Side-match Approach for Improving Histogram-Based Reversible Data Hiding

Cheng-Hsing Yang
Department of Computer Science, National Pingtung University of Education
Email: chyang@mail.npue.edu.tw

Meng-Hsuan Tsai
Department of Computer Science and Engineering, National Sun Yat-sen University
Email: emma7337@gmail.com

Min-Hao Wu
Department of Computer Science, National Pingtung University of Education
Email: bm097116@mail.npue.edu.tw

Chiu-Chih Jen
Department of Computer Science, National Pingtung University of Education
Email: bm098108@mail.npue.edu.tw

***Abstract*— Reversible data hiding has drawn considerable attention in recent years. Being reversible, the decoder can extract the hidden data and recover the original image completely. In this paper we used the side-match prediction to achieve a histogram-based reversible data hiding. The histogram is created by exploiting all difference values between pixels and their predictive values. Experimental results show that our method is capable of providing a great embedding capacity without making noticeable distortion. In the one-level hiding, our method remains image qualities larger than 48 dB and has the best capacity. Moreover, in the multilevel case our method performs better than other existed methods. Our method can successfully increase the embedding capacity from histogram-based data hiding and remain the image quality well.**

***Index Terms* — reversible data hiding, side match, prediction, histogram**

I. Introduction

With the rapid development of network technologies and the coming of the digital era, Internet has become indispensable for many people. By the development of Internet, a lot of new businesses are developed, such as e-commerce, e-learning, online game, and video-on-demand, etc. Many enterprises have expanded their traditional business activities in the Internet. On every day, thousands of multimedia data are transferred conveniently and efficiently over Internet. Because digital multimedia data, such as voices, videos, images, texts, etc., have the attributes of easy copy, modification, and distribution, the development of Internet has increased the problems of multimedia securities. How to protect the authentication and the ownership of multimedia data has become an important topic. Many researchers have paid high 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 for 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. Besides, 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 [2-18]. According to where the data are embedded, these approaches can be classified into three categories: the spatial domain [2-13], the frequency domains [14,15], and other compression types, such as vector quantization

(VQ) [16, 17]. In those developed reversible data hiding methods, two main technologies have been widely applied: the difference-expansion-based technology [8-14] and the histogram-based technology [2-7]. In 2006, Ni et al. presented a reversible data hiding method based on the histogram [2]. 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. But their method used the pixel values in the original image to create the histogram. The peak values of the histogram are not high enough. Some methods used the concept of prediction to increase the peak height [3-6].

In this paper, we propose a new reversible information hiding method based on the histogram for grayscale images. We used the side-match prediction to improve a histogram-based reversible data hiding. Our predictive difference values are as many as the pixel values, in our approach. All predictive difference values are transformed into histogram to create higher peak values and to improve the embedding capacity. Experimental results show that our histogram-based reversible hiding approach can raise a larger capacity and still remain a good image quality, compared to other histogram-based approaches. The remainder of this paper is as follows. In Section II, we introduce some related works of reversible information hiding technologies. Our proposed scheme is described in Section III and some experimental results are shown in Section IV. Finally, we bring some conclusions in Section V.

II. Related Works

A. Ni et al.'s method

Ni et al. proposed a reversible data hiding method in 2006 [2]. 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 assume 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.

Fig. 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.

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

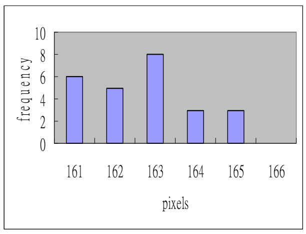

Figure 1. Ni et al.'s example.

Suppose the to-be-embedded data are 11001101.

Pixels shown in Fig. 2 are scanned from left to right and from top to down. All pixels with value equal to p = 163 can be used to embed one secret bit. The embedded results are shown in Fig. 3.

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

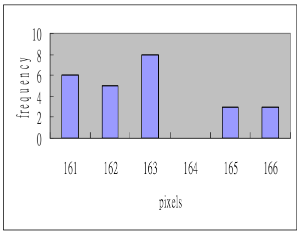

Figure 2. Results after shifting.

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

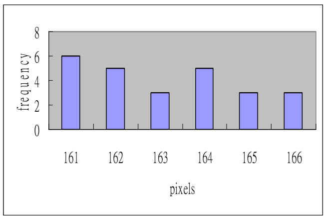

Figure 3. The embedded results.

B. Lin et al.'s method

In 2008, Lin et al. applied the histogram approach at a difference image to achieve the reversible data hiding [5]. Their method used the peak point of a histogram created from a difference image to create the free space for hiding messages. For an image H with $P \times Q$ pixels, a difference image with size $P \times (Q-1)$ is generated. In order to avoid serious distortions, the original image was divided into 4 × 4 non-overlapping blocks in their experiments. Each block generates a difference image with size 4 × 3. Let $H_{i,j}$ be a pixel value in image H at location (i, j) and $D_{i,j}$ be a difference value in difference image D at location (i, j). Their algorithm is as follows:

1. Generate the difference image D from the original image H. The formula is as follows:

$$D_{i,j} = \left| H_{i,j} - H_{i,j+1} \right|.$$

2. Generate the histogram of the difference image and find the peak point p.
3. If the value $D_{i,j}$ is greater than p, change the value to $D_{i,j}+1$.
4. Scan the difference image and embed a secret bit when the value $D_{i,j}$ is equal to p as follows:
   (a) If the to-be-embedded bit is 0, the value $D_{i,j}$ remains p.
   (b) If the to-be-embedded bit is 1, the value $D_{i,j}$ is set to $p+1$.
5. Use the original image and the embedded difference image to construct the stego-image S. Let $S_{i,j}$ be a pixel value of the stego-image. The formula is as follows:

$$S_{i,0} = \begin{cases} H_{i,0} & \text{if } H_{i,0} \le H_{i,1}, \\ H_{i,1} + D_{i,0} & \text{otherwise,} \end{cases}$$

$$S_{i,1} = \begin{cases} H_{i,0} + D_{i,0} & \text{if } H_{i,0} \le H_{i,1}, \\ H_{i,1} & \text{otherwise,} \end{cases}$$

$$S_{i,j} = \begin{cases} S_{i,j-1} + D_{i,j-1} & \text{if } H_{i,j-1} \le H_{i,j}, \\ S_{i,j-1} - D_{i,j-1} & \text{otherwise.} \end{cases}$$

6. Output the marked image S and the peak point p.

Fig. 4 shows an example of Lin et al.'s method. Fig. 4(a) is a 4 × 4 block H. Fig. 4(b) is the difference image D create from H, and the peak point is p = 1. Fig. 4(c) is the shifted difference image *D'* which is obtained by shifting all values larger than *p* = 1 in *D*. Suppose, the to-be-embedded secret data are 0110. Fig. 4(d) is the embedded difference image *D''* which embeds one secret bit to each pixel with value equal to *p* = 1 in *D'*. Fig. 4(e) shows the embedded image *S* after Step 5 is executed.

H

162	156	163	160
161	159	158	159
160	161	159	155
158	158	156	157

(a)

D

6	7	3
2	1	1
1	2	4
0	2	1

(b)

D'

7	8	4
3	1	1
1	3	5
0	3	1

(c)

D''

7	8	4
3	1	2
2	3	5
0	3	1

(d)

S

163	156	164	160
162	159	158	160
160	162	159	154
158	158	155	156

(e)

Figure 4. An example of Lin et al.'s method.

III. OUR PROPOSED METHOD

In this section, we describe our method in detail. We used the side-match prediction to achieve a histogram-based reversible data hiding. This method can increase the embedded capacity and the image quality remains as well. Fig. 5 shows the main concept of the side-match prediction. Our predictive method is to employ the neighboring pixels $H_{i,j-1}, H_{i-1,j-1}, H_{i-1,j}$ *and* $H_{i-1,j+1}$ to predict the pixel $H_{i,j}$.

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

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

A. Embedding algorithm

Input: Original image, secret message

Output: Stego-image, two pairs of peak and zero points

1. Input the original image $H = \{H_{0,0}, H_{0,1}, ..., H_{0,511}, H_{1,0}, ... H_{511,511}\}$.
2. Predict each pixel in the original image as follows. Assume $H_{i,j}$ is the pixel value in the original image, and $D_{i,j}$ is the predictive difference value.
   (a) If $i = j = 0$, then
   $$D_{i,j} = H_{i,j} - 128 \quad . \quad (1)$$
   (b) If $i = 0$ and $j \neq 0$, then
   $$D_{i,j} = H_{i,j} - H_{i,j-1} \,. \quad (2)$$
   (c) If $i \neq 0$ and $j = 0$, then
   $$D_{i,j} = H_{i,j} - \left\lfloor \frac{H_{i-1,j} + H_{i-1,j+1}}{2} \right\rfloor . \quad (3)$$
   (d) If $i \neq 0$ and $j = 511$, then
   $$D_{i,j} = H_{i,j} - \left\lfloor \frac{H_{i,j-1} + H_{i-1,j-1} + H_{i-1,j}}{3} \right\rfloor . \quad (4)$$
   (e) Else,

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

3. Create the histogram $H(x)$ with $x \in [-255, 255]$ from all predictive difference values.
4. Find two pairs of peak and zero point (P_1, Z_1) and (P_2, Z_2).
5. Let $D'_{i,j}$ be the new value of $D_{i,j}$ after the following shifting step and embedding step. 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. Convert the predictive difference values into pixel value. Assume $H'_{i,j}$ is the embedded pixel value in the stego-image.
   (a) If $i = j = 0$, then

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

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

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

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

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

   (d) If $i \neq 0$ and $j = 511$, then

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

   (e) Else,

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

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

As shown in Fig. 6, a 5 × 5 grayscale original image $H = \{H_{0,0}, H_{0,1}, ..., H_{0,4}, H_{1,0}, ..., H_{4,4}\}$ is given to explain our embedding algorithm. We predict the pixels first. For example, $D_{0,0} = H_{0,0} - 128 = 7 - 128 = -121$, $D_{0,1} = H_{0,1} - H_{0,0} = 6 - 7 = -1$, $D_{1,0} = H_{1,0} - \left\lfloor \frac{H_{0,0} + H_{0,1}}{2} \right\rfloor = 6 - \left\lfloor \frac{7+6}{2} \right\rfloor = 0$,

$$D_{1,1} = H_{1,1} - \left\lfloor \frac{H_{1,0} + H_{0,0} + H_{0,1} + H_{0,2}}{4} \right\rfloor = 7 - \left\lfloor \frac{6+7+6+5}{4} \right\rfloor = 1$$

and so on. The results are shown in Fig. 6. The histogram created from all predictive difference values is also shown in Fig. 7. According to the histogram, we find 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. 8.

7	6	5	6	6
6	7	6	6	5
8	7	6	7	5
6	5	6	7	6
7	6	5	4	5

Figure 6. The original image.

-121	–1	–1	1	0
0	1	0	1	–1
2	1	0	2	–1
–1	–1	0	1	0
2	0	–1	–2	0

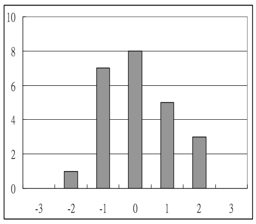

Figure 7. The predictive difference values and their histogram.

-121	–1	–1	2	0
0	2	0	2	–1
3	2	0	3	–1
–1	–1	0	2	0
3	0	–1	–3	0

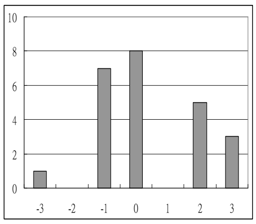

Figure 8. The predictive difference values and their histogram after shifting.

Then, we embedded the secret messages. The predictive difference values equivalent to P_1 or P_2 are used to embed secret messages. Assume the secret message I = $101001101000110_{(2)}$. After the secret message I is embedded, the predictive difference values and their histogram are shown in Fig. 9. Finally, the predictive difference values are inverted into pixel values. For example,

$$H'_{0,0} = D'_{0,0} + 128 = -121 + 128 = 7,$$

$$H'_{0,1} = D'_{0,1} + H_{0,0} = -2 + 7 = 5,$$

$$H'_{1,0} = D'_{1,0} + \left\lfloor \frac{H_{0,0} + H_{0,1}}{2} \right\rfloor = 0 + \left\lfloor \frac{7+6}{2} \right\rfloor = 6,$$

$$H'_{1,1} = D'_{1,1} + \left\lfloor \frac{H_{1,0} + H_{0,0} + H_{0,1} + H_{0,2}}{4} \right\rfloor = 2 + \left\lfloor \frac{6+7+6+5}{4} \right\rfloor = 8$$

and so on. The embedded results are shown in Fig. 10. That is also the stego-image.

-121	–2	–1	2	1
0	2	0	2	–2
3	2	1	3	–1
–2	–1	0	2	0
3	1	–2	–3	0

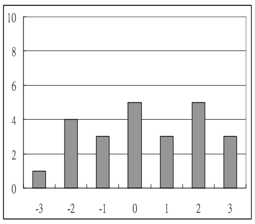

Figure 9. The predictive difference values and their histogram after embedding.

7	5	5	7	7
6	8	6	7	4
9	8	7	8	5
5	5	6	8	6
8	7	4	3	5

Figure 10. The stego-image.

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

Now, we perform extracting and reversing operations on the embedded results shown in Fig. 10. Assume $H'_{i,j}$ is an embedded pixel, $D'_{i,j}$ is an embedded predictive difference value, $D_{i,j}$ is a predictive difference value, and $H_{i,j}$ is an original pixel. For this example, (P_1, Z_1) = (0, 3) and (P_2, Z_2) $= (-1, -3)$. We process the first pixel value $H'_{0,0}$ first. $D'_{0,0} = H'_{0,0} - 128 = 7 - 128 = -121$. We can find that $D'_{0,0}$ does not belong to any situation of Step 4, so $D_{0,0} = D'_{0,0} = -121$. Finally, calculate the original pixel value $H_{0,0} = D_{0,0} + 128 = -121 + 128 = 7$. Next, the second pixel value $H'_{0,1}$ is processed. $D'_{0,1} = H'_{0,1} - H_{0,0} = 5 - 7 = -2$. We can find that $D'_{0,1} = -2$ is equal to $P_2 - 1$, so secret bit 1 is extracted and the predictive difference value is recovered as $D_{0,1} = D'_{0,1} + 1 = -2 + 1 = -1$. Finally, calculate the original pixel value $H_{0,1} = D_{0,1} + H_{0,0} = -1 + 7 = 6$. Similarly, other pixel values $H'_{i,j}$ are processed in order to generate their original pixel values $H_{i,j}$.

IV. Experimental Result and Discussion

In this section, we show the experimental results of our proposed schemes and some discussions of the overflow and underflow problems. The detailed experimental comparison between our schemes and other scholar's methods are shown in Section *A*. Section *B* shows some discussions.

A. Experimental results

In this subsection, we display the performance of our scheme. In our experiments, we used the random numbers as the secret messages. Five of 512 × 512 grayscale images, Airplane, Baboon, Boat, Lena and Peppers, are used as the cover images. As shown in Fig. 11, the side-match approaches of predicting pixel $H_{i,j}$ can come from different directions and have various methods. For example, from the left-up corner, {$H_{i,j-1}$, $H_{i-1,j-1}$, $H_{i-1,j}$, $H_{i-1,j+1}$}, {$H_{i,j-1}$, $H_{i-1,j-1}$, $H_{i-1,j}$} and {$H_{i,j-1}$, $H_{i-1,j}$} can be used to predict pixel $H_{i,j}$. Similarly, from the right-up corner, {$H_{i,j+1}$, $H_{i-1,j+1}$, $H_{i-1,j}$, $H_{i-1,j-1}$}, {$H_{i,j+1}$, $H_{i-1,j+1}$, $H_{i-1,j}$} and {$H_{i,j+1}$, $H_{i-1,j}$} can be used to predict pixel $H_{i,j}$. In the following experiment, we use {$H_{i,j+1}$, $H_{i-1,j}$} to predict $H_{i,j}$ from the right-up corner. Table 1 shows a comparison among Ni et al.'s method [2], Lin et al.'s method [5], and our scheme when the embedding algorithm is applied once. Two peak points are the first highest point and the second highest point chosen from the histogram. The averaged capacity of our scheme is about 7 times of Ni et al.'s method and the image qualities of both their and our approaches are similar. Compared to the averaged capacity of Lin et al.'s method, our averaged capacity is 11% higher and the image quality is better. Besides, we performed the multilevel data hiding on our scheme. As shown in Fig. 12, we perform a 180^{o} clockwise rotation on the test image after the hiding procedure is finished at each level. Table 2 and Table 3 show the embedding results of different levels and various images without and with the 180^{o} rotation, respectively. The results show that the rotation operation can improve the image qualities such that more secret data can be embedded based on the same image quality. Table 4 shows a comparison among Lin et al.'s method [4], Lin et al.'s method [5], Hsiao et al.'s method [8], and our scheme with PSNR values close to 30 dB. Our method includes the rotation operation. The number of levels of our scheme is shown in the parentheses. Table 4 shows that our averaged capacity is 0.57% higher, 23.5% higher, and 6.7% higher than that of Lin's et al.'s method [5], Hsiao's et al.'s method [8], and Lin's et al.'s method [4], respectively.

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

Figure. 11. The main concept of the side-match prediction.

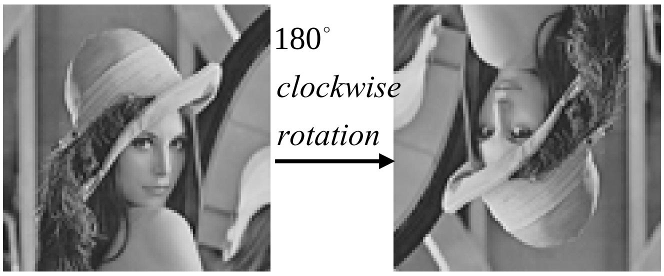

Figure. 12. The 180^o clockwise rotation of Lena

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

Images	Ni et al.'s method [2]	Lin et al.'s method [5]	Our scheme
Airplane	17,415	69,941	94,524
Baboon	5,796	38,465	23,928
Boat	11,109	56,713	59,598
Lena	5,760	65,349	73,686
Peppers	5,737	64,632	68,853
Average	9,163	59,022	64,118
PSNR	48.30	48.67	48.70

Table 2. Without the 180^o clockwise rotation.

Level		Airplane	Baboon	Boat	Lena	Pepper
1	capacity	94,524	23,928	59,598	55,583	68,853
	PSNR	49	48.34	48.66	48.62	48.75
2	capacity	148,907	45,375	100,462	98,857	118,595
	PSNR	43.56	42.97	43.4	43.42	43.78
3	capacity	197,268	62,767	133,354	131,874	158,442
	PSNR	40.16	39.78	40.43	40.42	40.6
6	capacity	287,945	109,830	207,783	208,771	247,900
	PSNR	34.59	33.91	34.29	34.51	34.9
9	capacity	349,613	145,262	258,015	262,967	306,457
	PSNR	31.08	30.46	31.33	31.18	31.21
12	capacity	396,062	175,019	297,053	303,317	351,457
	PSNR	28.74	28.09	28.66	28.68	29.07

Table 3. With the 180^o clockwise rotation.

Level		Airplane	Baboon	Boat	Lena	Pepper
1	capacity	94,524	23,928	59,598	73,686	68,853
	PSNR	49	48.34	48.66	48.79	48.75
2	capacity	158,218	44,814	100,188	129,037	119,600
	PSNR	44.96	44.53	44.24	45.06	44.81
3	capacity	199,636	63,972	132,407	173,169	159,889
	PSNR	41.72	41.03	41.13	41.78	41.44
6	capacity	291,619	112,262	206,644	265,422	246,220
	PSNR	36.24	35.7	36.02	36.52	36.16
9	capacity	353,363	149,460	256,635	330,350	303,720
	PSNR	32.5	32.37	32.23	32.5	32.77
12	capacity	397,070	180,186	294,889	379,078	348,280
	PSNR	30.2	29.87	29.86	30.27	30.12

Table 4. The comparison for multilevel data hiding.

Images	Lin's et al.'s method [5]	Hsiao's et al.'s method [8]	Lin's et al.'s method [4]	Our scheme (Level)
Airplane	362,847	286,488	367,392	409,755 (13)
Baboon	230,079	138,398	162,544	170,628 (11)
Boat	314,196	266,724	307,937	283,031 (11)
Lena	346,568	303,700	309,166	393,246 (13)

Peppers	342,175	303,736	356,450	348,280 (12)
Average	319,173	259,809	300,698	320,988
PSNR	30.19	30.00	30.26	30.10

B. Discussions

After secret messages are embedded in our method, the change of each pixel remains within ±1. Therefore, if pixel values are equal to 0 and 255 in the original image, they may become –1 and 256 in the stego-image and cause the underflow and the overflow problems. In order to avoid this problem, we used a pre-processing method [18]. 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. So, 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. From the experimental results shown in Table 5, we know that the overflow and underflow probabilities are very few. Therefore, the overhead is low. Moreover, the overhead can be embedded together with secret data. For fourteen grayscale images, Table 4 shows the bits of overhead, the pure loads, and the PSNR values.

Table 5. The bits of overhead, the pure load and the PSNR values.

Image	overhead	pure load	PSNR
Airplane	0	94,524	49
Baboon	0	23,928	48.34
Barb	0	41,885	48.5
Boat	0	59,598	48.66
Girl	0	68,044	48.74
Goldhill	0	50,250	48.58
Jet	0	94,524	49
Lena	0	73,686	48.79
Pepper	17	68,853	48.75
Sailboat	0	39,679	48.48
Ship	16	35,042	48.44
Tiffany	0	74,456	48.81
Toys	3	60,997	48.68
Zelda	23	54,938	48.62

ACKNOWLEDGEMENTS

This research was supported by the National Science Council of the Republic of China under the Grants NCS 98-2221-E-153-001.

REFERENCE

[1] S.P. Mohanty and B.K. Bhargava, “Invisible watermarking based on creation and robust insertion-extraction of image adaptive watermarks,” *Transactions on Multimedia Computing, Communications, and Applications*, Vol. 5, No. 12, pp. 1-24, 2008.

[2] Z. Ni, Y.Q. Shi, N. Ansari, and W. Su, “Reversible data hiding,” *IEEE Transaction on Circuits and Systems for Video Technology*, Vol. 16, No. 3, pp. 354-362, 2006.

[3] P. Tsai, Y.C. Hu, and H.L. Yeh, “Reversible image hiding scheme using predictive coding and histogram shifting,” *Signal Processing*, Vol. 89, Issue 6, pp. 1129-1143, 2009.

[4] C.C. Lin and N.L. Hsueh, “A lossless data hiding scheme based on three-pixel block differences,” *Pattern Recognition*, Vol. 41,

No. 4, pp. 1415-1425, 2008.

[5] C.C. Lin, W.L. Tai, and C.C. Chang, “Multilevel reversible data hiding based on histogram modification of difference images,” *Pattern Recognition*, Vol. 41, No. 12, pp. 3582-3591, 2008.

[6] E. Chrysochos, V. Fotopoulos, A.N. Skodras, and M. Xenos, “Reversible image watermarking based on histogram modification”, *11th Panhellenic Conference on Informatics with international participation*, Vol. B, pp. 93-104, 2007.

[7] S. Yousefi, H. Rabiee, E. Yousefi, and M. Ghanbari, "Reversible date hiding using histogram sorting and integer wavelet transform," *In Proceedings of IEEE DEST*, pp. 487-490, 2007.

[8] J.Y. Hsiao, K.F. Chan, and J.M. Chang, “Block-based reversible data embedding,” *Signal Processing*, Vol. 89, No. 4, pp. 556-569, 2009.

[9] D.M. Thodi and J.J. Rodriguez, “Expansion embedding techniques for reversible watermarking,” *IEEE Transactions on Image Processing*, Vol. 16, No. 3, pp. 723-730, 2007.

[10] H.J. Kim, V. Sachnev, Y.Q. Shi, J. Nam, and H.G. Choo, “A novel difference expansion transform for reversible data embedding,” *IEEE Transactions on Information Forensics and Security*, Vol. 3, No. 3, pp. 456-465, 2008.

[11] H.L. Jin, M. Fujiyoshi, and H. Kiya, “Lossless data hiding in the spatial domain for high quality image,” *IEICE Transactions on Fundamentals*, Vol. E90-A, No. 4, pp. 771-777, 2007.

[12] J. Tian, “Reversible data embedding using a difference expansion,” *IEEE Transactions on Circuits and System for Video Technology*, Vol. 13, No. 8, pp. 831-841, 2003.

[13] A.M. Alattar, “Reversible watermark using the difference expansion of generalized integer transform,” *IEEE Transactions on Image Processing*, Vol. 13, No. 8, pp. 1147-1156, 2004.

[14] C.C. Chang, C.C. Lin, C.S. Tseng, and W.L. Tai, “Reversible hiding in DCT-based compressed images,” *Information Sciences*, Vol. 177, No. 13, pp. 2768-2786, 2007.

[15] S. Lee, C.D. Yoo, and T. Kalker, “Reversible image watermarking based on integer-to-integer wavelet transform,” *IEEE Transactions on Information Forensics Security*, Vol. 2, No. 3, pp. 321-330, 2007.

[16] C.C. Chang and C.Y. Lin, “Reversible steganography for VQ-compressed images using side matching and relocation,” *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 4, pp. 493-501, Dec. 2006.

[17] C.H. Yang and Y.C. Lin, “Reversible data hiding of a VQ index table based on referred counts”, *Journal of Visual Communication and Image Representation,* Vol. 20, No. 6, pp. 399-407, August 2009.

[18] H.W. Tseng and C.P. Hsieh, “Prediction-based reversible data hiding,” *Information Sciences*, Vol. 179, No. 14, pp. 2460-2469, June 2009.