

Reversible watermarking based on multiple prediction modes and adaptive watermark embedding

Shaowei Weng · Jeng-Shyang Pan

Published online: 20 July 2013
© Springer Science+Business Media New York 2013

Abstract A new reversible watermark scheme based on multiple prediction modes and adaptive watermark embedding is presented. Six prediction modes fully exploiting strong correlation between any pixel and its surrounding pixels, are designed in this paper. Under any prediction mode, each to-be-predicted pixel must be surrounded by several pixels (they constitute a local neighborhood, and any modification to this neighborhood is not allowed in the embedding process). This neighborhood has three main applications. The first one is that when it is exploited to interpolate some to-be-predicted pixel, the noticeable improvement in prediction accuracy is obtained. The second one is that its variance is employed to determine which classification (i.e., smooth or complex set) its surrounded pixel belongs to. For any to-be-predicted pixel, the number of embedded bits is adaptively determined according to this pixel's belonging. The last one is that we can accurately evaluate the classification of watermarked pixels by analyzing the local complexity of their corresponding neighborhoods on the decoding side. Therefore, the payload can be largely increased as each to-be-predicted pixel in the smooth set can possibly carry more than 1 bit. Meanwhile, the embedding distortion is greatly controlled by embedding more bits into pixels belonging to smooth set and fewer bits into the others in complex set. Experimental results reveal the proposed method is effective.

Keywords Reversible watermarking · Multiple prediction modes · Adaptive watermark embedding

S. Weng (✉)
School of Information Engineering, Guangdong University of Technology,
GuangZhou, People's Republic of China
e-mail: wswweiwei@126.com

J.-S. Pan
Shenzhen Graduate School, Harbin Institute of Technology,
ShenZhen, People's Republic of China

1 Introduction

In some applications, such as the fields of law enforcement, medical and military image system, any permanent distortion induced to the host images by data watermarking techniques is intolerable. In such cases, the original image is required to be recovered without any distortion after extraction of the embedded watermark. The watermarking techniques satisfying these requirements are referred to as reversible (or lossless) watermarking.

A considerable amount of research on reversible watermarking has been done over the last several years since the concept of reversible watermarking firstly appeared in the patent owned by Eastman Kodak [4]. In paper [12], reversible watermarking (RW) is classified into the following three categories: 1) fragile authentication 2) high embedding capacity and 3) semi-fragile authentication. The first two categories of RW are actually a special subset of fragile watermarking, namely, they cannot resist any attacks. From the literature up to today, it is observed that most of the RW scheme is grouped into the second category. Here, we provide a list containing multiple high-capacity RW schemes [1, 2, 8, 10, 13, 14, 16–20] which account for only part of the most representative ones over the last a few years.

RW based on difference expansion was first proposed by Tian [13]. Difference (between a pair of neighboring pixels) was shifted left by one unit to create a vacant least significant bit (LSB), and 1-bit watermark was appended to this LSB. This method was called Difference Expansion (DE). Alattar [1] generalized the DE technique by taking a set containing multiple pixels rather than a pair. Coltuc et al. proposed a threshold-controlled embedding scheme based on an integer transform for pairs of pixels [2]. In Thodi's work [14], histogram shifting was incorporated into Tian's method to produce a new algorithm called Alg. D2 with a overflow map. Weng et al. proposed an integer transform based on invariability of the sum of pixel pairs [19]. Weng et al. also proposed a new integer transform, and the embedding rate can approach to 1 bpp (bit per pixel) for a single embedding layer by overlapped pairing [18]. In Wang et al.'s method [17], a generalized integer transform and a payload-dependent location map were constructed to extend the DE technique to the pixel blocks of arbitrary length. Feng et al. presented a new reversible data hiding algorithm based on integer transform (proposed by [16]) and adaptive embedding [10]. Luo et al. first introduced an interpolation technique into RW to obtain the prediction-errors, which increased prediction-accuracy by using full-enclosing pixels to predict the current pixel [8]. In Li et al.'s method [7], an efficient reversible watermarking scheme was proposed by incorporating in prediction-error expansion two new strategies, namely, adaptive embedding and pixel selection. Wu et al. proposed reversible image watermarking on prediction errors by efficient histogram modification [20].

Besides, the third class (namely semi-fragile RW scheme) only constitutes a very small part of the literature. Up to today, there exist four RW schemes [3, 9, 15, 22] which have certain robustness against JPEG lossy compression. In a word, the number of the RW schemes capable of resisting attacks is very small, and these schemes are less resistant to the other attacks except JPEG lossy compression. Therefore, in RW, the watermark is not able to survive various types of attacks. However, in copyright protect, the watermark must be able to effectively resist common image attacks, such as low pass filtering or JPEG compression. In order

to achieve copyright protection, Lin et al. proposed a blind watermarking algorithm based on the significant difference of wavelet coefficient quantization in 2008 [5]. Lin et al.'s method is quite effective against JPEG compression, low-pass filtering, and Gaussian noise, and the PSNR (peak signal-to-noise ratio) value of a watermarked image is greater than 40 dB. In the following year, Lin et al. proposed a wavelet-tree-based watermarking method using distance vector of binary cluster for copyright protection [6]. Rosiyadi et al. proposed an efficient copyright protection scheme for e-government document images [11].

In this paper, we focus on high-capacity RW schemes. After four prediction modes in Wu et al.'s method [20] are further investigated, it is found that there still exist two drawbacks in their paper: (1) in each of four prediction modes, for each of some pixels (which account for one-fourth of all pixels), the prediction performance which is gained using its four diagonally-adjacent pixels along the 45 and 135 degrees to predict itself, is weak; (2) the strategy predicting one pixel along one single direction (either left-to-right horizontal (combining top and bottom neighbors) or top-to-bottom vertical direction (combining left and right neighbors)) is incomplete.

A novel reversible watermark scheme based on multiple prediction modes and adaptive watermark embedding is presented in this paper. We focus on designing new prediction modes so as to solve the problems efficiently existing in Wu et al.'s method. Besides, in this paper, since the structural characteristic which the prediction modes themselves have makes the adaptive embedding strategy can be combined with these prediction modes. This proposed method is efficient because the improvement of the prediction accuracy implies the reduction of the distortion introduced by prediction-error expansion, and otherwise, the embedding capacity for one single embedding layer also can be largely improved by the application of adaptive embedding strategy.

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 The related method

2.1.1 Comparison with Li et al.'s method

Li et al.'s method [7] is based on pixel-by-pixel prediction. In Li et al.'s method, for each pixel, the local complexity level of its four (namely right, bottom left, bottom and bottom right) pixels is firstly estimated, and meanwhile, it is predicted using the GAP (gradient-adjusted prediction [21]) predictor, and finally, the watermark whose length is determined by the already obtained level of complexity is embedded into it using prediction error expansion technique.

However, there exist several schemes composed of the proposed method, Luo et al.'s, and Wu et al.'s, in which any modification to some pixels (which usually account for a large part of all the pixels) is not allowed in a single embedding layer, their mission is to predict the others. Otherwise, GAP predictor is utilized in Li et al.'s method, while the interpolation technique in paper [8] adopted in the proposed

method. For each to-be-predicted pixel in the proposed method, its four surrounding neighboring pixels (i.e. top, left, right, bottom neighbors) are used to predict it. For this type of RW schemes with part of all pixels kept unchanged, the proposed method achieves much higher performance compared with the other studies, such as Luo et al.'s, and Wu et al.'s.

2.1.2 Comparison with Wu et al.'s method

It is known that the closer the distance between two pixel points, the stronger the correlation. Therefore, the correlation between any to-be-predicted pixel and its four pixels along two diagonal directions is weaker than that between this pixel and its four neighboring pixels (i.e. top, left, right, bottom neighbors).

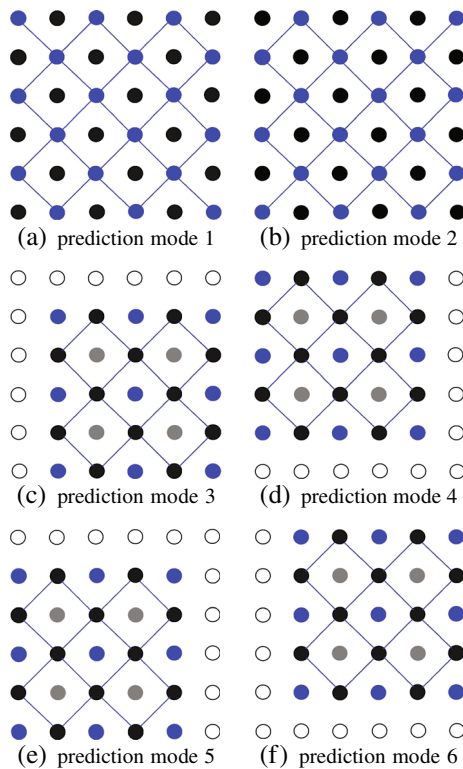
Relative to Wu et al.'s method, there exist three advantages in this paper: (1) four improved prediction modes is proposed, in which each of some pixels (which account for quarter of all pixel points) are predicted by its four neighboring pixels (i.e. top, left, right, bottom neighbors) rather than by its four pixels along two orthogonal direction like in Wu et al.'s method; (2) since prediction along two orthogonal directions is substantially more accurate than that along single direction, two new prediction modes are adopted so as to make almost all the pixels predicted by their four neighboring pixels; (3) the adaptive embedding strategy is combined with all the prediction modes so as to further improve embedding capacity.

The first two advantages aim to achieve the purpose of reducing the embedding distortion by improving prediction accuracy. The third advantage is based on the first two ones. That is to say, once prediction accuracy is improved, correspondingly, the values of prediction errors become smaller, and thus, the distortion introduced by the prediction errors expansion is correspondingly reduced. Otherwise, for each of six prediction modes, their structural characteristic allows the adaptive embedding strategy is combined with them. Adaptive embedding strategy is used to determine to embed 1 bit or 2 bits into each pixel in the to-be-predicted set. Since the values of the prediction errors are decreased, and hence, even if the prediction errors are expanded twice, the distortion is comparatively lower than that introduced by Wu et al.'s method.

2.2 Six embedding modes

Given an 8-bit graylevel I with size $M \times N$, all the pixel points are partitioned into two non-overlapped sets: the to-be-predicted set and the predicting set. Pixels marked in blue-filled circles in modes 1 and 2 (refer to Fig. 1a–b) are picked out to form the predicting set, the others (marked in black-filled circles) classified into the predicted set. Based on the principle that correlation becomes weaker as the temporal distance between pixels increases, for each pixel in the to-be-predicted set, its several surrounding neighboring pixels from the predicting set, which are most strongly related to it, are used to predict it. And meanwhile, no modification can be done to these pixels in the predicting set for a single embedding process. Otherwise, reversibility will be destroyed. In modes 1 and 2, the pixels in the first & last columns and the bottom & top rows have only two or three neighboring pixels. For instance, if some pixel lacks its left (or top) neighbor, we use its right (or bottom) neighbor to replace the absent one so as to obtain four neighbors and vice versa. Here, we consider that modes 1 and 2 are mutually symmetrical mode.

Fig. 1 Six prediction modes



For the residual four embedding modes, all the pixels marked in blue-filled circles remain unchanged during the embedding process. The pixels in black-filled circles have two roles. First, every four pixels forming a neighborhood is used to estimate their surrounded pixel (i.e., the pixel marked in gray-filled circle), and then, each of them will be predicted by its two neighboring pixels along some single direction (either horizontal or vertical) again. The benefit of double exploiting this kind of pixels is to further increase prediction accuracy. The symmetrical mode of mode 3 is considered as mode 4 in this paper, and vice versa. Similarly, mode 5 and mode 6 are also deemed mutually symmetrical.

For any to-be-predicted pixel x_c , if it has a surrounding four-pixel neighborhood, then it can be predicted by this neighborhood simply using the interpolation technique in [8]. The interpolated value of pixel x_c is calculated as

$$p_{i,j} = x'_0 \times \frac{\sigma_2}{\sigma_1 + \sigma_2} + x'_{90} \times \frac{\sigma_1}{\sigma_1 + \sigma_2} \tag{1}$$

where

$$\begin{aligned} x'_0 &= \frac{I_{i,j-1} + I_{i,j+1}}{2} \\ x'_{90} &= \frac{I_{i-1,j} + I_{i+1,j}}{2} \end{aligned} \tag{2}$$

and

$$\begin{aligned} \sigma_1 &= \frac{(I_{i,j-1} - \mu)^2 + (x'_0 - \mu)^2 + (I_{i,j+1} - \mu)^2}{3} \\ \sigma_2 &= \frac{(I_{i-1,j} - \mu)^2 + (x'_{90} - \mu)^2 + (I_{i+1,j} - \mu)^2}{3} \end{aligned} \tag{3}$$

x'_0 and x'_{90} are used to denote the interpolation values along two orthogonal directions (i.e., 0° diagonal and 90° diagonal), respectively. $I_{i-1,j}$, $I_{i,j-1}$, $I_{i,j+1}$ and $I_{i+1,j}$ are the top, left, right and bottom neighbors of x_c . These four pixels constitute a neighborhood resembling a diamond-shaped pixel cell denoted as I_C . The mean value in this neighborhood, denoted by μ , is calculated using the following formula: $\mu = \frac{I_{i-1,j} + I_{i+1,j} + I_{i,j-1} + I_{i,j+1}}{4}$. σ_1 and σ_2 are the variance estimation of interpolation errors in two diagonal directions, respectively. From (1), we can see obviously that more variance, which means weaker relation with the actual value, has less influence on the interpolated value.

Then the prediction error, denoted by $e_{i,j}$, is calculated via

$$e_{i,j} = x_c - p_{i,j} \tag{4}$$

2.3 Process of producing classification and embedding strategy for each to-be-predicted pixel

The local variance, denoted by σ , of cell I_C is used to estimate whether any to-be-predicted pixel has strong correlation with its surrounding cell or not. σ is obtained via

$$\sigma = \sqrt{\frac{(I_{i-1,j} - \mu)^2 + (I_{i,j-1} - \mu)^2 + (I_{i,j+1} - \mu)^2 + (I_{i+1,j} - \mu)^2}{4}} \tag{5}$$

When $\sigma < vT_h$, pixel x_c is regarded as having a strong relation with its cell, and therefore, it is classified into a smooth set, where vT_h is a predefined threshold which is used for distinguishing which classification x_c belongs to. Otherwise, x_c belongs to a complex set.

For any pixel in smooth set, if its prediction-error $e_{i,j}$ falls into the range of $[-pT_h, pT_h]$, $e_{i,j}$ is expanded twice according to (6), where pT_h is used for estimating the correlation between the current pixel and its predicted value, w_b represents 2-bit watermark, i.e., $w_b \in \{0, 1, 2, 3\}$.

$$e'_{i,j} = 4 \times e_{i,j} + w_b \tag{6}$$

If $e_{i,j} \geq pT_h$ and $e_{i,j} \leq -pT_h - 1$, it will be shifted by $3pT_h$. Correspondingly, the watermarked prediction-error is

$$e'_{i,j} = \begin{cases} e_{i,j} - 3 \times pT_h & e_{i,j} \leq -pT_h - 1 \\ e_{i,j} + 3 \times pT_h, & e_{i,j} \geq pT_h \end{cases} \tag{7}$$

For any pixel in complex set, if its prediction-error $e_{i,j}$ belongs to the inner region $[-pT_h, pT_h]$, then 1-bit watermark is embedded into $e_{i,j}$ to get the watermarked value as

$$e'_{i,j} = 2 \times e_{i,j} + b \tag{8}$$

where $b \in \{0, 1\}$. If $e_{i,j}$ belongs to the outer region $(-\infty, -pT_h) \cup [pT_h, \infty)$, $e_{i,j}$ will be shifted by pT_h according to the following formula

$$e'_{i,j} = \begin{cases} e_{i,j} - pT_h & d \leq -pT_h - 1 \\ e_{i,j} + pT_h, & d \geq pT_h \end{cases} \tag{9}$$

Finally, the watermarked value of x_c , i.e., y_c , is computed by $y_c = p_{i,j} + e'_{i,j}$.

2.4 Algorithm description

To guarantee reversibility, some overhead information should be recorded before data embedding. In embedding procedure, it must be embedded into the host image together with the payload so as to ensure reliable image reconstruction on the receiver side. For each embedding layer, the corresponding overhead information

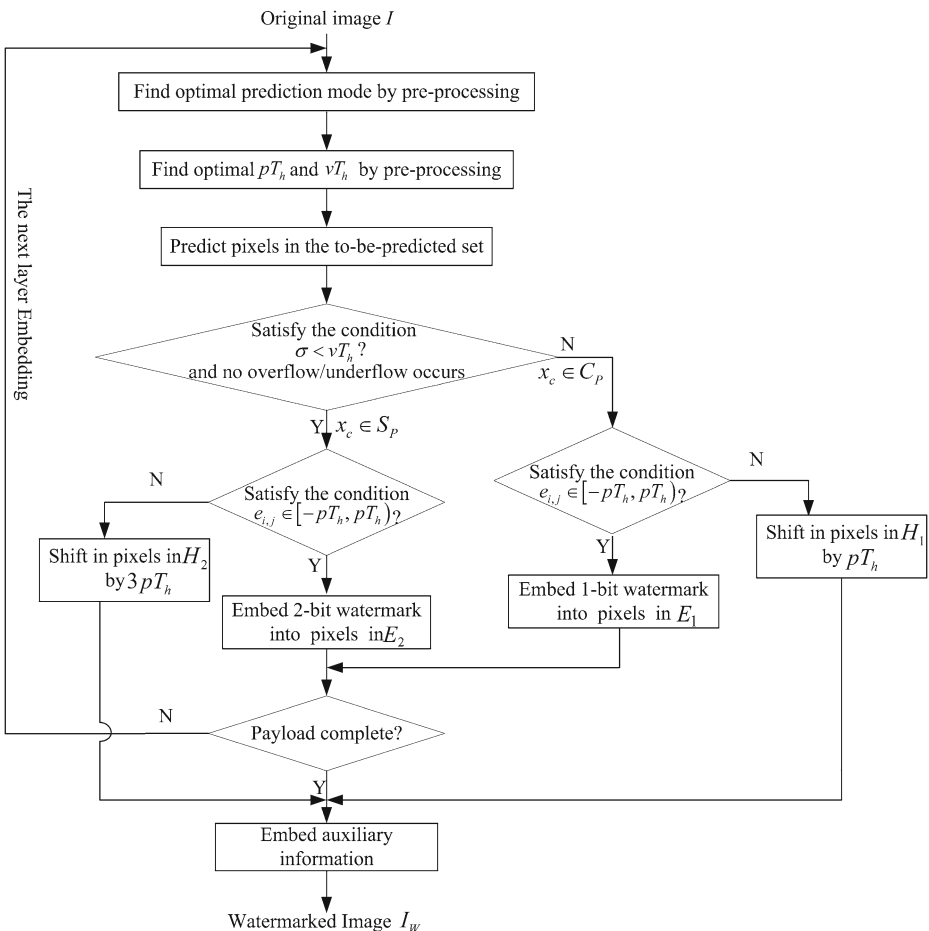
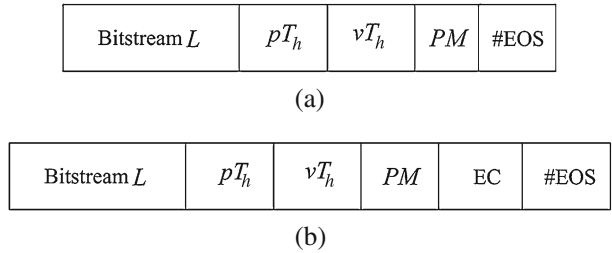


Fig. 2 Embedding mechanism

Fig. 3 Auxiliary information formats in an embedding layer: **a** the other embedding layers; **b** last embedding layer



is composed of five parts: the 8-bit, unsigned representation for the values of pT_h ; the 8-bit, unsigned representation for the values of vT_h ; the compressed location map which is utilized to solve potential overflow/underflow problem; the mode type (PM which is created to distinguish six prediction modes); end of symbol (EOS). We need 3 bits to save PM , since PM includes six different values (1/2/3/4/5/6 for mode 1/mode 2/mode 3/mode 4/mode 5/mode 6, respectively). A unique EOS symbol is added at the end of the overhead information.

A location map is generated in which the locations of the to-be-predicted pixels whose watermarked values belonging to $[0, 255]$ are marked by ‘1’ while the others are marked by ‘0’. This location map is compressed losslessly by an arithmetic encoder and the resulting bitstream is denoted by \mathcal{L} . L_S is the bit length of \mathcal{L} . Suppose that the size of the overhead information is denoted by L_Σ . The overhead information is embedded into the LSBs of the pixels after embedding the watermark by simple LSB replacement. The L_Σ LSBs to be replaced need to be embedded into the original image along with the payload so as to recover the original image. Figure 2 shows the block diagram of data embedding mechanism.

Figure 3 shows the two overhead information formats for the last embedding layer and the other embedding layers. Both of them consist of the compressed location map (L_S bits), pT_h (8 bits), vT_h (8 bits), PM (3 bits) and EOS (8 bits). 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 embedding capacity in the last layer and the number of layers, respectively. Therefore, for the other layers, the overhead information occupies L_S plus 27 bits, i.e., $L_\Sigma = L_S + 27$. While for the last layer, L_Σ consumes $L_S + 51$ bits in total.

2.4.1 Prediction mode determination

Before embedding, we must determine which mode is used in data embedding for the 1st embedding layer (also named single embedding). First, vT_h ($vT_h \in \mathbb{Z}$) is successively selected from 1 to 20. Next, for each value of vT_h , pT_h ($pT_h \in \mathbb{Z}$) runs from 1 to 40. Hence, there exist $800 = 20 \times 40$ combinations of vT_h and pT_h in total. We need to iterate all possible combinations. For any combination, repeat embedding procedures in Section 2.4.3 from mode 1 to mode 6, and simultaneously record six corresponding results (each of which contains a pair of values: PSNR and embedding rate).

Although vT_h is set to one value from 1 to 20, we are only able to select one integer from $[1,20]$ due to limited space in this paper. Here, we select $vT_h = 1$, and thus, six

curves corresponding to six modes are plotted in Fig. 4, respectively. In Fig. 4, for ‘Lena’, two curves respectively corresponding to modes 1 and 2 are very close to each other, while the others are also very close to each other. Hence, we use mode 2 to represent modes 1 and 2, while mode 6 to represent the other four modes. From Fig. 5, it can be seen that mode 2 achieves slight increase in PSNR relative to mode 6 when the embedding rate is low. Mode 6 is superior to mode 2 when the embedding rates are gradually increased. This is due to that at most half of all pixels are predicted in modes 1 and 2 so that the number of prediction errors is fewer. While in modes 3–6, almost three-fourth of all pixels are predicted.

Therefore, in the 1st embedding layer, mode 6 is determined as the final mode for ‘Lena’. Then, its symmetrical mode, i.e., mode 5, is selected for the 2nd embedding layer. The 2nd layer (also called double embedding) means we embed data once again into this already obtained watermarked image. That is to say, for ‘Lena’, mode 6 and mode 5 are alternately used for each embedding layer. Hence, for some single layer, the prediction mode is individually operated. However, prediction modes are mixed operated for multiple embedding.

As illustrated in Fig. 6, when an improved mode (e.g., mode 6) is combined with adaptive embedding in the proposed method, our method for single embedding can get almost the same performance with Wu et al.’s method having multiple embedding. It can achieve larger capacity through multiple embedding than Wu et al.’s method. Its PSNR value is about 0.5–4dB higher than that of Wu et al.’s method when the same amount of payloads is embedded.

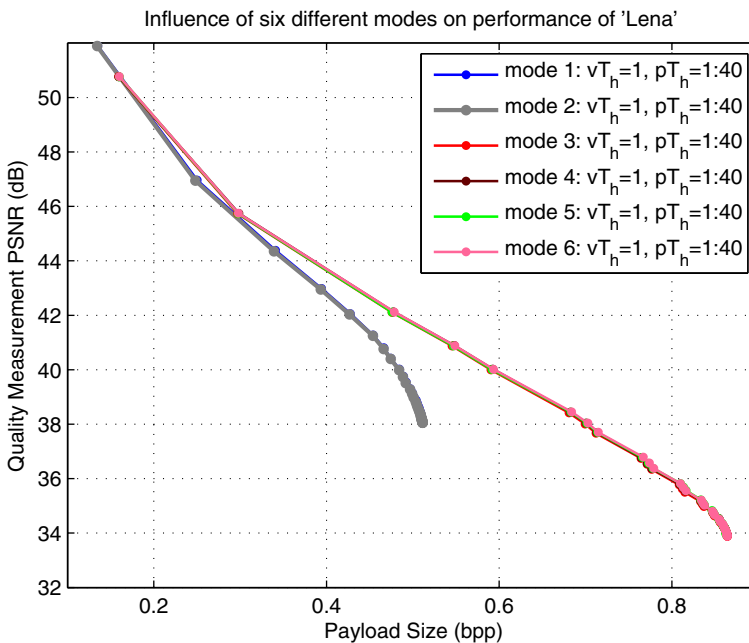


Fig. 4 Influence of six different modes on performance of ‘Lena’

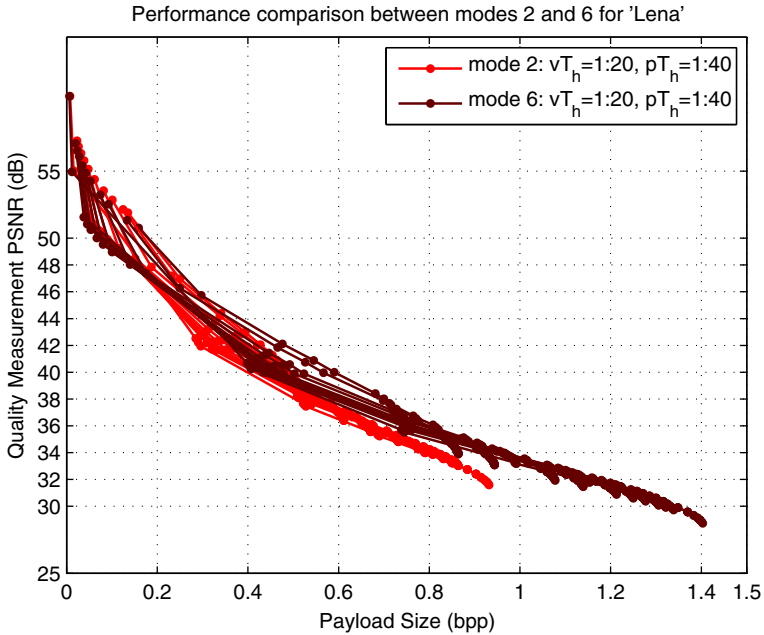


Fig. 5 Performance comparison between modes 2 and 6 for 'Lena'

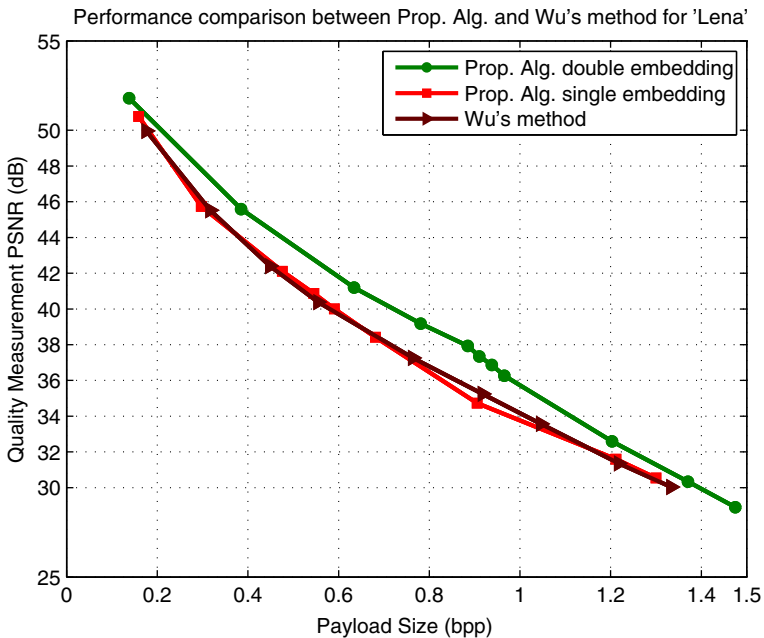


Fig. 6 Performance comparison between the proposed method and Wu et al.'s for 'Lena'

2.4.2 Threshold determination

Once the prediction mode is fixed, the best threshold for each single embedding layer indicates the corresponding embedding process with least distortion while keeping the same capacity. In fact, in the experiment, the threshold that gives the highest PSNR value at some given embedding rate is determined as the best one.

vT_h has an initial value, denoted by v_s , which is set to 0. $v_s = 0$ means that all the to-be-predicted pixels are classified into complex set. That is to say, each pixel in to-be-predicted set carries 1-bit at most, i.e., adaptive embedding loses its role. vT_h runs from this initial value to v_e (i.e., the upper limit of vT_h). Take ‘Lena’ for an example, Fig. 7 provides an intuitive illustration of determining v_e and p_e (namely the maximum value of pT_h). Note that the adopted mode in Figs. 7 and 8 is mode 2. As illustrated in Fig. 7, the corresponding curves are already very close to each other when vT_h is increased from 10 up to 20. If we increase the value of v_e over 20, it can also be very difficult to tell the difference between the curve representing $vT_h = 20$ and curves having vT_h over 20, and thus, v_e is set to 20 for ‘Lena’ in this paper. Similarly, pT_h also has an initial value, denoted by p_s , which is set to a value capable of ensuring the embedding rate larger than 0. In this paper, p_s is set to 1 for ‘Lena’. From Fig. 7, it can be seen that the upper limit of pT_h is selected as 40. It is not necessary to continue to increase this upper limit (namely exceeding 40), since both the increase in embedding rate and the decrease of PSNR value can almost be ignored.

First, vT_h is successively set to any value from v_s to v_e . Next, for each value of set $\{v_s, v_s + 1, \dots, v_e\}$, perform repeatedly embedding procedure in Section 2.4.3 from

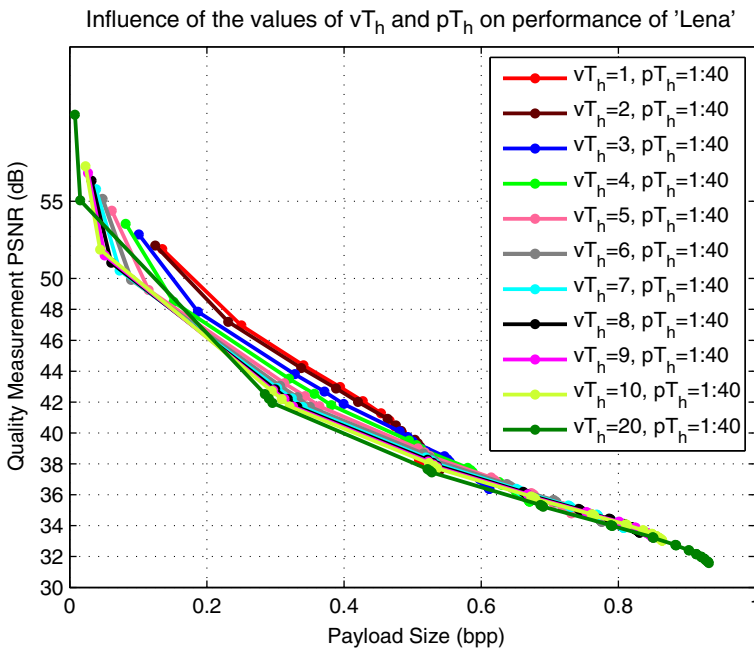


Fig. 7 Influence of the values of vT_h and pT_h on performance of ‘Lena’

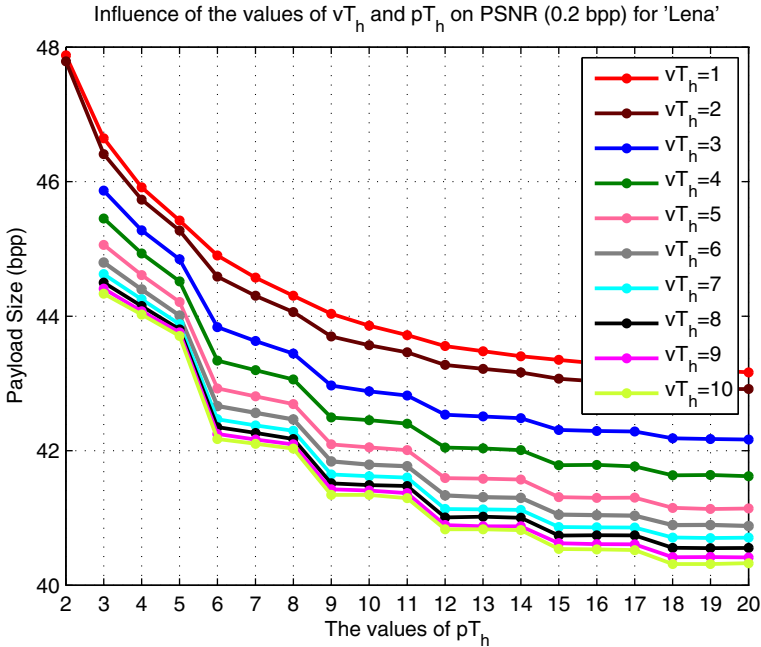


Fig. 8 Influence of the values of vT_h and pT_h on PSNR (0.2 bpp) for 'Lena'

$pT_h = p_s$ to p_e and compute the corresponding PSNR. Hence, the total number of PSNR values is $(v_e - v_s + 1) \times (p_e - p_s + 1)$. In the process of implementing embedding procedure, we stop data embedding and record PSNR once the given embedding rate is satisfied. Take the embedding rate = 0.2 bpp for example, ten curves corresponding to ten values of vT_h are displayed in Fig. 8, respectively. From Fig. 8, we can easily determine the most optimum combination of pT_h and vT_h which achieves the highest PSNR value at 0.2 bpp, i.e., $vT_h = 1$ and $pT_h = 2$.

For multiple embedding, first, two optimum threshold values (namely vT_{h1} and pT_{h1}) corresponding to single embedding at required embedding rates are experimentally determined according to Section 2.4.2. Then, implement embedding procedure with vT_{h1} and pT_{h1} in Section 2.4.3 and obtain the corresponding watermarked image. Similarly, two optimum threshold values corresponding to double embedding are vT_{h2} and pT_{h2} , respectively. vT_{h2} and pT_{h2} at some given embedding rate are obtained in the same way as single embedding. Embedding procedure having vT_{h2} and pT_{h2} is implemented so as to compute the final PSNR value. Two embedding rates are added to get final embedding rate.

2.4.3 Embedding procedure

To prevent the overflow/underflow, each to-be-predicted pixel should be contained in $[0, 255]$. We define

$$D = \{ x_c \in A : 0 \leq y_c \leq 255 \}$$

where $A = \{x_c \in \mathbb{Z} : 0 \leq x_c \leq 255\}$. Here, assume the length of the desired payload \mathcal{P} is \mathcal{P}_L .

Now, the embedding procedure is described as follows step by step.

Step 1 Pixel classification

Each to-be-predicted pixel is classified into one of three sets: NO_s , S_P and C_P .

S_P : It contains to-be-predicted pixels in smooth set whose watermarked values belong to the inner region $[0, 255]$, i.e., $S_P = \{x_c \in D : \sigma < vT_h\}$. S_P is further classified into two subsets: E_2 and H_2 .

E_2 : For $x_c \in S_P$, if its prediction-error $e_{i,j}$ falls in the range of $[-pT_h, pT_h)$, then it is classified into E_2 .

H_2 : The rest of pixels belongs to H_2 .

C_P : For any to-be-predicted pixel in complex set, if its watermarked value falls in the range of $[0, 255]$, then it belongs to set C_P , namely $C_P = \{x_c \in D : \sigma \geq vT_h\}$.

C_P is further classified into two subsets: E_1 and H_1 .

E_1 : For $x_c \in C_P$, if its prediction-error $e_{i,j}$ falls in the range of $[-pT_h, pT_h)$, then it is classified into E_1 .

H_1 : The rest of pixels belongs to H_1 .

NO_s : After any modification above (including (6) to (9)), if overflow/underflow occurs, then this pixel belongs to set NO_s .

Therefore, in the best case, the maximum payload capacity (denoted by \mathbf{P}_M) in an embedding layer is

$$\mathbf{P}_M = (2 \times |E_2| + |E_1| - L_\Sigma) \tag{10}$$

where $|\cdot|$ is the cardinality (bit length or numbers of elements) of a set.

Step 2 Watermark embedding in one embedding layer

Since there exist difference in embedding way for the overhead information and the payload, **Step 2** is divided into two parts so that two embedding ways are respectively described.

Step 2.1 Embedding watermark bits

For any pixel, if it is in NO_s , then it is kept unaltered, i.e., $y_c = x_c$. If x_c belongs to set E_2 , then 2-bit watermark is embedded into its prediction-error according to (6). If x_c belongs to H_2 , it will be shifted by $3pT_h$ according to (7). If x_c belongs to set E_1 , then 1-bit watermark is embedded into its prediction-error according to (8). If x_c belongs to H_1 , it will be shifted by pT_h according to (9). The detailed embedding procedure is listed below:

set $w_n = 1$

if $x_c \in S_P$

if $(e_{i,j} < pT_h) \&\& (e_{i,j} \geq -pT_h)$

$$y_c = p_{i,j} + 4 \times e_{i,j} + w_n$$

```

        wn = wn + 4;
    else
        if ( $e_{i,j} \geq pT_h$ )
             $y_c = x_c + 3 \times pT_h$ ;
        elseif ( $e_{i,j} \leq -pT_h - 1$ )
             $y_c = x_c - 3 \times pT_h$ ;
        end
    end
end
elseif  $x_c \in C_P$ 
    if ( $e_{i,j} < pT_h$ ) && ( $e_{i,j} \geq -pT_h$ )
         $y_c = p_{i,j} + 2 \times e_{i,j} + b_{wn}$ ;
         $wn = wn + 2$ ;
    else
        if ( $e_{i,j} \geq pT_h$ )
             $y_c = x_c + pT_h$ ;
        elseif ( $e_{i,j} \leq -pT_h - 1$ )
             $y_c = x_c - pT_h$ ;
        end
    end
end
elseif  $x_c \in NO_s$ 
     $y_c = x_c$ ;
end

```

Step 2.2 *Embedding overhead information*

The overhead information corresponding to the current layer is formed. Suppose that the payload for this layer is \mathcal{P}_C . We know $|\mathcal{P}_C| \leq \mathbf{P}_M$. After the first L_Σ pixels have been processed based on **Step 2.1**, the LSBs of y_c are appended to the payload \mathcal{P}_C and replaced by the overhead information. Hence, for this layer, the watermark bits are composed of two parts: the payload \mathcal{P}_C and the bit sequence \mathcal{C} consisting L_Σ appended bits. All the watermark bits are embedded according to **Step 2.1** for this layer.

Step 3 *Obtaining the watermarked image I_w*

For the convenience of description, **Step 3** is divided into two parts below. If $\mathcal{P}_L \leq \mathbf{P}_M$, go to **Step 3.1**. Otherwise, do **Step 3.2**.

Step 3.1 This embedding layer is marked as the last one. Then, we have $|\mathcal{P}_C| = \mathcal{P}_C$. After **Step 2** are performed, a new watermarked image I_w is obtained.

Step 3.2 we must perform multiple embedding so as to satisfy the desired payload \mathcal{P}_L . First, for this current layer, perform **Step 2** for the payload of length \mathbf{P}_M and the corresponding overhead information. Note that the size of the residual payload is $\mathcal{P}_L - \mathbf{P}_M$. Set $\mathcal{P}_L = \mathcal{P}_L - \mathbf{P}_M$. Go to **step 1** for next layer embedding.

2.4.4 Data extraction and image restoration

The detailed process consists of the following three steps.

Step 1 Extraction of the overhead information

The LSBs of the pixels in I_w are collected into a bitstream \mathcal{B} according to the same order as in embedding. By identifying the EOS symbol in \mathcal{B} , the bits from the start to the EOS, which comprise the compressed map \mathcal{L} , are decompressed by an arithmetic decoder to retrieve the location map. The location map is compressed losslessly by an arithmetic encoder so as to obtain its length, i.e., L_S .

Once L_S is known, vT_h , pT_h , PM will be extracted one by one according to their fixed lengths. If the current layer is the final one, we also will easily obtain the number of embedding layers and the embedded bits for the last layer by extracting EC, respectively.

Step 2 Extraction of the payload \mathcal{P}

For each watermarked pixel y_c , if its location is associated with ‘0’ in the location map, then it is ignored. Otherwise, according to the obtained prediction mode, the local variance σ of the cell I_C surrounding y_c is calculated via (5). $p_{i,j}$ corresponding to the obtained mode is obtained according to (2). If $\sigma < vT_h$, then $e_{i,j}$ is retrieved as follows.

$$e_{i,j} = \begin{cases} \lfloor \frac{e'_{i,j}}{4} \rfloor & e'_{i,j} \in [-4pT_h, 4pT_h - 1] \\ e'_{i,j} + 3 \times pT_h & e'_{i,j} \leq -4pT_h - 1 \\ e_{i,j} - 3 \times pT_h, & e_{i,j} \geq 4pT_h \end{cases} \tag{11}$$

where $e'_{i,j} = y_c - p_{i,j}$. Correspondingly, the embedded watermark is extracted by the following formula: $w_b = e'_{i,j} - 4 \lfloor \frac{e'_{i,j}}{4} \rfloor$. Otherwise, $e_{i,j}$ is retrieved as follows.

$$e_{i,j} = \begin{cases} \lfloor \frac{e'_{i,j}}{2} \rfloor & e'_{i,j} \in [-2pT_h, 2pT_h - 1] \\ e'_{i,j} + pT_h & e'_{i,j} \leq -2pT_h - 1 \\ e_{i,j} - pT_h, & e_{i,j} \geq 2pT_h \end{cases} \tag{12}$$

1-bit watermark is retrieved as $b = e'_{i,j} - 2 \lfloor \frac{e'_{i,j}}{2} \rfloor$. Finally, the retrieved prediction-error $e_{i,j}$ and the predicted value $p_{i,j}$ are substituted into (4) to obtain the original pixel value x_c . The detailed extraction procedure is listed below:

```

if location map == 1
    if  $\sigma < vT_h$ 
        if  $(e'_{i,j} \leq 4pT_h - 1) \&\& (e'_{i,j} \geq -4pT_h)$ 
    
```

```


$$e_{i,j} = \lfloor \frac{e'_{i,j}}{4} \rfloor;$$


$$w_b = e'_{i,j} - 4 \lfloor \frac{e'_{i,j}}{4} \rfloor;$$

elseif  $e'_{i,j} \leq -4pT_h - 1$ 

$$e_{i,j} = e'_{i,j} + 3 \times pT_h;$$

elseif  $e'_{i,j} \geq 4pT_h$ 

$$e_{i,j} = e'_{i,j} - 3 \times pT_h;$$

end
else
if  $(e'_{i,j} \leq 2pT_h - 1) \&\& (e'_{i,j} \geq -2pT_h)$ 

$$e_{i,j} = \lfloor \frac{e'_{i,j}}{2} \rfloor;$$


$$b = e'_{i,j} - 2 \lfloor \frac{e'_{i,j}}{2} \rfloor;$$

elseif  $e'_{i,j} \leq -2pT_h - 1$ 

$$e_{i,j} = e'_{i,j} + pT_h;$$

elseif  $e'_{i,j} \geq 2pT_h$ 

$$e_{i,j} = e'_{i,j} - pT_h;$$

end
end
end
 $x_c = e_{i,j} + p_{i,j};$ 
else
 $x_c = y_c;$ 
end

```

Step 3 Retrieve of the original image

If the number of extracted watermark bits is equal to the capacity calculated by EC, stop extraction and form \mathcal{P} ; else go the next layer extraction and repeat above steps. When the current layer is fully extracted, classify the extracted bitstream into watermark bits and restore the pixels by simple LSB replacement.

3 Experimental results

The proposed reversible watermarking scheme is carried out in MATLAB environment. The capacity vs. distortion comparisons among the proposed method, Wu et al.'s, Luo et al.'s and Feng et al.'s methods are shown in Figs. 9 and 12. The 'Lena', 'Baboon', 'Sailboat' and 'Goldhill' images with size 512×512 are downloaded

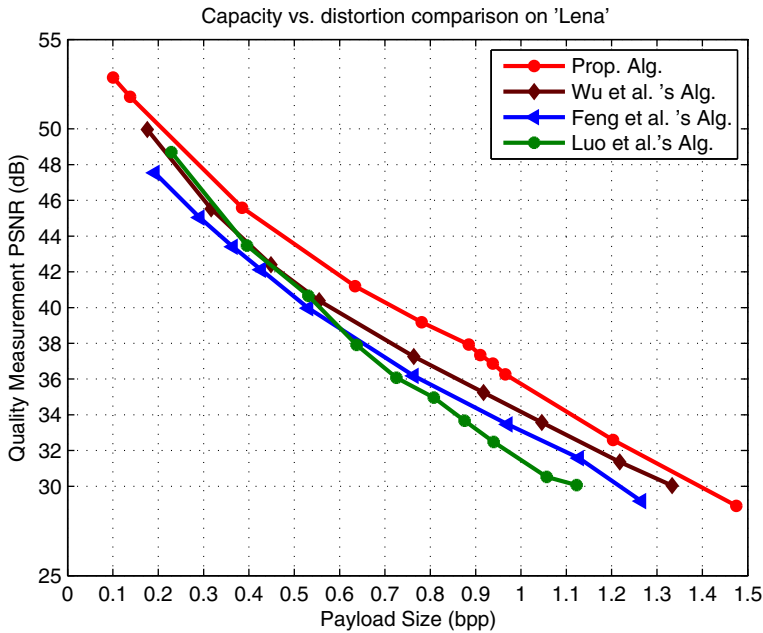


Fig. 9 Capacity vs. distortion comparison on 'Lena'

on the network (<http://sipi.suc.edu/database>). For the convenience of description, Wu, Luo and Feng are used as an abbreviation of Wu et al.'s, Luo et al.'s and Feng et al., respectively.

In paper [20], Wu have implemented a performance comparison between their proposed algorithm with two other RW ones [8, 10]. The algorithm [10] is based on adaptive embedding and integer transform, which embeds $n \log_2 k$ bits into a n -sized image block. The value of k is selected according to complexity level of blocks, so that more bits are adaptively embedded into low-complexity blocks, while fewer bits into blocks with high-complexity. Integer transform may also be considered as a predictor, which actually exploits the mean value to approximately predict each pixel in the block. Abundant experiments and analysis demonstrate that the integer transform itself produces higher distortion than that generated by our prediction-error expansion. Hence, it is certain that our method outperforms Feng's.

Wu have designed four prediction modes in their paper, but each mode has two disadvantages: the first one is that nearly half of all the pixel points are predicted along some single direction (either vertical or horizontal direction); the second one is that the pixels are predicted less accurately along two diagonal directions than that using four neighboring pixels. In the proposed method, to improve prediction performance, we predict almost each pixel with its four neighboring pixels by exploiting mode 1 or 2.

A similar situation also exists in Luo's paper [8] that nearly one quarter of the entire images are estimated along two diagonal directions. And therefore, from this analysis mentioned above, the proposed method can obtain superior performance than Luo's method.

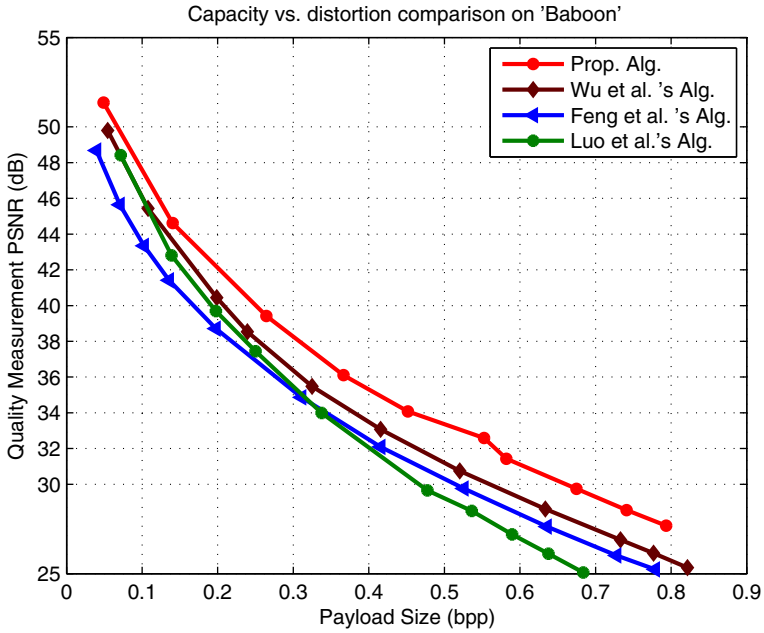


Fig. 10 Capacity vs. distortion comparison on 'Baboon'

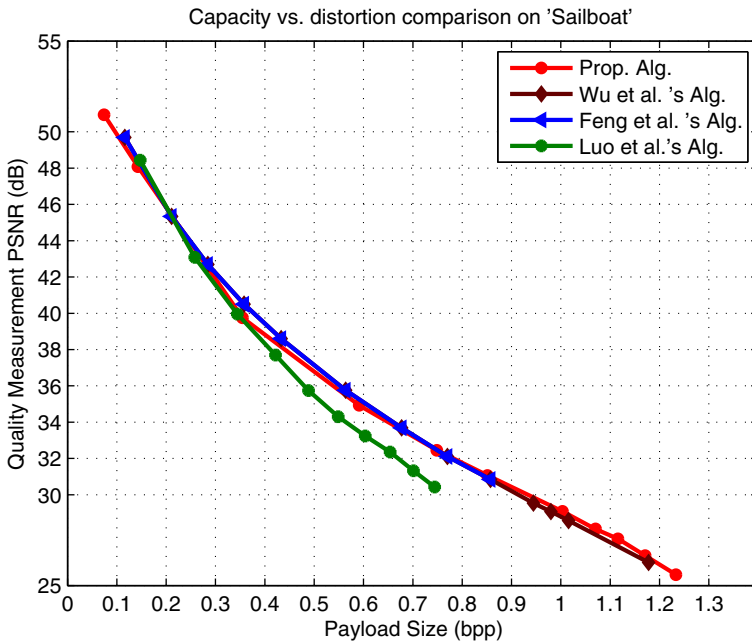


Fig. 11 Capacity vs. distortion comparison on 'Sailboat'

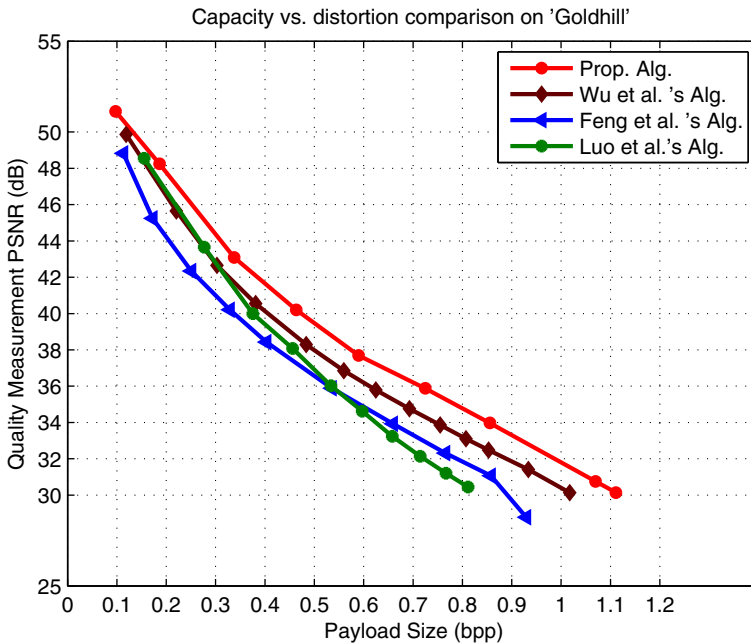


Fig. 12 Capacity vs. distortion comparison on ‘Goldhill’

Before embedding, we will test all six embedding modes according to Section 2.4.2 so as to select the optimum mode capable of achieve the global minimum distortions for the given bit-rates.

For ‘Lena’, it also can be seen from Fig. 9 that our method performs well, and significantly outperforms Wu’s, Peng’s and Luo’s at all embedding rates. The PSNR value of our method is about 0.5–4dB higher than those of Wu’s, Peng’s and Luo’s when the same amount of payloads are embedded. ‘Baboon’ is a typical image with large areas of complex texture, so the obtained bit-rate is slightly lower at the same PSNR. Figure 10 shows that the proposed method achieves higher embedding capacity with lower embedding distortion than the other three methods.

Our method obtains very similar experimental results with Feng’s and Wu’s at low embedding rates due to the high complexity of ‘Sailboat’. As the embedding rates is increased, our method can obtain slightly higher PSNR values than the others (refer to Fig. 11). For ‘Goldhill’, the proposed method can also offer better performance than the other three methods at almost all embedding rates as illustrated in Fig. 12.

4 Conclusions

A new reversible watermark scheme based on multiple prediction modes and adaptive watermark embedding is presented. For any to-be-predicted pixel, the proposed method can adaptively determine the number of embedded bits by analyzing the local complexity of its surrounding neighborhood. The proposed method can achieve bit rates close to 1 bpp for a single embedding layer as nearly half of all pixel points

can possibly carry 2-bit watermark. A series of experiments demonstrate that the proposed method can achieve better performance than the other three methods at all embedding rates.

Acknowledgement This work was supported in part by National NSF of China (No. 61201393, No. 61272498, No. 61001179).

References

1. Alattar AM (2004) Reversible watermark using the difference expansion of a generalized integer transform. *IEEE Trans Image Process* 13(8):1147–1156
2. Coltuc D, Chassery JM (2007) Very fast watermarking by reversible contrast mapping. *IEEE Sig Process Lett* 14(4):255–258
3. Gao XB, An LL (2009) Reversibility improved lossless data hiding. *Signal Process* 89(10):2053–2065
4. Honsinger CW, Jones P, Rabbani M, Stoffe JC (2001) Lossless recovery of an original image containing embedded data. US patent: 6278791
5. Lin WH, Horng SJ, Kao TW, Fan P, Lee CL, Pan Y (2008) An efficient watermarking method based on significant difference of wavelet coefficient quantization. *IEEE Trans Multimedia* 10(5):746–757
6. Lin W-H, Wang Y-R, Horng S-J (2009) A wavelet-tree-based watermarking method using distance vector of binary cluster. *Expert Syst Appl* 36(6):9869–9878
7. Li XL, Yang B, Zeng TY (2011) Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection. *IEEE Trans Image Process* 20(12):3524–3533
8. Luo L, Chen Z, Chen M, Zeng X, Xiong Z (2010) Reversible image watermarking using interpolation technique. *IEEE Trans Inf Forensics Security* 5(1):187–193
9. Ni ZC, Shi YQ, Ansari N, Su W, Sun QB (2008) Robust lossless image data hiding designed for semi-fragile image. *IEEE Trans Circ Syst Video Tech* 18(4):497–509
10. Peng F, Li X, Yang B (2012) Adaptive reversible data hiding scheme based on integer transform. *Signal Process* 92(1):54–62
11. Rosiyadi D, Horng S-J, Fan PZ, Wang X, Khan MK, Pan Y (2012) An efficient copyright protection scheme for e-government document images. *IEEE Multimedia* 19(3):62–73
12. Shi YQ, Ni ZC, Zou D, Liang CY, Xuan GR (2004) Lossless data hiding: fundamentals, algorithms and applications. In: *Proceedings of IEEE international symposium on circuits and systems*, vol 2, pp 33–36
13. Tian J (2003) Reversible data embedding using a difference expansion. *IEEE Trans Circ Syst Video Tech* 13(8):890–896
14. Thodi DM, Rodriguez JJ (2007) Expansion embedding techniques for reversible watermarking. *IEEE Trans Image Process* 16(3):721–730
15. Vleeschouwer C, De Deraigle JF, Macq B (2003) Circular interpretation of bijective transformations in lossless watermarking for media asset management. *IEEE Trans Multimedia* 5(1):97–105
16. Wang X, Li XL, Yang B (2010) High capacity reversible image watermarking based on in transform. In: *Proceedings of ICIP*
17. Wang X, Li XL, Yang B, Guo ZM (2010) Efficient generalized integer transform for reversible watermarking. *IEEE Sig Process Lett* 17(6):567–570
18. Weng SW, Zhao Y, Ni RR, Pan JS (2009) Parity-invariability-based reversible watermarking. *IET Electron Lett* 1(2):91–95
19. Weng SW, Zhao Y, Pan JS, Ni RR (2008) Reversible watermarking based on invariability and adjustment on pixel pairs. *IEEE Sig Process Lett* 45(20):1022–1023
20. Wu H-T, Huang JW (2012) Reversible image watermarking on prediction errors by efficient histogram modification. *Signal Process* 92(12):3000–3009
21. Wu X, Memon N (1997) Context-based, adaptive, lossless image coding. *IEEE Trans Commun* 45(4):437–444
22. Zou DK, Shi YQ, Ni ZC, Su W (2006) A semi-fragile lossless digital watermarking scheme based on integer wavelet transform. *IEEE Trans Circ Syst Video Tech* 16(10):1294–1300



Shaowei Weng was born in 1980, and received her PhD from Institute of Information Science at Beijing Jiaotong University in July 2009. 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. In addition, she participates in 973 and 863 projects as the backbone. She publishes more than 10 papers, and applies one national patent.



Jeng-Shyang Pan received the BS degree in Electronic Engineering from the National Taiwan University of Science and Technology, Taiwan in 1986, the MS degree in Communication Engineering from the National Chiao Tung University, Taiwan in 1988, and the PhD degree in Electrical Engineering from the University of Edinburgh, U.K. in 1996. Currently, he is a Professor in the Department of Electronic Engineering, National Kaohsiung University of Applied Sciences, Taiwan. Professor Pan has published more than 50 journal papers and 120 conference papers. He joins the editorial board for LNCS Transactions on Data Hiding and Multimedia Security, Springer, International Journal of Knowledge-Based Intelligent Engineering Systems, IOS Press, and International Journal of Hybrid Intelligent System, Advanced Knowledge International. He is the Co-Editors-in-Chief for International Journal of Innovative Computing, Information and Control. His current research interests include data mining, information security and image processing.