Multimed Tools Appl (2017) 76:23699–23720
DOI 10.1007/s11042-016-4133-4

High-fidelity lossless data hiding based on predictors selection

Cheng-Hsing Yang[1] · **Chi-Yao Weng**[1] · **Yih-Kai Lin**[1] · **Kuan-Liang Liu**[1]

Received: 3 November 2015 / Revised: 16 September 2016 / Accepted: 7 November 2016 /
Published online: 22 November 2016
© Springer Science+Business Media New York 2016

Abstract Reversible data hiding based on prediction methods is a data hiding technique wherein secret bits can be efficiently hidden into cover images. In this paper, we propose a reversible data hiding method based on multiple prediction methods and local complexity. At each level of data hiding algorithm, we evaluate four prediction methods to decide which method should be chosen to embed secret messages. We propose two tactics to evaluate and select prediction methods. When a prediction method is chosen to perform a shifting and embedding process, a threshold based on local complexity is used to determine which pixel should join the shifting and embedding process. If the local complexity of a pixel is smaller than the threshold, the pixel will join the process; otherwise, the pixel will cease to join the process. Therefore, more pixels will avoid executing pixel shifting. Doing so results in stego-images with lower distortion. The experimental results show that our embedding capacity and quality is superior to those of other approaches.

Keywords Reversible data hiding · Variance · Image quality · Prediction

1 Introduction

Reversible data hiding is an application of data hiding that can not only embed secret messages into images, but also can restore the original images after secret messages have been extracted [5, 24]. Thus far, reversible data hiding has been used in applications such as military maps, medical images [1, 20], and legal texts.

Several reversible data hiding schemes have been proposed [2, 8–18, 21–23, 25–27, 29–34]. The reversible data hiding (RDH) technique can be divided into three domains, spatial domain [1, 2, 8–18, 21–23, 25–27, 29–34], frequency domain [3],

✉ Chi-Yao Weng
cyweng@mail.nptu.edu.tw

[1] Department of Computer Science, National Pingtung University, Pingtung, Taiwan 900

and compression domain [28]. For the spatial domain, the RDH techniques could be performed into three categories, liked as DE-based (difference-expansion-based) [2, 14, 26] and HS-based (histogram-based) [22, 32], pixel-value ordering [29]. According to these studies, many studies have used a variety of techniques to enhance and expand the capacity and/or quality of difference-expansion-based or histogram-based reversible data hiding methods. Tian (2003) presented a reversible data hiding method based on difference expansion [26]. He partitioned an image into pairs of pixel values, selected expandable difference numbers for difference expansion, and embedded a payload which included an authentication hash. In 2004, Alettar improved upon Tian's method by utilizing a quad-based algorithm to enlarge the data embedding capacity [2]. Subsequently, some authors used prediction and selection methods to enhance the difference-expansion-based data hiding approach. For example, Lee et al. (2010) predicted a pixel by its left-hand and upper adjacent pixels [13]. In 2011, Yang and Zeng programmed a method of choosing a pixel by the character of its local pixels [16]. Conversely, Ni et al. (2006) presented a reversible data hiding method based on the histogram [22]. Their method guaranteed that the change in each pixel in the stego-image remained within ±1. Doing so ensured that the peak signal-to-noise ratio (PSNR) value of the stego-image would be at least 48 dB.

However, because Ni et al.'s method (2006) used pixel values in the original image to create a histogram, the peak points of the histogram were not high enough. Therefore, many researchers applied prediction methods to improve the structure of the histogram. For example, Lin et al.'s difference-image scheme [18], Tsai et al.'s predictive coding scheme [27], and Yang and Tsai's interleaving prediction scheme [31]. Li et al.'s difference-image scheme sorted out many methods and provided a framework of histogram-based reversible data hiding with which to create a framework for data hiding and date extraction [15]; however, the testability of this framework was a minor problem.

Usually, steganography is one of the most important components of an information hiding algorithm. Some detection approaches have been proposed such as the spatial rich model (SRM) [7] and maxSRMd2 [4]. More methods, such as syndrome-trellis codes (STCs) and matrix embedding, have been proposed for minimizing the distortion in steganography and passing the detection of hidden information [6, 19].

In this paper, we propose a reversible data hiding scheme based on four prediction methods and local complexity. Here, secret data is embedded into the cover image, level by level [18]. At each level of data hiding alogrithm, an interleaving grouping approach is applied to divide the cover image into four groups [31]. Next, one of the four prediction methods is chosen to apply to each group within a level. In processing the pixels of each group, the predicted error is computed between the current pixel value and its predicted pixel value. Then, the local complexity, i.e., pixel variance, is analyzed to determine whether or not the predicted error will join the process of pixel shifting and data embedding. After pixel shifting and data embedding have been executed within one level, the difference between a cover pixel and a stego-pixel remains within ±1.

The rest of this paper is organized as follows. In Section 2, the interleaving prediction method proposed by Yang and Tsai in 2010 is introduced [31], the predicted error expansion method proposed by Lee et al. in 2010 is explained [13], and a general framework proposed by Li et al. in 2013 is introduced [15]. In Section 3, our reversible data hiding is presented. The experimental result of our scheme is demonstrated in Section 4. Finally, the conclusion is given in Section 5.

2 Related work

2.1 Yang-Tsai's interleaving prediction method

Yang and Tsai proposed a reversible data hiding scheme based on interleaving prediction and histogram shifting [31]. The interleaving prediction method is used to promote the altitude of the peak point in histogram. The idea likes a chessboard, as shown in Fig. 1, and contains two stages. In the first stage, pixels with white-color are processed and predicted by their neighboring pixels of black-color. After that, in the second stage, black pixels are processed and predicted by their neighboring white pixels.

The detailed steps of the first stage are as follows. Let $P_{i,j}$ be a white pixel, where (i, j) is the location, and $D_{i,j}$ be the predicted error between $P_{i,j}$ and its predicted value. All predicted errors are collected to generate a histogram $HS(D)$. Then, the following steps are executed, where each $D_{i,j}$ is changed into a new value $D'_{i,j}$.

Step 1: Find two pairs of peak and zero points ($Peak_1$, $Zero_1$) and ($Peak_2$, $Zero_2$) from the histogram $HS(D)$, such that $Zero_2<Peak_2<Peak_1<Zero_1$.

Step 2: Shift the value of the predicted error $D_{i,j}$ by 1 in the following cases.

Case A: Change all values in the range of [$Zero_2$ +1, $Peak_2$-1] to the left by 1 unit. This indicates that $D'_{i,j}$ is set to $D_{i,j}$-1 as $D_{i,j} \in$ [$Zero_2$ +1, $Peak_2$-1].

Case B: Change all values in the range of [$Peak_1$ +1, $Zero_1$-1] to the right by 1 unit. This shows that $D'_{i,j}$ is set to $D_{i,j}$ +1 as $D_{i,j} \in$ [$Peak_1$ +1, $Zero_1$-1].

Step 3: Conceal a secret bit when the predicted error $D_{i,j}$ is equal to $Peak_1$ or $Peak_2$ as the following two cases.

Case A: If the to-be-embedded bit is 0, predicted error $D_{i,j}$ is unchanged. It indicates that $D'_{i,j}$ is set to $D_{i,j}$.

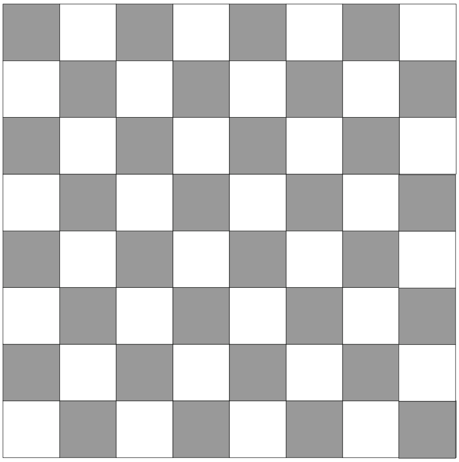


Fig. 1 A chessboard with *white pixels* and *black pixels*

Case B: If the to-be-embedded bit is 1,

$$D'_{i,j} = \begin{cases} D_{i,j}+1, & \text{if } D_{i,j} = Peak_1 \\ D_{i,j}-1, & \text{if } D_{i,j} = Peak_2 \end{cases}$$

Step 4 Transform all the predicted errors $D'_{i,j}$ into pixel values by running the inversed interleaving prediction. Then, output the stego-image.

The outputted stego-image is used as the input image of the second stage. The second stage processes all black pixels. After the second stage is processed, another two pairs of peak and zero points ($Peak_1$, $Zero_1$) and ($Peak_2$, $Zero_2$) are created and the final stego-image is obtained.

2.2 Lee et al.'s prediction-error expansion method

Lee et al.'s method belonged to category of difference expansion [13]. This work took the concept of the predicted error and a threshold TH into consideration. For a current pixel $P_{i,j}$, they used its left and up adjacent pixels to create a predicted value $\hat{P}_{i,j}$ and calculated the predicted error between $P_{i,j}$ and $\hat{P}_{i,j}$. To be reversible, the first row and first column of the image is unused for data hiding and the image is processed from left to right and from top to bottom. The detailed embedding algorithm of Lee et al.'s approach is described below.

Assume that the left and up adjacent pixels of the current pixel $P_{i,j}$ are (P_1, P_2). The predicted value $\hat{P}_{i,j}$ is defined as Eq. (1).

$$\hat{P}_{i,j} = \left\lfloor \frac{P_1 + P_2}{2} \right\rfloor \tag{1}$$

The predicted error is estimated by $d = |\hat{P}_{i,j} - P_{i,j}|$. A bit s is embedded into pixel $P_{i,j}$ according the following cases, where $P'_{i,j}$ is the resulted stego-pixel.

Case A: If $d \le TH$, bit s can be hidden by the following function:

$$P'_{i,j} = \begin{cases} \hat{P}_{i,j} + 2 \times d + s, \text{if } \hat{P}_{i,j} \le P_{i,j} \\ \hat{P}_{i,j} - 2 \times d + s, \text{otherwise} \end{cases} \tag{2}$$

Case B: If $d > TH$, the current pixel cannot tolerate embedding a secret bit, and this pixel needs to run the following function:

$$P'_{i,j} = \begin{cases} P_{i,j} + (TH+1), \text{if } \hat{P}_{i,j} \le P_{i,j} \\ P_{i,j} - TH, \text{ otherwise} \end{cases} \tag{3}$$

2.3 Li et al.'s general framework method

Li et al. proposed a general framework to construct histogram-based reversible data hiding in 2013 [15]. By the proposed framework, one can get a reversible data hiding algorithm by simply designing the so-called shifting and embedding functions. One of their embedding algorithms is described below.

Fig. 2 A 3x3 block

x_1	x_2	x_3
x_4	x_5	x_6
x_7	x_8	x_9

For a 3 × 3 block $x = x_1, x_2,\ldots, x_9$ shown in Fig. 2, they took the following linear predictor with non-uniform weight to predict x_5.

$$\hat{x}_5 = \frac{1}{16}(x_1 + x_3 + x_7 + x_9) + \frac{3}{16}(x_2 + x_4 + x_6 + x_8) \tag{4}$$

The prediction-error is denoted as $e_5 = x_5 - \hat{x}_5$.

Utilize the smooth pixels for reversible data embedding whereas ignoring the noisy ones will significantly reduce the embedding distortion. Then they took the following function

$$\mathrm{C}(\mathbf{x}) = \max\{x_1, \ldots, x_4, x_6, \ldots, x_9\} - \min\{x_1, \ldots, x_4, x_6, \ldots, x_9\} \tag{5}$$

to measure the local complexity of pixel x_5 and used an integer-valued parameter s to select smooth pixels.

For an integer threshold $t > 0$, take $t_l = \left\lfloor \frac{t}{2} \right\rfloor$ and $t_r = \left\lceil \frac{t}{2} \right\rceil$. They then defined as following:

(1) $S = \{\mathbf{x} \in Z^9 : -t_l \leq e_5 < t_r, C(\mathbf{x}) < s\}$ and $T = Z^9 - S$.
(2) For $\mathbf{x} \in T$, $g(x) = \{\, (x_1, \cdots, x_4, x_5 + t_r, x_6, \cdots, x_9)$,if $e_5 \geq^5 t_r$ and $C(x) < s$ $(x_1, \cdots, x_4, x_5 - t_r, x_6, \cdots, x_9)$,if $e_5 <^5 - t_l$ and $C(x) < s$ x ,if $C(x) \geq s$
(3) For $\mathbf{x} \in S$ and $m \in \{0,1\}$, $fm(\mathbf{x}) = (x_1,\ldots, x_4, x_5 + \lfloor e_5 \rfloor + m, x_6,\ldots, x_9)$.

In the above definitions, S means the set of blocks where each block will be used to embed a secret bit m. T is the set of blocks where each block will do a shifting operation or nothing.

2.4 Our proposed methods

Our method takes four different prediction methods and their corresponding variance calculations into consideration. Firstly, the cover image is divided into four groups – Group1, Group2, Group3, and Group4 – using an interleaving grouping method. Figure 3 shows a grouping result, wherein each cell indicates a pixel and the number in the cell indicates the group number of the cell. Then, each stage with four processes is used to process the four groups, respectively. At each stage, the four prediction methods are evaluated, and the one with the largest capacity or the highest efficiency ratio is selected. Using the chosen prediction method, all pixels' predicted errors and variance values are created using their neighboring pixels. A threshold TH is determined by variance values to decide whether or not a pixel will join the shifting and embedding process, is given in following subsections.

Fig. 3 Interleaving grouping with four groups

1	2	1	2	1
3	4	3	4	3
1	2	1	2	1
3	4	3	4	3
1	2	1	2	1

2.5 Data hiding algorithm

We take the first stage, which processes Group1, to describe our data hiding method. As shown in the gray part of Fig. 3, one Group1 pixel, says $P_{i,j}$, is surrounded by eight neighboring pixels which belong to the other groups. Four prediction errors and four variance values of the four prediction methods are calculated as follows.

(1) Chessboard prediction: For each pixel $P_{i,j}$ in Group1, its neighboring Group2 and Group3 pixels are chosen to predict $P_{i,j}$. The predicted value $\hat{P}_{i,j}$ is the averaged gray value of the chosen pixels. The predicted error $D_{i,j}$ is defined as $P_{i,j} - \hat{P}_{i,j}$. Besides, a variance value $V_{i,j}$ is defined as the variance of the chosen pixels.

(2) Edge prediction: For each pixel $P_{i,j}$ in Group1, the difference of its neighboring Group2 pixels and the difference of its neighboring Group3 pixels are calculated. The two pixels with the smaller difference value are chosen to predict $P_{i,j}$. Then, the predicted error $D_{i,j}$ and the variance value $V_{i,j}$ are calculated.

(3) Squared prediction: For each pixel $P_{i,j}$ in Group1, $P_{i,j}$ is predicted by the averaged gray value of its neighboring Group2, Group3, and Group4 pixels are chosen to predict $P_{i,j}$. Then, the predicted error $D_{i,j}$ and the variance value $V_{i,j}$ are calculated.

(4) Max-min-omitted prediction: For each pixel $P_{i,j}$ in Group1, its neighboring Group2 and Group3 pixels excluding the two pixels with the maximal value and the minimal value are chosen to predict $P_{i,j}$. Then, the predicted error $D_{i,j}$ and the variance value $V_{i,j}$ are calculated.

2.5.1 Algorithm data-embedding-of-the-first-stage

Input: Cover image *I*, secret message *S*, embedding cost *C*, tactic *A* or tactic *B*.

Output: Stego Image *I'*, the couple data ($Peak_1$, $Zero_1$) and ($Peak_2$, $Zero_2$), a variance threshold *TH*, prediction method *M*.

Step 1: For each pixel $P_{i,j}$ in Group1, a predicted error $D_{i,j}$ is calculated by each of the following four prediction methods M_1, M_2, M_3, and M_4:

- M_1: Chessboard prediction

Case A: If $P_{i,j}$ is has only two neighboring pixels P_1 and P_2 belonging to Group2 and Group3, the predicted error $D_{i,j}$ is given as

$$D_{i,j} = P_{i,j} - \left\lfloor \frac{P_1 + P_2}{2} \right\rfloor. \tag{6}$$

Case B: If $P_{i,j}$ has only three neighboring pixels P_1, P_2 and P_3 belonging to Group2 and Group3, the predicted error $D_{i,j}$ is given as

$$D_{i,j} = P_{i,j} - \left\lfloor \frac{P_1 + P_2 + P_3}{3} \right\rfloor. \tag{7}$$

Case C: If $P_{i,j}$ is has four neighboring pixels P_1, P_2, P_3, and P_4 belonging to Group2 and Group3, the predicted error $D_{i,j}$ is given as

$$D_{i,j} = P_{i,j} - \left\lfloor \frac{P_1 + P_2 + P_3 + P_4}{4} \right\rfloor. \tag{8}$$

- M_2: Edge prediction

Case A: The same to Eq. (6).

Case B: The same to Eq. (7).

Case C: If $P_{i,j}$ has two neighboring Group2 pixels P_1 and P_2 and two neighboring Group3 pixels P_3 and P_4, the predicted error $D_{i,j}$ is given as

If $|P_1\text{-}P_2| > |P_3\text{-}P_4|$, $D_{i,j} = P_{i,j} - \left\lfloor \frac{P_3 + P_4}{2} \right\rfloor$;

else,

$$D_{i,j} = P_{i,j} - \left\lfloor \frac{P_1 + P_2}{2} \right\rfloor. \tag{9}$$

- M_3: Squared prediction

Case A: If $P_{i,j}$ has only three neighboring pixels P_1, P_2, and P_3, the predicted error $D_{i,j}$ is given as

$$D_{i,j} = P_{i,j} - \left\lfloor \frac{P_1 + P_2 + P_3}{3} \right\rfloor. \tag{10}$$

Case B: If $P_{i,j}$ has only five neighboring pixels P_1, P_2, P_3, P_4, and P_5, the predicted error $D_{i,j}$ is given as

$$D_{i,j} = P_{i,j} - \left\lfloor \frac{P_1 + P_2 + P_3 + P_4 + P_5}{5} \right\rfloor. \tag{11}$$

Case C: If $P_{i,j}$ has eight neighboring pixels P_1, P_2, P_3, P_4, P_5, P_6, P_7, and P_8, the redicted error $D_{i,j}$ is given as

$$D_{i,j} = P_{i,j} - \left\lfloor \frac{P_1 + P_2 + P_3 + P_4 + P_5 + P_6 + P_7 + P_8}{8} \right\rfloor. \tag{12}$$

- M_4: Max-min-omitted prediction

Case A: The same to Eq. (6).

$$D_{i,j} = P_{i,j} - \left\lfloor \frac{P_2 + P_3}{2} \right\rfloor. \tag{13}$$

Case B: The same to Eq. (7).

Case C: If $P_{i,j}$ has four neighboring Group2 and Group3 pixels P_1, P_2, P_3, and P_4 with $P_{1\le}P_{2\le}P_{3\le}P_4$, the predicted error $D_{i,j}$ is given as

Step 2: For each prediction method M_i, a histogram is created from its predicted errors and two pairs of peak and zero points are ($Peak_1$, $Zero_1$) and ($Peak_2$, $Zero_2$) with $Zero_{2i}<Peak_{2i}<Peak_{1i}<Zero_{1i}$, $i = 1, 2, 3, 4$.

Step 3: Let the number of pixels falling in $Peak_{1i}$, $Peak_{2i}$, [$Peak_{1i}$+1, $Zero_{1i}$], and [$Zero_{2i}$, $Peak_{2i}$-1] be J_{1i}, J_{2i}, I_{1i}, and I_{2i}, respectively. Select a prediction method M in the following two cases.

Case A: If tactic A is inputted, the selected prediction method M is defined as

$$M = \arg \max_{M_i, i=1,\dots 4} \{J_{1i} + J_{2i}\}. \tag{14}$$

Case B: If tactic B is inputted, the selected prediction method M is the method with the maximal efficiency ratio, where the efficiency ratio R_i is defined as

$$R_i = \frac{J_{1i} + J_{2i}}{I_{1i} + I_{2i}}, i = 1,\ 2,\ 3,\ 4. \tag{15}$$

Step 4: The selected method M is used to process Group1 to embed secret data. Using method M, let its predicted errors $D_{i,j}$, two pairs of peak and zero points ($Peak_1$, $Zero_1$) and ($Peak_2$, $Zero_2$) with $Zero_2<Peak_2<Peak_1<Zero_1$ have been calculated. Additionally, for each pixel $P_{i,j}$ in Group1, a variance value $V_{i,j}$ is calculated by the selected prediction method. The variance calculations of the four prediction methods are similar, here only the chessboard prediction's calculation is shown below:

Case A: If $P_{i,j}$ has only two neighboring pixels P_1 and P_2 belonging to Group2 and Group3, the variance value $V_{i,j}$ is given as

$$V_{i,j} = 2 \cdot \left[\left(P_1 - \left\lfloor \frac{P_1 + P_2}{2} \right\rfloor \right)^2 + \left(P_2 - \left\lfloor \frac{P_1 + P_2}{2} \right\rfloor \right)^2 \right]. \tag{16}$$

Case B: If $P_{i,j}$ has only three neighboring pixels P_1, P_2, and P_3 belonging to Group2 and Group3, the variance value $V_{i,j}$ is given as

$$V_{i,j} = \left\lfloor \frac{4}{3} \cdot \left[\left(P_1 - \left\lfloor \frac{P_1 + P_2 + P_3}{3} \right\rfloor \right)^2 + \left(P_2 - \left\lfloor \frac{P_1 + P_2 + P_3}{3} \right\rfloor \right)^2 + \left(P_3 - \left\lfloor \frac{PP_1 + P_2 + P_3}{3} \right\rfloor \right)^2 \right] \right\rfloor. \tag{17}$$

Case C: If $P_{i,j}$ has four neighboring pixels P_1, P_2, P_3, and P_4 belonging to Group2 and Group3, the variance value $V_{i,j}$ is given as

$$\begin{aligned} V_{i,j} = & \left(P_1 - \left\lfloor \frac{P_1+P_2+P_3+P_4}{4} \right\rfloor\right)^2 \\ & + \left(P_2 - \left\lfloor \frac{P_1+P_2+P_3+P_4}{4} \right\rfloor\right)^2 \\ & + \left(P_3 - \left\lfloor \frac{P_1+P_2+P_3+P_4}{4} \right\rfloor\right)^2 \\ & + \left(P_4 - \left\lfloor \frac{P_1+P_2+P_3+P_4}{4} \right\rfloor\right)^2. \end{aligned} \tag{18}$$

Step 5: Two arrays V_{peak}, and V_{shift} are created, where their rth elements are defined as

$$V_{peak}[r] = \left|\left\{P_{i,j} \middle| P_{i,j} \in Group1, D_{i,j} = Peak_1 or Peak_2, V_{i,j} = r\right\}\right| \tag{19}$$

$$V_{shift}[r] = \left|\left\{P_{i,j} \middle| P_{i,j} \in Group1, D_{i,j} \in [Peak_1+1, Zero_1] or \, [Zero_2, Peak_2 - 1], V_{i,j} = r\right\}\right| \tag{20}$$

where r is a non-negative integer and |.| means the size of the set.

Step 6: Using the embedding cost C, TH is the maximal non-negative value satisfying

$$\frac{V_{shift}[r]}{V_{peak}[r]} < C, 0 \le r \le TH. \tag{21}$$

Step 7: Use TH to distinguish whether a Group1's pixel $P_{i,j}$ will join the shifting and embedding process or not by setting the following flag:

$$Flag_{i,j} = \begin{cases} 1, & \text{if } V_{i,j} \le TH. \\ 0, & \text{if } V_{i,j} > TH. \end{cases}$$

If $Flag_{i,j} = 0$, it means that the pixel $P_{i,j}$ in Group1 will not join the shifting and embedding process; If $Flag_{i,j} = 1$, it means that the pixel $P_{i,j}$ in Group1 will join the shifting and embedding process.

Step 8: Run the following cases for each predicted error $D_{i,j}$ and its variance $V_{i,j}$.

Case A: If $Flag_{i,j} = 1$ and the predicted error $D_{i,j}$ is equal to $Peak_1$ or $Peak_2$, fetch a secret bit from S and do the following two cases:

Case A1: If to-be-embedded-bit is 0, $D'_{i,j}$ is set to $D_{i,j}$.

Case A2: If to-be-embedded-bit is 1,

$$\begin{cases} \text{If } D_{i,j} = Peak_1, & D'_{i,j} = D_{i,j} + 1 \\ \text{If } D_{i,j} = Peak_2, & D'_{i,j} = D_{i,j} - 1. \end{cases}$$

Case B: If $Flag_{i,j} = 1$ and the predicted error $D_{i,j}$ falls into the range of [$Zero_2 + 1$, $Peak_2$-1] or [$Peak_1 + 1$, $Zero_1$-1], shift predicted error $D_{i,j}$ by one unit as follows:

$$\begin{cases} \text{If } D_{i,j} \in [Peak_1 + 1, Zero_1 - 1], & D'_{i,j} = D_{i,j} + 1 \\ \text{If } D_{i,j} \in [Zero_2 + 1, Peak_2 - 1], & D'_{i,j} = D_{i,j} - 1. \end{cases}$$

Case C: If $Flag_{i,j} = 0$, do nothing, that is, $D'_{i,j}$ is set to $D_{i,j}$.

Step 9: Transform each predicted error $D'_{i,j}$ into pixel value $P'_{i,j}$ by the inverse of the prediction method M. Finally, all pixel values $P'_{i,j}$ form the stego-image I'.

Step 10: Output two pairs ($Peak_1$, $Zero_1$) and ($Peak_2$, $Zero_2$), a variance threshold TH, and a prediction method M.

The above algorithm, which processes Group1, is for the first stage of level-one data hiding. For reducing the time complexity, the second, third, and forth stages will use the same selected method M of the first stage. In Step 6, if none of r values satisfies Eq. (21), TH will not exist. In this case, none of secret data will be embedded in this stage. To avoid that none of data can be embedded in the next following levels, we increase the embedding cost C by a fixed value whenever a new level is processed. Note that the outputted parameters in Step 10 can be seen as secret data and can be embedded in the next stage. So, only the outputted parameters of the last stage cannot be embedded.

2.6 Data extracting and recovering algorithm

The data extracting and recovering algorithm is similar with the data embedding algorithm. Following shows the detailed steps of the data extracting and recovering algorithm for the last stage.

2.6.1 Algorithm Extracting-and-Recovering-of-the-Last-Stage

Input: Stego image I', two pairs ($Peak_1$, $Zero_1$) and ($Peak_2$, $Zero_2$), a variance threshold TH, a prediction method M.

Output: Cover image I, secret message S.

Step 1: For each pixel $P'_{i,j}$ of stego-image I', compute its predicted error $D'_{i,j}$ and variance value $V'_{i,j}$ by the prediction method M.

Step 2: For each predicted error $D'_{i,j}$ and its variance value $V'_{i,j}$, run the following data extracting and pixel recovering process.

If $V'_{i,j} > TH$, do nothing, that is, $D_{i,j} = D'_{i,j}$.

If $V'_{i,j \le} TH$, do the following cases.

Case A: If the predicted error $D'_{i,j}$ is equal to $Peak_1$ or $Peak_2$, obtain a secret bit 0 and recover the predicted error as $D_{i,j} = D'_{i,j}$.

Case B: If $D'_{i,j} = Peak_1 + 1$, extract a secret bit 1 and recover the predicted error as $D_{i,j} = D'_{i,j}$-1.

If $D'_{i,j} = Peak_2$-1, extract a secret bit 1 and recover the predicted error as $D_{i,j} = D'_{i,j} + 1$.

Case C: If $D'_{i,j}$ falls into the range of [Zero_2, Peak_2-2], recover the predicted error as $D_{i,j} = D'_{i,j} + 1$.

If $D'_{i,j}$ falls into the range of [$\mathrm{Peak}_1 + 2$, Zero_1], recover the predicted error as $D_{i,j} = D'_{i,j}$-1.

Step 3: Transform all predicted errors $D_{i,j}$ into pixels $P_{i,j}$ to form the cover image I.

Step 4: Output the cover image I and all extracted message S.

Note that variance value $V'_{i,j}$ in the data extracting and recovering process is the same to the variance value $V_{i,j}$ in the corresponding data hiding process. The reason is that both $V_{i,j}$ and $V'_{i,j}$

are calculated from neighboring pixels, which are unchanged after the corresponding data hiding process is finished. Therefore, $V'_{i,j}$ can be used to judge whether $D'_{i,j}$ joins the data extracting and pixel recovering process.

2.7 Overflow/Underflow

Overflow and underflow occur when the shifting and embedding process causes a pixel's value to be over 255 or under 0. To cope with this issue, we pre-modify the pixels that are susceptible to overflow or underflow, and use a location map to record this information. This information can be seen as a part of the secret data, given that it is embedded into the cover image.

In our histogram-based reversible data hiding method, each pixel will change by ±1, at most, at every level. Therefore, before embedding secret data into the image at a given level, we pre-transform pixels with values of 0 and 255 into pixels with values of 1 and 254, respectively, to prevent overflow and underflow as follows:

$$P_{i,j} = \begin{cases} 1, & \text{if } P_{i,j} = 0 \\ 254, & \text{if } P_{i,j} = 255. \end{cases}$$

Also, judge those modifications, the location map is created before those modifications as follows:

$$L[k] = \begin{cases} 0, & \text{if } P_{i,j} = 1 \ \text{ or } P_{i,j} = 254 \\ 1, & \text{if } P_{i,j} = 0 \text{ or } P_{i,j} = 255, \end{cases}$$

where k is an index for those recorded pixels. Therefore, in each level, the size of the location map is k bits if there are k pixels with values equal to values 0, 1, 254, or 255.

3 Experimental results

In this section, we provide the resultant of embedding capacity and image quality to demonstrate the performance of our proposed scheme. In our experiment, ten gray images with size

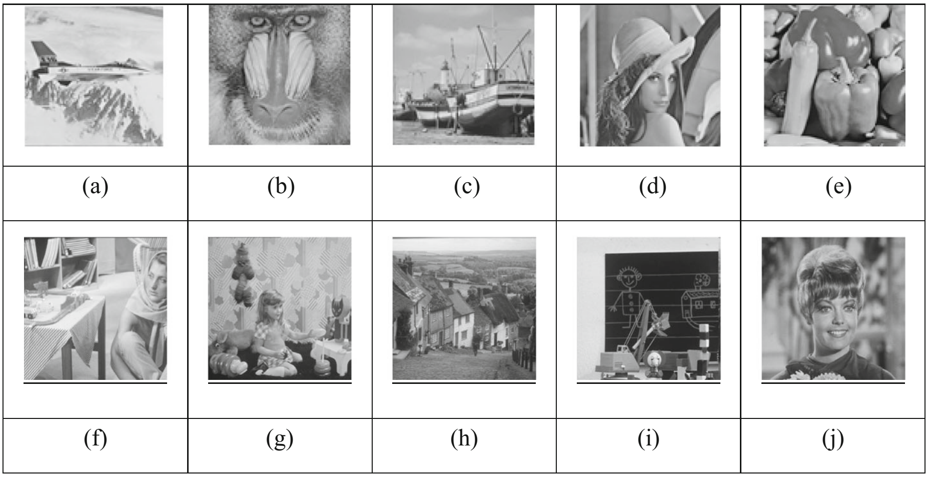


Fig. 4 The ten cover images with size 512 × 512; (**a**) Airplane; (**b**) Baboon; (**c**) Boat; (**d**) Lena; (**e**) Peppers; (**f**) Barb; (**g**) Girl; (**h**) Gold; (**i**) Toys; (**j**) Zelda

Table 1 Each image's capacity, PSNR, and the prediction M_i on different levels. (Tactic *A*, embedding cost $C = 30$)

Image		Airplane	Baboon	Boat	Lena	Peppers	Barb	Girl	Sailboat	Toys	Zelda
Level 1	Capacity	103,956	31,719	58,759	99,936	50,725	86,178	86,178	40,767	64,181	87,860
	PSNR	51.17	55.10	51.01	49.06	52.96	48.92	48.92	55.08	50.38	48.93
	M	M_1	M_4	M_2	M_1	M_2	M_2	M_2	M_2	M_2	M_2
Level 4	Capacity	272,609	130,702	192,922	255,913	172,365	231,638	231,638	170,258	178,750	225,523
	PSNR	43.12	43.69	41.60	40.92	42.05	39.14	39.14	43.58	40.61	39.28
	M	M_3	M_2	M_2	M_2	M_2	M_1	M_1	M_2	M_2	M_1
Level 7	Capacity	394,342	207,013	277,664	363,636	252,192	314,262	314,262	255,873	256,342	287,933
	PSNR	39.15	39.47	38.23	36.18	38.54	36.26	36.26	39.61	36.74	36.71
	M	M_1	M_2	M_2	M_2	M_2	M_2	M_2	M_2	M_2	M_1
Level 10	Capacity	479,299	270,876	353,120	432,191	329,451	378,355	378,355	337,084	315,602	347,859
	PSNR	35.84	37.38	35.63	34.48	36.32	33.83	33.83	37.23	34.323	34.10
	M	M_1	M_2	M_2	M_4	M_2	M_2	M_2	M_2	M_2	M_2
Level 20	Capacity	687,704	470,440	510,963	629,919	528,247	499,396	499,396	565,915	458,959	453,375
	PSNR	31.65	32.46	30.02	29.49	30.87	28.97	28.97	32.07	28.78	29.30
	M	M_4	M_4	M_2	M_2	M_2	M_2	M_2	M_2	M_2	M_2

Table 2 Each image's capacity, PSNR, and the prediction M_i on every level. (Tactic B, embedding cost $C = 30$)

Image		Airplane	Baboon	Boat	Lena	Peppers	Barb	Girl	Sailboat	Toys	Zelda
Level 1	Capacity	103,956	16,873	60,215	90,345	79,711	42,311	74,646	38,856	56,250	96,788
	PSNR	51.17	61.70	47.56	49.39	49.65	58.90	49.98	59.10	51.66	49.72
	M	M_1	M_3	M_1	M_4	M_4	M_3	M_3	M_1	M_4	M_1
Level 4	Capacity	282,135	61,429	207,521	239,069	209,481	149,425	148,459	138,433	75,632	192,284
	PSNR	43.30	53.90	39.95	41.37	40.23	50.71	43.12	50.52	44.48	40.22
	M	M_4	M_3	M_3	M_3	M_2	M_3	M_3	M_3	M_3	M_4
Level 7	Capacity	398,864	112,359	338,425	331,799	299,571	243,970	279,913	236,267	172,771	182,810
	PSNR	40.12	48.48	37.23	39.88	38.73	47.64	40.30	45.85	39.95	36.95
	M	M_4	M_1	M_3	M_3	M_3	M_3	M_1	M_1	M_1	M_3
Level 10	Capacity	507,270	144,627	466,401	416,004	362,713	318,176	371,309	291,936	218,396	235,411
	PSNR	37.64	47.72	34.51	38.00	37.27	45.14	38.35	42.11	37.03	35.23
	M	M_2	M_3	M_1	M_3	M_3	M_3	M_3	M_4	M_3	M_3
Level 20	Capacity	731,523	264,966	654,985	623,648	533,181	509,590	509,539	381,307	367,762	423,027
	PSNR	33.05	44.20	32.23	34.58	32.81	39.69	35.79	35.14	35.03	32.99
	M	M_1	M_3	M_4	M_3	M_3	M_3	M_3	M_4	M_3	M_3

Table 3 The compared resultant among our approach (embedding cost $C = 30$), Lin's, Hsiao's and Yang's schemes

Images	Lin et al.'s method [18]	Hsiao et al.'s method [17]	Lin and Hsueh's method [30]	Yang and Tsai's method [31]	Our approach (tactic *A*)	Our approach (tactic *B*)
Airplane	362,847	286,488	367,392	397,712	742,082	750,408
Baboon	230,079	138,398	162,544	184,468	576,014	575,917
Boat	314,196	266,724	307,937	294,944	510,963	513,744
Lena	346,568	303,700	309,166	420,986	596,903	604,839
Peppers	342,175	303,736	356,450	374,837	570,497	557,048
Average	319,173	259,809	300,698	334,589	599,291	600,391
Ratio	53.16 %	43.27 %	50.08 %	55.72 %	99.81 %	100 %
PSNR	30.19	30.00	30.26	30.25	30.27	32.75

Table 4 The compared resultant between our approach and other schemes, for an embedding capacity of 20,000 bits

Image	Hu et al. [10]	Luo et al. [11]	Li et al. [16]	Hong [21]	Li et al. [15]	Our approach (Tactic *A*)	Our approach (Tactic *B*)
Lena	52.86	53.83	54.82	54.92	55.93	61.32	62.02
Airplane	54.63	55.43	56.84	58.58	59.26	62.33	62.33
Peppers	50.64	52.19	52.55	52.16	53.31	57.32	61.42
Sailboat	50.69	52.17	53.25	52.03	53.19	57.78	57.63
Average	52.21	53.41	54.37	54.42	55.42	59.69	60.85

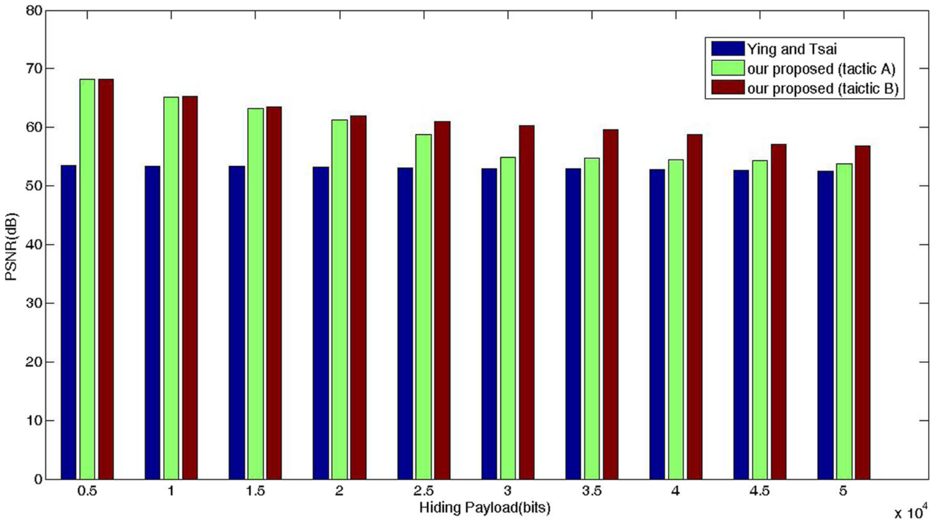


Fig. 5 The PSNR comparison of Lena image between our approach and Yang and Tsai's scheme based on various hiding payload

Table 5 Comparison between Lee et al.'s scheme and our proposed scheme with the same embedding ratio

	Lee et al.'s scheme [13]		Our approach (tactic *A*)		Our approach (tactic *B*)	
	ER(bpp)	PSNR	ER(bpp)	PSNR	ER(bpp)	PSNR
Lena	0.14	48.47	0.14	58.59	0.14	58.89
	0.98	32.17	0.98	46.92	0.98	46.30
Baboon	0.05	48.25	0.05	62.92	0.05	63.11
	0.62	30.02	0.62	46.48	0.62	47.39

512 × 512, which are depicted in Fig. 4, are used, and the secret message is obtained by random generation. To estimate the image quality, we applied the function of peak-signal-to-noise-ratio (PSNR) defined as Eq. (22). To estimate the embedding capacity, the function of ER (Embedded Ratio; bpp) is adopted, where ER = total embedded bits / size of cover image. In order to run the program efficiently, the embedding cost *C* is only inputted on the first level. Then, the *C* value is added by 0.5 on each of the following levels. Besides, for each level, the four stages will use the same prediction method which is chosen by the first stage.

$$PSNR = 10 \times \log_{10}\left(\frac{255^2}{MSE}\right), \tag{22}$$

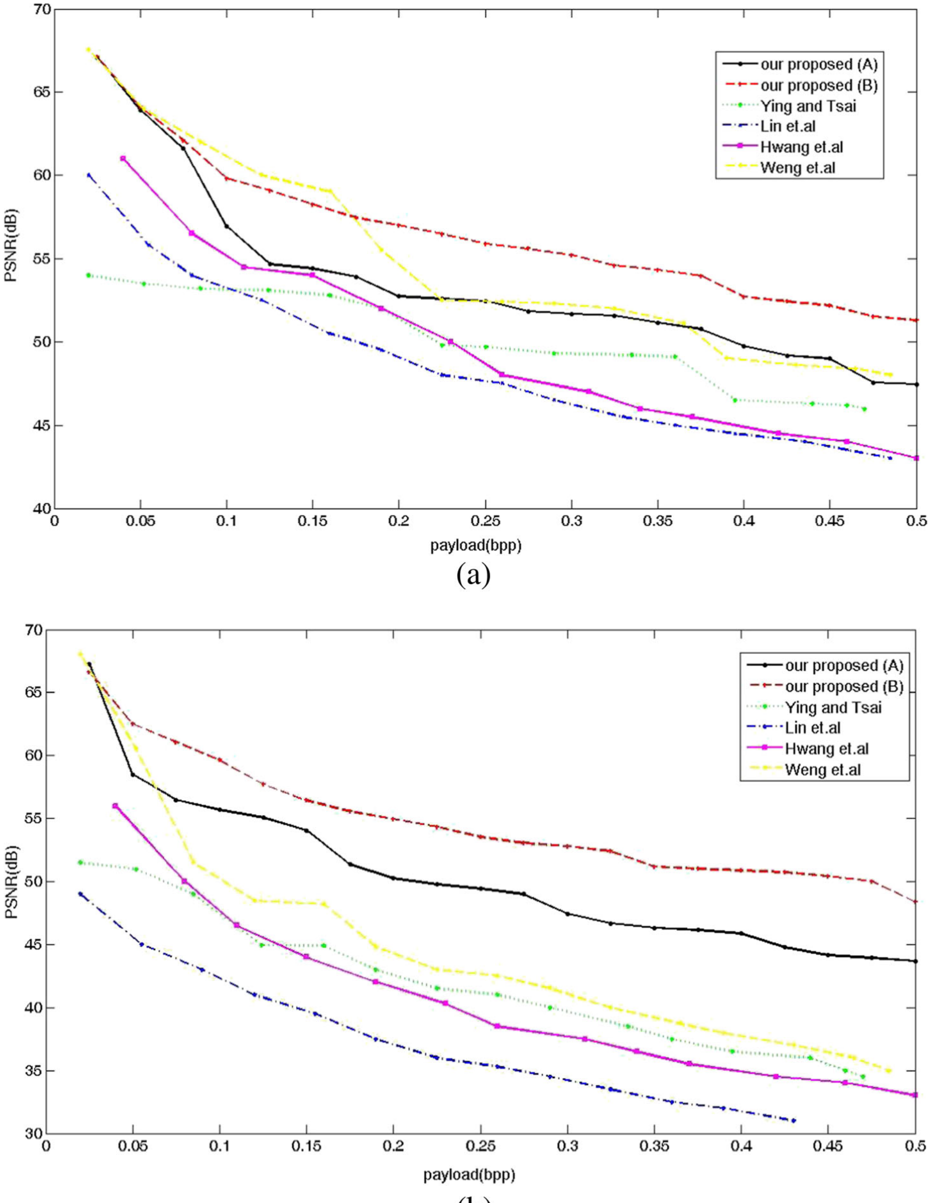
where *MSE* is the mean square error between the cover image and the stego-image.

Tables 1 and 2 show the experimental results using tactics *A* and *B*, respectively. Because every level tries to choose a better prediction method, our approach can embed more secret data on every level. Besides, because the prediction methods have different characteristics, it can avoid using the same prediction method on every level such that some regions of pixel have been changed too much.

Table 3 shows a comparison between experimental results using methods proposed by Lin et al. [18], Hsiao et al. [17], Lin and Hsueh [30], Yang and Tsai [31], and our approach. From Table 3, one can see that that our approach clearly has more capacity when the PSNR values are similar. We have embedded at least 44 % (≅ 99.81 % − 55.72) more capacity than other methods due to the fact that our approach chooses the more efficient prediction method at every level.

Table 4 shows a comparison between methods proposed by Hu et al. [10], Luo et al. [11], Li et al. [16], Hong [21], Li et al. [15], and our proposed scheme. Our scheme has demonstrated better image quality than that of other methods when the size of the embedded payload is 20,000 bits, because our approach prohibits some prediction errors from entering the process of pixel shifting. In addition, we use four different prediction methods. The threshold, *TH*, strategy in our proposed scheme has efficiently eliminated the distortion caused by pixel shifting. This enables superior PSNR performance compared to that of previous studies. Figure 5 shows a comparison of PSNR values between our approach and that of Yang-Tsai [31], based on various embedded capacities. The average PSNR value using Yang-Tsai's method is 52.52 dB, while the PSNR value using our tactic *A* approach is approximately 58.88 dB. Furthermore, using tactic *B* approach generated approximately 61.26 dB, which significantly improved upon Yang-Tsai's method by 6.36 and 8.74 dB, respectively.

Importantly, some overhead is needed in our approach. To record a peak point, an origin, *TH* value, and a prediction, *M*, require 9, 9, 8, and 2 bits, respectively, for a total of 46 bits



(a)

(b)

Fig. 6 Comparison results among our proposed scheme and other reversible schemes for images: (**a**) Lena; (**b**) Baboon

needed for a given stage. These overhead bits can be embedded into the image when the next stage is processed. Accordingly, if the data hiding program is executed for 20 levels, the required overhead will only be 3,680 bits, which is very small compared to the embedding capacity.

In addition, as shown in Table 5 we also compared the result with the previous work of Lee et al.'s scheme [13]. From this table, it is obvious that the averaged stego-image qualities of

Fig. 7 **a** Cover image Lena. **b** Stego-image Lena with capacity = 300,288 bits and PSNR = 39.05 dB

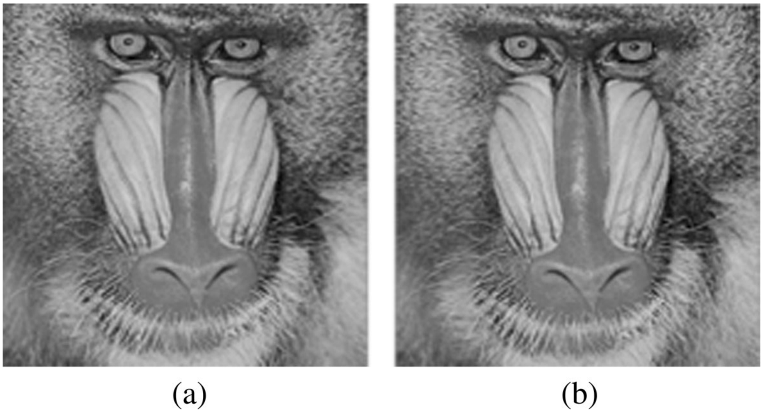


Fig. 8 **a** Cover image Baboon. **b** Stego-image Baboon with capacity = 317,015 bits and PSNR = 36.30 dB

Table 6 Experimental results of Lena using tactic *A* and different *C* values

Lena		$C = 5$	$C = 10$	$C = 15$	$C = 20$	$C = 30$
Level 1	Capacity	86,633	88,673	89,745	90,322	99,936
	PSNR	55.18	54.35	51.95	50.93	49.06
	M	M_1	M_1	M_2	M_4	M_1
Level 4	Capacity	292,764	297,028	279,401	246,138	255,913
	PSNR	46.749	45.31	44.25	41.70	40.92
	M	M_1	M_2	M_1	M_2	M_2
Level 7	Capacity	472,866	457,745	431,411	364,555	363,636
	PSNR	43.47	40.71	40.96	39.72	36.18
	M	M_1	M_2	M_3	M_1	M_2
Level 10	Capacity	607,127	573,514	570,844	492,707	432,191
	PSNR	41.641	40.10	39.59	37.95	34.48
	M	M_1	M_2	M_1	M_1	M_4
Level 20	Capacity	954,564	931,998	895,478	768,600	629,919
	PSNR	38.33	36.47	33.76	32.15	29.49
	M	M_1	M_2	M_2	M_2	M_2

Table 7 Experimental results of Baboon using tactic A and different C values

Baboon		$C=5$	$C=10$	$C=15$	$C=20$	$C=30$
Level 1	Capacity	13,713	24,115	27,504	27,940	31,719
	PSNR	63.59	59.828	58.07	57.227	55.10
	M	M_4	M_4	M_4	M_4	M_4
Level 4	Capacity	52,966	113,578	124,508	123,461	130,702
	PSNR	55.18	49.28	47.328	45.93	43.69
	M	M_4	M_2	M_2	M_2	M_2
Level 7	Capacity	110,206	186,375	202,841	203,005	207,013
	PSNR	50.12	45.361	43.09	41.46	39.47
	M	M_2	M_2	M_2	M_2	M_2
Level 10	Capacity	162,637	245,686	259,990	256,177	270,876
	PSNR	47.27	42.89	40.736	39.38	37.38
	M	M_2	M_4	M_4	M_4	M_2
Level 20	Capacity	343,170	447,590	464,752	463,457	470,440
	PSNR	40.02	36.14	34.89	33.59	32.46
	M	M_1	M_2	M_2	M_2	M_4

Lean and Baboon in our scheme are larger than that of Lee et al.'s scheme. As shown in Fig. 6, our scheme has better image quality than previous works of Yang and Tsai [31], Lin et al. [18], Hwang et al. [12], and Weng et al. [29] with the same embedding capacities.

As shown in Figs. 7 and 8, the original images and the stego-images are shown for comparison when the capacities are approaching to 300,000 bits using tactic A. It is visually difficult to distinguish between the original image and the stego-image for both Lena and Baboon by human eyes.

As shown in Tables 6 and 7, we used different C values to embed the secret bits into the image. Experimental results show that smaller C values would have larger capacities for the same PSNR value. Note that a smaller C value has a stricter sieving condition. When C is smaller, it has less capacity during low levels. However, it keeps the image with high quality. Therefore, it has better capacity and PSNR at high level.

Table 8 shows the temporal occurrences of overflow and underflow when 20 levels are executed. Compared to the embedding capacity of our approach, this is relatively little data, which can be easily embedded into the image.

4 Conclusions

We propose a reversible data hiding method based on four candidates for prediction methods and local complexity in order to enhance stego-image quality. We evaluate the four prediction methods

Table 8 The occurring times of overflow and underflow using tactic A and $C=30$ to embed 20 levels

Images	Airplane	Baboon	Boat	Lena	Peppers
Overflow	0	21	8	0	1644

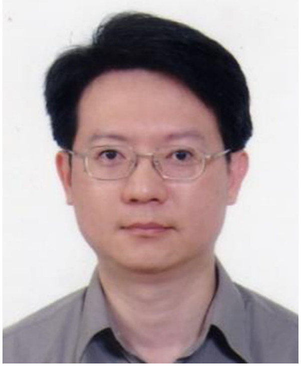
by calculating their capacity or efficiency ratios to decide which prediction method will be used. Additionally, we use a variance strategy to determine a threshold, *TH*, for selecting which pixel should join the process of pixel shifting and data concealing. This variance strategy has efficiently improved upon histogram-based approaches to obtain a high-quality image. From the experimental results, our method has proven to enable a larger image capacity and better quality than those of previous works.

Acknowledgement This research was partially supported by the Ministry of Science and Technology of the Republic of China under the Grants NSC 101-2221-E-153-002-MY2, MOST 103-2221-E-153-005, and MOST 105-2221-E-153-010.

References

1. Al-Qershi QM, Khoo BE (2011) High capacity data hiding schemes for medical images based on difference expansion. J Syst Softw 31(4):787–794
2. Alttar AM (2004) Reversible watermark using the difference expansion of a generalized integer transform. IEEE Trans Image Process 13(8):1147–1156
3. Chang CC, Pai PY, Yeh CM, Chan YK (2010) A high payload frequency-based reversible data hiding method. Inform Sci 180(11):2286–2298
4. Denemark T, Sedighi V, Holub V, Cogranne R, Fridrich J (2014) Select-channel-aware rich model for steganalysis of digital images. Proc IEEE Int Work Inf Forensics Secur 48–53
5. Feng JB, Lin IC, Tsai CS, Chu YP (2006) Reversible watermarking: current status and key issues. Int J Netw Secur 2(3):161–170
6. Filler T, Fridrich J (2011) Minimizing additive distortion in stegannography using syndrome-trellis codes. IEEE Trans Inf Forensics Secur 6(3):920–935
7. Fridrich J, Kodovsky J (2012) Rich models for steganalysis of digital images. IEEE Trans Inf Forensics Secur 7(3):868–882
8. Fu DS, Jing ZJ, Zhao SG, Fan J (2014) Reversible data hiding based on prediction-error histogram shifting and EMD mechanism. AEU Int J Electron Commun 68(10):933–974
9. Hong W (2012) Adaptive reversible data hiding method based on error energy control and histogram shifting. Opt Commun 285(2):101–108
10. Hsiao JY, Chan KF, Chang JM (2009) Block-based reversible data embedding. Signal Process 89(4):556–569
11. Hu Y, Lee HK, Li J (2009) DE-based reversible data hiding with improved overflow location map. IEEE Trans Circuits Syst Video Technol 19(2):250–260
12. Hwang HJ, Kim HJ, Sachnev V, Joo SH (2010) Reversible watermarking method using optimal histogram pair shifting based on prediction and sorting. KSII Trans Internet Inf Syst 4(4):555–670
13. Lee CF, Chen HL, Tso HK (2010) Embedding capacity raising in reversible data hiding based on prediction of difference expansion. J Syst Softw 83(10):1864–1872
14. Lee CC, Wu HC, Tsai CS, Chu YP (2008) Adaptive lossless steganography with centralized difference expansion. Pattern Recogn 141(6):2097–2106
15. Li X, Li B, Yang B, Zeng T (2013) General framework to histogram-shifting-based reversible data hiding. IEEE Trans Image Process 22(6)
16. Li X, Yang B, Zeng T (2011) Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection. IEEE Trans Image Process 20(12):3524–3533
17. Lin CC, Hsueh NL (2008) A lossless data hiding scheme based on three-pixel block differences. Pattern Recogn 41(4):1415–1425
18. Lin CC, Tai WL, Chang CC (2008) Multilevel reversible data hiding based on histogram modification of difference images. Pattern Recogn 41(12):3582–3591
19. Liu W, Liu G, Dai Y (2014) Efficient alternative form of syndrom-trellis codes for large relative payloads. J Comput Inf Syst 10(21):9037–9044
20. Lou DC, Hu MC, Li Liu C (2010) Multiple-layer data hiding scheme for medical image. Comput Stand Interfaces 31(2):329–335

21. Luo L, Chen Z, Chen M, Zeng X, Xiong Z (2010) Reversible image watermarking using interpolation technique. IEEE Trans Inf Forensic Secur 5(1):187–193
22. Ni Z, Shi YQ, Ansar N, Su W (2006) Reversible data hiding. IEEE Trans Circuits Syst Video Technol 16(3):354–362
23. Pan Z, Hu S, Ma X, Wang L (2015) Reversible data hiding based on local histogram shifting with multilayer embedding. J f Vis Commun Image Represent 31(8):64–74
24. Shi YQ, Ni Z, Zou D, Liang C, Xuan G 2004() Lossless data hiding: fundamentals, algorithms, and applications. Proc IEEE ISCAS 33–36
25. Tai WL, Yeh CM, Chang CC (2009) Reversible data hiding based on histogram modification of pixel differences. IEEE Trans Circuits Syst Video Technol 19(6):906–910
26. Tian J (2003) Reversible data embedding using a difference expansion. IEEE Trans Circuits Syst Video Technol 16(3):890–896
27. Tsai P, Hu YC, Yeh HL (2009) Reversible image hiding scheme using predictive coding. Signal Process 89(6):1129–1143
28. Tu TY, Wang CH (2015) Reversible data hiding with high payload based on referred frequency for VQ compressed codes index. Signal Process 108:278–287
29. Wang X, Ding J, Pei Q (2015) A novel reversible data hiding based on pixel value ordering and dynamic pixel block. Inform Sci 310(20):16–35
30. Weng CY, Yang CH, Fan CI, Liu KL, Sun HM (2013) Histogram-based reversible information hiding improved by prediction with the variance to enhance image quality. In The 8th Asia Joint Conference on Information Security, Seoul, Korea, July 25–26
31. Yang CH, Tsai MH (2010) Improving histogram-based reversible data hiding by interleaving prediction. IET Image Process 4(4):223–234
32. Yousefl S, Rablee H, Yousefl E, Ghanbarl M ()2007 Reversible data hiding using histogram sorting and integer transform. Proc IEEE DEST 487–490
33. Zhao A, Luo H, Lu ZM, Pan JS (2011) Reversible data hiding based on multilevel histogram modification and sequential recovery. AEU Int J Electron Commun 65(10):814–826
34. Zhao Z, Luo H, Lu ZM, Pan JS (2011) Reversible data hiding base on multilevel histogram modification and sequential recovery. AEU Int J Electron Commun 65:814–826



Cheng-Hsing Yang received his B.S. and M.S. degrees in Applied Mathematics from National Chung Hsing University, Taichung, Taiwan, in 1990 and 1992, respectively, and the Ph.D. degree in Electrical Engineering from National Taiwan University, Taipei, Taiwan, in 1997. From 1998 to 2003, he was an assistant professor of the Department of Information Management at Kun Shan University, Tainan, Taiwan. From August 2003 to July 2005, he was an assistant professor, from August 2005 to July 2009, he was an associate professor, and now a full professor of the Department of Computer Science at National Pingtung University, Pingtung, Taiwan. His research interests include information hiding and image watermarking.

Chi-Yao Weng received his M.S. degrees in Computer Science from National Pingtung University of Education, Pingtung, Taiwan, in 2007, and the Ph.D. degree in Computer Science from National Tsing Hua, Hsinchu, Taiwan, in 2011. During 2011 to 2015, he was a postdoctoral fellow of the Computer Science and Engineering at National Sun Yat-sen University and the Computer Science at National Tsing Hua University, respectively. He is currently an assistant professor of the Department of Computer Science at National Pingtung University, Pingtung, Taiwan. His research interests include multimedia application, information hiding and image watermarking.

Yih-Kai Lin received the Ph.D. degree in the Electrical Engineering from National Taiwan University, Taiwan, in 2003. He is presently an associate professor of Computer Science at National Pingtung University. His current research interests include image processing, pattern recognition, and data compression.

Kuan-Liang Liu received his B.S. degree in the Department of Information Management, Shih Hsin University, Taipei, Taiwan, in 2012. In 2014, he received his M.S. degrees in Computer Science from National Pingtung University of Education, Pingtung, Taiwan. His research interests include information hiding and digital watermarking.