

Reversible watermarking using optional prediction error histogram modification

Bo Ou^{a,b,*}, Yao Zhao^{a,b,c}, Rongrong Ni^{a,b}

^a Institute of Information Science, School of Computer Science, Beijing Jiaotong University, Beijing 100044, China

^b Beijing Key Laboratory of Advanced Information Science and Network Technology, Beijing 100044, China

^c State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, China

ARTICLE INFO

Article history:

Received 15 October 2011

Received in revised form

8 April 2012

Accepted 15 April 2012

Communicated by X. Gao

Available online 11 May 2012

Keywords:

Reversible watermarking

Prediction error

Optional histogram shifting

ABSTRACT

To reduce the distortion of high payload watermarking scheme, a reversible watermarking based on optional prediction-error histogram modification is proposed to improve the watermarked image fidelity at high embedding rate. By considering the pixel compensation during the multiple layer embedding, an optional predictor is designed to generate the most appropriate prediction error histogram, which results in less distortions at the same embedding rate. Unlike other histogram based schemes, the generated prediction error histogram can be tuned through the selection of threshold for each layer to strike the balance between capacity and pixel compensation. Compared with the other schemes, the proposed scheme introduces less distortion and obtains high image quality for the high embedding rate. The experiments on the standard and military images verify the effectiveness of the proposed scheme.

© 2012 Elsevier B.V. All rights reserved.

1. Introduction

The increasing applications of multimedia technology, especially in military and medical fields, have made the reversible watermarking becomes a hot research spot in recent years. Reversible watermarking, which is also called lossless data hiding, is a technology that embeds the covert information in the digital image and allows the receiver to recover the original image after the extraction of covert information. It is widely applied to copyright protection, ownership allegation and integrity authentication.

Dozens of high payload reversible watermarking techniques had been proposed over the past few years including difference expanding (DE) [2,3,11,12], histogram shifting (HS) [1,5,6,8–10,12,13], prediction error expanding (PEE) [7,14–18], and integer transform [19–24]. Besides, some methods [4,25–27] are proposed recently to investigate the robustness of reversible watermarking. At present, two kinds of method are quite effective and widespread. One is HS based scheme which is proposed by Ni et al. [6] firstly. In [6], Ni et al. employ peak/zero points in histogram of the given image to embed the watermark data. In their schemes, the amount of embedding capacity is equal to the population of peak points in histogram and the lower bound of PSNR (peak signal-to-noise ratio) of the marked image versus original image is guaranteed to be

higher than 48.13 dB. The work [6] is considered to be fundamental for histogram based reversible watermarking schemes, afterwards this method had been developed in many aspects and is extended into the prediction-error (PE) based histogram shifting. In [8], Tsai et al. utilize the residual histogram of the predicted errors of the host image to embed the watermark data. Later, Tai et al. [9] present a reversible data hiding method using the binary structure to solve the problem of communicating pairs of peak points. By utilizing the distribution of pixel differences, the method [9] can easily achieve the high embedding capacity while keeping the distortion low. Another mainstream reversible method is PEE based scheme which combines the strategy of expanding and histogram shifting. Since the modifications are small after doubling the prediction error, PEE based scheme is effective as it not only causes less embedding distortion at low embedding rate but obtains high embedding rate for a single layer. One representative method of PEE based schemes is the method [7] proposed by Sachnev et al. In [7], Sachnev et al. sort the prediction-error and start the embedding from the pixels with smallest in the sorted order. Recently, Luo et al. [5] utilize interpolation errors, which is the differences between the interpolation value and original pixel value, to embed secret data for reversible watermarking. Since the modification of pixels was slight, this method can preserve high image quality even a large amount of covert data is embedded. However, there still exist two drawbacks in the histogram shifting schemes: (1) the image quality degraded seriously after multiple layer embedding when a high payload watermarking was required, such as 1.0 bpp for lena image and 0.5 bpp for baboon image and (2) the principle to take advantages

* Corresponding author at: Institute of Information Science, School of Computer Science, Beijing Jiaotong University, Beijing 100044, China.

E-mail addresses: Gomes19851019@gmail.com, 09112055@bjtu.edu.cn (B. Ou), yzhao@bjtu.edu.cn (Y. Zhao), rrni@bjtu.edu.cn (R. Ni).

of pixel compensation have not been fully discussed yet in the previous histogram shifting based reversible watermarking schemes. Note that the pixel compensations will inevitably occur during the consecutive histogram shifting operations because some pixels will be added/subtracted by 1 at first and subtracted/added by 1 later. The larger pixel compensations indicate the low embedding distortion at the same embedding rate. Thus, how to make use of the pixel compensation to balance the conflict between capacity and distortion make significant sense, especially for the high payload watermarking schemes.

To solve the above problems, a novel reversible watermarking scheme based on optional histogram modification is proposed. In this paper, we focus on discussing a method to modify prediction error histogram and utilize it to improve the watermarked image fidelity at high embedding level. Different from other methods, we utilize the threshold to modify the generated prediction error histogram and take pixel compensations into account by introducing the distortion increment. The strategy is efficient because the embedding distortion is compensated partly after the multiple optional histogram shifting operations and the overhead information consumes little embedding capacity. The rest of paper is organized as follows. The details of proposed scheme are described in Section 2, and the performance comparisons with other methods are given in Section 3. Finally, Section 4 concludes this paper.

2. The proposed method

2.1. Optional prediction error histogram modification

For an 8-bit gray-scale host image X with $H \times W$ pixels, X is defined as

$$X = \{x_{i,j} | i \in [1, H], j \in [1, W]\} \quad (1)$$

where H and W denote the height and width of X , respectively, $x_{i,j}$ denotes the gray-scale value of pixel, $x_{i,j} \in [0, 255]$. For reversibility during prediction process, X is divided into two sub-images X_1 and X_2 which consist of the pixels of odd rows and even rows of the given image, respectively:

$$X_1 = \{x_{2i-1,j} | i \in [1, H_1], j \in [1, W]\}$$

$$X_2 = \{x_{2i,j} | i \in [1, H_1], j \in [1, W]\} \quad (2)$$

where $H_1 = \lfloor H/2 \rfloor$. In general, the predictor utilizing full-closing pixels as the prediction context obtains more accurate predictions than predictors using half-closing pixels. Thus, we adopt the prediction pattern as shown in Fig. 1 to involve the neighbor pixels as many as possible. During the embedding procedure these two sub-images are processed for data hiding by fixed order, e.g., use X_1 as the context to predict X_2 and obtain the watermarked sub-image X'_2 first, then use X'_2 to obtain X'_1 . Obviously, this order must be inverse when data extraction is processing. This situation is due to the fact that when we recalculate the predicted values, the prediction context should be the same as in the data embedding. Since both sub-images are processed similarly, we take the pixel $x_{i,j}$ in X_2 whose context is given as $C_{i,j} = \{x_k | k \in [1, 6]\}$ for example to illustrate the optional histogram modification algorithm. In the proposed scheme, we define two predictors to calculate the prediction errors. The first predictor, which is similar to the mean value predictor, is defined as

$$\hat{x}_{i,j}^A = \frac{x_2 + x_5}{4} + \frac{x_1 + x_3 + x_4 + x_6}{8} \quad (3)$$

Considering that the abrupt changes appear in texture areas, the context of $x_{i,j}$ may contain the edge pixels and results in a bigger prediction error. In the second predictor, we try to obtain a better

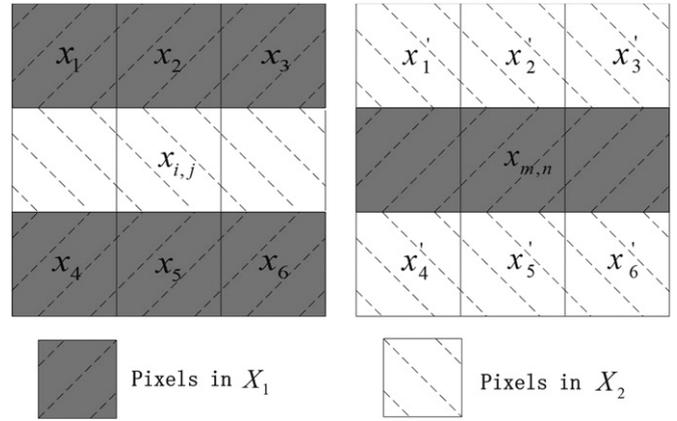


Fig. 1. Prediction pattern of X_1 and X_2 : x' denotes the watermarked pixel, and the subscript 1, ..., 6 denote the prediction context.

estimated value when we predict the pixels of texture by excluding the edge pixels from the prediction context. That is say, for the texture area not all the six neighbor pixels are used for prediction procedure in the second predictor. Thus, we use a measure of regularity R_k based on squared difference to distinguish the edge pixels

$$R_k = (x_k - \hat{x}_{i,j}^A)^2 \quad (4)$$

There are two reasons why we use the squared difference rather than absolute difference: firstly, by magnifying the difference we can easily distinguish the abrupt pixels and secondly, it tends to assign larger weighted value to the homologous pixels, and consequently maximizes the similarity between the prediction and its context. Then we select the pixels in C whose R_k below the threshold Th into an new prediction context with the regularity set $C_R = \{R_l | l \in [0, L(C)]\}$, where $L(c)$ is the length of set. Thus, based on R_l the second predictor is formulated as

$$\hat{x}_{i,j}^B = \sum_{l=1}^{L(C)} w_l x_l \quad \text{if } C_R \neq \emptyset, x_l \in C_R, \forall R_l \neq 0 \quad (5)$$

where the associated weighted value w_l is calculated in the following manner:

$$w_l = \left(\frac{1}{R_l} \right) / \left(\sum_{R \in C_R} \frac{1}{R} \right) \quad (6)$$

Eq. (6) is similar to a version of weighted mean, where the context pixels with smaller R_l contribute more than others to the final prediction. From (5), it is clear that the type B predictor with w_l make accurate results under the condition of Th when predicting the pixels of texture. However, two extreme cases need to be noticed: (1) the large value of Th results in the fact that abrupt pixels are hardly excluded and (2) when the pixel to be predicted is the abrupt pixel compared with its context, most of neighbor pixels may be excluded, e.g., when $x_{i,j} = 104$ with the context 141, 64, 58, 82, 115, 55 and $Th=4$, $\hat{x}_{i,j}^B = 86.75$, thus none of neighbor pixels are used and the type A predictor is utilized. By tuning the value of Th , the final prediction $\hat{x}_{i,j}$ varies as

$$\hat{x}_{i,j} = \begin{cases} \hat{x}_{i,j}^B & \text{if } C_R \neq \emptyset, x_l \in C_R, \forall R_l \neq 0 \\ \hat{x}_{i,j}^A & \text{else} \end{cases} \quad (7)$$

Particularly, when Th is equaled to 0, only the type A predictor is utilized. The prediction error $e_{i,j}$ is obtained as

$$e_{i,j} = x_{i,j} - \text{round}(\hat{x}_{i,j}) \quad (8)$$

where function $\text{round}(\cdot)$ rounds the elements into the nearest integers.

2.1.1. Pixel compensation

Because these two predictors generate different predictions in general, the features of the prediction error histogram varies along with Th . The features include the populations of peak points and how sharply the prediction error histogram is, which indicate the embedding capacity and embedding distortion, respectively. Such variations in prediction error histogram determine the degree of pixel compensation.

The pixel compensation is referred to the decrease of embedding distortion on a single pixel during multiple histogram shifting operations and it satisfies

$$x_{i,j}^k - x_{i,j} \leq x_{i,j}^{k-1} - x_{i,j}, \quad k \geq 2$$

That is to say the new modified pixel value $x_{i,j}^k$ in the k th histogram shifting operation is more closed to original pixel value compared to the previous modified value $x_{i,j}^{k-1}$. Of course, the compensation quantity $x_{i,j}^k - x_{i,j}^{k-1}$ is needed to take into account and will be discussed in the selection of Th . An example of the pixel compensation during the multiple layer embedding is illustrated by Fig. 2. From Fig. 2, it is clearly that after one histogram shifting operation $x_{i,j}$ can be compensated when $Th=0$ in the next histogram shifting operation, but if $Th=100$ the distortion of $x_{i,j}$ is not reduced. We call the above modification of prediction error histogram varying along with the threshold Th as optional prediction method. The purpose of the proposed method is to optimize features of the prediction error histogram by tuning the value of Th and utilizes the pixel compensation

between the multiple histogram shifting operations to improve the performance in turn.

2.1.2. Threshold determination

To illustrate the mechanism that how Th is chosen to maximize the compensation quantity, we will given the description of capacity and distortion for each single layer embedding in the following. The embedding capacity Cap_k can be estimated by the prediction error histogram as

$$\text{Cap}_k = \text{hist}_k(LP) + \text{hist}_k(RP) \tag{9}$$

where $\text{hist}_k(\cdot)$ denotes the population of bins in the prediction error histogram, LP (Left Peak point) and RP (Right Peak point) denote the two highest bins in the prediction error histogram. Of course, such estimation on capacity must follow the previous order (first X_2 and then X_1) because we cannot predict the pixels of X_1 correctly before the embedding procedure of X_2 . Since the histogram shifting operation modifies pixel at most by 1, after k histogram shifting operations, the accumulative distortion impact on the pixel $f(x_{i,j})$ is measured as an impact of MSE (mean square error)

$$f(x_{i,j}; \Psi_1, \dots, \Psi_{k-1}, \Psi_k) = \left(\sum_{l=1}^k \Psi_l(x_{i,j}) \right)^2 \tag{10}$$

where the function $\Psi_l(\cdot) \in \{-1, 0, 1\}$ denotes the modification of pixel in one histogram shifting operation. In this paper, we assumed that the each layer are fully embedded for the simplicity of analysis. Note that when we try to optimize the performance in the k th layer embedding, the distortion of the previous $k-1$ histogram shifting operations is already fixed, so the distortion

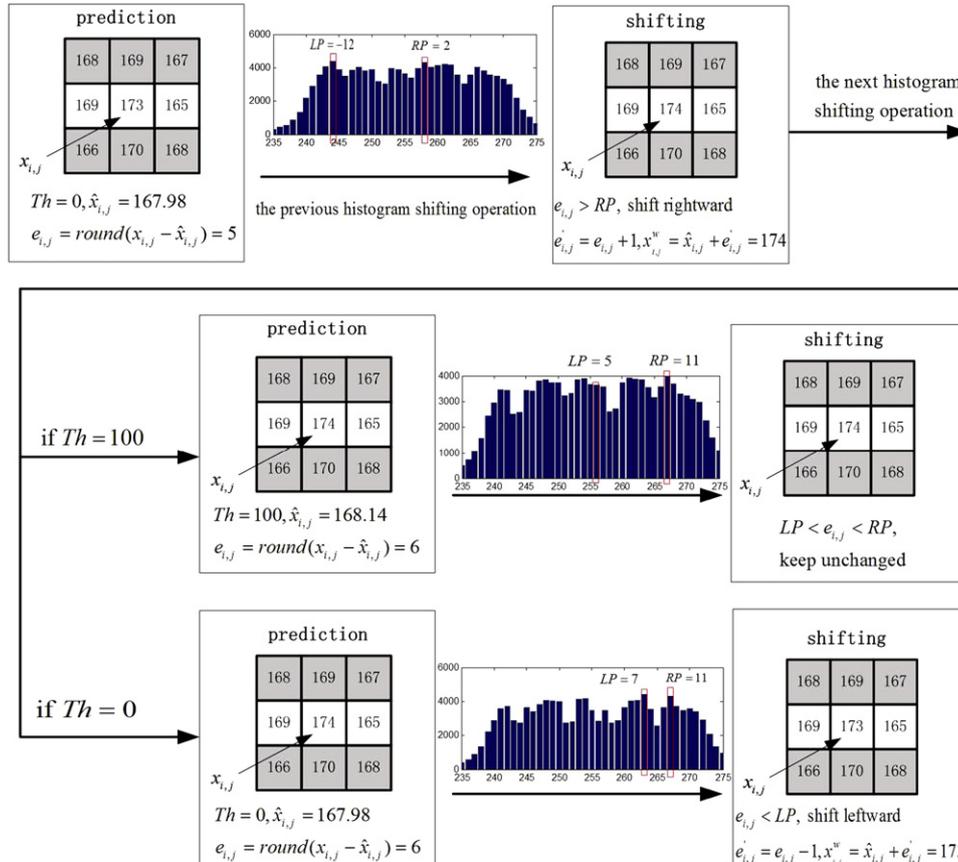


Fig. 2. An example of pixel compensation between the consecutive histogram shifting operation: when $Th=0$ in the next operation, the pixel compensation occurs. LP, RP are the two peak points of prediction error histogram.

of $x_{i,j}$ can also be represented as

$$f(x_{i,j}; \Psi_k | \Psi_1, \dots, \Psi_{k-1}) = f(x_{i,j}; \Psi_1, \dots, \Psi_{k-1}) + \Psi_k^2(x_{i,j}) + 2\Psi_k(x_{i,j}) \cdot \left(\sum_{l=1}^{k-1} \Psi_l(x_{i,j}) \right) \quad (11)$$

Compared with the $k-1$ th histogram shifting operation, the distortion increment $D_k(i,j)$ on $x_{i,j}$ is

$$D_k(i,j) = \Psi_k(x_{i,j})^2 + 2\Psi_k(x_{i,j}) \cdot \left(\sum_{l=1}^{k-1} \Psi_l(x_{i,j}) \right) \quad (12)$$

Provided that multiple layer embedding is required ($k \geq 2$), because the values of $\Psi_1, \dots, \Psi_{k-1}$ are already fixed, the values of $f(x_{i,j}; \Psi_1, \dots, \Psi_{k-1})$ and $(\sum_{l=1}^{k-1} \Psi_l(x_{i,j}))$ are constant values $C_{k-1}^2(i,j)$, $C_{k-1}(i,j)$. Thus, $D_k(i,j) \in \{0, 1-2C_{k-1}(i,j), 1+2C_{k-1}(i,j)\}$. Obviously, the compensation on $x_{i,j}$ occurs only $D_k(i,j)$ is negative and the compensation quantity is equal to $D_k(i,j)$ on this condition. Thus, the relationship of MSE and the total D_k of all the pixels is denoted as

$$\text{MSE} = \sum_i^H \sum_j^W (f(x_{i,j}; \Psi_k | \Psi_1, \dots, \Psi_{k-1}))^2 = \sum_{l=1}^k D_l \quad (13)$$

The best Th for each single layer indicates the corresponding prediction error histogram with least distortion D_k while keeping the same capacity. Of course, note that high payload and low distortion are two conflict requirements for reversible watermarking, the capacity may be reduced along with the decrease of the distortion. To make a trade-off of capacity and pixel compensation, the Th is preferentially selected as

$$Th = \arg \max_{Th_i} \left(\frac{\text{Cap}_k}{D_k} \right) \quad (14)$$

where Cap_k is the capacity in the k th layer.

2.1.3. Watermark embedding and extracting

The watermark embedding and extracting processes are the same as the other HS schemes and formulats (15)–(18) are cited from Luo et al.'s method [5]. The watermark bit $b, b \in \{0, 1\}$, is embedded via (15)

$$e'_{i,j} = \begin{cases} e_{i,j} + \text{symbol}(e_{i,j}) \times b, & e_{i,j} \in LP \text{ or } RP \\ e_{i,j} + \text{symbol}(e_{i,j}) \times 1, & e_{i,j} \in (LN, LP) \text{ or } e_{i,j} \in (RP, RN) \\ e_{i,j} & \text{else} \end{cases} \quad (15)$$

where LN, RN denote the integers with no prediction error satisfying: $e = LN, e = RN$, respectively, and the function $\text{symbol}(\cdot)$ is defined as

$$\text{symbol}(e_{i,j}) = \begin{cases} 1, & e_{i,j} \geq RP \\ -1, & e_{i,j} \leq LP \end{cases} \quad (16)$$

Finally, the watermarked pixel $x_{i,j}^w$ is obtained

$$x_{i,j}^w = \text{round}(\hat{x}_{i,j}) + e'_{i,j} \quad (17)$$

At the receiver, the process of recovery will be operated at the inverse order. If the prediction error equals to LP or RP , a watermark bit "0" is extracted; if $LP-1$ or $RP+1$ is encountered, "1" is extracted. The inverse modification on prediction error is

$$e_{i,j} = \begin{cases} e'_{i,j} - \text{symbol}(e'_{i,j}) \times b, & e'_{i,j} \in [LP-1, LP] \text{ or } e'_{i,j} \in [RP, RP+1] \\ e'_{i,j} - \text{symbol}(e'_{i,j}) \times 1, & e'_{i,j} \in [LNLP-1] \text{ or } e'_{i,j} \in [RP+1, RN] \\ e'_{i,j} & \text{else} \end{cases} \quad (18)$$

Finally, the original pixel is recovered as

$$x_{i,j} = \text{round}(\hat{x}_{i,j}) + e_{i,j} \quad (19)$$

2.2. Algorithm description

To guarantee reversibility, some auxiliary information should be recorded before secret data embedding. For each layer embedding a corresponding auxiliary information L is composed of three parts: the value of Th ; the values of two pairs of peak points, namely LPs, RP s, to tell whether the pixel embeds watermark bit; a location map to classify boundary pixels with the value of 1 or 255 which may cause ambiguities. Note that the marked boundary pixels for each layer embedding will keep unchanged to avoid the over/underflow cause by increment/decrease modification on them. Denote the length of bit stream L after the compression of arithmetic coding as L_s . L_s is embedded in the LSBs (Least Significant Bit) of pixels in the last three rows of the given image by simple LSB replacement. For reversibility, these L_s replaced LSBs will be appended after watermark bit stream P . In addition, because only the last layer may not be fully embedded, an extra information EC , which records the number of layers and how many bits are embedded in the last layer, is appended after L of the last layer for the calculation of capacity. Fig. 3 shows the block diagram of embedding procedure and extracting procedure.

Fig. 4 shows the two extra information formats for the last embedding layer and the other embedding layers. Both of them consist of LP s and RP s ($4 \times 9 = 36$ bits), Th (8 bits), EOS (end of symbol, 8 bits) and the location map. As the size of other parts in format is fixed, it is clearly to distinguish the location map when we extract L . The difference in format is that EC is added in the type B format, which consists of 18 bits ($2^{18} = 512 \times 512$) and 6 bits to illustrate that how many watermark bits and how many layers are embedded, respectively.

2.2.1. Embedding procedure

The embedding procedure is described as follows:

- Step (1) Embed the payload P and the replaced LSBs. For every pixel in embedding area, calculate the prediction error by optional histogram shifting algorithm as discussed in Section 2.1. Then using (15), embed watermark bit in its prediction error and decrement/increment it. The embedding order should be fixed, e.g., first get X_2 and then X_1 . Only after an exhausted embedding in X_2, X_1 can be processed.
- Step (2) Embed auxiliary information L_s . Choose the auxiliary information and form L according to the embedding layer. Embed the compressed auxiliary information L_s into the LSBs of pixels in the boundaries of given images. For standard images, the last three rows of given image are used to restore L_s as its size is only dozens of bits.
- Step (3) Obtain watermarked image. If P is not completed and this layer is full embedded, go to step (1) for next layer embedding; if P is completed, EC is also embedded in the last embedding layer by LSB replacements and the watermarked image is obtained.

2.2.2. Extracting procedure

The extracting procedure is processed as the inverse order:

- Step (1) Extract auxiliary information L . Collect the LSBs of pixels in the specific area of watermarked image and form L after decompression. Differently only the first extracting layer's LSBs contains EC .

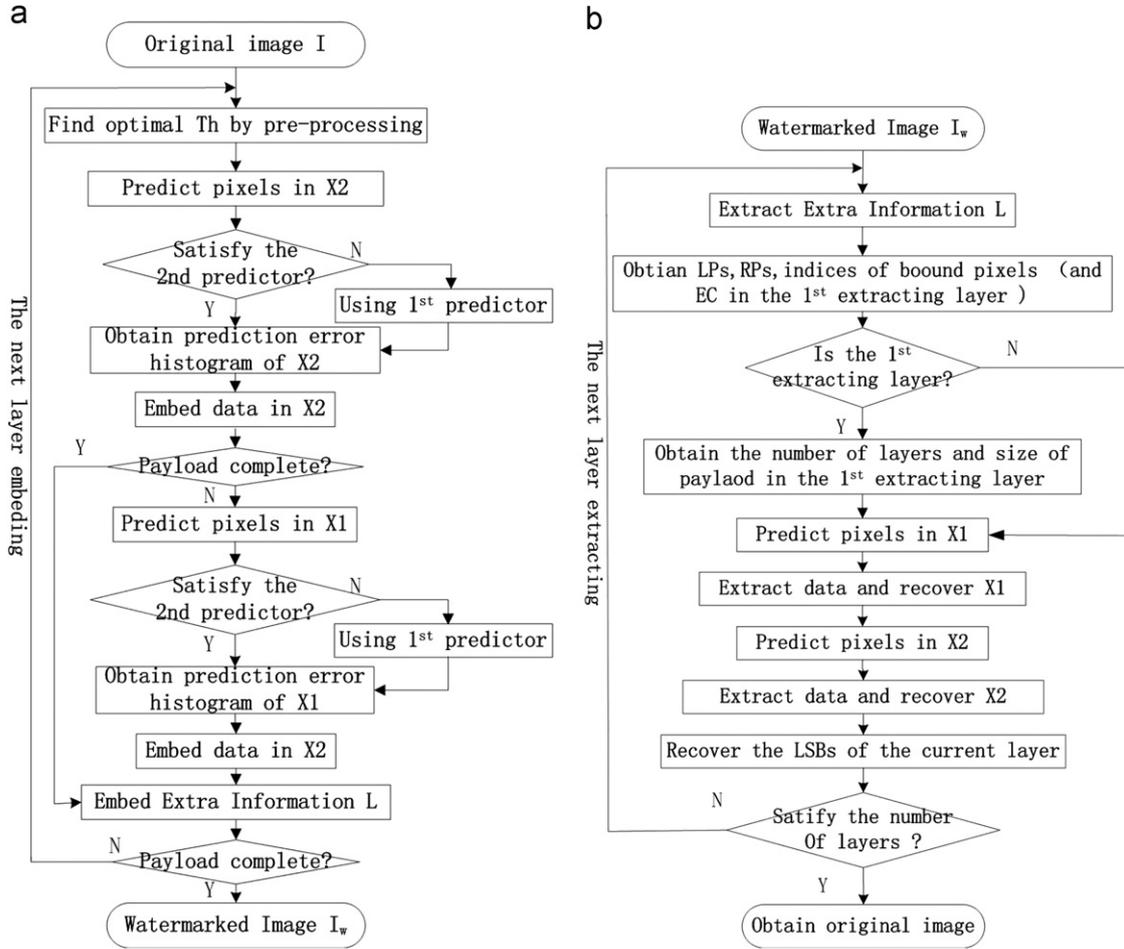


Fig. 3. (a) Embedding procedure. (b) Extracting procedure.

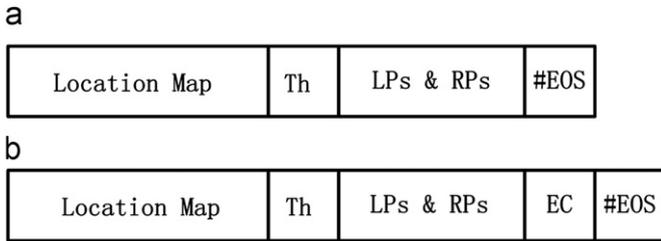


Fig. 4. Extra information formats: (a) in the other embedding layer and (b) in the last embedding layer.

- Step (2) Data extraction and recovery. Predict every pixel in embedding area as the order discussed in Section 2.1, extract watermark bit and recover it using (18) and (19).
- Step (3) Obtain payload P . If the number of extracted watermark bits is equal to the capacity calculated by EC , stop extraction and form P ; else go the next layer extraction and repeat above steps. When the current layer is fully extracted, classify the extracted bit stream into watermark bits and restore the pixels in the last three rows by simple LSB replacement.

3. Experiments

In this section, we will discuss the performance of the proposed method. Fig. 5 gives the comparisons with the previous

methods: Luo et al.'s [5], Sachnev et al.'s [7], Tsai et al.'s [8], and Tai et al.'s [9]. All experiments are tested on commonly used standard images sized 512×512 : lena, baboon, airplane, and sailboat. Compared with these methods, our method can obtain higher PSNR improvement at high payload by utilizing pixel compensation.

Compared with method [5] method, our method achieves better performance for simple texture images, but for complex texture image we are nearly the same. For lena image, when the embedding rate is nearly 0.3 bpp, the PSNR of both methods is almost the same, but as the embedding rate increase, our method outperforms method [5]. When the embedding rate is 1.14 bpp, the improvements are over 1.75 dB. Even the embedding rate is 1.20 bpp, PSNR can still be 31.0 dB high. For baboon image, the improvements on the PSNR of ours is slightly below method [5] at low embedding level. Only when the embedding rate is about 0.3 bpp, ours begins to outperform method [5]. At 0.5 bpp the improvement is about 0.97 dB. For the other two images airplane and sailboat, our method always outperforms the method [5]. The method [7] is based on the prediction error expanding technique. In their scheme, the sorting technique is utilized to reduce the huge distortion caused by expanding technique. At low capacity the method [7] is nearly the same as histogram shifting schemes (the proposed method and the method [5]), but it clearly outperforms in a range of 0.4 ~ 0.8 bpp, especially in smooth image as airplane. However, as the capacity increases, the image fidelity degrades seriously than ours because its expanding distortion is magnified in its second embedding layer than the distortion caused by conventional histogram shifting. In Fig. 5(c) and (d),

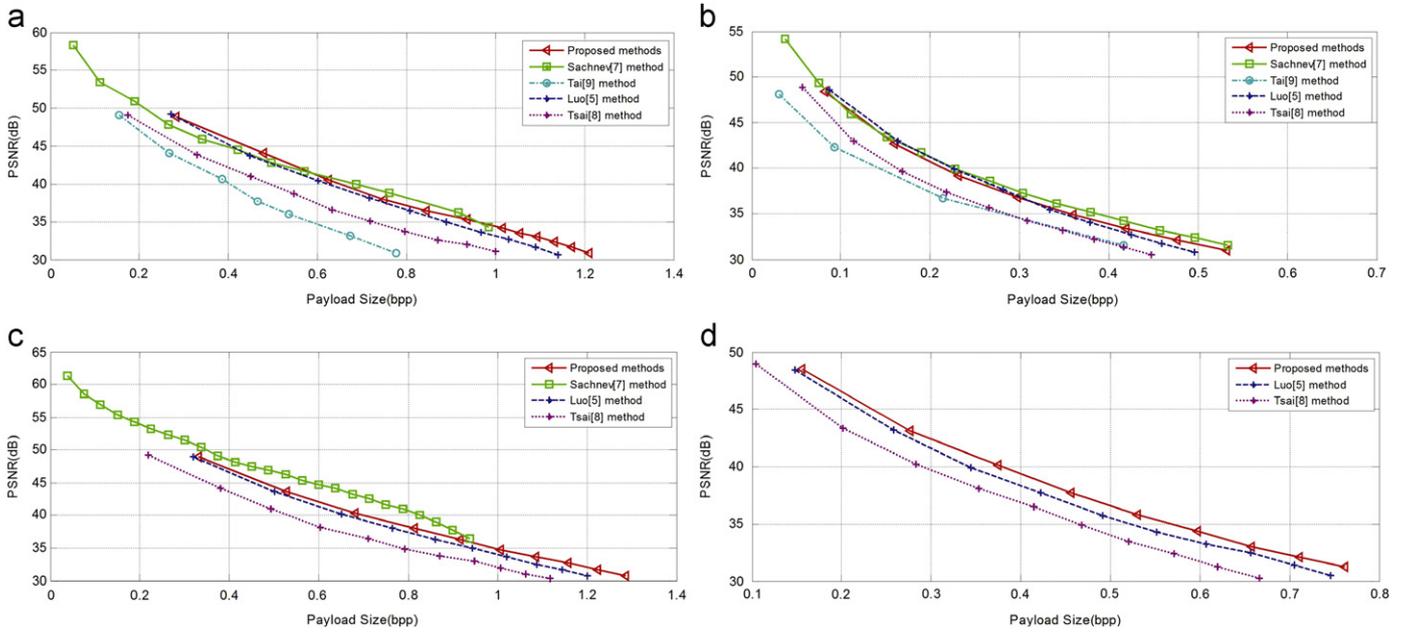


Fig. 5. The performance evaluation of proposed method compared with recent methods over standard test images: (a) lena, (b) baboon, (c) airplane and (d) sailboat.

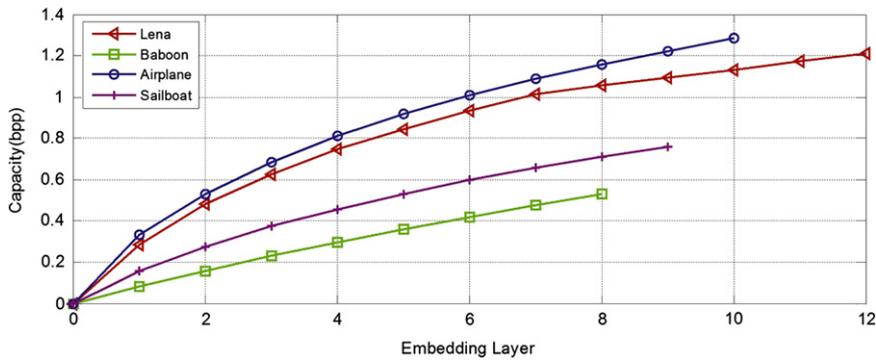


Fig. 6. Capacity versus embedding layers for test images.

the performance curves of method [9] and method [7] are not plotted as these experimental results are not mentioned in their papers. Fig. 6 gives the embedding level versus the pure payload. From Fig. 4, it is seen that the proposed method has better performances on smooth image than texture image, and obtains higher improvements of PSNR at high embedding rate (after multiple layer embedding) than at low embedding rate. Besides, Table 1 gives the detailed experimental results on the four images.

To illustrate the compensation of our method on image quality at the high embedding level, Table 2 gives results with optimal Th compared to the results with four selected values of Th . Since we just consider optimization of the current embedding layer, the experiment is taken on the assumption that the former embedding layers are processed with optimal Th already. Compared with other Th , ΔD is the reduction of distortion for a single layer by utilizing optimal Th . For the same embedding rate, the larger ΔD , the more improvements on PSNR. As Table 2 shown, the stimulation with optimal Th is advantageous as it yields more pixel compensations (the larger value of ΔD) at the same embedding rate. In Table 2, for a single layer the effect of pixel compensation on texture image is less than smooth one as ΔD is smaller on average. This result demonstrates that the pixel compensation definitely improve the quality of watermarked image by reducing

D as indicated in Section 2.1. It is worth noting that ΔD of two different layers are not comparable because the accumulation of $C_{k-1}(i,j)$ amplifies the difference. Besides, as the table shown, the improvement on PSNR is weak for a single layer embedding, but gradually become stronger as it accumulates after multiple layer embedding finally. That is why the proposed method performs better at high embedding rate.

In addition, the disadvantages of previous work are inability to control the values of prediction error. Our methods are able to remove this shortcoming and make the selection mechanism of prediction error histogram into computable criteria. Fig. 7 depicts variations of Cap_k/D_k by tuning the value of Th . Considering there exist many combinations of Cap and D , Th should be iteratively adjusted for 256 times in theory in order to achieve the maximum value of Cap_k/D_k . Here, the value of Cap_k/D_k indicates the balance of higher embedding capacity with less distortion. The largest value of Cap_k/D_k is selected as the optimal selection of Th . A drawback of our simulation is that much more time are consumed compared to the other HS schemes during the embedding process as we considering the optimization of Th . which is implemented on a Intel Core2 Duo 3.00 GHz CPU with 2 G RAM. The computable complexity of our method is $O(EL \cdot N)$, where EL , N are the numbers of embedding layers and the candidates of Th , respectively. It can be seen that the runtime is mainly determined by

Table 1

Detail results of the standard images. *C*, *Th*, *V* and *L_s* refer to the embedding capacity, the selection of threshold, the values of *LPs* and *RP*s and the compressed extra information, respectively. The unit of *C* and *L_s* is bits. “–” refers to the unavailable.

Images	Pure capacity <i>C</i> for the embedding layer $EL=0, \dots, 9$								
	1	2	3	4	5	6	7	8	9
Lena									
<i>C</i>	74,160	125,803	163,709	196,267	221,617	245,262	266,312	276,536	286,759
<i>Th</i>	34 ²	46 ²	18 ²	0	44 ²	0	2 ²	69 ²	0
<i>V</i>	{0, -1, 0, -1}	{3, 2, 3, -3}	{4, 3, 4, -4}	{-1, -2, -2, -3}	{-8, -9, 6, -6}	{8, -4, -9, -10}	{5, 4, 11, 7}	{1, -8, 4, 0}	{13, 9, 3, 1}
<i>L_s</i>	98	130	98	98	130	130	98	98	98
Baboon									
<i>C</i>	21,574	41,812	60,713	78,014	94,450	110,074	124,938	139,487	–
<i>Th</i>	37 ²	36 ²	48 ²	54 ²	54 ²	67 ²	1	9 ²	–
<i>V</i>	{0, -1, 0, -1}	{3, -4, 3, -4}	{6, -7, 6, 5}	{9, 2, 9, -8}	{7, -5, -3, -5}	{-13, -14, -1, -9}	{-8, -9, 16, -18}	{16, -19, 5, 4}	–
<i>L_s</i>	98	98	98	98	98	98	98	98	–
Airplane									
<i>C</i>	86,717	138,569	178,913	213,307	240,570	263,900	284,715	303,353	320,917
<i>Th</i>	86 ²	0	0	0	0	0	0	0	0
<i>V</i>	{1, 0, 1, 0}	{-1, -2, -2, -3}	{2, 1, 4, 3}	{5, -5, 1, 0}	{8, -1, 8, -2}	{-3, -4, -5, -6}	{-7, -11, -9, -10}	{3, -9, 7, -13}	{6, 1, 5, 4}
<i>L_s</i>	258	354	370	514	570	658	754	1,130	1,114
Sailboat									
<i>C</i>	40,868	72,457	98,147	119,721	138,936	156,474	172,307	186,332	199,544
<i>Th</i>	52 ²	54 ²	0	79 ²	0	0	0	0	0
<i>V</i>	{1, 0, 1, 0}	{-2, -3, -3, -4}	{5, 4, 5, 4}	{-6, -7, -1, -2}	{2, 1, 9, 2}	{5, -1, -9, -10}	{-4, -5, -6, -7}	{-8, -15, 9, -15}	{14, 13, 6, -3}
<i>L_s</i>	130	130	130	130	130	130	130	130	130

Table 2

The improvements of PSNR between the optimal selection of *Th* and four selected value of *Th*. *EL*, *ER*, ΔP and ΔD refer to the sequence number of embedding layer, embedding rate in a single layer, improvements of PSNR and reductions of *D*, respectively. “–” refers to the value of *Th* is optimal already.

<i>EL</i>	<i>Th</i>	Lena			Baboon			Airplane			Sailboat		
		<i>ER</i>	ΔP	ΔD									
2	0	0.197	0.3309	56,980	0.0702	0.2523	64,762	–	–	–	0.1185	0.0743	27,595
	50 ²	0.1967	0.0042	1073	0.0728	0.0012	870	0.1947	0.0933	16,785	0.1205	0.0026	332
	100 ²	0.1969	0.0035	606	0.0728	0.0042	934	0.1947	0.0967	17,073	0.1201	0.0102	2391
	150 ²	0.1967	0.0055	1214	0.0729	0.0034	517	0.1948	0.0934	16,696	0.1201	0.0108	2090
3	0	0.1431	0.0227	8065	0.059	0.0889	44,681	–	–	–	–	–	–
	50 ²	0.1446	0.0047	1104	0.0614	0.0011	746	0.1476	0.0534	9838	0.0959	0.1501	44,783
	100 ²	0.1446	0.0064	1511	0.0615	0.0025	983	0.1475	0.0602	12,560	0.0961	0.1461	42,459
	150 ²	0.1446	0.0066	1537	0.0615	0.0025	957	0.1475	0.0604	12,529	0.0961	0.1464	42,915
4	0	–	–	–	0.0513	0.183	153,313	–	–	–	0.0817	0.0111	7890
	50 ²	0.1239	0.0008	3236	0.053	0.0013	1150	0.1268	0.0325	16,315	0.0819	0.0111	7689
	100 ²	0.1236	0.006	7046	0.053	0.0016	1357	0.1265	0.0436	23,640	0.0823	0	351
	150 ²	0.1236	0.0059	6895	0.053	0.0014	782	0.1264	0.069	22,585	0.0823	0.0006	300
5	0	0.0967	0.0778	71,892	0.0459	0.1059	123,998	–	–	–	–	–	–
	50 ²	0.0965	0.0629	54,256	0.0471	0.0006	370	0.0937	0.5093	564,790	0.0716	0.0226	27,825
	100 ²	0.0966	0.0622	53,808	0.0471	0.001	458	0.0936	0.5144	569,535	0.0715	0.0275	33,000
	150 ²	0.0965	0.0645	56,094	0.0471	0.0011	962	0.0936	0.4662	535,989	0.0715	0.0276	33,101
6	0	–	–	–	0.0428	0.0345	22,806	–	–	–	–	–	–
	50 ²	0.0833	0.3064	383,800	0.0442	0.0001	289	0.0848	0.0337	47,263	0.0617	0.0963	156,123
	100 ²	0.0855	0.2841	353,034	0.0442	0.0005	594	0.0848	0.0375	52,705	0.0621	0.091	151,804
	150 ²	0.0844	0.2942	368,923	0.0442	0.0005	870	0.0848	0.0386	52,817	0.0619	0.0937	155,542

EL and *N*, but is irrelevant the content of image. Of course, the runtime of high resolution images is more than the runtime of low high resolution images as the much more pixels are processed. Table 3 gives the average runtime comparisons of embedding process for the different value of *EL* and *N* on the sized 512 × 512 standard images. From the table, the average runtime

of a single layer with $n=50, 100, 150, 256$ are 0.88 min, 1.77 min, 2.60 min, 4.38 min, respectively. In fact, through our experiments, the most of neighbor differences fall into the range of 0–100 and the optimal results are determined by the prediction errors with the most populations. So the selection of *Th* can be limited in the range of $\{1^2, 2^2, 3^2, \dots, 100^2\}$ in practical applications to reduce

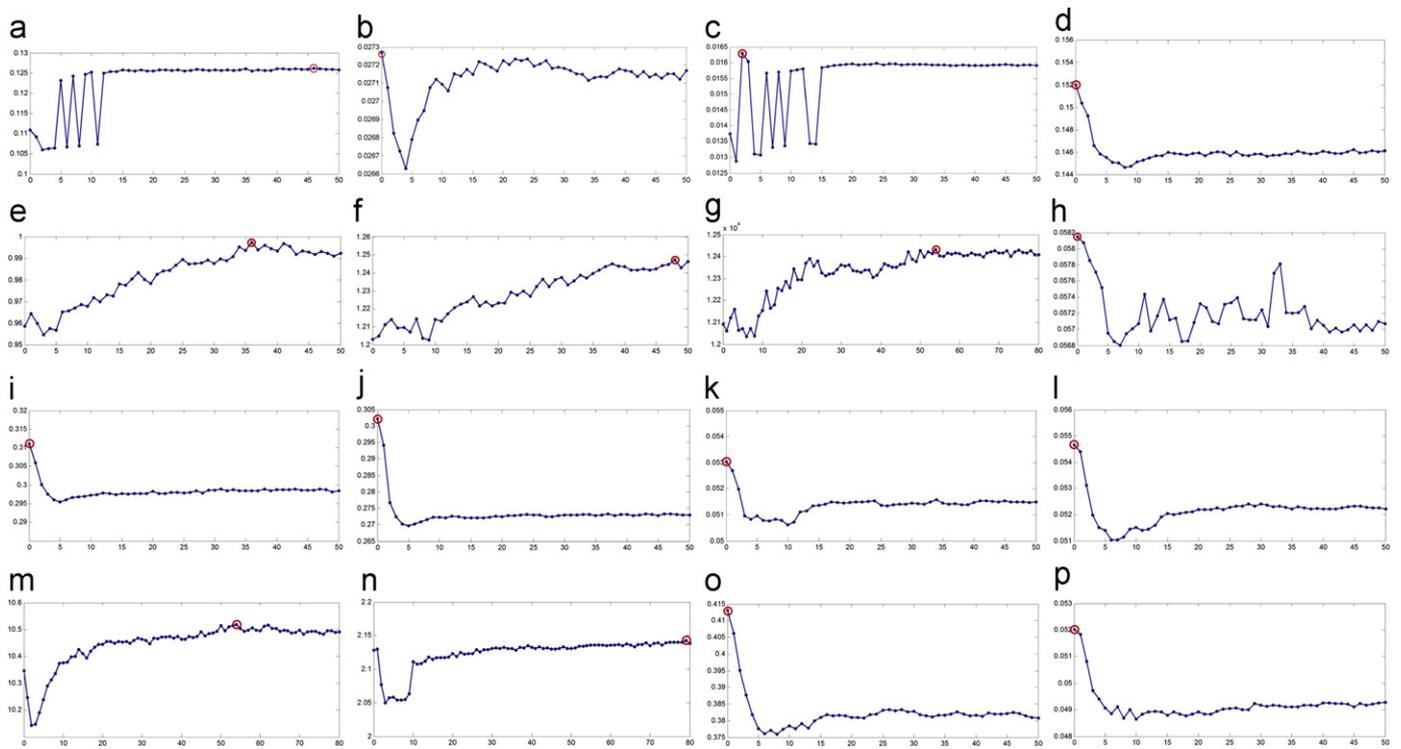


Fig. 7. Corresponding Cap_k/D_k (y -axis) for a range of Th (x -axis) at different embedding rate: (a–d) lena, (e–h) baboon, (i–l) airplane and (m–p) lake. The optimal selection of Th is marked by a circle. (a) 0.48 bpp, (b) 0.75 bpp, (c) 1.0 bpp, (d) 1.2 bpp, (e) 0.16 bpp, (f) 0.27 bpp, (g) 0.32 bpp, (h) 0.49 bpp, (i) 0.68 bpp, (j) 0.92 bpp, (k) 1.16 bpp, (l) 1.29 bpp, (m) 0.28 bpp, (n) 0.46 bpp, (o) 0.6 bpp, and (p) 0.76 bpp.

Table 3

The average runtime comparisons of embedding process for different value of EL and N on the sized 512×512 standard images. The unit of runtime is minute (min).

N	EL						
	1	2	3	4	5	6	7
50	0.88	1.76	2.65	3.53	4.4	5.26	6.17
100	1.81	3.59	5.39	7.16	8.92	10.63	12.38
150	2.61	5.16	7.77	10.38	13.03	15.61	18.23
256	4.51	8.85	13.25	17.59	21.87	26.25	30.66

Table 4

The runtime comparisons of a single layer for tested images (lena, baboon, airplane, and sailboat). The unit of runtime is second (s).

Images	Proposed method		Luo et al.'s [5]	Sachnev et al.'s [7]
	$Th=100$	$Th=150$		
Lena	103.90	152.45	5.37	10.14
Baboon	99.14	148.54	5.23	9.44
Airplane	102.69	162.71	4.85	9.07
Sailboat	103.07	157.36	4.79	10.22

the more than half of the computable complexity. In addition, the watermark image for such a high embedding rate is not often necessary in commercial use. So, 5–7 layers are enough for the size of payload. Table 4 shows the runtime comparisons with Luo et al.'s [5] and Sachnev et al.'s [7] for a single layer on tested image. It can be seen that the runtime of Luo et al.'s [5] and Sachnev et al.'s [7] are far more less than ours as their theoretical computation complexity is $O(1)$.

At last, another experiment is tested on the ten military images as Fig. 8 shown and the curve of performances are plotted

in Fig. 9. The average embedding rate is 0.814 bpp with 31.32 dB on average. It illustrates that our method definitely satisfies the practical applications because 0.5 bpp for the sized 512×512 image is equaled to 130k bits, not mention to the high-resolution images in the internet. Fig. 10 shows the gain in PSNR of proposed method comparing with Luo et al.'s method [5] for 150 gray-scale images. All the tested images are downloaded from the USC-SIPI image database at the web site of <http://sipi.usc.edu/database/>. In principle, Luo's method is kind of HS scheme, which is very similar with ours as it need to multiple layer embedding to obtain high embedding rate. Different from their embedding process, our method put emphasize on the optimization of embedding process by utilizing pixel compensation during multiple layer embedding and hence to reduce the distortion at the high embedding rate. That is to say, our method is better than Luo's at high embedding rate because the effect of pixel compensation become stronger as the layers increase. The fact is confirmed by our experiments as shown in Fig. 10. It is clearly that our method gains more improvements of PSNR at high embedding rate compared with Luo et al.'s [5]. In addition, a phenomenon is found through the abundant experiments that our method performs better for the kind of images which need multiple layer embedding for the high payload. But for the too flat or two complex images, the difference of performances between ours is very small. The reason is that for the too flat images only the high embedding rate as 1.0 bpp is easily achieved by 2–3 layers embedding and for the complex images the correlation of neighborhood pixels are too weak to modified the prediction-error histogram by tuning Th .

4. Conclusion

In this paper, a reversible watermarking scheme using optional prediction error histogram modification is proposed. Two different predictors adaptively selected by Th are employed to make



Fig. 8. The 10 military images used for the experiments: A1–A10.

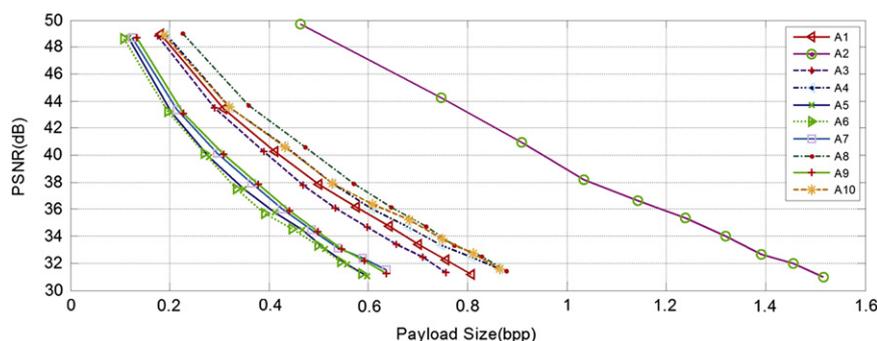


Fig. 9. The performances of the 10 military images.

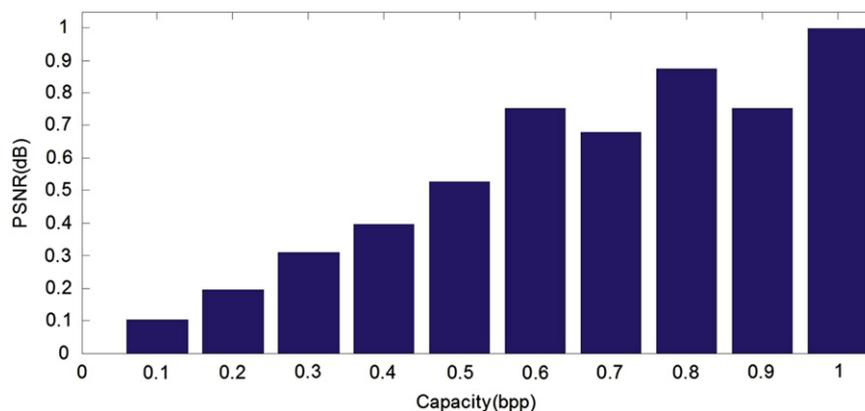


Fig. 10. Gain in PSNR of proposed method comparing with Luo's method [5] at 10 specific capacity (0.1 bpp ~ 1.0 bpp), for 150 gray-scale images.

the selection mechanism of prediction errors into computable criteria, and higher PSNR at high embedding rate can be achieved by utilizing pixel compensation and optimizing the prediction error histogram during multiple layer embedding process.

Acknowledgment

This work was supported in part by 973 Program (2011CB302204), National Natural Science Funds for Distinguished Young Scholar (61025013), National NSF of China (61073159),

Sino-Singapore JRP (2010DFA11010), and Fundamental Research Funds for the Central Universities (2012JBM042).

References

- [1] C.-C. Chang, T.D. Kieu, A reversible data hiding scheme using complementary embedding strategy, *Inf. Sci.* 180 (16) (2010) 3045–3058.
- [2] Y. Hu, H.-K. Lee, J. Li, DE-based reversible data hiding with improved overflow location map, *IEEE Trans. Circ. Syst. Video Technol.* 19 (2) (2009) 250–260.
- [3] H.-J. Kim, V. Sachnev, Y.Q. Shi, J. Nam, H.-G. Choo, A novel difference expansion transform for reversible data embedding, *IEEE Trans. Inf. Forensic Secur.* 3 (3) (2008) 456–465.

- [4] C.C. Lin, S.C. Chen, N.L. Hsueh, Adaptive embedding techniques for VQ-compressed images, *Inf. Sci.* 179 (1–2) (2009) 140–149.
- [5] L.X. Luo, Z.Y. Chen, M. Chen, X. Zeng, Z. Xiong, Reversible image watermarking using interpolation technique, *IEEE Trans. Inf. Forensics Secur.* 5 (1) (2010) 187–193.
- [6] Z. Ni, Y.Q. Shi, N. Ansari, S. Wei, Reversible data hiding, *IEEE Trans. Circ. Syst. Video Technol.* 16 (3) (2006) 354–362.
- [7] V. Sachnev, H.-J. Kim, J. Nam, S. Suresh, Y.Q. Shi, Reversible watermarking algorithm using sorting and prediction, *IEEE Trans. Circ. Syst. Video Technol.* 19 (7) (2009) 989–999.
- [8] P. Tsai, Y.C. Hu, H.L. Yeh, Reversible image hiding scheme using predictive coding and histogram shifting, *Signal Process.* 89 (2009) 1129–1143.
- [9] W.-L. Tai, C.-M. Yeh, C.-C. Chang, Reversible data hiding based on histogram modification of pixel differences, *IEEE Trans. Circ. Syst. Video Technol.* 19 (6) (2009) 906–910.
- [10] H.W. Tseng, C.P. Hsieh, Prediction-based reversible data hiding, *Inf. Sci.* 179 (14) (2009) 2460–2469.
- [11] S.W. 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.
- [12] 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.
- [13] J. Hwang, J. W. Kim, J. U. Choi, A reversible watermarking based on histogram shifting, in: *Int. Workshop on Digital Watermarking, Lecture Notes in Computer Science*, vol. 4283, 2006, pp. 348–361.
- [14] M. Chen, Z. Chen, X. Zeng, Z. Xiong, Reversible data hiding using additive prediction-error expansion, in: *Proceeding of the 11th Workshop on Multimedia and Security*, 2009, pp. 19–24.
- [15] M. Fallahpour, Reversible image data hiding based on gradient adjusted prediction, *IEICE Electron. Express* 5 (20) (2008) 870–876.
- [16] K.S. Kim, M.J. Lee, H.Y. Lee, H.K. Lee, Reversible data hiding exploiting spatial correlation between sub-sampled images, *Pattern Recognition* 42 (11) (2009) 3083–3096.
- [17] M. Fallahpour, Reversible image data hiding based on gradient adjusted prediction, *IEICE Electron. Express* 5 (20) (2008) 870–876.
- [18] M. Fallahpour, D. Megias, M. Ghanbari, Subjectively adapted high capacity lossless image data hiding based on prediction errors, *Multimedia Tools Appl.* 52 (2–3) (2011) 513–527.
- [19] A.M. Alattar, Reversible watermark using the difference expansion of a generalized integer transform, *IEEE Trans. Image Process.* 13 (8) (2004) 1147–1156.
- [20] D. Coltuc, A. Tremeau, Simple reversible watermarking schemes, in: *Proc. SPIE, Security, Steganography, Watermarking of Multimedia Contents*, vol. 5681, 2005, pp. 561–568.
- [21] S. Lee, C.D. Yoo, T. Kalker, Reversible image watermarking based on integer-to-integer wavelet transform, *IEEE Trans. Inf. Forensic Secur.* 2 (3) (2007) 321–330.
- [22] D. Coltuc, J.M. Chassery, Very fast watermarking by reversible contrast mapping, *IEEE Signal Process. Lett.* 14 (4) (2007) 255–258.
- [23] X. Wang, X. Li, B. Yang, Z. Guo, Efficient generalized integer transform for reversible watermarking, *IEEE Signal Process. Lett.* 17 (6) (2010) 567–570.
- [24] C.C. Chang, T.D. Kieu, W.-C. Wu, A lossless data embedding technique by joint neighboring coding, *Pattern Recognition* 42 (7) (2009) 1597–1603.
- [25] L. An, X.-B. Gao, Y. Yuan, D.-C. Tao, Robust lossless data hiding using clustering and statistical quantity histogram, *Neurocomputing* 77 (1) (2012) 1–11.
- [26] L. An, X.-B. Gao, Y. Yuan, D.-C. Tao, C. Deng, F. Ji, Content-adaptive reliable robust lossless data embedding, *Neurocomputing* 79 (1) (2012) 1–11.
- [27] X. Gao, L. An, Y. Yuan, D. Tao, X. Li, Lossless data embedding using generalized statistical quantity histogram, *IEEE Trans. Circ. Syst. Video Technol.* 21 (8) (2011) 1061–1070.



Bo Ou was born in Hunan, China in 1985. He received the BS degree in Beijing Jiaotong University, Beijing, China in 2008. He is currently pursuing the PhD degree at the Institute of Information Science, Beijing Jiaotong University. His research interests include image processing, digital watermarking, etc.



Yao Zhao received the BE degree from Fuzhou University in 1989 and the ME degree from the Southeast University in 1992, both from the Radio Engineering Department, and the PhD degree from the Institute of Information Science, Beijing Jiaotong University (BJTU) in 1996. He became an associate professor at BJTU in 1998 and became a professor in 2001. From 2001 to 2002, he worked as a senior research fellow in the Information and Communication Theory Group, Faculty of Information Technology and Systems, Delft University of Technology, Netherlands. He is now the director of the Institute of Information Science, Beijing Jiaotong University. His research interests include image/video coding, fractals, digital watermarking, and content based image retrieval. Now he is leading several national research projects from 973 Program, 863 Program, the National Science Foundation of China, and Fok Ying Tong Education Foundation. He is a member of the IEEE.



Rongrong Ni was born in 1976, and received her PhD from Institute of Information Science at Beijing Jiaotong University in April 2005. Her research interests include image processing, data hiding and digital watermarking, pattern recognition, computer vision, etc. Now she is in charge of a NSFC (Natural Science Foundation of China) project and a Beijing NSF (Natural Science Foundation) project. In addition, she participates in 973 and 863 projects as the backbone. She publishes more than 30 papers, and applies four national patents.