histograms of (a) $PE_{\max}$ and (b) $d_{\max}$ with 2×2 sized blocks, for the standard 512×512 sized gray-scale image Lena.



(a) Before data embedding.

(b) After data embedding.

Fig. 4. Histogram of $d_{\max}$, before and after data embedding, in which the bins 1 and 0 (the red ones in (a)) are used for expansion embedding and the other bins (the green ones in (a)) are shifted. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Let us first introduce some notations. Take

$$d_{\max} = x_u - x_v \tag{9}$$

where

$$\begin{cases} u = \min\bigl(\sigma(n), \sigma(n-1)\bigr), \\ v = \max\bigl(\sigma(n), \sigma(n-1)\bigr). \end{cases} \tag{10}$$

Here, instead of the difference $PE_{\max} = x_{\sigma(n)} - x_{\sigma(n-1)}$ utilized in Li et al.'s work which is always positive, the new difference $d_{\max}$ taking value in $(-\infty, +\infty)$ is considered in our work. And, unlike the case of $PE_{\max}$ (see Fig. 3 for illustration), the histogram of $d_{\max}$ is a Laplacian-like distribution centered at 0 since both the two differences $x_{\sigma(n)} - x_{\sigma(n-1)}$ and $x_{\sigma(n-1)} - x_{\sigma(n)}$ are counted considering the order of $\sigma(n)$ and $\sigma(n-1)$. In this way, the blocks with $x_{\sigma(n)} = x_{\sigma(n-1)}$ can be exploited by our method to embed data by modifying $d_{\max}$, while they are not utilized in Li et al.'s work.

The relationship between $PE_{\max}$ and $d_{\max}$ is analyzed as follows. Clearly, by (1) and (9), one can verify that

- if $\sigma(n) < \sigma(n-1)$, then $u = \sigma(n)$, $v = \sigma(n-1)$, and $d_{\max} = PE_{\max}$. In this case, by the definition of σ, we have $x_{\sigma(n)} > x_{\sigma(n-1)}$ and thus $d_{\max} > 0$;
- if $\sigma(n) > \sigma(n-1)$, then $u = \sigma(n-1)$, $v = \sigma(n)$, and $d_{\max} = -PE_{\max}$. In this case, $d_{\max} \leqslant 0$.

Hence, the histograms of $PE_{\max}$ and $d_{\max}$, $h_{PE_{\max}}$ and $h_{d_{\max}}$, satisfy

$$\begin{cases} h_{PE_{\max}}(0) = h_{d_{\max}}(0), \\ h_{PE_{\max}}(k) = h_{d_{\max}}(k) + h_{d_{\max}}(-k) \quad (k > 0). \end{cases} \tag{11}$$

By using $d_{\max}$, the embedding rules in (2) and (3) can be reformulated in an equivalent way, in which $d_{\max}$ is first modified to $\tilde{d}_{\max}$ with

$$\tilde{d}_{\max} = \begin{cases} d_{\max}, & \text{if } d_{\max} = 0, \\ d_{\max} + b, & \text{if } d_{\max} = 1, \\ d_{\max} + 1, & \text{if } d_{\max} > 1, \\ d_{\max} - b, & \text{if } d_{\max} = -1, \\ d_{\max} - 1, & \text{if } d_{\max} < -1 \end{cases} \tag{12}$$

where $b \in \{0, 1\}$ is a data bit to be embedded, and then the marked value of $x_{\sigma(n)}$ is computed by

$$\tilde{x} = x_{\sigma(n-1)} + |\tilde{d}_{\max}| = \begin{cases} x_{\sigma(n)}, & \text{if } d_{\max} = 0, \\ x_{\sigma(n)} + b, & \text{if } d_{\max} = 1, \\ x_{\sigma(n)} + 1, & \text{if } d_{\max} > 1, \\ x_{\sigma(n)} + b, & \text{if } d_{\max} = -1, \\ x_{\sigma(n)} + 1, & \text{if } d_{\max} < -1. \end{cases} \tag{13}$$

Comparing (12) with (5), one can observe that the embedding rules of Li et al. and Lee et al. are exactly the same. That is, in Li et al. and Lee et al.'s methods, the two differences $d_{\max}$ and d are expanded or shifted in the same way. In this light, the extension of Lee et al.'s method presented in Section 2.2 can be directly applied to Li et al.'s. Specifically, we propose to improve Li et al.'s embedding rules (2) and (12) as follows (see Fig. 4 for illustration). Instead of taking bins 1 and -1 in the histograms of $d_{\max}$ for expansion embedding, we utilize here the bins 1 and 0. First, $d_{\max}$ is modified to

$$\tilde{d}_{\max} = \begin{cases} d_{\max} + b, & \text{if } d_{\max} = 1, \\ d_{\max} + 1, & \text{if } d_{\max} > 1, \\ d_{\max} - b, & \text{if } d_{\max} = 0, \\ d_{\max} - 1, & \text{if } d_{\max} < 0. \end{cases} \tag{14}$$

Then, the marked value of the maximum $x_{\sigma(n)}$ is determined as

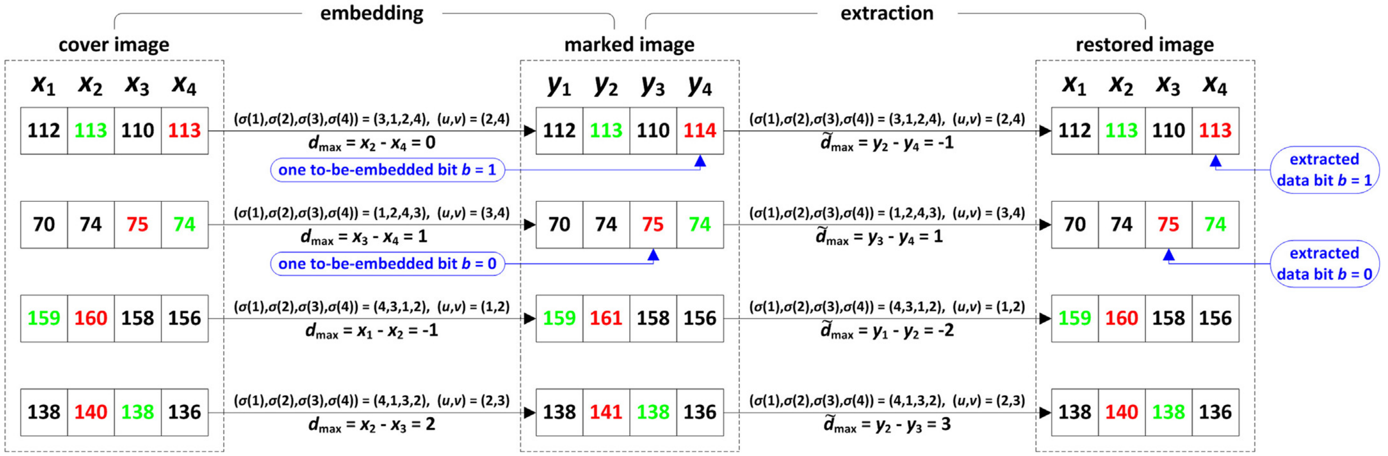


Fig. 5. Illustration of the improved maximum-modification-based data embedding. The maximum and the second largest value of each block are marked in red and green, respectively. For data embedding, the upper two blocks are expanded to carry hidden data, and the lower two blocks are shifted. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$$\tilde{x} = x_{\sigma(n-1)} + |\tilde{d}_{\max}| = \begin{cases} x_{\sigma(n)} + b, & \text{if } d_{\max} = 1, \\ x_{\sigma(n)} + 1, & \text{if } d_{\max} > 1, \\ x_{\sigma(n)} + b, & \text{if } d_{\max} = 0, \\ x_{\sigma(n)} + 1, & \text{if } d_{\max} < 0. \end{cases} \qquad (15)$$

That is, the marked value of X is $(y_1, \dots, y_n)$, where $y_{\sigma(n)} = \tilde{x}$ and $y_i = x_i$ for every $i \neq \sigma(n)$.

Notice that in this improved method, the mapping σ also keeps unchanged. As a result, the decoder can realize data extraction and image restoration as follows according to the marked value $(y_1, \dots, y_n)$. Suppose here $\tilde{d}_{\max} = y_u - y_v$, where (u, v) is defined in (10).

- If $\tilde{d}_{\max} > 0$, we know that $y_u > y_v$. Thus, $\sigma(n) < \sigma(n-1)$, $u = \sigma(n)$ and $v = \sigma(n-1)$:
  - if $\tilde{d}_{\max} \in \{1, 2\}$, the hidden data bit is $b = \tilde{d}_{\max} - 1$ and the original maximum is $x_{\sigma(n)} = y_u - b$;
  - if $\tilde{d}_{\max} > 2$, there is no hidden data in this block and the original maximum is $x_{\sigma(n)} = y_u - 1$.
- If $\tilde{d}_{\max} \leqslant 0$, we know that $y_u \leqslant y_v$. Thus, $\sigma(n) > \sigma(n-1)$, $u = \sigma(n-1)$ and $v = \sigma(n)$:
  - if $\tilde{d}_{\max} \in \{0, -1\}$, the hidden data bit is $b = -\tilde{d}_{\max}$ and the original maximum is $x_{\sigma(n)} = y_v - b$;
  - if $\tilde{d}_{\max} < -1$, there is no hidden data in this block and the original maximum is $x_{\sigma(n)} = y_v - 1$.

Fig. 5 illustrates the above embedding and extraction procedures with some blocks of size $n = 4$.

For the improved method, the EC is clearly $h_{d_{\max}}(1) + h_{d_{\max}}(0)$. On the other hand, Li et al.'s EC is $h_{PE_{\max}}(1)$. Hence, as $h_{d_{\max}}$ is a Laplacian-like distribution centered at 0, one can conclude that Li et al.'s EC is increased by the proposed improvement since, according to (11),

$$h_{PE_{\max}}(1) = h_{d_{\max}}(1) + h_{d_{\max}}(-1) < h_{d_{\max}}(1) + h_{d_{\max}}(0). \qquad (16)$$

Fig. 6 shows some experimental results comparing Li et al.'s maximum-modification-based method (dotted) and our proposed improvement (solid). According to this figure, one can observe that our method can increase Li et al.'s EC, e.g., from 15 000 bits to 19 000 bits when taking 2 × 2 sized blocks. At the same time, our method can achieve a larger PSNR for a given EC when using the same sized blocks. Moreover, for example, when EC is 10 000 bits, instead of using necessarily small sized blocks (i.e., 2 × 2) in Li et al.'s method, our improvement has the opportunity of taking larger sized blocks (i.e., 3 × 3). This may better exploit the image redundancy and yield a larger PSNR.

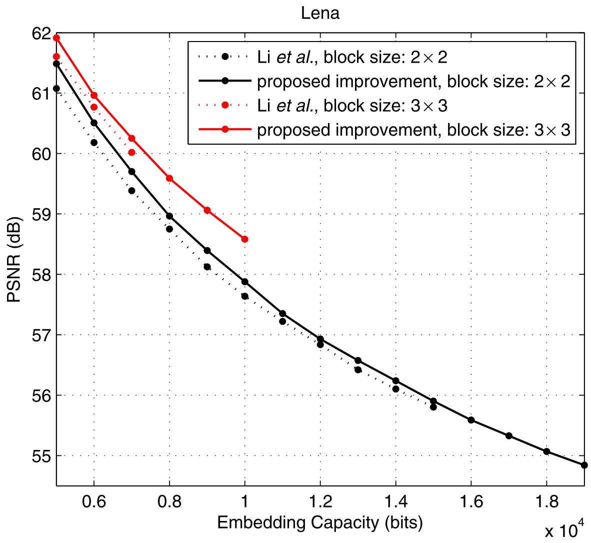


Fig. 6. Performance comparison between Li et al.'s maximum-modification-based method (dotted) and our proposed improvement (solid), for the standard 512 × 512 sized gray-scale image Lena.

3.2. Improved minimum-modification-based data embedding

Our method presented in Section 3.1 can be directly applied to Li et al.'s minimum-modification-based data embedding. The details are given below. First, we define

$$d_{\min} = x_s - x_t \qquad (17)$$

where

$$\begin{cases} s = \min\big(\sigma(1), \sigma(2)\big), \\ t = \max\big(\sigma(1), \sigma(2)\big). \end{cases} \qquad (18)$$

Clearly, one can verify that

- if $\sigma(1) > \sigma(2)$, then $s = \sigma(2)$ and $t = \sigma(1)$. In this case, one must have $x_{\sigma(1)} < x_{\sigma(2)}$ and $d_{\min} > 0$;
- if $\sigma(1) < \sigma(2)$, then $s = \sigma(1)$, $t = \sigma(2)$, and $d_{\min} \leqslant 0$.

Then, the difference $d_{\min}$ is modified to

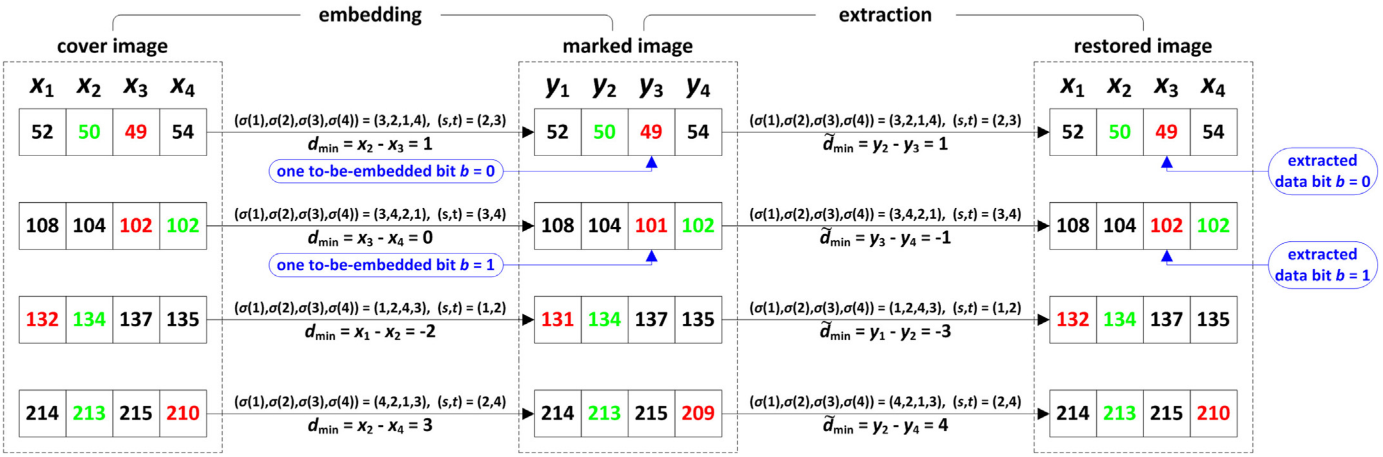


Fig. 7. Illustration of the improved minimum-modification-based data embedding. The minimum and the second smallest value of each block are marked in red and green, respectively. For data embedding, the upper two blocks are expanded to carry hidden data, and the lower two blocks are shifted. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$$\tilde{d}_{\min} = \begin{cases} d_{\min} + b, & \text{if } d_{\min} = 1, \\ d_{\min} + 1, & \text{if } d_{\min} > 1, \\ d_{\min} - b, & \text{if } d_{\min} = 0, \\ d_{\min} - 1, & \text{if } d_{\min} < 0 \end{cases} \tag{19}$$

where $b \in \{0, 1\}$ is a data bit to be embedded. Finally, the marked value of the minimum $x_{\sigma(1)}$ can be determined as

$$\tilde{x} = x_{\sigma(2)} - |\tilde{d}_{\min}| = \begin{cases} x_{\sigma(1)} - b, & \text{if } d_{\min} = 1, \\ x_{\sigma(1)} - 1, & \text{if } d_{\min} > 1, \\ x_{\sigma(1)} - b, & \text{if } d_{\min} = 0, \\ x_{\sigma(1)} - 1, & \text{if } d_{\min} < 0. \end{cases} \tag{20}$$

In summary, the marked value of X is $(y_1, \ldots, y_n)$, where $y_{\sigma(1)} = \tilde{x}$ and $y_i = x_i$ for every $i \neq \sigma(1)$.

Based on the fact that the mapping σ is unchanged, the decoder can realize data extraction and image restoration as follows according to the marked value $(y_1, \ldots, y_n)$. Suppose here $\tilde{d}_{\min} = y_s - y_t$, where (s, t) is defined in (18).

- If $\tilde{d}_{\min} > 0$, we know that $y_s > y_t$. Thus, $\sigma(1) > \sigma(2)$, $s = \sigma(2)$ and $t = \sigma(1)$:
  - if $\tilde{d}_{\min} \in \{1, 2\}$, the hidden data bit is $b = \tilde{d}_{\min} - 1$ and the original minimum is $x_{\sigma(1)} = y_t + b$;
  - if $\tilde{d}_{\min} > 2$, there is no hidden data in this block and the original minimum is $x_{\sigma(1)} = y_t + 1$.
- If $\tilde{d}_{\min} \leqslant 0$, we know that $y_s \leqslant y_t$. Thus, $\sigma(1) < \sigma(2)$, $s = \sigma(1)$ and $t = \sigma(2)$:
  - if $\tilde{d}_{\min} \in \{0, -1\}$, the hidden data bit is $b = -\tilde{d}_{\min}$ and the original minimum is $x_{\sigma(1)} = y_s + b$;
  - if $\tilde{d}_{\min} < -1$, there is no hidden data in this block and the original minimum is $x_{\sigma(1)} = y_s + 1$.

To better explain the above embedding and extraction procedures, some examples with block size $n = 4$ are shown in Fig. 7.

3.3. Proposed data embedding and extraction procedures

First of all, we remark that, the same as Li et al.'s method, we estimate the noise level of a block X by

$$NL = x_{\sigma(n-1)} - x_{\sigma(2)}. \tag{21}$$

Only the blocks satisfying $NL < T$ will be selected for data embedding, where T is a given threshold. By block selection, smooth blocks are selected while the rough ones are not utilized in data embedding. This technique is proved very favorable to the embedding performance.

By combining the improved maximum- and minimum-modification-based embedding methods and by using the block-selection technique, a new RDH method is presented in this section. In the new method, by modifying both the maximum $x_{\sigma(n)}$ and minimum $x_{\sigma(1)}$ according to (15) and (20), at most two bits can be embedded into a block at the same time. Besides, as $x_{\sigma(n-1)}$ and $x_{\sigma(2)}$ are unchanged after data embedding, the noise level NL is invariant and it can be obtained by decoder, which guarantees the reversibility when using the block-selection technique. The data embedding procedure is described in detail as follows. A flowchart illustrating this procedure is given in Fig. 8.

Step 1 *(Image partition)* Divide the cover image into non-overlapped blocks $\{X_1, \ldots, X_N\}$ such that each block contains n pixels. For each block X_i, sort its value $(x_1, \ldots, x_n)$ in ascending order to obtain $(x_{\sigma(1)}, \ldots, x_{\sigma(n)})$. Here, we omit the block-index i in definition of pixel values of X_i for clarity.

Step 2 *(Location map construction)* A location map LM sized N is constructed in this step. For each X_i, we take $LM(i) = 1$ if $x_{\sigma(1)} = 0$ or $x_{\sigma(n)} = 255$, meaning that the modification in this block may occur overflow/underflow. Otherwise, we take $LM(i) = 0$. Afterward, the location map is losslessly compressed to get CLM using arithmetic coding to reduce its length. Denote the length of CLM as l_{CLM}.

Step 3 *(Secret message embedding)* The secret message is embedded into the cover image in this step. For each X_i, calculate its noise level NL by (21).

- If $LM(i) = 1$, the block is marked in location map as overflow/underflow and we do nothing with it.
- If $LM(i) = 0$ and $NL \geqslant T$, the block is a noisy one and we do nothing with it as well.
- If $LM(i) = 0$ and $NL < T$, the block is considered as a smooth one. In this case, the differences $d_{\max}$ and $d_{\min}$ are computed using (9) and (17). Then the maximum and minimum are shifted or expanded to embed data according to (15) and (20).

The step will stop when the message is completely embedded, and we denote p_{end} as the index of the last data-carrying block.

Step 4 *(Auxiliary information and location map embedding)* In this step, the auxiliary information used for blind decoding and the compressed location map are embedded into the rest part of the cover image, i.e., the blocks $\{X_{p_{\text{end}}+1}, \ldots, X_N\}$. First, record

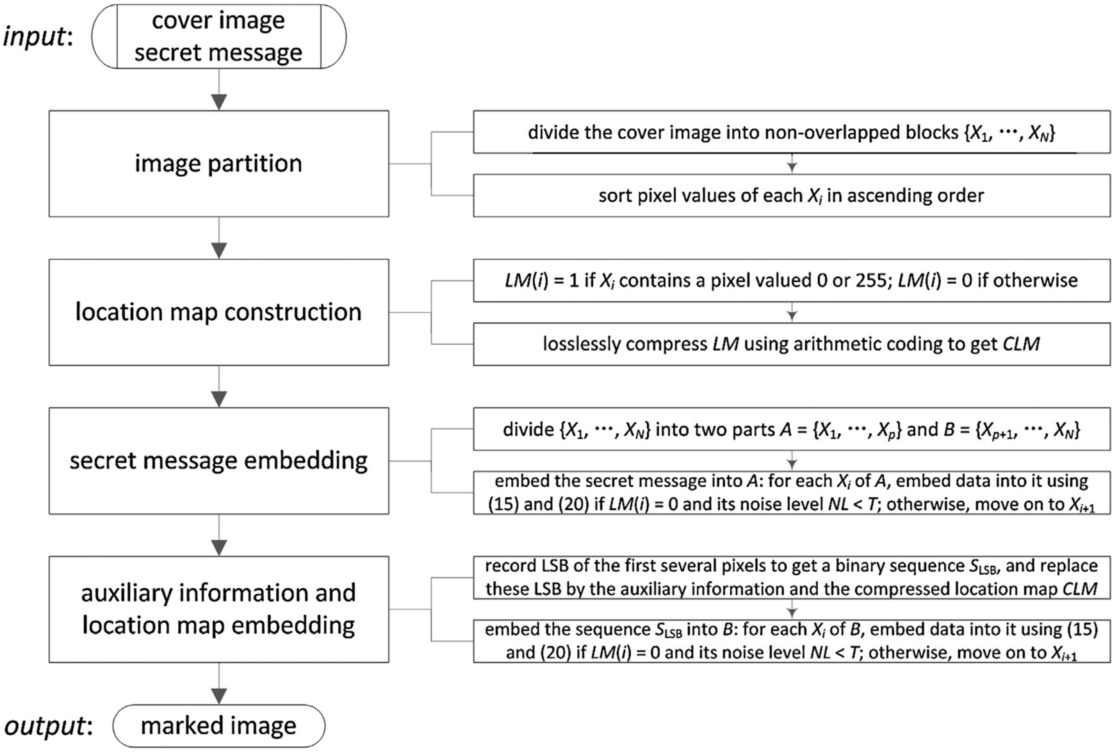

input:
cover image
secret message
image partition
divide the cover image into non-overlapped blocks {X1, ···, XN}
sort pixel values of each Xi in ascending order
location map construction
LM(i) = 1 if Xi contains a pixel valued 0 or 255; LM(i) = 0 if otherwise
losslessly compress LM using arithmetic coding to get CLM
secret message embedding
divide {X1, ···, XN} into two parts A = {X1, ···, Xp} and B = {Xp+1, ···, XN}
embed the secret message into A: for each Xi of A, embed data into it using (15) and (20) if LM(i) = 0 and its noise level NL < T; otherwise, move on to Xi+1
auxiliary information and location map embedding
record LSB of the first several pixels to get a binary sequence SLSB, and replace these LSB by the auxiliary information and the compressed location map CLM
embed the sequence SLSB into B: for each Xi of B, embed data into it using (15) and (20) if LM(i) = 0 and its noise level NL < T; otherwise, move on to Xi+1
output:
marked image

Fig. 8. Flowchart of data embedding.

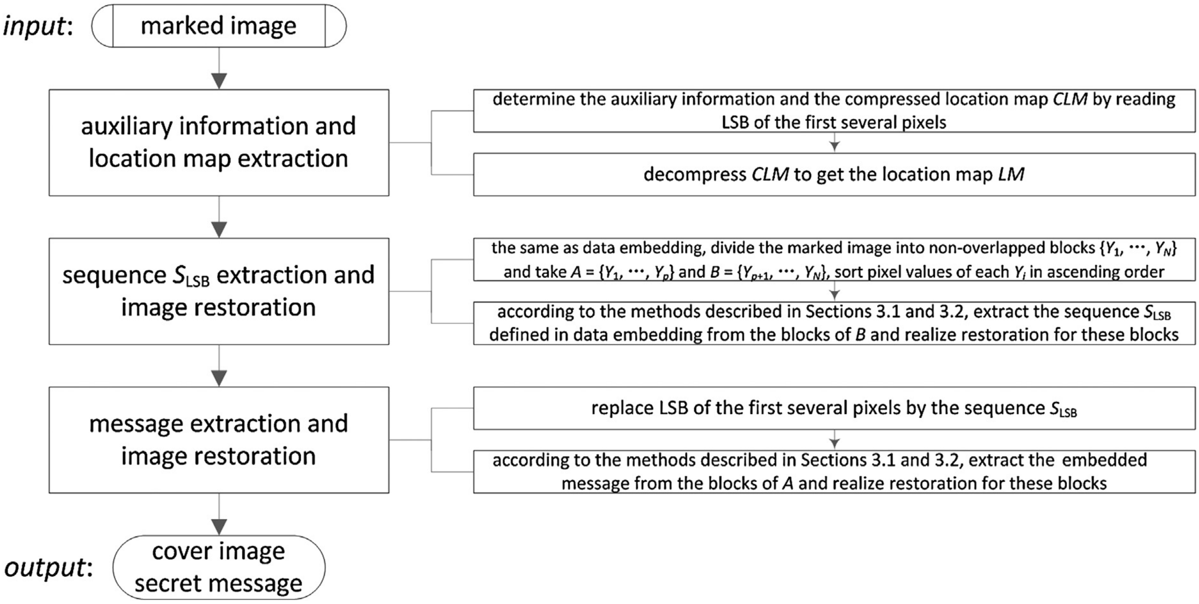

input:
marked image
auxiliary information and location map extraction
determine the auxiliary information and the compressed location map CLM by reading LSB of the first several pixels
decompress CLM to get the location map LM
sequence SLSB extraction and image restoration
the same as data embedding, divide the marked image into non-overlapped blocks {Y1, ···, YN} and take A = {Y1, ···, Yp} and B = {Yp+1, ···, YN}, sort pixel values of each Yi in ascending order
according to the methods described in Sections 3.1 and 3.2, extract the sequence SLSB defined in data embedding from the blocks of B and realize restoration for these blocks
message extraction and image restoration
replace LSB of the first several pixels by the sequence SLSB
according to the methods described in Sections 3.1 and 3.2, extract the embedded message from the blocks of A and realize restoration for these blocks
output:
cover image
secret message

Fig. 9. Flowchart of data extraction.

the LSB of the first $12 + 2\lceil \log_2 N \rceil + l_{\text{CLM}}$ pixels of the image to get a binary sequence S_{LSB}. Then, replace these LSB by the auxiliary information and the compressed location map including

- block size parameters n_1 (2 bits) and n_2 (2 bits), where $n = n_1 \times n_2$,
- noise level threshold T (8 bits),
- end position p_{end} ($\lceil \log_2 N \rceil$ bits),
- length of compressed location map l_{CLM} ($\lceil \log_2 N \rceil$ bits),
- the compressed location map *CLM* (l_{CLM} bits).

Finally, embed the sequence S_{LSB} into $\{X_{p_{\text{end}}+1}, \ldots, X_N\}$ using the same method in Step 3.

In this embedding procedure, the noise level threshold T is selected from 1 and increased by 1 iteratively, to obtain a smallest threshold such that all the hidden data including secret message, auxiliary information and compressed location map can be successfully embedded. Moreover, the same as Li et al.'s method, the embedding procedure is repeated 16 times with $n_1, n_2 \in \{2, 3, 4, 5\}$ to determine the best block size. By this means, the one which achieves the largest PSNR is taken as the final block size.

The corresponding data extraction procedure is detailed as follows. Its flowchart is shown in Fig. 9.

Step 1 *(Auxiliary information and location map extraction)* Read LSB of first $12 + 2\lceil \log_2 N \rceil$ pixels of marked image to get the auxiliary information including the values of n_1, n_2, T, p_{end} and l_{CLM}. Then read LSB of next l_{CLM} pixels to get the compressed location map *CLM*. Determine the location map *LM* by decompressing *CLM*.

Step 2 *(Sequence S_{LSB} extraction and image restoration)* The same as data embedding, divide the marked image into non-overlapped blocks $\{Y_1, \ldots, Y_N\}$ such that each Y_i contains n pixels. In this step, for blocks $\{Y_{p_{\text{end}}+1}, \ldots, Y_N\}$, we will extract the sequence S_{LSB} defined in Step 4 of data embedding and meanwhile realize the restoration. For a block Y_i $(i > p_{\text{end}})$ whose values in ascending order are $(y_{\sigma(1)}, \ldots, y_{\sigma(n)})$:

- If $LM(i) = 0$ and $y_{\sigma(n-1)} - y_{\sigma(2)} < T$, according to the decoding methods described in Sections 3.1 and 3.2, extract data and recover the original values from $y_{\sigma(n)}$ and $y_{\sigma(1)}$.
- Otherwise, there is no hidden data in the block and the original values of $y_{\sigma(n)}$ and $y_{\sigma(1)}$ are themselves.

Here, in any case, for every $i \in \{2, \ldots, n-1\}$, the original value of $y_{\sigma(i)}$ is itself since it is unchanged in data embedding. This step will stop if the sequence S_{LSB} is extracted.

Step 3 *(Message extraction and image restoration)* Replace LSB of the first $12 + 2\lceil \log_2 N \rceil + l_{\text{CLM}}$ image pixels by the sequence S_{LSB} extracted in Step 2. Then use the same method of Step 2 to extract the hidden data from the blocks $\{Y_1, \ldots, Y_{p_{\text{end}}}\}$, and meanwhile to realize restoration for those blocks. Finally, the embedded data is extracted and the cover image is recovered.

4. Experimental results

The proposed method is evaluated by comparing it with Li et al.'s original PVO-based method [33], Sachnev et al.'s method [9], Hong's method [16] and Tsai et al.'s method [36]. Ten standard gray-scale images including Lena, Baboon, Airplane (F-16), Peppers, Fishing boat, Sailboat on lake, Elaine, Barbara, Tiffany and Man are used in our experiment. Except Barbara, all the images are downloaded from the USC-SIPI database.[1] The image size is 1024 × 1024 for Man and 512 × 512 for each one of the other images. Fig. 10 shows the performance comparison. For our method, we vary EC from 5000 bits to its maximum with a step size of 1000 bits. The figure illustrates that the proposed method outperforms the state-of-the-art works in most cases. However, it should be mentioned that for the image Man, our method is in fact worse than Li et al.'s original PVO-based method. It is mainly due to the negative effect of location map and we will discuss this issue later.

For Li et al.'s PVO-based method, our algorithm is a direct improvement of their work. One can observe that, according to Fig. 10, except for the rough image Baboon and the image Man containing a lot of zero-valued pixels, our method can provide a larger PSNR for every image whatever the EC is. Referring to Table 1, it can be seen that our method improves Li et al.'s by 0.42 dB in average for an EC of 10 000 bits. Moreover, the proposed method can increase Li et al.'s EC especially for smooth images. For example, for the smooth image Airplane, Li et al.'s EC can be increased significantly from 38 000 bits to 52 000 bits. For the ten test images, our method can increase Li et al.'s EC by 14% in average. Our advantage lies in the utilization of new differences $d_{\max}$ and $d_{\min}$, and the new histogram-modification-strategy.

Sachnev et al.'s method is based on PEE incorporating with an embedding-position-selection strategy. For this method, the integer average of the four neighboring pixels is used to predict the center one. Moreover, a sorting technique is used 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. Referring to Fig. 10, one can see that in most cases, our method is better than this well-performed method. However, for some images, when EC approaches its maximum (e.g., for Baboon when EC is larger than 8000 bits, for Airplane when EC is larger than 47 000 bits), Sachnev et al.'s method can achieve a larger PSNR. The reason is that in such cases, since EC is high, our method should necessarily take 2 × 2 sized blocks and cannot well exploit the advantage of pixel correlation. Besides, the smooth blocks are insufficient in such cases and our method should necessarily use rough blocks to embed data. This is also unfavorable to the embedding performance. But, for moderate EC, our method is better. Referring to Table 1, our method improves Sachnev et al.'s by 2.35 dB in average for an EC of 10 000 bits.

Table 1
Comparison of PSNR (dB) between the proposed method and the methods of Li et al. [33], Sachnev et al. [9], Hong [16] and Tsai et al. [36], for an EC of 10 000 bits.

	Proposed	Li et al. [33]	Sachnev et al. [9]	Hong [16]	Tsai et al. [36]
Lena	60.47	59.86	58.18	58.77	58.45
Baboon	53.55	53.50	54.15	52.90	51.97
Airplane (F-16)	62.96	61.61	60.38	62.07	58.50
Peppers	58.98	58.55	55.55	56.04	55.34
Fishing boat	58.27	57.85	56.15	56.53	54.88
Sailboat on lake	58.87	58.18	56.65	57.79	55.55
Elaine	57.36	56.84	56.12	56.99	55.97
Barbara	60.54	59.98	58.15	58.34	56.16
Tiffany	60.75	60.39	58.31	57.27	56.71
Man	61.16	61.95	55.74	51.60	53.27
Average	**59.29**	**58.87**	**56.94**	**56.83**	**55.68**

Hong's method [16] is also based on PEE incorporating with an embedding-position-selection strategy. According to Fig. 10, as expected, this method performs similarly to Sachnev et al.'s and it is better than ours only for the images Airplane and Barbara when our EC approaches its maximum. In most cases, our method can achieve a larger PSNR. Referring to Table 1, our method improves Hong's by 2.46 dB in average for an EC of 10 000 bits.

Tsai et al.'s work [36] is a recently proposed method which is based on a new residual histogram generation strategy. In this method, the differences between each pixel and its two neighbors are exploited to generate a residual histogram and embed data. Though this recently proposed method is proved better than many existing up-to-date algorithms such as [28,30,37], our method can provide a superior performance according to Fig. 10. More specifically, referring to Table 1, our method improves Tsai et al.'s by 3.61 dB in average for an EC of 10 000 bits.

The proposed method is also compared with [33,9,16,36] on the Kodak image data set[2] which contains 24 color images sized 512 × 768 or 768 × 512. These images are taken from digital cameras and they are changed to gray-scale format before testing. The experimental results are reported in Table 2. According to this table, one can see that our method improves Li et al.'s, Sachnev et al.'s, Hong's and Tsai et al.'s by 1.64 dB, 1.94 dB, 4.24 dB and 2.15 dB in average, respectively. The experimental results confirm that our algorithm outperforms the prior arts for various kinds of images.

We now give the parameters used in our method, for the ten standard test images and the 24 Kodak images with a fixed EC of 10 000 bits. These parameters, including block size (n_1, n_2), noise level threshold T, auxiliary information size l_{AI}, and compressed location map size l_{CLM}, are shown in Table 3. According to this table, we can conclude that:

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

[2] http://r0k.us/graphics/kodak/.

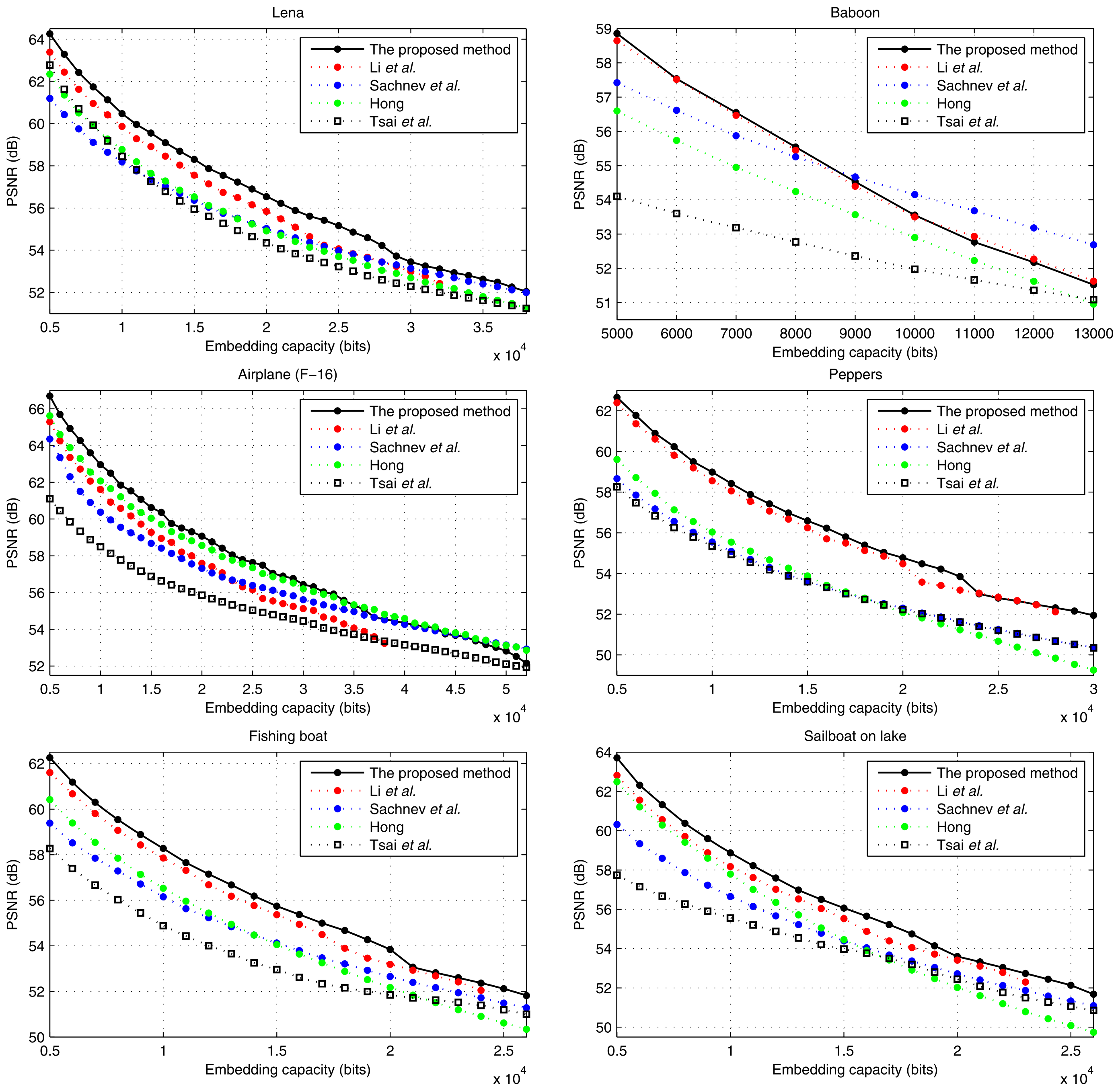


Fig. 10. Performance comparison between the proposed method and the methods of Li et al. [33], Sachnev et al. [9], Hong [16] and Tsai et al. [36].

- There is no general law for the block size. Recall here that in our method, for a given EC, the block size is adaptively selected to maximize the PSNR.
- For a given EC, the noise level threshold T is large for a rough image (e.g., 65 for Baboon) while small for a smooth one (e.g., 4 for Airplane). So, compared to a rough image, a smooth one can provide a larger maximum EC.
- The auxiliary information contains only a few bits (about 40 bits in our test cases).
- The compressed location map is usually small in size. However, when l_{CLM} is large, it has a negative effect on the performance since the overhead information is large. For example, for the image Man, the embedded payload size is actually $10\,000 + 44 + 11\,216 = 21\,260$ which is about two times of EC. In this case, our method is even worse than Li et al.'s original PVO-based method. For another example, for the image kodim08, the embedded payload size is $10\,000 + 44 + 7992 = 18\,036$, and in this case, our method is also worse than Li et al.'s.

Finally, to better illustrate the performance of the proposed method, we also conduct experiments on a large database BossBase v1.01[3] [38] containing 10 000 gray-scale images. The images in BossBase are never-compressed cover images coming from several digital cameras. All these images are created from full-resolution color images in RAW format (CR2 or DNG). The images are then resized such that the smaller side is 512 pixels long, then

[3] http://www.agents.cz/boss/BOSSFinal/.

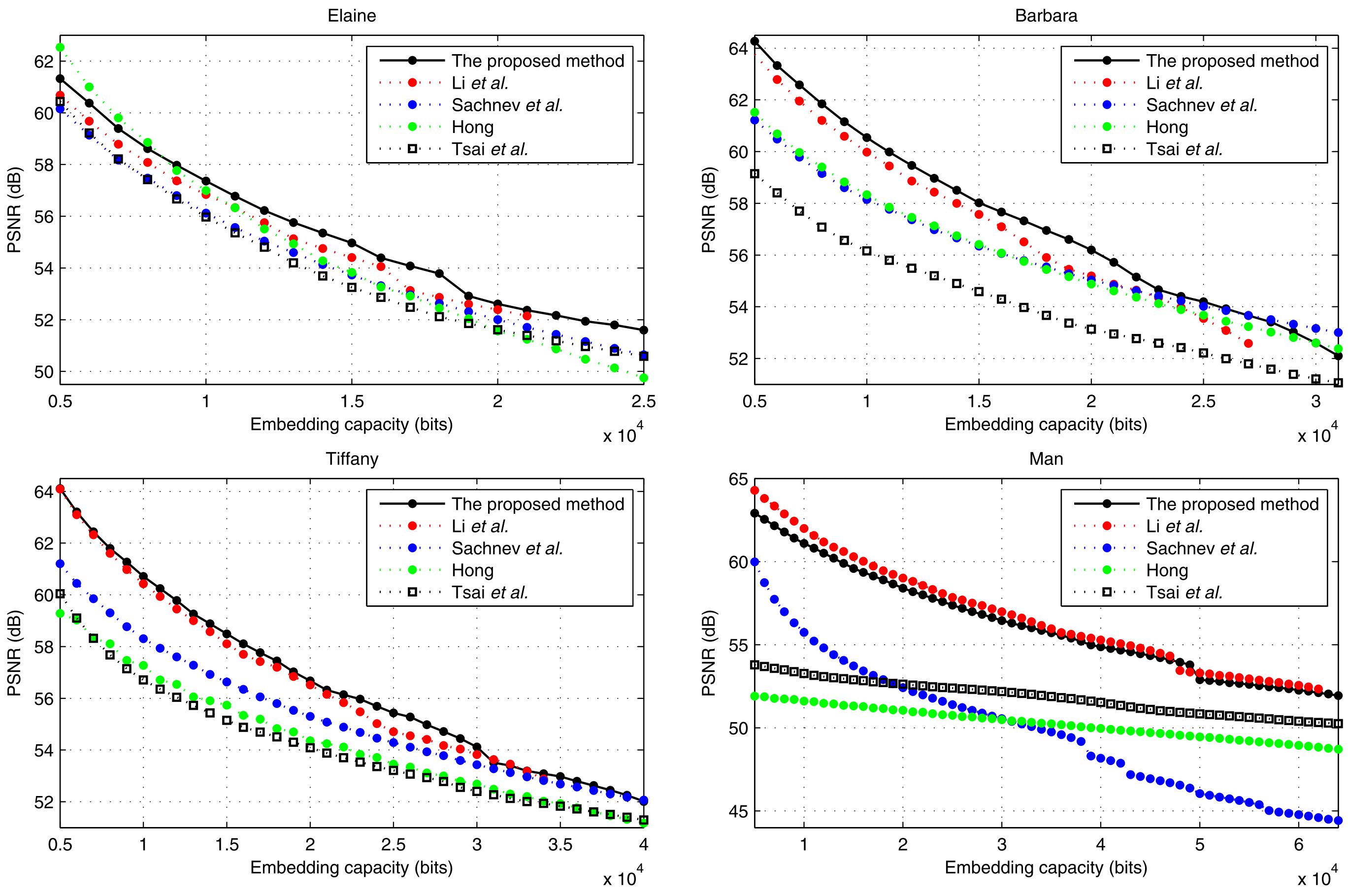


Fig. 10. (*continued*)

Table 2
Comparison of the PSNR (dB) between the proposed method and the methods of Li et al. [33], Sachnev et al. [9], Hong [16] and Tsai et al. [36], for the Kodak image data set with an EC of 10 000 bits.

	Proposed	Li et al. [33]	Sachnev et al. [9]	Hong [16]	Tsai et al. [36]
kodim01	62.85	59.20	61.83	59.30	58.14
kodim02	64.07	62.81	61.84	59.56	59.74
kodim03	65.35	63.82	63.60	60.76	63.66
kodim04	63.55	62.00	61.85	62.04	62.24
kodim05	61.64	59.78	61.73	58.31	59.99
kodim06	65.00	61.56	65.27	62.56	64.01
kodim07	65.10	63.61	63.08	60.05	63.06
kodim08	56.49	56.55	58.39	50.34	53.00
kodim09	63.69	62.81	60.54	60.63	62.02
kodim10	63.14	62.28	60.66	58.93	61.54
kodim11	65.18	62.43	62.88	60.60	63.97
kodim12	64.48	63.01	62.76	59.99	62.82
kodim13	53.88	52.71	57.65	49.22	54.15
kodim14	61.28	59.54	60.83	57.52	59.25
kodim15	64.72	63.23	62.25	60.27	61.19
kodim16	64.96	62.99	63.01	60.21	64.07
kodim17	63.73	62.05	61.54	61.12	62.81
kodim18	60.71	59.84	58.85	56.49	58.19
kodim19	63.35	62.51	60.74	60.48	61.69
kodim20	64.88	61.59	52.79	58.20	60.30
kodim21	63.62	62.62	60.72	57.51	62.10
kodim22	62.56	61.57	60.34	57.23	60.33
kodim23	64.58	63.32	62.23	59.64	62.93
kodim24	61.50	59.24	58.27	57.64	57.49
Average	**62.93**	**61.29**	**60.99**	**58.69**	**60.78**

Table 3
Parameters used in the proposed method with an EC of 10 000 bits: block size (n_1, n_2), noise level threshold T, auxiliary information size l_{AI}, and compressed location map size l_{CLM}.

	(n_1, n_2)	T	l_{AI}	l_{CLM}
Lena	(4, 3)	9	42	40
Baboon	(3, 2)	65	44	40
Airplane (F-16)	(5, 2)	4	42	40
Peppers	(3, 4)	13	42	72
Fishing boat	(2, 5)	16	42	136
Sailboat on lake	(2, 4)	10	42	40
Elaine	(4, 3)	24	42	64
Barbara	(3, 3)	8	42	40
Tiffany	(3, 4)	8	42	920
Man	(4, 5)	31	44	11 216
kodim01	(3, 2)	4	44	320
kodim02	(5, 3)	6	42	64
kodim03	(3, 5)	4	42	88
kodim04	(5, 3)	8	42	376
kodim05	(3, 2)	4	44	1504
kodim06	(3, 2)	2	44	824
kodim07	(5, 4)	6	42	88
kodim08	(3, 3)	60	44	7992
kodim09	(5, 5)	9	40	312
kodim10	(4, 5)	10	42	952
kodim11	(5, 2)	4	44	192
kodim12	(3, 5)	5	42	424
kodim13	(3, 2)	51	44	7568
kodim14	(3, 3)	9	44	776
kodim15	(5, 4)	7	42	1112
kodim16	(3, 5)	6	42	40
kodim17	(4, 4)	8	42	368
kodim18	(4, 3)	15	42	1008
kodim19	(5, 4)	14	42	72
kodim20	(2, 5)	2	44	5040
kodim21	(5, 4)	10	42	760
kodim22	(3, 4)	9	42	840
kodim23	(5, 5)	8	40	600
kodim24	(2, 5)	10	44	5168

they are cropped to 512 × 512 pixels, and finally converted to gray-scale. For each BossBase image, we employ the proposed method and Li et al.'s to embed data with an EC of 10 000 bits. The obtained average PSNR on this database are 62.76 dB (our method) and 62.07 dB (Li et al.), respectively. So, the proposed method out-

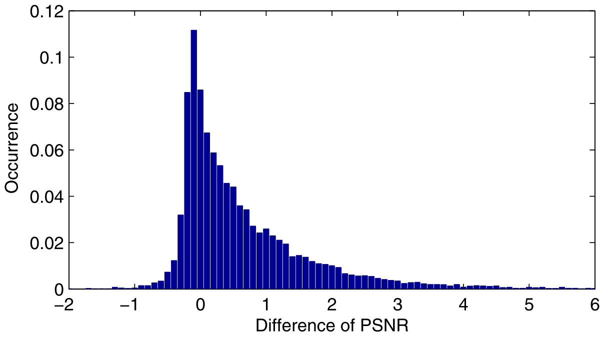


Fig. 11. Distribution of PSNR difference between the proposed method and Li et al. [33] on the database of BossBase v1.01.

performs Li et al.'s with an average PSNR increase of 0.69 dB. The probability distribution of PSNR difference between the proposed method and Li et al.'s is shown in Fig. 11, demonstrating the superiority of our method on a large database.

5. Conclusion

In this paper, as an extension to Li et al.'s work [33], we presented an improved PVO-based RDH method. Based on PVO, new differences are computed and new histogram-modification-strategy is used to embed data. By the proposed approach, the image redundancy can be better exploited and a superior embedding performance is achieved. Extensive experiments verified that the proposed method outperformed Li et al.'s original PVO-based method and the state-of-the-art works [9,16,36].

References

[1] R. Caldelli, F. Filippini, R. Becarelli, Reversible watermarking techniques: An overview and a classification, EURASIP J. Inf. Secur. 2010 (2010), article ID 134546.
[2] J. Fridrich, M. Goljan, R. Du, Lossless data embedding – new paradigm in digital watermarking, EURASIP J. Appl. Signal Process. 2002 (2) (2002) 185–196.
[3] M.U. Celik, G. Sharma, A.M. Tekalp, E. Saber, Lossless generalized-LSB data embedding, IEEE Trans. Image Process. 14 (2) (2005) 253–266.
[4] M.U. Celik, G. Sharma, A.M. Tekalp, Lossless watermarking for image authentication: A new framework and an implementation, IEEE Trans. Image Process. 15 (4) (2006) 1042–1049.
[5] J. Tian, Reversible data embedding using a difference expansion, IEEE Trans. Circuits Syst. Video Technol. 13 (8) (2003) 890–896.
[6] D.M. Thodi, J.J. Rodriguez, Expansion embedding techniques for reversible watermarking, IEEE Trans. Image Process. 16 (3) (2007) 721–730.
[7] Y. Hu, H.K. Lee, J. Li, DE-based reversible data hiding with improved overflow location map, IEEE Trans. Circuits Syst. Video Technol. 19 (2) (2009) 250–260.
[8] X. Li, B. Yang, T. Zeng, Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection, IEEE Trans. Image Process. 20 (12) (2011) 3524–3533.
[9] V. Sachnev, H.J. Kim, J. Nam, S. Suresh, Y.Q. Shi, Reversible watermarking algorithm using sorting and prediction, IEEE Trans. Circuits Syst. Video Technol. 19 (7) (2009) 989–999.
[10] L. Luo, Z. Chen, M. Chen, X. Zeng, Z. Xiong, Reversible image watermarking using interpolation technique, IEEE Trans. Inf. Forensics Secur. 5 (1) (2010) 187–193.
[11] W. Hong, T.-S. Chen, A local variance-controlled reversible data hiding method using prediction and histogram-shifting, J. Syst. Softw. 83 (12) (2010) 2653–2663.
[12] X. Gao, L. An, Y. Yuan, D. Tao, X. Li, Lossless data embedding using generalized statistical quantity histogram, IEEE Trans. Circuits Syst. Video Technol. 21 (8) (2011) 1061–1070.
[13] D. Coltuc, Improved embedding for prediction-based reversible watermarking, IEEE Trans. Inf. Forensics Secur. 6 (3) (2011) 873–882.
[14] C.-C. Chang, Y.-H. Huang, H.-Y. Tsai, C. Qin, Prediction-based reversible data hiding using the difference of neighboring pixels, AEÜ, Int. J. Electron. Commun. 66 (9) (2012) 758–766.
[15] H.-T. Wu, J. Huang, Reversible image watermarking on prediction errors by efficient histogram modification, Signal Process. 92 (12) (2012) 3000–3009.
[16] W. Hong, Adaptive reversible data hiding method based on error energy control and histogram shifting, Opt. Commun. 285 (2) (2012) 101–108.
[17] G. Coatrieux, W. Pan, N. Cuppens-Boulahia, F. Cuppens, C. Roux, Reversible watermarking based on invariant image classification and dynamic histogram shifting, IEEE Trans. Inf. Forensics Secur. 8 (1) (2013) 111–120.
[18] S. Weng, Y. Zhao, J.S. Pan, R. Ni, Reversible watermarking based on invariability and adjustment on pixel pairs, IEEE Signal Process. Lett. 15 (2008) 721–724.
[19] D. Coltuc, Low distortion transform for reversible watermarking, IEEE Trans. Image Process. 21 (1) (2012) 412–417.
[20] X. Wang, X. Li, B. Yang, Z. Guo, Efficient generalized integer transform for reversible watermarking, IEEE Signal Process. Lett. 17 (6) (2010) 567–570.
[21] F. Peng, X. Li, B. Yang, Adaptive reversible data hiding scheme based on integer transform, Signal Process. 92 (1) (2012) 54–62.
[22] Z. Ni, Y.Q. Shi, N. Ansari, W. Su, Reversible data hiding, IEEE Trans. Circuits Syst. Video Technol. 16 (3) (2006) 354–362.
[23] S.K. Lee, Y.H. Suh, Y.S. Ho, Reversible image authentication based on watermarking, in: Proc. of the IEEE ICME, 2006, pp. 1321–1324.
[24] X. Li, B. Li, B. Yang, T. Zeng, General framework to histogram-shifting-based reversible data hiding, IEEE Trans. Image Process. 23 (6) (2013) 2181–2191.
[25] L. Kamstra, H.J.A.M. Heijmans, Reversible data embedding into images using wavelet techniques and sorting, IEEE Trans. Image Process. 14 (12) (2005) 2082–2090.
[26] W. Hong, An efficient prediction-and-shifting embedding technique for high quality reversible data hiding, EURASIP J. Adv. Signal Process. 2010 (2010), article ID 104835.
[27] M. Fallahpour, Reversible image data hiding based on gradient adjusted prediction, IEICE Electron. Express 5 (20) (2008) 870–876.
[28] W. Hong, T.S. Chen, C.W. Shiu, Reversible data hiding for high quality images using modification of prediction errors, J. Syst. Softw. 82 (11) (2009) 1833–1842.
[29] G. Coatrieux, C.L. Guillou, J.M. Cauvin, C. Roux, Reversible watermarking for knowledge digest embedding and reliability control in medical images, IEEE Trans. Inf. Technol. Biomed. 13 (2) (2009) 158–165.
[30] Y.-C. Li, C.-M. Yeh, C.-C. Chang, Data hiding based on the similarity between neighboring pixels with reversibility, Digit. Signal Process. 20 (4) (2010) 1116–1128.
[31] K.-L. Chung, Y.-H. Huang, W.-M. Yan, W.-C. Teng, Distortion reduction for histogram modification-based reversible data hiding, Appl. Math. Comput. 218 (9) (2012) 5819–5826.
[32] X.-T. Wang, C.-C. Chang, T.-S. Nguyen, M.-C. Li, Reversible data hiding for high quality images exploiting interpolation and direction order mechanism, Digit. Signal Process. 23 (2) (2013) 569–577.
[33] X. Li, J. Li, B. Li, B. Yang, High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion, Signal Process. 93 (1) (2013) 198–205.
[34] M.J. Weinberger, G. Seroussi, G. Sapiro, The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS, IEEE Trans. Image Process. 9 (8) (2000) 1309–1324.
[35] X. Wu, N. Memon, Context-based, adaptive, lossless image coding, IEEE Trans. Commun. 45 (4) (1997) 437–444.
[36] Y.-Y. Tsai, D.-S. Tsai, C.-L. Liu, Reversible data hiding scheme based on neighboring pixel differences, Digit. Signal Process. 23 (3) (2013) 919–927.
[37] Z. Zhao, H. Luo, Z.-M. Lu, J.-S. Pan, Reversible data hiding based on multilevel histogram modification and sequential recovery, Int. J. Electron. Commun. 65 (10) (2011) 814–826.
[38] P. Bas, T. Filler, T. Pevny, Break our steganographic system – the ins and outs of organizing boss, in: Proc. 13th Int. Workshop on Information Hiding, in: Lect. Notes Comput. Sci., vol. 6958, Springer, 2011, pp. 59–70.

Fei Peng received the B.S. degree from Peking University, Beijing, China, in 2011. He is currently working toward the Master degree of Computer Science at Peking University. His research interest is information hiding.

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 in 2007–2009. His 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.