Multimed Tools Appl (2013) 67:571–591
DOI 10.1007/s11042-012-1047-7

Reversible data hiding using side-match predictions on steganographic images

Fu-Hau Hsu · Min-Hao Wu · Shiuh-Jeng Wang

Published online: 16 March 2012

Abstract Information hiding is an important method to achieve multi-media security. Recently, many researchers have paid attention to the reversible data hiding scheme, which can completely recover original multi-media files after the embedded data are extracted. In this paper, a side-match approach is proposed to achieve more capacity in histogram-based reversible data hiding for grayscale images. The histogram is created by exploiting the difference in all the values between pixels and their predictedive values. Experimental results show that our method is capable of providing a great embedding capacity without causing noticeable distortion. In one-level hiding, where it has the best capacity, our method conserves image qualities larger than 48 dB. Furthermore, in multilevel hiding, a rotation strategy is proposed to further improve image qualities. Experimental results show that our method performs better than other existing methods in multilevel hiding cases.

Keywords Information hiding · Reversible data hiding · Side match · Prediction · Histogram

1 Introduction

With the rapid development of network technologies and the coming of the digital era, the Internet has become indispensable for many people. Through the development of the Internet, many new businesses have been developed, such as e-commerce, e-learning, online games, video-on-demand, etc. Furthermore, many existing enterprises have expanded their traditional business activities to the Internet. Every day, thousands of multimedia data are transferred conveniently and efficiently over the Internet. Because digital multimedia data, such as audio, videos, images, texts, etc., have the attributes of easy copying, modification and distribution, the development of the Internet has increased the problems of multimedia security. A way to protect the authentication and ownership of multimedia data has become

F.-H. Hsu · M.-H. Wu
Department of Computer Science and Information Engineering, National Central University, Jhongli 320, Taiwan

S.-J. Wang (✉)
Department of Information Management, Central Police University, Taoyuan 333, Taiwan
e-mail: sjwang@mail.cpu.edu.tw

an important topic, so many researchers have paid a great deal of attention to this topic. One of the most important approaches to this topic is the technique of information hiding.

Many approaches to information hiding have been proposed with different attributes, such as capacity, imperceptibility, undetectability, robustness and reversibility. These attributes are used for various applications, such as secret communication, copyright protection, tampering detection and other human-centered approaches. Information hiding techniques can be categorized into two types: methods in the spatial domain and methods in the frequency domain. In the spatial domain, the secret messages are embedded by changing image pixels directly. On the other hand, in the frequency domain, the image is first transformed into its frequency domain, and then the secret messages are embedded in the transformed coefficients.

Recently, reversible data hiding has drawn many researchers' attention. Reversibility allows original media to be completely recovered from stego-media after the embedded message is extracted. Many reversible data hiding approaches have been proposed [1–3, 5, 7–18, 21–24]. According to how the data are embedded, these approaches can be classified into three categories: the spatial domain [1, 3, 5, 7–9, 11–18, 24], the frequency domain [10, 22] and other compression types, such as vector quantization (VQ) [2, 21, 23]. In those developed reversible data hiding methods, two main technologies have been widely applied: difference-expansion-based technology [1, 5, 7, 9, 15, 16, 18] and histogram-based technology [8, 11–14, 17, 24]. In 2006, Ni et al. presented a reversible data hiding method based on the histogram [14]. Their method guarantees that the change of each pixel in the stego-image remains within ±1. Therefore, the PSNR value of the stego-image is at least 48 dB. However, their method used the pixel values in the original image to create the histogram, So the peak values of the histogram were not high enough. Some methods used predictive concepts to increase the peak height [3, 4, 11–13, 17, 24]. In 2008, Lin et al. proposed a lossless data hiding scheme based on three-pixel block differences [12]. Their lossless method embeds a message into a cover image using the two differences—between the first and second pixels, as well as between the second and third pixels—in a three-pixel block. In the best case, a three-pixel block can embed two bits "11" and only the central pixel needs to be increased or decreased by 1. In 2008, Lin et al. proposed a multilevel reversible data hiding scheme based on the histogram of difference images [24]. They applied their lossless data hiding method to a cover image many times to enlarge the capacity. In 2008, Hong et al. [4] proposed a reversible data hiding scheme based on histogram-shifting of prediction errors (HSPE). Their method provided advantages of using the median edge detector to design a scheme based on histogram shifting. In 2009, Li et al. proposed a reversible data hiding scheme based on the similarity between neighboring pixels [11]. The difference between two adjacent pixels has a high probability of being a small value. They employed the histogram of the pixel difference sequence to increase the embedding capacity. In 2010, Fallahpour et al. [3] proposed an reversible data hiding approach using the image prediction errors, where the most well-known prediction algorithms, such as the median edge detector (MED) [19], gradient adjacent prediction (GAP) [20], and Jiang et al.'s prediction [6], are tested for the adaptive shifted prediction error (ASPE). From the above developed histogram-based data hiding approaches, we know that Ni et al. provided the first histogram-based approach but not applied the predictive methods to enhance the height of the histogram. Other papers applied the predictive methods to enhance the height. However, sometimes the predictive methods are too simple to create an efficient height. Moreover, no paper providing any strategy to different levels when multi-level data hiding approaches are applied.

In this paper, we propose a new reversible information hiding method based on the histogram for grayscale images. We use the side-match prediction to advance the capacity embedding in histogram-based reversible data hiding methods. All predictive error values

are transformed into histograms to create higher peak values. When multilevel hiding is performed, a rotation strategy is proposed to make image quality evaluations higher in empirical experiments. The remainder of this paper is organized in the following way: in Section 2, we introduce some related works of reversible information hiding technology, our proposed scheme is described in Section 3, and some experimental results are shown in Section 4. Finally, we offer some conclusions in Section 5.

2 Related works

In this section, we review Ni et al.'s and Li et al.'s histogram-based reversible data hiding methods [11, 14].

2.1 Ni et al.'s method

Ni et al. proposed a reversible data hiding method in 2006 [14]. Their method uses the histogram of an original image to embed secret messages. In the histogram, they find multiple pairs of peak and zero points, where a peak point corresponds to the pixel value which a maximal number of pixels in the cover image assumes, and a zero point corresponds to the pixel value which no pixel in the cover image assumes. To use a pair of peak and zero points to embed the secret messages, their algorithm is as follows:

1. Generate the histogram $H(x)$ with $x \in [0, 255]$ of the original image.
2. In the histogram $H(x)$, find a peak point p and a zero point z, where $p,z \in [0, 255]$.
3. Shift the values between the peak point and the zero point as follows:
   (a) If $p>z$, move the whole part of the histogram $H(x)$ with $x \in [z+1, p-1]$ to the left by 1 unit.
   (b) If $p<z$, move the whole part of the histogram $H(x)$ with $x \in [p+1, z-1]$ to the right by 1 unit.
4. Scan the image and embed one secret bit when a pixel with value p is met:
   (a) If the to-be-embedded bit is 0, the pixel value remains p.
   (b) If the to-be-embedded bit is 1, the pixel value is set to $p+1$ and $p-1$ when p is smaller than z and p is greater than z, respectively.
5. Output the stego-image, peak point p, and zero point z.

Figure 1 shows an example of Ni et al.'s method. The table in the left is an image with 5 × 5 pixels. The diagram in the right is the histogram, where a peak point $p=163$ and a zero point $z=166$ are found. Then, pixel values belonging to [164, 165] are moved to the right by 1 unit. The results are shown in Fig. 2.

Suppose the to-be-embedded data are 11001101. Pixels shown in Fig. 2 are scanned from left to right and from top to bottom. All pixels with a value equal to $p=163$ can be used to embed one secret bit. The embedded results are shown in Fig. 3.

2.2 Li et al.'s method

In 2009, Li et al. provided a histogram-based reversible data hiding method using the adjacent pixel difference (APD) approach, which transforms the cover image into an intermediary difference sequence to increase the frequency of the peak points [11]. A natural image usually

162	165	163	161	161
164	162	165	161	161
162	163	164	164	161
163	163	161	162	165
163	163	163	163	162

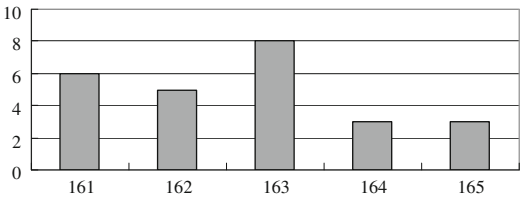


Fig. 1 Ni et al.'s example

contains several smooth areas. The difference between two adjacent pixels has a high probability of being a small value, often close to zero. Various calculations can be applied to an image to generate various difference sequences such that the peak points of these sequences have higher frequencies than the peak points of the histogram of the untransformed image. The APD employs the histogram of the pixel difference sequence to increase the embedding capacity.

Let P be the original gray image with n pixels and P_i be the pixel value with index value i in the scanned sequence. The embedding algorithm is as follows:

Input: Original image, secret message.
Output: Stego-image, two pairs of peak and zero points.

1. Generate the difference image P' from the original image P. The formula is as follows:

$$P_i' = \begin{cases} P_i & ,if\ i = 0 \\ P_{i-1} - P_i & ,if\ 1 \le i < n \end{cases}.$$

2. Generate the histogram of the difference image and find the pairs of peak and zero points (PP_1, CZP_1) and (PP_2, CZP_2) that satisfy $CZP_1 < PP_1 < PP_2 < CZP_2$. If no zero point exists, a minimum frequency point is selected, and the extra information of these coefficient values is stored. Then, the minimum frequency point is cleared to become the zero point.
3. Shift the histogram as follows:
   (a) P_i'' is set to $P_i' - 1$ if $P_i' \in [CZP_1 + 1, PP_1 - 1]$.
   (b) P_i'' is set to $P_i' + 1$ if $P_i' \in [PP_2 + 1, CZP_2 - 1]$.
4. Embed a secret bit when P_i' is equal to PP_1 or PP_2 as follows:
   (a) If the to-be-embedded bit is 0, P_i'' is set to P_i'.
   (b) If the to-be-embedded bit is 1, P_i'' is set to $P_i' - 1$ and $P_i' + 1$ when P_i' is equal to PP_1 and PP_2, respectively.

Fig. 2 Results after shifting

162	166	163	161	161
165	162	166	161	161
162	163	165	165	161
163	163	161	162	166
163	163	163	163	162

162	166	164	161	161
165	162	166	161	161
162	164	165	165	161
163	163	161	162	166
164	164	163	164	162

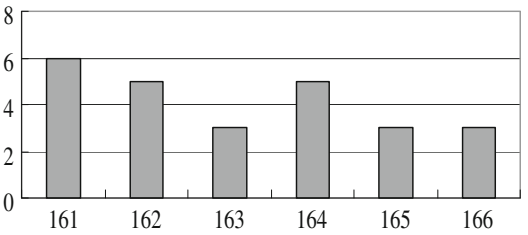


Fig. 3 The embedded results

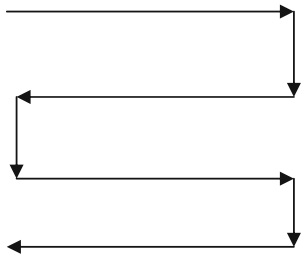
5. Convert each embedded predictive error value P_i'' into its embedded pixel value P_i''' as follows:

$$P_i''' = \begin{cases} P_i'' & ,if\ i = 0 \\ P_{i-1} - P_i'' & ,otherwise \end{cases}.$$

6. Generate the stego-image from P'''.

As shown in Fig. 4, a 6×4 grayscale image is provided to explain the embedding algorithm. Assume the secret data are "100010010". In Step 1, the original image is scanned

110	110	110	113	110	110
114	113	111	113	111	113
115	113	113	111	113	113
108	112	113	110	113	112



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
P	110	110	110	113	110	110	113	111	113	111	113	114	115	113	113	111	113	113	112	113	110	113	112	108
P′	110	0	0	-3	3	0	-3	2	-2	2	-2	-1	-1	2	0	2	-2	0	1	-1	3	-3	1	4
P″	110	-1	0	-4	4	0	-4	2	-3	3	-3	-2	-2	2	0	3	-3	0	1	-2	4	-4	1	5
P	110	110	110	113	110	110	113	111	113	111	113	114	115	113	113	111	113	113	112	113	110	113	112	108
P‴	110	111	110	114	109	110	114	111	114	110	114	115	116	113	113	110	114	113	112	114	109	114	112	107

Fig. 4 An example of the data embedding process of the APD method

in inverse s-order. Then, a one-dimensional difference sequence P' is generated. In Step 2, two pairs of peak points and zero points, (0, –4) and (2, 5), are selected. In Step 3, values of P' are shifted. In Step 4, secret data are embedded, and the P'' sequence is generated. The secret data are embedded in the grays boxes of the sequence. In Step 5, the other values of the sequence are obtained through the first value $P_0''' = P_0'' = 110$. For example, $P_1''' = P_0 - P_1'' = 110 - (-1) = 111$. Then, the stego-image is constructed from the P''' sequence. Each pixel difference between the original image and stego-image is at most one.

3 Our proposed method

In this section, we propose a new histogram-based reversible data hiding scheme. We use the side-match prediction approach to achieve histogram-based reversible data hiding. This method can increase the embedding capacity and conserves the quality of the image. Figure 5 shows the main concept of the side-match prediction, where $i>0$ and $j>0$. Our predictive method is to employ the neighboring pixels $H_{i,\,j\text{-}1}$, $H_{i\text{-}1,\,j\text{-}1}$, $H_{i\text{-}1,\,j}$, and $H_{i\text{-}1,\,j+1}$ to predict the pixel $H_{i,\,j}$. Figure 6 shows the flowchart of our method. In the embedding process, predictive error values created by the side-match prediction are used to generate a histogram to embed secret data. In the extracting and reversing process, the side-match prediction is applied to the stego-image, and the created histogram is processed for extracting and reversing. The detail of the embedding algorithm and the extracting and reversible algorithm are shown in the following subsections.

3.1 Embedding algorithm

Suppose that the original image is an $n \times m$ gray image. $H_{i,\,j}$ and $H'_{i,j}$ are the pixel values at location (i, j) before and after being processed, respectively. $D_{i,\,j}$ is the predictive error value at location (i, j), and $D'_{i,j}$ is the embedded predictive error value at location (i, j). The embedding algorithm is as follows:

Input: Original image, secret message.
Output: Stego-image, two pairs of peak and zero points.

1. Input the original image $H=\{H_{0,0}, H_{0,1}, \ldots, H_{0,n\text{-}1}, \ldots, H_{1,0}, \ldots, H_{n\text{-}1,n\text{-}1}\}$.
2. From left to right and from top to bottom, predict each pixel $H_{i,\,j}$ in the original image to create its predictive error value $D_{i,\,j}$ as follows:

   (a) If $i=j=0$, then

$$D_{i,j} = H_{i,j} - 128. \tag{1}$$

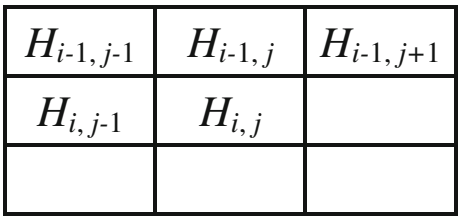


Fig. 5 The main concept of the side-match prediction

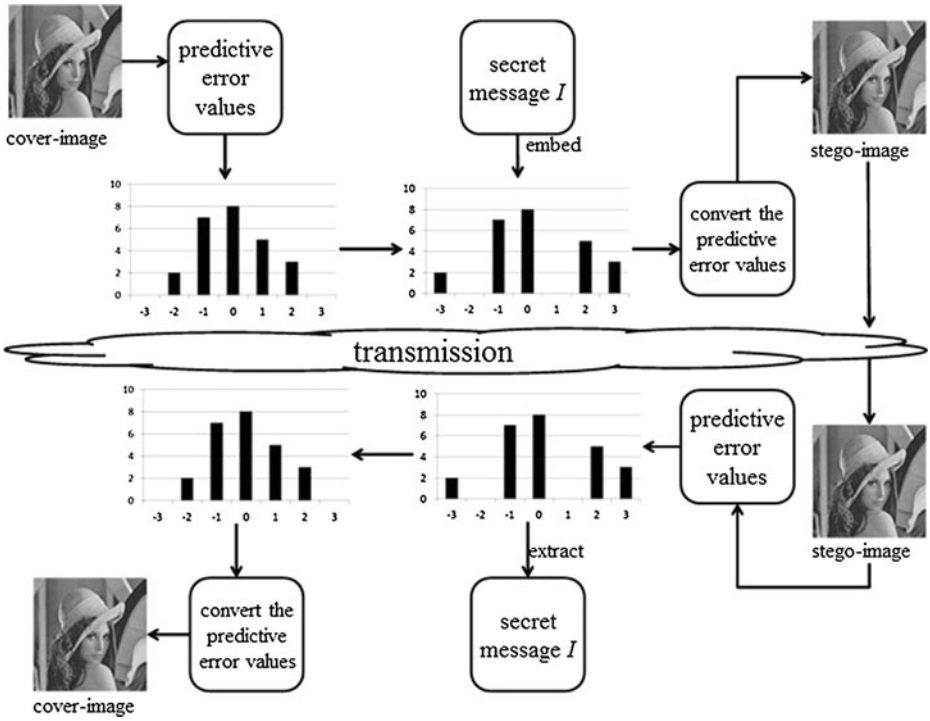


Fig. 6 The flowchart of our proposed scheme

(b) If $i=0$ and $j\neq 0$, then

$$D_{i,j} = H_{i,j} - H_{i,j-1}. \tag{2}$$

(c) If $i\neq 0$ and $j=0$, then

$$D_{i,j} = H_{i,j} - \left\lfloor \frac{H_{i-1,j} + H_{i-1,j+1}}{2} \right\rfloor. \tag{3}$$

(d) If $i\neq 0$ and j=$\underline{m-1}$, then

$$D_{i,j} = H_{i,j} - \left\lfloor \frac{H_{i,j-1} + H_{i-1,j-1} + H_{i-1,j}}{3} \right\rfloor. \tag{4}$$

(e) Else,

$$D_{i,j} = H_{i,j} - \left\lfloor \frac{H_{i,j-1} + H_{i-1,j-1} + H_{i-1,j} + H_{i-1,j+1}}{4} \right\rfloor. \tag{5}$$

3. Create the histogram $H(x)$ with $x \in [-255, 255]$ from all predictive error values.
4. Find two pairs of peak and zero point (P_1, Z_1) and (P_2, Z_2) satisfying $Z_2<P_2<P_1<Z_1$.
5. Shift the histogram as follows:

(a) $D'_{i,j}$ is set to $D_{i,j}+1$ if $D_{i,j} \in [P_1+1, Z_1 - 1]$.

(b) $D'_{i,j}$ is set to $D_{i,j} - 1$ if $D_{i,j} \in [Z_2+1, P_2 - 1]$.

6. Embed the secret message as follows:
   (a) If the to-be-embedded bit is 0, $D'_{i,j}$ is set to $D_{i,j}$.
   (b) If the to-be-embedded bit is 1, $D'_{i,j}$ is set to $D_{i,j}+1$ and $D_{i,j}-1$ when $D_{i,j}$ is equal to P_1 and P_2, respectively.
7. From left to right and from top to bottom, convert each embedded predictive error value $D'_{i,j}$ into its embedded pixel value $H'_{i,j}$.
   (a) If $i=j=0$, then

$$H'_{i,j} = D'_{i,j} + 128. \tag{6}$$

   (b) If $i=0$ and $j\neq 0$, then

$$H'_{i,j} = D'_{i,j} + H_{i,j-1}. \tag{7}$$

   (c) If $i\neq 0$ and $j=0$, then

$$H'_{i,j} = D'_{i,j} + \left\lfloor \frac{H_{i-1,j} + H_{i-1,j+1}}{2} \right\rfloor. \tag{8}$$

   (d) If $i\neq 0$ and $j=\underline{m}-1$, then

$$H'_{i,j} = D'_{i,j} + \left\lfloor \frac{H_{i,j-1} + H_{i-1,j-1} + H_{i-1,j}}{3} \right\rfloor. \tag{9}$$

   (e) Else,

$$H'_{i,j} = D'_{i,j} + \left\lfloor \frac{H_{i,j-1} + H_{i-1,j-1} + H_{i-1,j} + H_{i-1,j+1}}{4} \right\rfloor. \tag{10}$$

8. Output stego-image, (P_1, Z_1), and (P_2, Z_2).

As shown in Fig. 7, a 5×5 grayscale original image $H=\{H_{0,0}, H_{0,1}, \ldots, H_{0,4}, H_{1,4}, \ldots, H_{4,4}\}$ is given to explain our embedding algorithm. We predict the pixels first. For example,

$$D_{0,0} = H_{0,0} - 128 = 126 - 128 = -2,$$
$$D_{0,1} = H_{0,1} - H_{0,0} = 125 - 126 = -1,$$

$$D_{1,0} = H_{1,0} - \left\lfloor \frac{H_{0,0} + H_{0,1}}{2} \right\rfloor = 125 - \left\lfloor \frac{126 + 125}{2} \right\rfloor = 0,$$

Fig. 7 The original image

126	125	124	125	125
125	126	125	125	124
127	126	125	126	124
125	124	125	126	125
126	125	124	123	124

–2	–1	–1	1	0
0	1	0	1	–1
2	1	0	2	–1
–1	–1	0	1	0
2	0	–1	–2	0

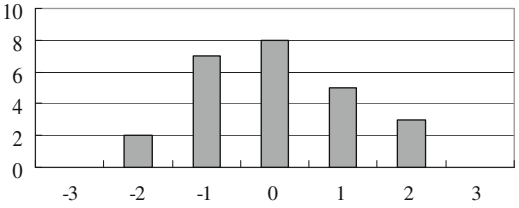


Fig. 8 The predictive error values and their histogram

$$D_{1,1} = H_{1,1} - \left\lfloor \frac{H_{1,0}+H_{0,0}+H_{0,1}+H_{0,2}}{4} \right\rfloor = 126 - \left\lfloor \frac{125+126+125+124}{4} \right\rfloor = 1$$

and so on. The results are shown in Fig. 8. The histogram created from all predictive error values is also shown in Fig. 8. According to the histogram, we have found two pairs of peak and zero points: $(P_1, Z_1)=(0, 3)$ and $(P_2, Z_2)=(-1, -3)$. The results and the histogram after shifting are shown in Fig. 9.

Then, we embed the secret messages. The predictive error values equivalent to P_1 or P_2 are used to embed the secret messages. Assume secret message $I=1010011010001110_{(2)}$. After secret message I is embedded, the predictive error values and their histogram are shown in Fig. 10. Finally, the predictive error values are converted into pixel values. For example,

$$H'_{0,0} = D'_{0,0} + 128 = -3 + 128 = 125,$$
$$H'_{0,1} = D'_{0,1} + H_{0,0} = -2 + 126 = 124,$$

$$H'_{1,0} = D'_{1,0} + \left\lfloor \frac{H_{0,0}+H_{0,1}}{2} \right\rfloor = 0 + \left\lfloor \frac{126+125}{2} \right\rfloor = 125,$$

$$H'_{1,1} = D'_{1,1} + \left\lfloor \frac{H_{1,0}+H_{0,0}+H_{0,1}+H_{0,2}}{4} \right\rfloor = 2 + \left\lfloor \frac{125+126+125+124}{4} \right\rfloor = 127,$$

–3	–1	–1	2	0
0	2	0	2	–1
3	2	0	3	–1
–1	–1	0	2	0
3	0	–1	–3	0

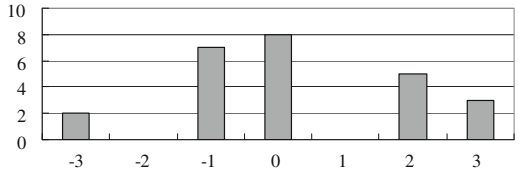


Fig. 9 The predictive error values and their histogram after shifting

–3	–2	–1	2	1
0	2	0	2	–2
3	2	1	3	–1
–2	–1	0	2	0
3	1	–2	–3	0

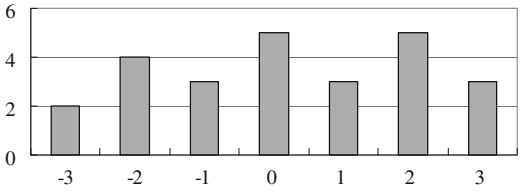


Fig. 10 The predictive error values and their histogram after embedding

3.2 Extracting and reversing the algorithm

In the extracting and reversing process, the secret message is extracted, and the embedded pixel values are reversed. The extracting and reversing algorithm is shown as follows:

Input: Stego-image H', two pairs of peak and zero points (P_1, Z_1) and (P_2, Z_2).
Output: Original image, secret message.

1. Process each pixel of stego-image H' from left to right first and then from top to bottom by following Step 2 to Step 4 repeatedly.
2. Create predictive error value $D'_{i,j}$ from H'.
3. Use (P_1, Z_1) and (P_2, Z_2) to extract the secret message and recover the predictive error values as follows:
   (a) If the $D'_{i,j}$ is equal to P_1 or P_2, the secret bit 0 is extracted, and the predictive error is recovered as $D_{i,j} = D'_{i,j}$.
   (b) If the $D'_{i,j}$ is equal to $P_1 + 1$ or $P_2 - 1$, the secret bit 1 is extracted and the predictive errors are recovered as $D_{i,j} = D'_{i,j} - 1$ or $D_{i,j} = D'_{i,j} + 1$, respectively.
   (c) If $D'_{i,j}[P_1 + 2, Z_1]$, recover the predictive error as $D_{i,j} = D'_{i,j} - 1$.
   (d) If $D'_{i,j} \varepsilon [Z_2, P_2 - 2]$, recover the predictive error as $D_{i,j} = D'_{i,j} + 1$.
4. Convert predictive error values $D_{i,j}$ into original pixel value $H_{i,j}$.
5. Output the original image and the secret message.

Now, we perform extracting and reversing operations on the embedded results shown in Fig. 11. The two inputted pairs of peak and zero points are (P_1, Z_1)=

Fig. 11 The stego-image

125	124	124	126	126
125	127	125	126	123
128	127	126	127	124
124	124	125	127	125
127	126	123	122	124

Fig. 12 A pixel $H_{i,\ j}$ and its 8 adjacent pixels

$H_{i\text{-}1,\ j\text{-}1}$	$H_{i\text{-}1,\ j}$	$H_{i\text{-}1,\ j+1}$
$H_{i,\ j\text{-}1}$	$H_{i,\ j}$	$H_{i,\ j+1}$
$H_{i+1,\ j\text{-}1}$	$H_{i+1,\ j}$	$H_{i+1,\ j+1}$

(0, 3) and $(P_2,\ Z_2)$=(−1, −3). We process the first pixel value $H'_{0,0}$ first. $D'_{0,0} = H'_{0,0} - 128 = 125 - 128 = -3$. We find that $D'_{0,0} = -3$ is in $[Z_2,\ P_2 - 2]$, so $D_{0,0} = D'_{0,0} + 1 = -2$. Then, we calculate the original pixel value $H_{0,0} = D_{0,0} + 128 = -2 + 128 = 126$. Next, the second pixel value $H'_{0,1}$ is processed. $D'_{0,1} = H'_{0,1} - H_{0,0} = 124 - 126 = -2$. We find that $D'_{0,1} = -2$ is equal to $P_2 - 1$, so secret bit 1 is extracted, and the predictive error value is recovered as $D_{0,1} = D'_{0,1} + 1 = -2 + 1 = -1$. Finally, we calculate the original pixel value $H_{0,1} = D_{0,1} + 126 = -1 + 126 = 125$. Similarly, other pixel values $H'_{i,j}$ are processed in order to generate their original pixel values $H_{i,\ j}$.

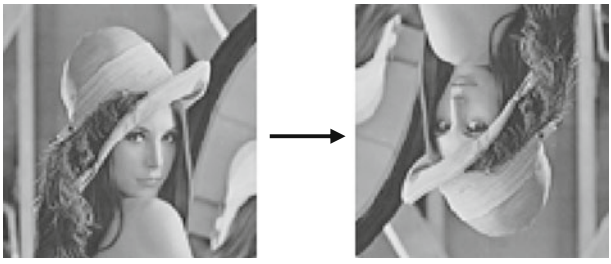
3.3 Processing directions of side-match approaches

Side-match approaches can be performed from various directions. As shown in Fig. 12, predicting pixel $H_{i,\ j}$ by side-match approaches can come from different directions and has various choices. For example, from the upper-left corner, $\{H_{i,\ j-1}, H_{i-1,\ j-1}, H_{i-1,\ j}, H_{i-1,\ j+1}\}$, $\{H_{i,\ j-1}, H_{i-1,\ j-1}, H_{i-1,\ j}\}$, $\{H_{i,\ j-1}, H_{i\text{-}1,\ j}\}$ and $\{H_{i,\ j-1}\}$ are some possible choices to predict pixel $H_{i,j}$. Similarly, from the upper-right corner, $\{H_{i,\ j+1}, H_{i\text{-}1,\ j+1}, H_{i\text{-}1,\ j}, H_{i\text{-}1,\ j\text{-}1}\}$, $\{H_{i,\ j+1}, H_{i\text{-}1,\ j+1}, H_{i\text{-}1,\ j}\}$, $\{H_{i,\ j+1}, H_{i\text{-}1,\ j}\}$ and $\{H_{i,\ j+1}\}$ are some possible choices to predict pixel $H_{i,\ j}$. The algorithm shown in Section 3.1 is the case $\{H_{i,\ j-1}, H_{i-1,\ j-1}, H_{i-1,\ j}, H_{i-1,\ j+1}\}$ which is from the upper-left corner. Other cases can be implemented similarly. The experimental results of this paper are created by using $\{H_{i,\ j+1}, H_{i\text{-}1,\ j}\}$ to predict $H_{i,\ j}$ from the upper-right corner.

3.4 Image-rotation strategies for multilevel data hiding

Our histogram-based reversible data hiding scheme guarantees that the change of each pixel in the stego-image remains within ±1. Therefore, the stego-image has good image quality.

Fig. 13 The clockwise 180° rotation of Lena



(a)

Pixels	10	20	30	40	50	60	70	80	90
Predictive error values	↓	10	10	10	10	10	10	10	10
Shifting and embedding		11	11	11	11	11	11	11	11
Stego-results	10	21	31	41	51	61	71	81	91

(b)

Pixels	10	21	31	41	51	61	71	81	91
Predictive error values	↓	11	10	10	10	10	10	10	10
Shifting and embedding		12	11	11	11	11	11	11	11
Stego-results	10	22	32	42	52	62	72	82	92

(c)

Pixels	10	21	31	41	51	61	71	81	91
Predictive error values	-11	-10	-10	-10	-10	-10	-10	-10	↓
Shifting and embedding	-12	-11	-11	-11	-11	-11	-11	-11	
Stego-results	9	20	30	40	50	60	70	80	91

Fig. 14 An example to explain the rotation operation: **a** Embedded from the left-to-right direction in the first step. **b** Embedded from the left-to-right direction in the second step. **c** Embedded from the right-to-left direction in the second step

The image quality of the stego-image is usually evaluated by the peak signal to noise ratio (PSNR) which is defined as:

$$\mathrm{PSNR} = 10 \times \log_{10} \frac{255^2}{\mathrm{MSE}} (\mathrm{dB}), \tag{11}$$

where MSE is the mean square error between the original image and the stego-image. For the 512×512 gray image, the MSE is defined as:

$$\mathrm{MSE} = \frac{1}{512 \times 512} \sum_{i=0}^{511} \sum_{j=0}^{511} \left(H_{i,j} - H'_{i,j} \right)^2, \tag{12}$$

where $H_{i,j}$ and $H'_{i,j}$ are the pixel values of the original image and the stego-image, respectively. If the difference value of each pixel between the original image and the stego-image remains ±1, the PSNR value is larger than $10 \times \log_{10} \frac{255 \times 255}{1} = 10 \times 4.8130 = 48.13$ dB. To increase the embedding capacity, the embedding procedure can be applied to the cover images many times. The final embedded results are still reversible. This is called the multilevel data hiding approach [12, 24]. The multilevel data hiding approach can increase the embedding capacity, but the image quality of the embedded results will decline.

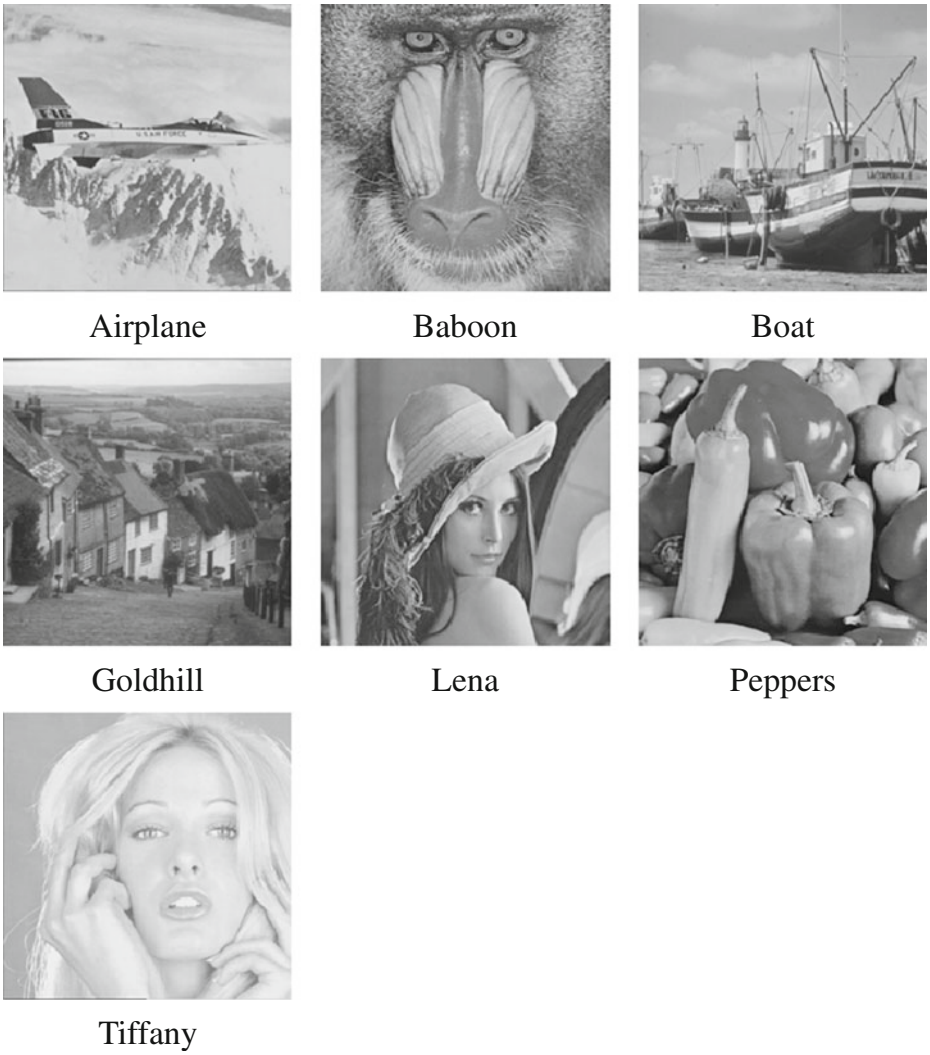


Fig. 15 Seven tested images

We use our reversible data hiding method to perform the multilevel data hiding scheme. Moreover, an image-rotation strategy is proposed to further advance image quality. As shown in Fig. 13, we perform a clockwise 180° rotation on the test image after the embedding procedure is finished at each level. A simple example is given in Fig. 14 to explain our idea. We know that a natural image usually contains image-gradient parts, that is, the gray values of neighboring pixels are increasing or decreasing. Figure 14 shows a 9-pixel artificial example where the gray values of the 9 pixels increase from left to right. With the predicted left-to-right direction, the first-level embedded results are shown in Fig. 14(a). Most pixels increase by 1 for the reason that the shifting operation will increase the predictive error values by 1 based on the assumption that 0 is a peak point. Based on that same assumption, Fig. 14(b) shows the second-level embedded results also using the

Table 1 The comparison among Ni et al.'s, Tsai et al.'s, Li et al.'s, and our proposed methods

Images	Ni et al.'s method [14]		Tsai et al.'s method [17]		Li et al.'s method [11]		Our scheme	
	Capacity	bpp	Capacity	bpp	Capacity	bpp	Capacity	bpp
Airplane	17,042	0.07	65,526	0.25	79,363	0.30	94,524	0.36
Boat	10,567	0.04	44,628	0.17	49,681	0.19	59,598	0.23
Goldhill	5,336	0.02	34,962	0.13	48,977	0.19	50,250	0.19
Lena	5,760	0.02	48,412	0.18	60,785	0.23	73,686	0.28
Peppers	5,737	0.02	48,108	0.18	62,206	0.24	68,836	0.26
Sailboat	8,186	0.03	31,704	0.12	35,430	0.14	39,679	0.15
Average	8,771	0.03	45,223	0.17	56,074	0.21	64,429	0.25
PSNR	48.3	0.03	48.79	0.17	48.47	0.21	48.58	0.25

predicted left-to-right direction. Again, most pixels increase by 1. Compared to the original pixels, most pixels have increased by 2. However, as shown in Fig. 14(c), if a predicted right-to-left direction is used in the second-level data hiding, most pixels will decrease by 1. Therefore, compared with the original pixels, most pixels remain unchanged after the second-level data hiding. From the above observations, we believe that our image-rotation strategy has significantly lifted the quality of multilevel data hiding.

3.5 Overflow and underflow solutions

After secret messages are embedded in an image, the change of each pixel remains within ±1. Therefore, if pixel values are equal to 0 or 255 in the original image, they may become –1 and 256 in the stego-image and cause underflow and overflow problems. In order to avoid this problem, we use a pre-processing method [5]. When the pixel values are equal to 0 or 255 in the original image, they are changed into 1 or 254 in advance, respectively. For each pixel value 1, a flag bit is needed to record that its original pixel value is 0 or 1. Similarly, one flag bit is needed for each pixel value 254. Therefore, for each pixel with values 1 or 254, if the pixel is changed from 0 or 255, the flag bit is set to 1; otherwise, the flag bit is set to 0.

Table 2 The comparison between Fallahpour et al.'s [3] and our proposed methods

Images	GAP		Jiang		MED		Our scheme	
	Capacity	PSNR	Capacity	PSNR	Capacity	PSNR	Capacity	PSNR
Lena	57,949	49.20	53,963	48.80	53,836	48.68	73,686	48.79
Barbara	47,363	49.00	43,477	48.50	43,995	48.55	41,885	48.50
Boat	56,850	49.20	53,275	48.60	53,313	48.63	59,598	48.66
Goldhill	41,466	48.90	39,408	48.50	40,318	48.50	50,250	48.57
Baboon	19,277	48.50	17,792	48.30	17,932	48.30	23,928	48.34
Peppers	45,669	49.00	39,189	48.50	39,608	48.51	68,853	48.74
Zelda	57,908	49.20	51,039	48.60	51,347	48.61	54,936	48.61
Average	46,640	49.00	42,592	48.54	42,907	48.54	53,305	48.60

Table 3 The experimental results of different layers without using the clockwise 180° rotation

Level	Indicator	Airplane	Baboon	Boat	Goldhill	Lena	Peppers	Tiffany
1	Capacity	94,524	23,928	59,598	50,250	73,686	68,853	74,456
	PSNR	49.00	48.34	48.66	48.57	48.62	48.75	48.80
2	Capacity	148,907	45,375	100,462	90,222	98,857	118,595	122,390
	PSNR	43.56	42.97	43.40	43.31	43.42	43.78	43.26
3	Capacity	197,268	62,767	133,354	122,267	131,874	158,514	160,882
	PSNR	40.16	39.77	40.43	40.31	40.42	40.60	40.05
6	Capacity	287,945	109,830	207,783	200,137	208,771	247,965	241,786
	PSNR	34.59	33.90	34.29	34.46	34.51	34.90	34.61
9	Capacity	349,613	145,262	258,015	255,721	262,967	307,009	296,931
	PSNR	31.08	30.46	31.33	30.95	31.18	31.23	31.06
12	Capacity	396,062	175,007	297,053	300,303	303,317	352,586	336,204
	PSNR	28.74	28.08	28.66	28.74	28.68	28.92	28.71

4 Experimental results

In this section, we display the performance of our scheme. In our experiments, we used random numbers as secret messages and 512×512 grayscale images as the cover images. Figure 15 shows the seven 512×512 grayscale test images Airplane, Baboon, Boats, Goldhill, Lena, Peppers and Tiffany. Table 1 shows a comparison among Ni et al.'s method [14], Tsai et al.'s method [17], Li et al.'s method [11], and our scheme when the embedding algorithm is applied once, where the capacity represents the embedded bit number and the bbp indicates bits per pixel. Our method has the largest capacity. The average capacity of our scheme is about 7 times of Ni et al.'s method, but the image qualities of their and our approaches are similar. Furthermore, our average capacity is 13% higher than that of Li et al.'s method. Table 2 shows a comparison between Fallahpour et al.'s method [3] and our scheme. In Fallahpour et al.'s method, they used the main peak in the adaptive shifted prediction error (ASPE) with three prediction types: the GAP method [20], Jiang et al.'s

Table 4 The experimental results of different layers using the clockwise 180° rotation

Level	Indicator	Airplane	Baboon	Boat	Goldhill	Lena	Peppers	Tiffany
1	Capacity	94,524	23,928	59,598	50,250	73,686	68,853	74,456
	PSNR	49.00	48.34	48.66	48.57	48.79	48.75	48.80
2	Capacity	158,218	44,814	100,188	90,607	129,037	119,600	126,168
	PSNR	44.96	44.52	44.24	45.10	45.06	44.81	44.19
3	Capacity	199,636	63,972	132,407	124,028	173,169	159,889	163,487
	PSNR	41.72	41.03	41.13	41.85	41.78	41.44	40.87
6	Capacity	291,619	112,262	206,644	200,797	265,422	245,638	243,564
	PSNR	36.24	35.70	36.02	36.72	36.52	36.16	35.68
9	Capacity	353,363	149,460	256,635	257,023	330,350	301,922	298,835
	PSNR	32.50	32.37	32.23	32.92	32.50	32.52	32.26
12	Capacity	397,070	180,186	294,889	299,921	379,078	344,721	340,473
	PSNR	30.20	29.87	29.86	30.47	30.27	29.91	29.66

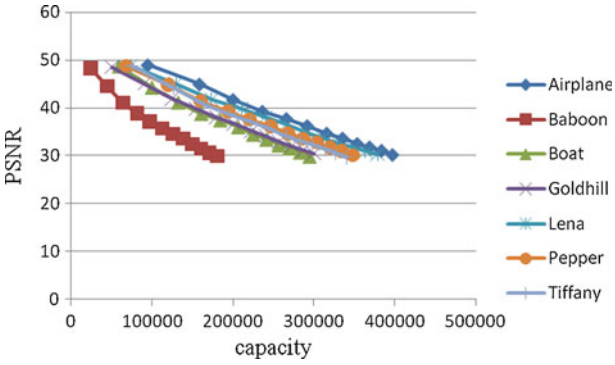


Fig. 16 Relationships between capacities and PSNR values in our scheme with the clockwise 180° rotation for one to twelve levels

method [6], and the MED method [19]. Our average capacity is 12.50% higher, 20.10% higher, 19.51% higher than that of the GAP method, Jiang et al.'s method, and the MED method. Tables 3 and 4 show the embedding results of different levels without and with the 180° rotations, respectively. The results show that the rotation operation can advance the image qualities so that more secret data can be embedded based on the same image quality. Figure 16 shows the relationship between capacities and PSNR values for multilevel data embedding when levels are between 1 and 12. It demonstrates that smooth images, such as Airplane, have larger embedding capacities and better PSNR values than that of complex images, such as Baboon. Table 5 shows the bit numbers of overhead for pre-processing the overflow and underflow problems at different levels. Most images have low overhead. Table 6 shows a comparison among Lin and Hsueh's method [12], Zeng et al.'s method [24], Li et al.'s method [11], Hsiao et al.'s method [5] and our scheme with PSNR values close to 30 dB, where Hsiao et al.'s method is difference-expansion-based. Our method includes the rotation operation, and the capacity has excluded the overhead of processing overflow and underflow problems. The number of levels of our scheme is shown in the parentheses. As shown in Table 6, our average capacity is 7% higher, 18% higher, 21% higher and 17% higher than that of Lin and Hsueh's method [12], Zeng et al.'s method [24],

Table 5 The bit numbers of overhead for pre-processing the overflow and underflow problems

Level	Airplane	Baboon	Boat	Goldhill	Lena	Peppers	Tiffany
1	0	0	0	0	0	17	0
2	0	0	0	0	0	46	0
3	0	0	0	0	0	139	0
4	0	0	0	0	0	229	0
5	0	0	0	0	0	341	0
6	0	0	0	0	0	567	0
7	0	4	1	0	0	867	0
8	0	8	1	0	0	1,185	0
9	0	10	1	0	0	1,572	0
10	0	23	2	0	0	1,753	0
11	0	32	2	0	0	2,164	0
12	0	38	3	0	0	2,254	0
13	0	48	12	0	0	2,453	0
14	0	61	15	1	0	2,835	0

Table 6 The comparison for multilevel data hiding when the image quality is close to 30 dB

Images	Lin and Hsueh's method [12]		Zeng et al.'s method [24]		Li et al.'s method [11]		Hsiao et al.'s method [5]		Our scheme (Level)	
	Capacity	bpp	Capacity	bpp	Capacity	bpp	Capacity	bpp	Capacity	bpp
Airplane	366,784	1.40	338,492	1.29	310,773	1.19	286,488	1.09	409,755 (13)	1.56
Baboon	161,118	0.61	128,462	0.49	144,628	0.55	138,398	0.53	180,186 (12)	0.69
Boat	307,193	1.17	275,795	1.05	226,535	0.86	266,724	1.02	283,031 (11)	1.08
Goldhill	302,784	1.16	no data		236,208	0.90	245,370	0.94	299,921 (12)	1.14
Lena	308,474	1.18	281,451	1.07	300,867	1.15	303,700	1.16	393,246 (13)	1.50
Peppers	322,744	1.23	248,036	0.95	281,645	1.07	303,736	1.16	348,280 (12)	1.33
Tiffany	331,962	1.27	314,142	1.20	274,922	1.05	318,288	1.21	340,473 (12)	1.30
Average	300,151	1.14	264,396	1.01	253,654	0.97	266,101	1.02	322,127	1.23
PSNR	30.26		30.34		30.08		30		30.02	

Li et al.'s method [11] and Hsiao et al.'s method [5], respectively. Figure 17 demonstrates the compared results of the proposed scheme with other reversible schemes for the seven images. Table 7 shows a comparison among Kim et al.'s [8], Luo et al.'s [13] and our scheme when PSNR values are close to 48 db, 44 db, or 40 db. The parameter embedding level (EL for short) is the shifting distance used in Kim et al.'s and Luo et al.'s methods. As shown in Table 7, our capacities are the largest in all cases.

5 Conclusions

In this paper, we propose a side-match prediction method to strengthen the lossless data hiding technique based on the histogram. In our proposed side-match prediction methods, pixels can be predicted more precisely to create higher peak points than that of Li et al.'s method [11]. The predictive error values to create the histogram are as many as the pixels in the original image. Furthermore, an image-rotation scheme is proposed to come up with better results for multi-level data hiding. Experimental results show that our proposed schemes have better embedded results than that of existing approaches for single-level and multi-level data hiding.

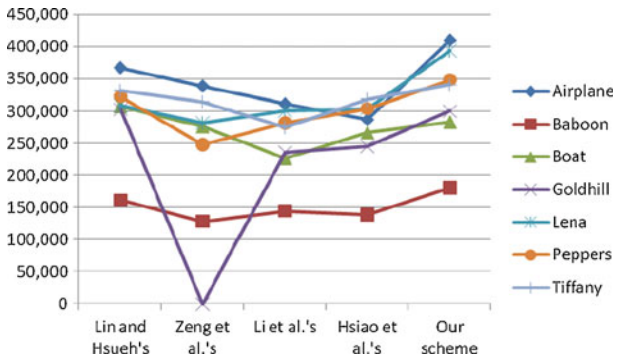


Fig. 17 Comparison of the results of the proposed scheme with other reversible schemes of the seven images

Table 7 Comparisons of our, Kim et al.'s [8] and Luo et al.'s [13] results with the 3*3 block partition

Test image	Indicator	EL=0		Our scheme in the first level	EL=1		Our scheme in the second level	EL=2		Our scheme in the third level
		Kim et al.'s	Luo et al.'s		Kim et al.'s	Luo et al.'s		Kim et al.'s	Luo et al.'s	
Airplane	capacity	32,436	39,853	94,524	83,011	96,501	158,218	116,858	130,636	199,636
	PSNR	48.96	49.03	48.99	43.63	43.83	44.95	40.84	41.18	41.71
Barbara	capacity	15,865	19,788	41,885	45,964	56,054	72,841	72,461	85,033	98,177
	PSNR	48.79	48.83	48.50	43.12	43.25	44.92	40.02	40.23	41.52
Lena	capacity	22,581	28,860	73,686	64,564	79,787	129,037	100,401	118,372	173,169
	PSNR	48.86	48.92	48.79	43.92	43.55	45.06	40.44	40.79	41.78
Truck	capacity	14,418	18,184	27,569	26,848	33,674	51,244	46,803	57,340	72,608
	PSNR	48.78	48.82	48.37	42.96	43.05	44.41	39.66	39.81	40.91

Acknowledgement This research was partially supported by the National Science Council of the Republic of China under the Grants NSC 98-2221-E-015-001-MY3, NSC 100-2221-E-015 -001 -MY2, and NSC 99-2918-I-015-001. We are also indebted to Prof. C.H. Yang for the discussions in the revised version toward a better quality presentation.

References

1. Alattar AM (2004) Reversible watermark using the difference expansion of generalized integer transform. IEEE Trans Image Process 13(8):1147–1156
2. Chang CC, Lin CY (2006) Reversible steganography for VQ-compressed images using side matching and relocation. IEEE Transactions on Information Forensics and Security 1(4):493–501
3. Fallahpour M, Megias D, Ghanbari M (2011) Subjectively adapted high capacity lossless image data hiding based on prediction errors. Multimed Tool Appl 52:513–527
4. Hong W, Chen TS, Shiu CW (2008) Reversible data hiding based on histogram shifting of prediction errors. Proceedings of the International Symposium on Intelligent Information Technology Application Workshop, 2008, Dec. 2008, 292–295
5. Hsiao JY, Chan KF, Chang JM (2009) Block-based reversible data embedding. Signal Process 89(4):556–569
6. Jiang J, Guo B, Yang S (2000) Revisiting the JPEG-LS prediction scheme. IEE Proc Vis Image Signal Process 147:575–580
7. Jin HL, Fujiyoshi M, Kiya H (2007) Lossless data hiding in the spatial domain for high quality image. IEICE Transactions on Fundamentals E90-A(4):771–777
8. Kim KS, Lee MJ, Lee HY, Lee HK (2009) Reversible data hiding exploiting spatial correlation between sub-sampled images. Pattern Recogn 42(11):3083–3096
9. Kim HJ, Sachnev V, Shi YQ, Nam J, Choo HG (2008) A novel difference expansion transform for reversible data embedding. IEEE Transactions on Information Forensics and Security 3(3):456–465
10. Lee S, Yoo CD, Kalker T (2007) Reversible image watermarking based on integer-to-integer wavelet transform. IEEE Transactions on Information Forensics Security 2(3):321–330
11. Li YC, Yeh CM, Chang CC (2010) Data hiding based on the similarity between neighboring pixels with reversibility. Digit Signal Process 20(4):1116–1128
12. Lin CC, Hsueh NL (2008) A lossless data hiding scheme based on three-pixel block differences. Pattern Recogn 41(4):1415–1425
13. Luo H, Yu F-X, Chen H, Huang Z-L, Li H, Wang P-H (2011) Reversible data hiding based on block median preservation. Inform Sci 181(2):308–328
14. Ni Z, Shi YQ, Ansari N, Su W (2006) Reversible data hiding. IEEE Trans Circ Syst Video Tech 16 (3):354–362
15. Thodi DM, Rodriguez JJ (2007) Expansion embedding techniques for reversible watermarking. IEEE Trans Image Process 16(3):723–730
16. Tian J (2003) Reversible data embedding using a difference expansion. IEEE Trans Circ Syst Video Tech 13(8):831–841
17. Tsai P, Hu YC, Yeh HL (2009) Reversible image hiding scheme using predictive coding and histogram shifting. Signal Process 89(6):1129–1143
18. Tseng HW, Hsieh CP (2009) Prediction-based reversible data hiding. Inform Sci 179(14):2460–2469
19. Weinberger MJ, Seroussi G, Sapiro G (2000) The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS. IEEE Trans Image Process 9:1309–1324
20. Wu X, Memon N (1997) Context-based, adaptive, lossless image coding. IEEE Trans Comm 45:437–444
21. Yang CH, Lin YC (2009) Reversible data hiding of a VQ index table based on referred counts. J Vis Comm Image Represent 20(6):399–407
22. Yang CH, Lin YC (2010) Fractal curves to improve the reversible data embedding for VQ-indexes based on locally adaptive coding. J Vis Comm Image Represent 21:334–342
23. Yang CH, Wang WJ, Huang CT, Wang SJ (2011) Reversible steganography based on side match and hit pattern for VQ-compressed images. Inform Sci 181(11):2218–2230
24. Zeng X, Ping L, Li Z (2009) Lossless data hiding scheme using adjacent pixel difference based on scan path. J Multimed 4(3):145–152

Fu-Hau Hsu received his Ph.D. degree in Computer Science from Stony Brook University, New York, USA in 2004. He is an assistant professor in the Department of Computer Science and Information Engineering at National Central University, Taiwan, R.O.C. His research interests include system security, mobile device security, web security, information hiding, operating system, and networking.

Min-Hao Wu received the M.S. degree in Computer Science from Nation Pingtung University of Education, Pingtung, Taiwan, in 2010, and is currently pursuing the Ph.D. degree in the Department of Computer Science and Information Engineering at National Central University, Taoyuan, Taiwan, R.O.C. His current research interests include data hiding, image watermarking, and images processing.

Shiuh-Jeng Wang received his PhD degree in Electrical Engineering at National Taiwan University, Taipei, Taiwan in 1996. Dr. WANG was a visiting scholar of Computer Science Dept. at Florida State University (FSU), USA in 2002 and 2004. He also was a visiting scholar of Dept. of Computer and Information Science and Engineering at University of Florida (UF) in 2004, 2005, 2010 and 2011. He served the editor-in-chief of the journal of Communications of the CCISA in Taiwan from 2000-2006. He has been elected as the Director of Chinese Cryptology and Information Security Association (CCISA) since 2000. Dr. Wang academically toured the CyLab with School of Computer Science in Carnegie Mellon University, USA, in 2007 for international project collaboration inspection. He is also the authors of eight books (in Chinese versions): Information Security, Cryptography and Network Security, State of the Art on Internet Security and Digital Forensics, Eyes of Privacy –Information Security and Computer Forensics, Information Multimedia Security, Computer Forensics and Digital Evidence, Computer Forensics and Security Systems, and Computer and Network Security in Practice, published in 2003, 2004, 2006, 2007, and 2009, respectively. Prof. WANG has published over 250 papers in referred Journals/Conference proceedings/Technique reports so far. He is a full professor and a member of the IEEE, ACM. He served a lot of academic and reputable journals in the position of guest-editors. Prof. WANG is currently with the Editor-in-Chief of Journal at JITAS (http://jitas.im.cpu.edu.tw/). He was the lead editor- IEEE J-SAC (IEEE Journal on Selected Areas in Communications), at http://www.comsoc.org/livepubs/sac/index.html.