The Journal of Systems and Software 86 (2013) 2700–2709

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

Reversible data hiding based on PDE predictor

Bo Ou[a], Xiaolong Li[b], Yao Zhao[a,*], Rongrong Ni[a]

[a] Institute of Information Science, Beijing Jiaotong University, Beijing 100044, China
[b] Institute of Computer Science and Technology, Peking University, Beijing 100871, China

ARTICLE INFO

Article history:
Received 25 May 2012
Received in revised form 24 April 2013
Accepted 18 May 2013
Available online 12 June 2013

Keywords:
Reversible data hiding
Prediction-error expansion (PEE)
Partial differential equation(PDE)

ABSTRACT

In this paper, we propose a prediction-error expansion based reversible data hiding by using a new predictor based on partial differential equation (PDE). For a given pixel, PDE predictor uses the mean of its four nearest neighboring pixels as initial prediction, and then iteratively updates the prediction until the value goes stable. Specifically, for each pixel, by calculating the gradients of four directions, the direction with small magnitude of gradient will be weighted larger in the iteration process, and finally a more accurate prediction can be obtained. Since PDE predictor can better exploit image redundancy, the proposed method introduces less distortion for embedding the same payload. Experimental results show that our method outperforms some state-of-the-art methods.

1. Introduction

Reversible data hiding is a special information hiding technique which enables decoder to perfectly recover original digital content after the extraction of hidden message. Generally, reversible data hiding possess no robustness as it is fragile to attacks. It can be utilized for the purposes of copyright protection, ownership allegation and integrity authentication, especially in military, medical and judicial fields where any permanent distortion of original content is unacceptable.

Initial reversible data hiding algorithms (referred as Type-I) are based on lossless compression (Fridrich et al., 2001, 2002; Celik et al., 2005, 2006; Qian and Zhang, 2012). The main idea of these algorithms is to losslessly compress images to create vacant space for data embedding, and the embedding capacity is determined by compression ratio. However, the performance of this type of algorithms is not satisfactory since the introduced distortion is usually large.

Another type of reversible data hiding algorithms referred as Type-II (Tian, 2003; Alattar, 2004; Kamstra and Heijmans, 2005; Kim et al., 2008) is based on difference expansion (DE) technique. DE was first proposed by, in which the original image is divided into low-pass and high-pass bands, and a data bit is embedded into a pixel pair by expanding the difference between the two pixels in a pair. To prevent overflow/underflow, e.g., restrict the marked pixel value in the range [0, 255] for a gray-scale image, a location map is constructed to mark pixel pairs that may cause overflow/underflow problem. The location map needs to be embedded into cover image as well, and would consume the embedding capacity.

Type-III algorithms are based on prediction-error expansion (PEE) technique (Thodi and Rodriguez, 2007; Hu et al., 2009; Sachnev et al., 2009; Tai et al., 2009; Hong et al., 2009, 2010; Luo et al., 2010; Li et al., 2011; Coltuc, 2011). PEE was first proposed by, in which a pixel is first predicted by its context, and then the difference between the prediction and its original value is used for data embedding. PEE has drawn much attention due to the excellent embedding performance and capacity control capability. Sachnev et al. (2009) sorted pixels according to the local variance in order to process smooth pixels first. By using the sorted prediction-errors, the embedding distortion is significantly reduced especially at low capacities. Recently, Li et al. (2011) proposed a novel PEE algorithm by using adaptive embedding. Different from other PEE algorithms, it adaptively embeds two bits into expandable pixels with low local complexities, and one bit to those with high local complexities. Consequently, it can achieve a higher capacity for the same embedding distortion.

Type-IV algorithms are based on histogram shifting (HS) technique (Ni et al., 2006; Lee et al., 2006; Gao et al., 2009, 2011; Feng and Fan, 2012; An et al., 2012) which is first proposed by Ni et al. (2006). In Ni et al.'s method, data bits are embedded into the pixels with the most population in histogram, and the embedding capacity equals to the number of these pixels. Gao et al. (2009) divided image into non-overlapped blocks, and then selected the suitable blocks for data embedding. In a recent method, Gao et al. (2011) proposed a new method based on the statistical quantity histogram to ensure a stable performance for diverse images.

* Corresponding author. Tel.: +86 010 51688667; fax: +86 010 51688667.
E-mail addresses: 09112055@bjtu.edu.cn (B. Ou), lixiaolong@pku.edu.cn (X. Li), yzhao@bjtu.edu.cn (Y. Zhao), rrni@bjtu.edu.cn (R. Ni).

0164-1212/$ – see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jss.2013.05.077

Besides, another type of algorithms, namely, integer-transform based reversible data hiding (Lee et al., 2007; Weng et al., 2008; Wang et al., 2010; Coltuc, 2012), also attracts a lot of attention. Compared with other algorithms, the main advantage of this type of algorithms is that the embedding capacity is much larger, e.g., usually more than one bit per pixel (bpp). Weng et al. (2008) proposed an integer transform by utilizing the invariability of pixel pair. Through pairwise difference adjustment, a highly compressible location map can be generated, and hence the capacity used for secret message is increased. Wang et al. (2010) proposed a more generalized integer transform and introduced a payload-independent location map. Most recently, Coltuc (2012) proposed a novel integer transform for reversible data hiding, in which data bits are simultaneously embedded into the current pixel and its context pixels.

Among the above techniques, PEE is widely accepted nowadays as it provides an effective way for data embedding. However, the predictors employed in current PEE methods are not image-dependent since they assign fixed weights to the context pixels when calculating predictions, and therefore cannot ensure accurate predictions for various images. Seen form this point of view, PEE can be further improved.

In this paper, we propose a PEE based reversible data hiding by designing a content-dependent predictor based on partial differential equation (PDE), namely PDE predictor, to achieve accurate predictions for various images. Compared with other predictors, PDE predictor can adaptively assign weights to context pixels according to local changes, and can better exploit image redundancy. As PDE evolution is constrained by a defined parameter, the prediction at decoder can be guaranteed to be same as the one at encoder. Since the derived prediction-error histogram is sharper, the proposed method demonstrates a superior capacity–distortion performance. Experimental results verify that the capacity–distortion performance of the proposed method is better than some state-of-the-art works. Besides, its wide applicability has been explored as well at the end of the paper.

The rest of paper is organized as follows. In Section 2, PEE principles are briefly introduced. The proposed algorithm is presented in details in Section 3. The experimental results and analysis are given in Section 4. Finally, Section 5 concludes this paper.

2. Preliminary

The basic mechanism of PEE can be summarized as follows. At encoder, for each pixel x, the prediction value $\hat{x}$ is obtained using a predictor. Then, the integer-valued prediction-error e is calculated as $e = x - \text{round}(\hat{x})$ where the function round$(\cdot)$ rounds the element into its nearest integer.

The corresponding prediction-error histogram is formed by counting the frequencies of prediction-errors. Usually, prediction-error histogram is a Laplacian-like distribution with the highest bin at zero or close to zero. According to the prediction-error histogram, prediction-errors can be classified into two sets: the inner-region errors $e \in [T_l, T_r)$ and the outer-region errors $e \in (-\infty, T_l) \cup [T_r, +\infty)$, where T_l and T_r $(T_l < 0 < T_r)$ are two integer-valued thresholds determined by the required capacity. For each prediction-error e, if $e \in [T_l, T_r)$, it will be embedded data through expansion

$$e^w = 2e + w$$

where $w \in \{0, 1\}$ is a to-be-embedded data bit. Accordingly, the marked pixel value x^w is obtained as

$$x^w = \text{round}(\hat{x}) + e^w = x + e + w.$$

If $e \in (-\infty, T_l) \cup [T_r, +\infty)$, it will be shifted as

$$x^w = \begin{cases} x + T_r, & \text{if} \quad e \geq T_r \\ x + T_l, & \text{if} \quad e < T_l \end{cases}.$$

Through expansion/shifting operations, the inner region and outer region are changed to $[2T_l, 2T_r)$ and $(-\infty, 2T_l) \cup [2T_r, +\infty)$, respectively. The average distortion in terms of mean square error (MSE) for a pixel is

$$D = \begin{cases} \frac{1}{2}[e^2 + (e+1)^2] = e^2 + e + \frac{1}{2}, & \text{if} \quad e \in [T_l, T_r) \\ T_r^2, \text{if} \quad e \in [T_r, +\infty) \\ T_l^2, \text{if} \quad e \in [-\infty, T_l) \end{cases}.$$

Institutively, D is related to the concentration degree of prediction-error histogram. That is, the more concentrated the prediction-error histogram is, the less embedding distortion is for embedding the same size of data bits. After secret data bits are embedded completely, the marked image is obtained.

Notice that in the above embedding process, for a gray-scale image, some pixel values may be outside the range [0, 255]. To prevent this overflow/underflow problem, a location map would be constructed, in which the pixels which may be out of the range after modification are marked with “1”, while the other pixels are marked with “0”. Then the location map is appended after/before the message bits *MS*. In general, the location map would be losslessly compressed into a smaller size L_s. So the actual size of payload is $|L_s| + |MS|$.

At decoder, for each pixel, the original prediction is retrieved by using the same context pixels, and the marked prediction-error is calculated as $e^w = x^w - \text{round}(\hat{x})$. Then, the recovery of pixels and data extraction are processed as follows.

If $e^w \in [2T_l, 2T_r)$, the original pixel is recovered as $x = \text{round}(\hat{x}) + \lfloor e^w/2 \rfloor$, where $\lfloor \cdot \rfloor$ is the floor function. The data bit w is extracted as the least significant bit (LSB) of marked prediction-error $w = LSB(e^w)$, where the function $LSB(\cdot)$ denotes the LSB of an element.

If $e^w \in (-\infty, 2T_l) \cup [2T_r, +\infty)$, there is no data bit to be extracted, and the original pixel is restored through the inverse shifting

$$x = \begin{cases} x^w - T_r, & \text{if} \quad e^w \geq 2T_r \\ x^w - T_l, & \text{if} \quad e^w < 2T_l \end{cases}.$$

After all the embedded data is extracted, the cover image is recovered accordingly.

3. Proposed algorithm

It is well known that an accurate prediction method leads to an effective PEE based reversible data hiding scheme. Usually, the prediction of a pixel is obtained as the weighted mean of its context pixels

$$\hat{x} = \sum_{i=1}^{k} w_i x_i \tag{1}$$

where the context of x includes $\{x_1, \ldots, x_k\}$, and w_i denotes the weighted value for x_i. Then, to achieve an more accurate prediction, it is to minimize l^2-error between the original and predicted values

$$\begin{aligned} &\underset{0 \leq w_1, \ldots, w_k \leq 1}{arg \min} \left\| x - \hat{x} \right\|_2^2 \\ &\text{subject to} \sum_{i=1}^{k} w_i = 1. \end{aligned} \tag{2}$$

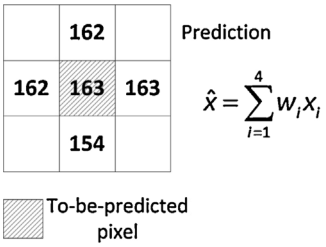


Fig. 1. Example of pixel prediction.

In this light, finding optimal weighted values $\{w_1^*, \ldots, w_k^*\}$ becomes the key issue of prediction. Conventional predictors set the fixed weights for context pixels to predict each pixel, but may not be accurate when predicting the pixel located in a rough region. For example, as shown in Fig. 1, we use four rhombus neighbor pixels $\{x_1, x_2, x_3, x_4\} = \{162, 162, 163, 154\}$ to predict the center one $x = 163$. By mean prediction method in Sachnev et al. (2009), it is predicted as $(162 + 162 + 163 + 154)/4 = 160.25$ by equally assigning weights as $w_1 = w_2 = w_3 = w_4 = 0.25$, and the l^2-error is 7.56. Evidently, in this case, such assignments are not reasonable as the context pixel 154 is less correlated with the center one and should be weighted less. A natural idea to improve the prediction accuracy is to give larger weight to the more similar context pixels, e.g., by setting $w_4 = 0.1$ and $w_1 = w_2 = w_3 = 0.3$, the prediction is 161.5 and the l^2-error is reduced to 2.25.

For accurate prediction, the weighted values to context pixels should be adaptively determined by image content. We argue that such content-dependent predictor can better exploit image redundancy, and hence yields a superior embedding performance for PEE than content-independent predictors. In the following, we will design a content-dependent predictor based on PDE to get a more accurate prediction for each pixel by considering the image's own characteristic.

This section is organized as follows. The PDE predictor is introduced in Section 3.1. An improved two-pass testing to prevent overflow/underflow problem is given in Section 3.2. The parameter determination is discussed in Section 3.3. Finally, the details of embedding and extracting procedure are described in Sections 3.4 and 3.5, respectively.

3.1. PDE Predictor

PDE is an effective mathematical tool proposed by Perona and Malik (1990) in the work, in which it is used to design a high quality edge detector in the scale-space. The general idea of PDE is to implement anisotropic diffusion by encouraging intra-region smoothing in preference to inter-region smoothing. As for the prediction in PEE, such property can also be utilized to make context pixels with high correlations being weighted larger than the ones with low correlations. Here, we use four nearest pixels to predict the center one, and its PDE prediction is calculated as

$$\begin{aligned} x_{i,j}^{t+1} &= x_{i,j}^t + \lambda \sum_D (c_D \nabla_D x_{i,j}^t) \\ &= x_{i,j}^t + \lambda (c_N \nabla_N x_{i,j}^t + c_S \nabla_S x_{i,j}^t + c_W \nabla_W x_{i,j}^t + c_E \nabla_E x_{i,j}^t) \end{aligned} \quad (3)$$

where $D \in \{N, S, W, E\}$ denotes the gradient direction (north, south, west and east), t is the iteration number, (i, j) denotes the pixel location in image, λ $(0 < \lambda \leq 0.25)$ is used to make the prediction stable after numerical iterations, ∇ calculates the gradient of each direction, and c_D is the weighted value for a gradient direction. For PDE prediction, the initial value of prediction is important and will ultimately affect the accuracy. Here, we choose the mean value of four neighboring pixels as the initial estimation $(t = 0)$, i.e., we take

$$x_{i,j}^0 = \frac{x_{i-1,j} + x_{i+1,j} + x_{i,j-1} + x_{i,j+1}}{4}. \quad (4)$$

Table 1
Comparison of the entropies using MED, mean-value, IP and PDE predictors.

Images	Lena	Baboon	Airplane	Barbara	House	Tiffany
MED	3.1520	4.3460	2.9032	3.7032	3.1496	3.0717
Mean-value	2.8752	4.1469	2.7282	3.5663	3.0786	2.8952
IP	3.0344	4.3259	2.9736	3.7399	3.3848	3.0080
PDE	2.8204	4.1097	2.6212	3.4881	2.9594	2.8088

The gradient of each direction is represented as the nearest-neighbor difference

$$\begin{aligned} \nabla_N x_{i,j}^t &= x_{i-1,j} - x_{i,j}^t, \\ \nabla_S x_{i,j}^t &= x_{i+1,j} - x_{i,j}^t, \\ \nabla_W x_{i,j}^t &= x_{i,j-1} - x_{i,j}^t, \\ \nabla_E x_{i,j}^t &= x_{i,j+1} - x_{i,j}^t. \end{aligned} \quad (5)$$

The associated weighted value c_D is

$$c_D = g(\left\| \nabla_D x_{i,j}^t \right\|) \quad (6)$$

where, for a pre-selected integer K, $g(\cdot)$ is a non-negative convex function chosen as (see Perona and Malik, 1990)

$$g(\nabla x) = \frac{1}{1 + (||\nabla x||/K)^2}. \quad (7)$$

Then the prediction would be updated in every iteration until it goes stable. More specifically, the PDE prediction, $\hat{x}_{i,j}$, is taken as $\hat{x}_{i,j} = \lim_{t \to +\infty} x_{i,j}^t$.

We now illustrate the superiority of PDE predictor. In doing so, we use the entropy E to measure the concentration degree of prediction-error histogram as

$$E = -\sum_e p_r(e) \ln p_r(e) \quad (8)$$

where $p_r(e)$ is the frequency of prediction-error e. The smaller the entropy, the more concentrated the prediction-error histogram. Table 1 gives the entropies of prediction-error histogram generated by four different predictors including median edge detector (MED) based predictor, mean-value predictor, interpolation-based predictor (Luo et al., 2010) (denoted as IP) and PDE predictor. From the table, it is clear that PDE predictor always gets the minimum entropy among the listed ones.

To further verify the superiority of PDE predictor, we test these four predictors on all the 1338 images of UCID database (Schaefer and Stich, 2004) as shown in Fig. 2. For better comparison, we compute the ratio of entropy value obtained by PDE predictor to that obtained by the compared predictor. Evidently, the ratio below 1 indicates that the entropy of PDE is lower than the compared one, and the smaller the ratio, the more the PDE outperforms the compared predictor. From Fig. 1, one can observe that, in most cases, PDE predictor achieves a smaller entropy than the other three. That is to say, the prediction-error histogram provided by PDE predictor is more sharply distributed. Among the three predictors, by comparison with PDE predictor, mean-value predictor is slightly worse and has a comparable entropy for each image, while MED and IP are obviously worse.

For different types of images, the performance of PDE predictor depends on the selection of K as shown in Fig. 3. The optimal K is the value that leads to the smallest entropy of prediction-error

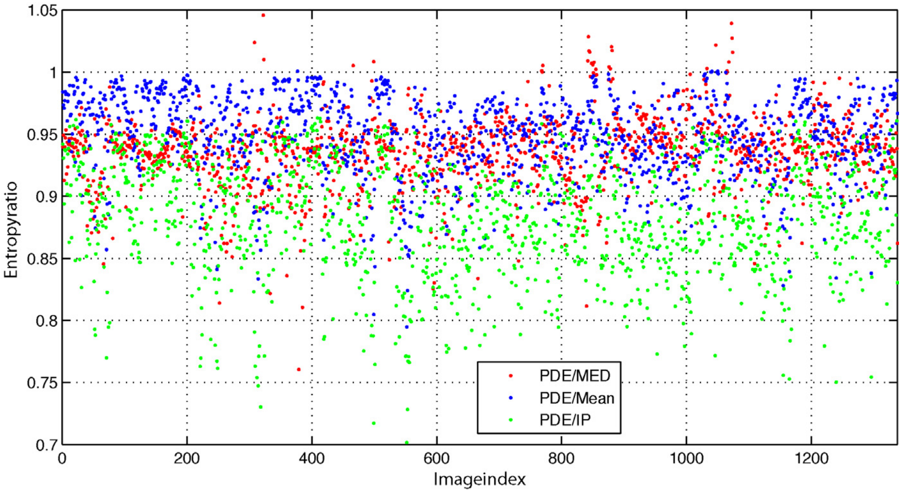


Fig. 2. Entropy ratios for 1338 images of UCID database, where we compare the entropy of PDE with that of MED, mean-value and IP as E_{PDE}/E_{MED}, E_{PDE}/E_{Mean} and E_{PDE}/E_{IP}, respectively.

histogram, and hence indicates the best performance of PDE predictor. It can be seen that K is image-dependent, and the optimal threshold is located at the valley of curve while the worst one at the peak. In our experiments, to find the optimal K adaptively, we use the greedy-like search which starts from the smallest value of K, and terminates until the conditions of $E_{K-1} > E_K$ and $E_{K+1} > E_K$ are both satisfied, where E_K denotes the entropy of prediction-error histogram for K.

3.2. Prevent overflow/underflow

To solve the overflow/underflow problem, an improved two-pass testing inspired by Sachnev et al.'s (2009) method is proposed to construct a more effective location map. In their method, the pixels are classified into three sets as a result of twice test on encoder (see Sachnev et al., 2009).

- Type-(a), if the pixel can be modified twice, it does not need to be marked in the location map.
- Type-(b), if the pixel can be modified only once, it is marked with "1" in the location map.
- Type-(c), if the pixel can not be modified, it is marked with "0" and will be keep unchanged during the embedding procedure.

Considering the classification of expandable (E) and shiftable (S) pixels in PEE, the pixels are in fact classified into six categories: E_a, E_b, E_c, S_a, S_b and S_c. The purpose of test is to simulate the value of x^w after modification. As for expandable pixels, such stimulation is based on the equation $x^w = x + e + w$. Obviously, the value of x^w would be varied with the to-be-embedded bit w. In the method Sachnev et al. (2009), all the expandable pixels are tested hardest. "Hardest" means that the simulated expansion is proceeded to make the value x^w more close to the gray-scale bound. For example, on the encoder, let $(x, e) = (251, 0)$ and the inner-region is $[-1, 1)$. Then, we use the $x^w = x + e + w$ to test whether a pixel is modifiable or not. To test hardest, w is selected as 1 so that x^w is obtained as the most probable overflow value $x^w = 252$. Similarly, if $(x, e) = (1, -1)$, w is selected as 0 to obtain the most probable underflow value as $x^w = 0$.

In their method, only the pixels which are still in the range [0, 255] after twice tests are used for data embedding. For expandable pixels, both E_c and E_b pixels are discarded, and will be marked with

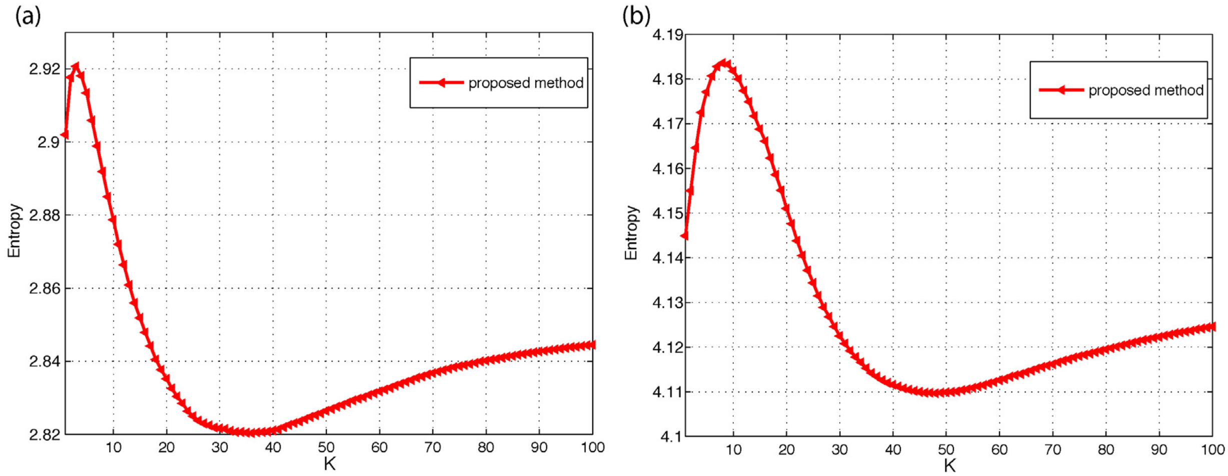


Fig. 3. Entropy of prediction-error histogram using PDE predictor (Y-axis) varies with the threshold K (X-axis). (a) Lena (b) Baboon. The optimal thresholds for these two images are 36 and 48, respectively.

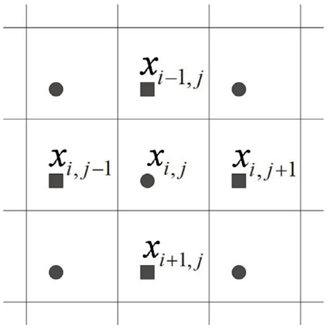


Fig. 4. Prediction pattern in double-layer embedding, where center pixel is predicted by its four nearest neighbors.

“1” and “0” in the location map, respectively. In fact, E_b pixels can also be utilized for data embedding, and discarding these pixels will lead to capacity decrease to some degree.

To make E_b pixels available for data embedding, we propose an improved two-pass testing by utilizing the odd-even property between the pixel and its prediction, which can retrieve the test bit w at decoder. The difference is that on the encoder we use the data bit $w \in \{0, 1\}$ rather than the extreme bit for the modification of E_b. On the decoder, to avoid the misclassification of E_b, the test values of E_b are needed to be revised before test. According to the odd-even property, the original test bit w can be retrieved at decoder as

$$w = \begin{cases} 1, & \text{if } \; x^w + \mathrm{round}(\hat{x})\text{is odd} \\ 0, & \text{if } \; x^w + \mathrm{round}(\hat{x})\text{is even} \end{cases}. \tag{9}$$

According to the retrieved data bits, the expandable pixels, which are not embedded hardest, can be distinguished. So, at decoder, we can use the revised test value x_r^w for test, which is obtained by

$$x_r^w = \begin{cases} x^w + 1, & \text{if } \; e \geq 0 \;\text{ and }\; w = 0 \\ x^w - 1, & \text{if } \; e < 0 \;\text{ and }\; w = 1 \\ x^w, & \qquad \text{otherwise} \end{cases}. \tag{10}$$

Notice that we do not change the pixel value but its test value. Through such revision, the revised test value x_r^w is same as the first test result at encoder. Therefore, the same classification of pixels can be obtained. For example, on decoder, let $(x, \hat{x}, w) = (251, 250, 0)$ be a sample of E_b. So, we have $e = 1$. By testing hardest, the twice testing results are 253 and 257 $(253 = 250 + 2 \times 1 + 1$, $257 = 250 + 2 \times (2 \times 1 + 1) + 1)$, respectively. Then to embed it with a data bit 0, the marked value $x^w = 250 + 2 \times 1 + 0 = 252$. Obviously, on decoder, the pixel may be misclassified without revision since the marked value is not the same as the first testing result on decoder. To classify it correctly, we first retrieve data bit $w = 0$ by the parity check of 252 + 250 according to Eq. (7), and then recover the test value as $x_r^w = 252 + 1 = 253$ according to Eq. (8). Thus, the classification of decoder is the same as that of encoder. It can be expected that capacity could be increased for images which contain abundant pixel values near the upper/lower boundary (i.e., much more E_b pixels).

3.3. Determination of embedding parameters

In our method, the payload size is equaled to the populations of pixels whose prediction-errors belong to inner region $[T_l, T_r)$.

Fig. 5. Six 512 × 512 sized test images: Lena, Baboon, Airplane, Barbara, House and Tiffany.

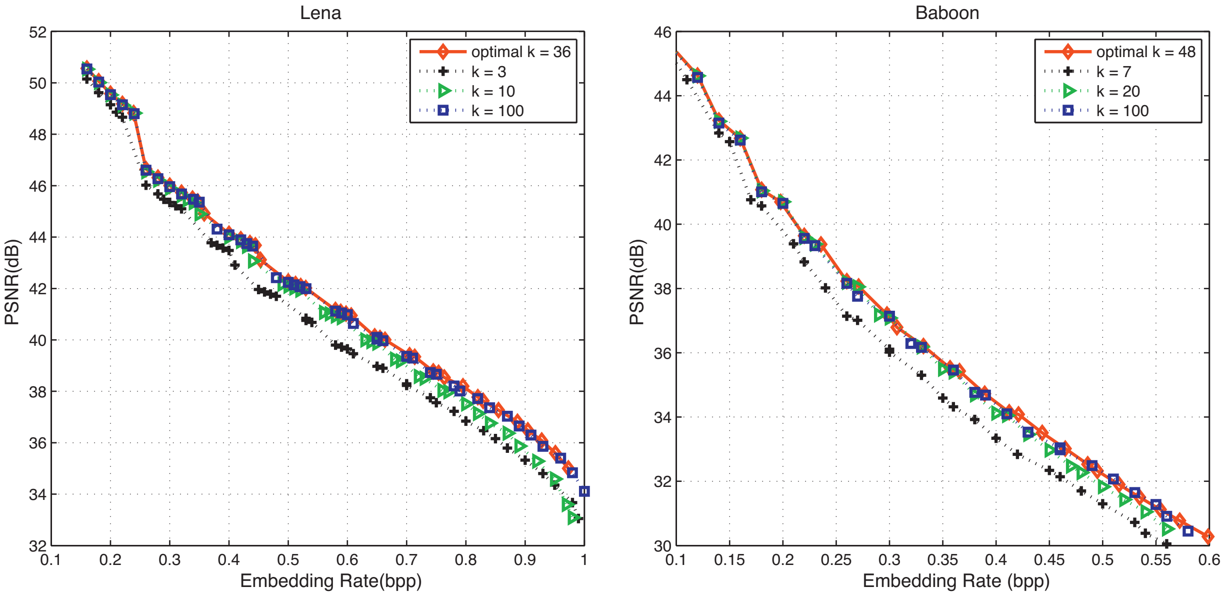


Fig. 6. Performance of the proposed method for different threshold *K*.

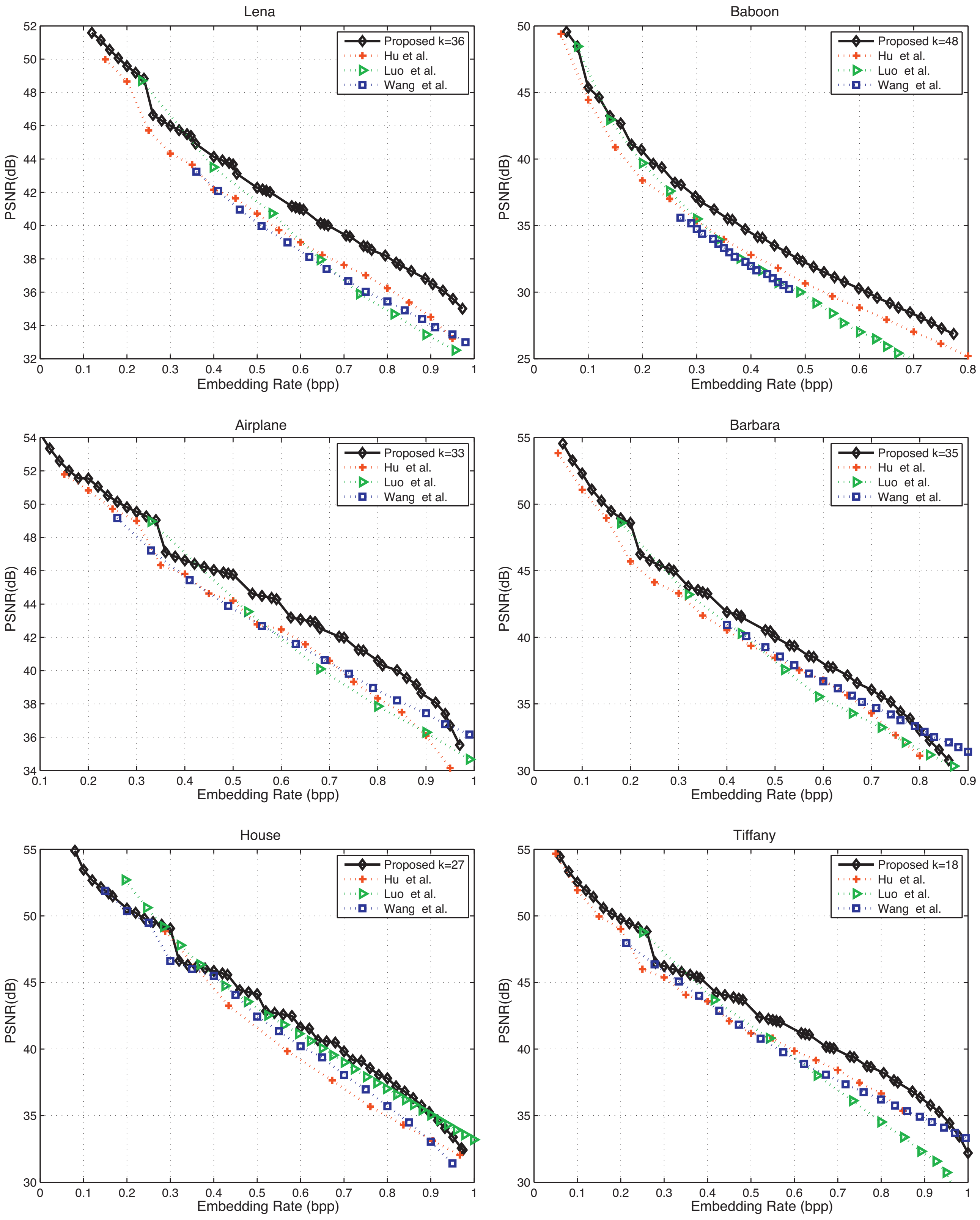


Fig. 7. Performance evaluation of the proposed method compared with recent methods over standard test images. The threshold is set as the optimal value.

Here, we use $N(T_l, T_r)$ to denote the number of these pixels. To meet the payload size and minimize the distortion as much as possible, the inner region $[T_l, T_r)$ should be determined before data embedding. In our method, the size of auxiliary information is set as 50 bits, which contains the values of T_l and T_r, the size of compressed location map L_s and the size of secret message bits C. The 50 bits are embedded into the LSB of first 50 pixels and the replaced LSB will be taken as a part of payload. The thresholds T_l and T_r should satisfy the condition of $N(T_l, T_r) \geq C + L_s + 50$. Notice that the size of location map would be increased as the range $[T_l, T_r)$ expands, and

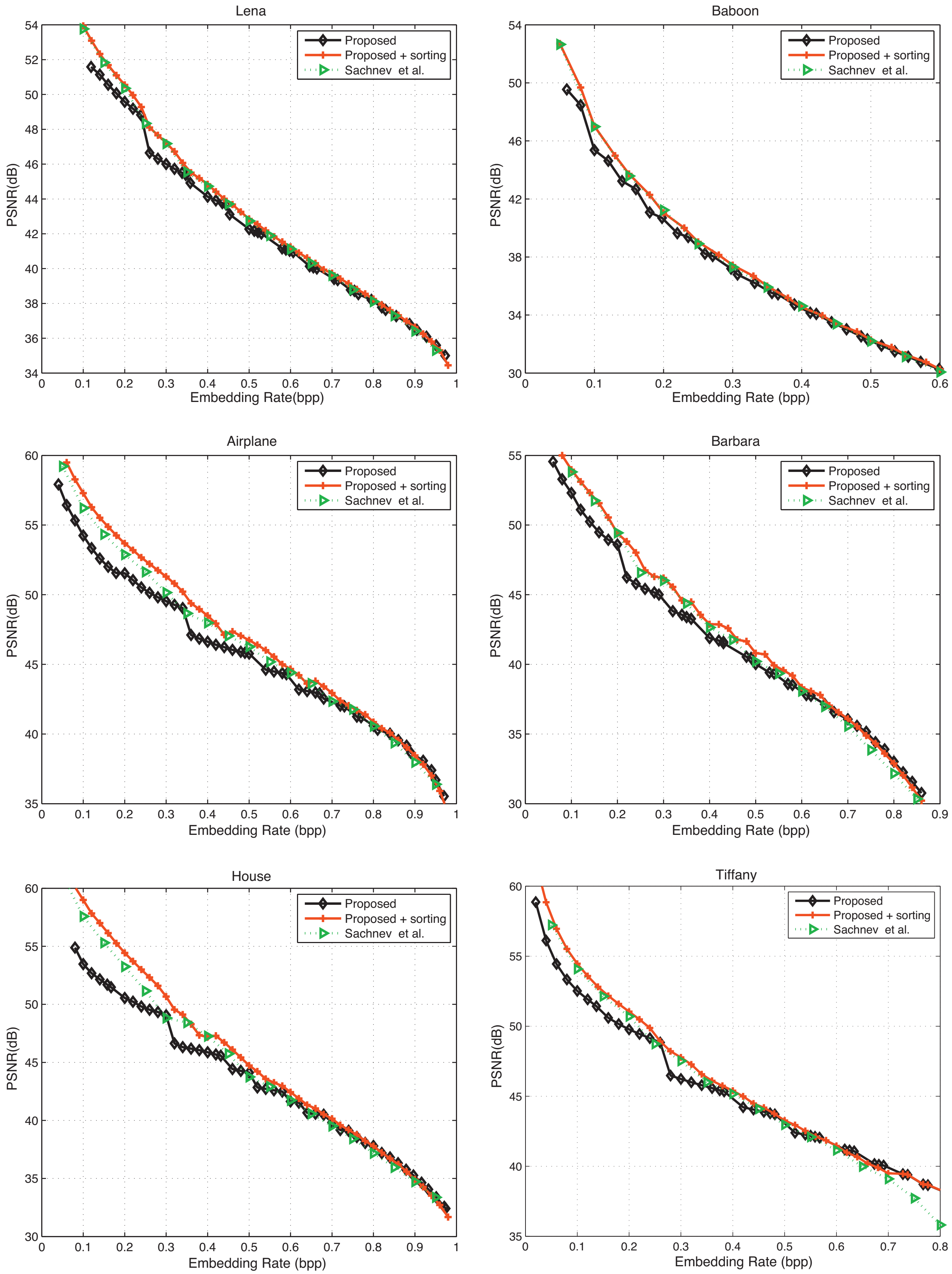


Fig. 8. Performance of hybrid PDE based method.

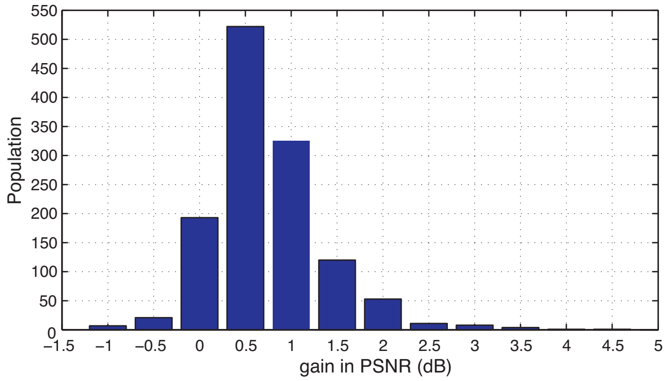


Fig. 9. Histogram of PSNR gains on UCID image database for the embedding rate 0.5 bpp.

an iterative process to adjust T_l and T_r is required to meet the least capacity.

3.4. Embedding procedure

Now we summarize the above proposed ideas, and give the details of embedding and extracting procedures. Since full-enclosing context is helpful for obtaining a more accurate prediction, we utilize the double-layer embedding strategy to embed data in twice embedding. In this way, the pixels are divided into two classes as circle pixels and square pixels (see the prediction pattern shown in Fig. 4). Firstly, circle pixels are embedded, then the marked circle pixels are taken as context for the prediction of square ones. At decoder, this order is inverse to ensure reversibility. Here, the half size of message bits is embedded in once embedding. The embedding procedure is described as follows.

Step (1) Skip the first 50 circle pixels, and empty the LSB of these pixels to make room for embedding auxiliary information.

Step (2) Collect all the remaining circle pixels, and obtain the prediction $\hat{x}$ of each pixel x by using PDE predictor. Obtain prediction-errors and count them to form prediction-error histogram. Then, determine the initial values of T_l and T_r to satisfy the condition $N(T_l, T_r) \geq C/2$.

Step (3) For each pixel, use improved two-pass testing as illustrated in Section 3.2. Here, the pixels of inner region are tested by the hardest bit to estimate the max size of compressed location map L_{max}.

- if it belongs to type-(b) or type-(c), mark it with “1” or “0” in the location map, respectively.
- if it belongs to type-(a), it dose not need to be marked in the location map.

Step (4) Check whether the least capacity condition $N(T_l, T_r) \geq C/2 + L_{max} + 50$ is met or not. If satisfied, record T_l, T_r, and the size of $C/2$ and L_s. Embed them in the LSB of first 50 pixels. If not, expand the inner range $[T_l, T_r)$ outwards, and go to step (3).

Table 2
Comparison of time cost for a single-level embedding, where the unit of runtime is second.

Images	Lena	Baboon	Airplane	Barbara	House	Tiffany	Average
Luo et. al.'s	0.27	0.28	0.31	0.28	0.27	0.28	0.28
Sachnev et. al.'s	8.17	8.30	8.35	8.25	8.22	8.17	8.24
Proposed (t = 300)	92.00	92.00	92.00	93.00	93.00	94.00	92.67
Proposed (t = 500)	103.00	108.00	98.00	102.00	99.00	101.00	101.83

Step (5) For all modifiable pixels, process data embedding by using PEE as described in Section 2. Notice that data bits are embedded into E_a pixels first and then E_b pixels to guarantee that the location map is restored in E_a. The first L_{max} sized E_a pixels are used for the embedding of actual compressed location map L_s. Then, to embed message bits, for each pixel,

- if the prediction-error e falls into the inner region $[T_l, T_r)$, expand it and embed a bit.
- if e belongs to outer region, add T_l or T_r to e, accordingly.

Stop embedding when the number of embedded bits is equaled to $C/2 + 50$. Embed the auxiliary information in the LSB of first 50 circle pixels. Save the actual L_s in the first L_{max} sized E_a pixels.

Step (6) Process square pixels in a similar way by repeating steps (1)–(5). After message bits are completely embedded, the marked image is obtained.

3.5. Extracting procedure

The extracting procedure begins by processing square pixels.

Step (1) Extract the auxiliary information T_l, T_r, and the size of $C/2$ and L_s from the first 50 square pixels.

Step (2) Collect the remaining square pixels. According to Eq. (7) and Eq. (8), obtain the revised test value x_r^w for E_b pixels, and then E_a can be definitely classified by testing. Extract the compressed location map L_s from E_a pixels.

Step (3) By consulting the location map, E_b and E_c pixels can be clearly classified. Then, follow the same order in embedding procedure to extract data bits, i.e., process E_a pixels at first. For each pixel,

- if e belongs to inner range $[2T_l, 2T_r)$, restore it by the inverse expansion operation and extract the embedded data bit as illustrated in Section 2.
- if e belongs to outer range $[-255, T_l) \cup [T_r, 255]$, restore it by the inverse shifting operation.

Check whether the number of extracted bits is equaled to $C/2 + 50$ or not. If not, process the next pixel. Else, stop extracting. For E_c pixels, keep them unchanged.

Step (4) After all the embedded bits are extracted, find the replaced LSB to recover the first 50 pixels by LSB replacement.

Step (5) Process circle pixels in a similar way by repeating steps (1)–(4). When all the message bits are extracted, the original image is recovered.

4. Experiments

In this section, the proposed method is evaluated by comparing it with four recently proposed methods: Hu et al. (2009), Sachnev et al. (2009), Luo et al. (2010) and Wang et al. (2010). The first three methods are based on PEE, and the last one is based on integer-transform. In these methods, Hu et al. choose MED to calculate prediction, Luo et al. utilize interpolation by slightly modifying the weights of casual pixels, and Sachnev et al. use the mean value of four neighboring pixels as the prediction. The comparison is conducted on six 512 × 512 sized standard images (see Fig. 5).

The capacity–distortion curve of the proposed method for different K is given in Fig. 6, by selecting the peak, valley and two medium points on the curve in Fig. 3. It can be seen that the major difference of performance curve are between the optimal threshold and the worst one, meanwhile, the performance of medium value of K is comparable to that of optimal one. Obviously, K directly contributes to the prediction accuracy of PDE predictor, because the detection of inter region and intra region depends on K.

Fig. 7 gives the capacity–distortion curve comparisons with the recent methods: Hu et al. (2009), Luo et al. (2010) and Wang et al. (2010). Luo et al.'s (2010) method aims to achieve accurate predictions, in which the weighted average of context pixels are used for prediction. It performs well at low embedding rate since the modification of pixel value is at most 1 for one layer embedding, but degrades seriously after multiple-layer embedding. Compared with Luo et al.'s method, the gain of ours in PSNR increases as the embedding rate increase, but when the embedding rate is near the upper bound of 1.0 bpp, the gain drops because doubling the large magnitude of prediction-error leads to high distortion. Wang et al.'s method (Luo et al., 2010) is a generalized DE algorithm by incorporating integer transform, in which the mean-value of pixel block is used for prediction. In our stimulations, the block size of method (Wang et al., 2010) is 4 × 4. Hu et al.'s (2009) method applies MED to exploit spatial redundancy of half-enclosing pixels. Compared with these two methods, our method performs better in most cases.

On the other hand, we try to illustrate that PDE prediction can be incorporated into PEE based schemes by replacing the conventional predictor with PDE predictor. To this end, we combine PDE prediction with sorting technique and compare the hybrid PDE algorithm with Sachnev et al. (2009) in Fig. 8. Here, Sachnev et al.'s (2009) is a hybrid algorithm using sorting technique and PEE, and obtain many benefits from utilization of sorting. Compared with theirs, the PDE without sorting performs worse at low embedding rate, when embedding rate increases, the performances of two methods are similar. When we combine sorting in our scheme, PDE can be directly improved as well and the improvements are mainly at low embedding rate. Meanwhile, the performance of hybrid PDE is slightly better than theirs. It is concluded that sorting technique plays an important role in PEE based scheme for reducing the embedding distortion.

Besides, a large-scale comparison on the UCID image database (Schaefer and Stich, 2004) containing 1338 images is given in Fig. 9, in which only the Sachnev et al.'s (2009) method is compared. The comparison is evaluated by the PSNR gains at the embedding rate 0.5 bpp, where the PSNR gain is calculated by subtracting the PSNR of Sachnev et al.'s from that of the proposed method combining sorting technique. It can be seen that PSNR gains are over zero in most cases, indicating that the proposed method outperforms Sachnev *et al*.'s. The average PSNR gain is 0.74 dB.

The time cost of PDE predictor is determined by the image size n and the iteration number t, thus the computational complexity is $O(nt)$, while that of traditional predictors is $O(n)$. But in practice, much more time would be consumed as we should consider the optimization of parameter K. The average time cost for a single-level embedding on the standard images is given in Table 2. The proposed and the compared algorithms are implemented by Matlab 7.9.0, and the experiments are ran on a personal PC. It can be seen that the runtime of the proposed method varies with images. Usually, the smooth image is associated with a small optimal K, leading to a less searching time accordingly. Besides, the iteration number also affects the time cost as the larger t results in more computations in the prediction. According to our experiments, t should be set over 100 so that the ultimate prediction becomes stable. Although our time cost is much larger than the prior arts, it is still acceptable in this age as it can be reduced by a high-performance PC or an effective programming language such as C++.

5. Conclusions

In this paper, we propose a new reversible data hiding scheme based on partial differential equation. Unlike the conventional predictors in previous PEE schemes, PDE predictor iteratively updates the prediction to yield a more accurate one. It well exploits the spatial correlation of pixels by considering the gradient of local context, and gives the priority to the evolution of direction with a small magnitude of gradient. Experimental results show that the PDE predictor generates a more concentrated prediction-error histogram with small entropy compared with the common predictors such as MED, mean-value predictor and IP, and the proposed method outperforms some of recent state-of-the-art reversible data hiding schemes (Hu et al., 2009; Sachnev et al., 2009; Luo et al., 2010; Wang et al., 2010).

Acknowledgement

This work was supported in part by 973 Program (2011CB302204), National Natural Science Funds for Distinguished Young Scholar (61025013), National NSF of China (61073159, 61272355), PCSIRT (IRT 201206), Fundamental Research Funds for the Central Universities (2012JBM042).

References

Alattar, A.M., 2004. Reversible watermark using the difference expansion of a generalized integer transform. IEEE Transactions on Image Processing 13 (8), 1147–1156.

An, L., Gao, X., Li, X., Tao, D., Deng, C., Li, J., 2012. Robust reversible watermarking via clustering and enhanced pixel-wise masking. IEEE Transactions on Image Processing 21 (8), 3598–3611.

Celik, M.U., Sharma, G., Tekalp, A.M., Saber, E., 2005. Lossless generalized–LSB data embedding. IEEE Transactions on Image Processing 14 (2), 253–266.

Celik, M.U., Sharma, G., Tekalp, A.M., 2006. Lossless watermarking for image authentication: a new framework and an implementation. IEEE Transactions onImage Processing 15 (4), 1042–1049.

Coltuc, D., 2011. Improved embedding for prediction-based reversible watermarking. IEEE Transactions on Information Forensics and Security 6 (3), 873–882.

Coltuc, D., 2012. Low distortion transform for reversible watermarking. IEEE Transactions on Image Processing 21 (1), 412–417.

Feng, G., Fan, L., 2012. Reversible data hiding of high payload using local edge sensing prediction. Journal of Systems and Software 85 (2), 392–399.

Fridrich, J., Goljan, M., Du, R., 2001. Invertible authentication. In: Security and Watermarking of Multimedia Contents III, vol. 4314 of SPIE, pp. 197–208.

Fridrich, J., Goljan, M., Du, R., 2002. Lossless data embedding – New paradigm in digital watermarking. EURASIP Journal on Applied Signal Processing 2002 (2), 185–196.

Gao, X., An, L., Li, X., Tao, D., 2009. Reversibility improved lossless data hiding. Signal Processing 89 (10), 2053–2065.

Gao, X., An, L., Yuan, Y., Tao, D., Li, X., 2011. Lossless data embedding using generalized statistical quantity histogram. IEEE Transactions on Circuits and Systems for Video Technology 21 (8), 1061–1070.

Hong, W., Chen, T.S., Shiu, C.W., 2009. Reversible data hiding for high quality images using modification of prediction errors. Journal of Systems and Software 82 (11), 1833–1842.

Hong, W., Chen, T.S., Chang, Y.P., Shiu, C.W., 2010. A high capacity reversible data hiding scheme using orthogonal projection and prediction error modification. Signal Processing 90 (11), 2911–2922.

Hu, Y., Lee, H.K., Li, J., 2009. DE-based reversible data hiding with improved overflow location map. IEEE Transactions on Circuits and Systems for Video Technology 19 (2), 250–260.

Kamstra, L., Heijmans, H.J.A.M., 2005. Reversible data embedding into images using wavelet techniques and sorting. IEEE Transactions on Image Processing 14 (12), 2082–2090.

Kim, H.J., Sachnev, V., Shi, Y.Q., Nam, J., Choo, H.G., 2008. A novel difference expansion transform for reversible data embedding. IEEE Signal Transactions on Information Forensics and Security 4 (3), 456–465.

Lee, S.K., Suh, Y.H., Ho, Y.S., 2006. Reversible image authentication based on watermarking. In: Proc. IEEE ICME, pp. 1321–1324.

Lee, S., Yoo, C.D., Kalker, T., 2007. Reversible image watermarking based on integer-to-integer wavelet transform. IEEE Signal Transactions on Information Forensics and Security 2 (3), 321–330.

Li, X., Yang, B., Zeng, T., 2011. Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection. IEEE Transactions on Image Processing 20 (12), 3524–3533.

Luo, L., Chen, Z., Chen, M., Zeng, X., Xiong, Z., 2010. Reversible image watermarking using interpolation technique. IEEE Signal Transactions on Information Forensics and Security 5 (1), 187–193.

Ni, Z., Shi, Y.Q., Ansari, N., Su, W., 2006. Reversible data hiding. IEEE Transactions on Circuits and Systems for Video Technology 16 (3), 354–362.

Perona, P., Malik, J., 1990. Scale-space and edge detection using anisotropic diffusion. IEEE Transactions on Pattern Analysis and Machine Intelligence 12 (7), 629–639.

Qian, Z., Zhang, X., 2012. Lossless data hiding in jpeg bitstream. Journal of Systems and Software 85 (2), 309–313.

Sachnev, V., Kim, H.J., Nam, J., Suresh, S., Shi, Y.Q., 2009. Reversible watermarking algorithm using sorting and prediction. IEEE Transactions on Circuits and Systems for Video Technology 19 (7), 989–999.

Schaefer, G., Stich, M., 2004. Ucid – an uncompressed colour image database. In: Storage and Retrieval Methods and Applications for Multimedia, SPIE, pp. 472–480.

Tai, W.L., Yeh, C.M., Chang, C.C., 2009. Reversible data hiding based on histogram modification of pixel differences. IEEE Transactions on Circuits and Systems for Video Technology 19 (6), 906–910.

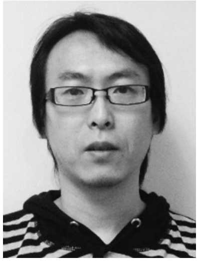
Thodi, D.M., Rodriguez, J.J., 2007. Expansion embedding techniques for reversible watermarking. IEEE Transactions on Image Processing 16 (3), 721–730.

Tian, J., 2003. Reversible data embedding using a difference expansion. IEEE Transactions on Circuits and Systems for Video Technology 13 (8), 890–896.

Wang, X., Li, X., Yang, B., Guo, Z., 2010. Efficient generalized integer transform for reversible watermarking. IEEE Signal Processing Letters 17 (6), 567–570.

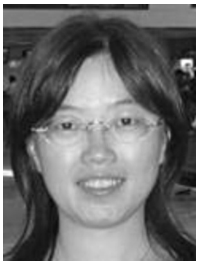
Weng, S., Zhao, Y., Pan, J.S., Ni, R., 2008. Reversible watermarking based on invariability and adjustment on pixel pairs. IEEE Signal Processing Letters 15, 721–724.

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.



Xiaolong Li received the BS degree from Peking University, Beijing, China, MS degree from Ecole Polytechnique, Palaiseau, France, and PhD 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 Univerity in 2007–2009. His research interests are image processing and information hiding.

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 senior 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, and 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 4 national patents.