A Location-Map Free Reversible Watermarking With Capacity Control

Wen-Shyong Hsieh  
Department of Computer and Communication  
Shu-Te University, Taiwan  
wshsieh@stu.edu.tw

Jui-Ming Kuo  
Department of Computer Science and Engineering,  
National Sun Yat-sen University, Taiwan  
g6326852@gmail.com

Abstract—Reversible watermarking techniques enable the extraction of the embedding bits from a watermarked image in a lossless way. It exploits the high spatial correlation among neighboring pixels. Application in reversible watermarking includes military and medical images. However, images occurs overflow or underflow problems during the imbedding process. Since pixels value may be out of range between [0:255]. Most methods require a location map to solve such problems. In this study, we propose a location map free reversible watermarking algorithm. First, a prediction threshold value is computed, histogram shifting scheme based on the prediction threshold value to solve overflow and underflow problems, second, another threshold value is adopted to achieve capacity control, image quality is better in different payload length with this control. The experimental results reveal that the performance of our proposed method outperforms that proposed by FUJIYOSHI et al. For example, with the same imbedding capacity, the PSNR of our scheme is higher than FUJIYOSHI et al. by 3 dB. Furthermore, our algorithm provides higher embedding capacity compared with FUJIYOSHI et al.

Keywords—reversible watermarking; capacity control; predicted value; histogram shifting;

I. INTRODUCTION

Because of the popularity of computers and the development of Internet, digital images and video audio become more accessible. However digital media is easy to be copied and modified, when the digital media of intellectual property owner suffer from infringement, the owner is hard to proof his Intellectual Property. In order to proof the Intellectual Property, the owner can put digital signature on the digital media. The research of verify media intellectual property rights so that rightful ownership can be declared is watermarking technique.

Depending on whether the human eye can recognize, digital watermarking technology can be divided into two categories, one is the visible watermark technology, and the other is the invisible watermark technology. The visible watermark technology embeds watermark like a translucent loge into a media, its main purpose is to declare the ownership, to prevent illegal use. However, the disadvantage is that reduce the commercial value of the media. In addition, visible watermark easily be overwritten or removed via signal processing approach.

Currently watermark technology research and development, mainly focusing on the invisible watermark technology development. Invisible watermark can be divided into two categories according to the embedding data domains, namely, the frequency domain[14] and spatial domain[1], using the frequency domain watermark is Robustness, but computational intensity requires large amount than that embedding data in spatial domain, and the capacity is lower. On the contrary, the spatial domain watermark is high capacity, but the watermarked image is fragile. In this article, we mainly introduce the spatial domain watermark scheme.

Some traditional watermarking technique does not recover original image. However, data hiding in medical and military images[5], because of their specific requirements, sensitive in image quality. Therefore, reversible watermarking has been proposed to restore the image after watermark was extracted.

There are two important objectives for reversible watermarking techniques, the embedding capacity and the watermarked image quality. It is difficult to achieve these two objectives at the same time. In general, an improved technique embeds the same capacity with lower distortion or vice versa. Reversible watermarking techniques also have to solve the location map problem caused by overflow and underflow, location map is additional burden for watermarking techniques[7], many watermarking method try to reduce the size of location map. In this paper, we proposed a novel watermarking technique that don’t need location map anymore.

II. RELATED WORKS

Tian [1] proposed a reversible watermarking scheme by difference expansion (DE). He used the redundancy in the digital images to find extra storage space. His method divides image pixels into pairs, then watermark bits into each pairs through the difference expansion technique. Since Tian’s method embed those pairs of pixels that will not cause overflow and underflow problem. To check the positions of watermarked pairs, need to construct a location map. The location map size is half of cover image (0.5bpp), Compression can reduce the location map size, but still a large overhead. In addition, the embedding capacity is at most 0.5bpp in single round, a higher embedding capacity is achieved by multi-layer embedding.

Alattar [3] extended Tian’s method using difference expansion of vectors of adjacent pixels to watermarking bits. Location map used to identify different vectors. He simulated...
results using quad-based algorithm and his method has better performance than Tian[1], because the location map need 1/N bpp without compression.

Kamstra and Heijmans [11] used the variance of neighboring pixels to sorting, due to the high correlation of image pixels, improve the performance and reduce the location map size.

Ni et al. [4] developed a histogram shifting method. Firstly, scan the image to build a histogram of pixels. Next, find the pair of peak and zero points from the histogram. Embedding data into peak pixels and shifting others pixels to zero points. The advantages are low computational complexity and execution time. But the image capacity is limited by the number of the peak points.

Fallahpour et al. [5] introduce a highly efficient reversible data hiding system. Dividing the image into four or sixteen non-overlapping image tiles. Find the pair of peak and zero points of image histogram from each tile. The frequency of the peak point determined the embedding capacity. This applies in the special case like medical images. With the special properties of medical images, this method can result in 30%-200% capacity improvement.

Wang et al. [10] proposed a novel framework that design 2D reversible data hiding scheme, two prediction methods are used to compute different prediction error for one pixel then forming a planar, channel is defined to represent a slash. Peaks in each channel are selected to embed data. This method has better performance than conventional histogram shifting method and can be further extended into a multi-dimensional framework.

Thodi and Rodriguez [6] proposed a reversible watermarking method which is used prediction error expansion and histogram shifting scheme. This method more exploits the neighborhood pixels, the prediction errors are Laplacian distribution. The histogram shift technique improves the image capacity and distortion, and have the ability to embed more watermark into the zero prediction errors. Resulting in a better performance in capacity than with difference expansion (DE) [1].

Sachnev et al. [7] proposed a reversible watermarking algorithm using sorting and prediction. The scheme sorts the prediction errors base on their local variance separately. They can embed watermark bits according to local variance in ascending order. Because the local variance is proportional to the magnitude of prediction errors. The location map size is reduced which increasing the image capacity. The performance is better in low payload. Combine with double embedding scheme and prediction using a rhombus pattern, ideal image capacity reaches 1b/pixel. Comparing with the research of Thodi and Rodriguez [6], there are Significant improvements in both capacity and image quality from investigation of Sachnev et al. [7].

Lee et al. [12] proposed a reversible watermarking scheme based on prediction and difference expansion without using a location map. To solve overflow and underflow problems, the scheme shrinkage the histogram of image pixels by narrowing down the pixels value that close to value 0 and 255 and record those pixels position as a n-bit string regarded as payload length also embedding to image. The real image capacity is affected by the size of the n-bit string. The n-bit string don’t needed in extraction process, the advantage compared with location map is don’t worry about where to hide a n-bit string.

In 2010, FUJIYOSHI [9] proposes a reversible data hiding method, used no image-dependent parameter or any image-dependent location map. Instead, the scheme used a threshold parameter to limit embedding range, maintain a certain degree of image capacity but lack of capacity control. The scheme sorted the watermarked positions base on the maximum absolute deviation- like parameters. The sorting technique increase the watermark embedding capacity slightly. Their performance is better than [8] , the experiment result of our proposed method will compare it in section 4.

III. PROPOSED METHOD

The proposed method used prediction values and rhombus pattern prediction scheme from Sachnev et al. method [7], the rhombus pattern scheme divides the original image into two planes: the half plane1 and half plane2 [10], each pixel is surrounded by the pixels of the other half plane. The two half planes are independent, in this way, the embedding capacity can reach 1 bpp(bit per pixel) in a best situation. See Fig.1. Half plane1 is the brown grids, half plane2 is the white grids.

![Fig. 1. The half plane1 and half plane2](image1)

A. Predicted value and prediction error

![Fig. 2. The rhombus prediction pattern](image2)

This method used rhombus pattern prediction scheme with the half plane1 and half plane2. Half plane1 is the brown grids, half plane2 is the white grids. Pixel $I$ used to embedding watermarking, for the purpose of reversibility, a predicted
value is computed use four neighboring pixels (i.e., $I_{i,j+1}$, $I_{i,j-1}$, $I_{i+1,j}$, and $I_{i-1,j}$) in white grids, see Fig. 2. Predicted values is not changed after embedding process, at extraction step, the same predicted values are computed.

With center pixel $I_{i,j}$ in half plane1 and four neighboring pixels $I_{i,j+1}$, $I_{i,j-1}$, $I_{i+1,j}$ and $I_{i-1,j}$ in half plane2, the predicted value $p_{i,j}$ is computed as follow:

$$p_{i,j} = \frac{I_{i,j+1} + I_{i,j-1} + I_{i+1,j} + I_{i-1,j}}{4} \quad (1)$$

For the pixels $I_{i,j}$ in half plane1, predicted value is computed by four neighboring pixels in half plane2.

According to the predicted value $p_{i,j}$, prediction error $e_{i,j}$ is computed as follow:

$$e_{i,j} = I_{i,j} - p_{i,j} \quad (2)$$

In this method, prediction error $e_{i,j}$ expand to embedding message bit $b$

$$E_{i,j} = 2e_{i,j} + b \quad (3)$$

Where (3) combine with histogram shifting scheme shows in (10)(11).

The watermarked pixel $W_{i,j}$ is computed as follow

$$W_{i,j} = p_{i,j} + E_{i,j} \quad (4)$$

In extraction process, predicted value $p_{i,j}$ and the watermarking pixel value $W_{i,j}$ is used for extraction of embedded bit and recover original pixel value.

$$E_{i,j} = W_{i,j} - p_{i,j} \quad (5)$$

Then, the watermark bits $b$ can be extracted as follow:

$$b = E_{i,j} \mod 2 \quad (6)$$

The original prediction error is computed as follow:

$$e_{i,j} = \left\lfloor \frac{E_{i,j}}{2} \right\rfloor \quad (7)$$

The original pixel value is computed as follow:

$$I_{i,j} = e_{i,j} + p_{i,j} \quad (8)$$

### B. Predicted threshold value

Predicted threshold value $T$ is main idea of proposed method. According to predicted threshold value $T$ and predicted value $p_{i,j}$, the histogram shifting scheme modifies the pixels value to avoid overflow and underflow problems caused by embedding process. When the half plane1 is used to embed data, the half plane2 is used to compute predicted threshold value, equally, when the half plane2 is used to embed data, the half plane1 is used to compute predicted threshold value.

Predicted threshold $T$ in half plane1 (brown grids) is computed as follow:

$$T_{p} = \frac{t_1 + t_2 + \ldots + t_n}{n} \quad (9)$$

Where $T$ is predicted threshold value.

The predicted threshold value of half plane1 is computed using (9), where $t_1$ - $t_2$ - $\ldots$ - $t_n$ are image pixels value in all half plane1, see Fig. 3.

### C. Predicted threshold value and predicted value

When prediction errors are distributed based on predicted threshold value and predicted value, prediction errors on both sides of zero is not equal.

![Fig. 4. The histogram of prediction errors based on predicted threshold value](image)
The relationship between \( p_{i,j} \cdot T \) and are used in this method. See Fig.4, the two pictures are asymmetric. When \( p_{i,j} \geq T \), the prediction errors \( e_{i,j} \) have more positive values, see Fig.4 (b). When \( p_{i,j} < T \), the prediction errors \( e_{i,j} \) have more negative values, see Fig.4 (a).

Since \( T \) is made to calculate the average using half plane 1 or half plane 2 in the image. If randomly choose half pixels of the half plane to calculate the average value, this value will very close to \( T \). If randomly choose quarter pixels of the half plane to calculate the average value, this value will be close to \( T \). If only randomly choose four pixels of the half plane, this value may close to \( T \), but not so obvious.

![Histogram of Prediction Errors of Lena Image](image)

For each pixel \( I_{i,j} \), the corresponding \( p_{i,j} \) is high spatial correlation, because \( p_{i,j} \) is the average of \( I_{i,j} \) four neighboring pixels, affecting \( p_{i,j} \) more closer to \( T \) than \( I_{i,j} \), see Fig.5.

When \( p_{i,j} \geq T \), the predicted value \( p_{i,j} \) have more probability to smaller than original pixel \( I_{i,j} \). Similarly, when \( p_{i,j} < T \), the predicted value \( p_{i,j} \) have more probability to greater than \( I_{i,j} \), mainly, the prediction errors \( e_{i,j} \) have more negative value.

**TABLE1 Prediction Errors Counts**

<table>
<thead>
<tr>
<th></th>
<th>Lena</th>
<th>Airplane</th>
<th>Cameraman</th>
<th>Barbara</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p_{i,j} \geq T )</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e=0</td>
<td>77738</td>
<td>104819</td>
<td>89402</td>
<td>74846</td>
</tr>
<tr>
<td>e=0</td>
<td>43657</td>
<td>37719</td>
<td>17908</td>
<td>46164</td>
</tr>
<tr>
<td>( p_{i,j} &lt; T )</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e=0</td>
<td>56514</td>
<td>31849</td>
<td>23339</td>
<td>54937</td>
</tr>
<tr>
<td>e=0</td>
<td>57050</td>
<td>41099</td>
<td>37951</td>
<td>60829</td>
</tr>
</tbody>
</table>

Table1 represented four images in Fig.9, their number of prediction error. It results in line with our above described. This feature contributes less distortion when embedding combine histogram shifting scheme.

**D. Histogram shifting scheme**

The histogram shifting scheme combined prediction error is widely used in most methods. The prediction errors close to zero that expansion to embedding watermarking, then histogram shifting scheme modify others prediction errors to avoid overlapping problems. The histogram shifting scheme feature is embedding most data in one side from the zero position of prediction error histogram according to predicted threshold value \( T \).

As shown in Fig.6, the \( k \) is threshold value that the range of prediction errors \(- \lfloor k/2 \rfloor \leq e_{i,j} \leq k \) are used for embedding watermark when predicted value \( p_{i,j} < T \), in Fig.6(a). The range of prediction errors \(- k \leq e_{i,j} \) and \( e_{i,j} \leq \lfloor k/2 \rfloor \) are used for embedding watermark when predicted value \( p_{i,j} \geq T \), in Fig.6(b).

![Histogram of Prediction Errors](image)

The reason to choose \( \lfloor k/2 \rfloor \) and \( k \) for embedding range in histogram of prediction errors are achieve equally embedding capacity around zero both side. In this way, watermarking scheme have large embedding capacity, reduce the amount of movement of all of the pixels Histogram shifting scheme is presented as follow:

\[
E_{i,j} = \begin{cases} 
2e_{i,j} + b + \lfloor k/2 \rfloor, & \text{if } e_{i,j} \in [\lfloor k/2 \rfloor, k] \\
\phantom{2}e_{i,j} + k + \lfloor k/2 \rfloor + 1, & \text{if } e_{i,j} > k \\
\phantom{2}e_{i,j}, & \text{if } e_{i,j} < \lfloor -k/2 \rfloor 
\end{cases} 
\]  

(10)

Prediction errors \( e_{i,j} \) are shifted using (10) base on predicted threshold value \( T \) and prediction value \( p_{i,j} < T \) means prediction errors \( e_{i,j} \) are shifted to the right, the prediction errors \( e_{i,j} \) belong to the range \([- \lfloor k/2 \rfloor, k] \) are expanded to embedding a message bit \( b \), then add \( \lfloor k/2 \rfloor \) to avoid underflow cause by expansion. For the prediction errors \( e_{i,j} > k \) are shifted \( k + \lfloor k/2 \rfloor + 1 \) to avoid overlapping problem cause by expansion, for the prediction errors \( e_{i,j} < \lfloor -k/2 \rfloor \) are stay unchanged.

\[
E_{i,j} = \begin{cases} 
2e_{i,j} + b - \lfloor k/2 \rfloor - 1, & \text{if } e_{i,j} \in [-k, \lfloor k/2 \rfloor] \\
\phantom{2}e_{i,j} - k - \lfloor k/2 \rfloor - 1, & \text{if } e_{i,j} < -k \\
\phantom{2}e_{i,j}, & \text{if } e_{i,j} > \lfloor k/2 \rfloor 
\end{cases} 
\]  

(11)
Prediction error $e_{i,j}$ are shifted using (11) base on predicted threshold value $T$ and prediction value $p_{i,j}$ : $p_{i,j} \geq T$ means prediction error $e_{i,j}$ are shifted to the left.

The prediction errors $e_{i,j}$ belong to the range $[-k : [k/2]]$ are expanded to embedding a message bit $b$, then add $- [k / 2] - 1$ to avoid overflow cause by expansion. For the prediction errors $e_{i,j} < -k$ are shifted $-k - [k / 2] - 1$ to avoid overlapping problem cause by expansion, for the prediction errors $e_{i,j} > [k / 2]$ stay unchanged. In this condition, some images like Lena or Airplane is more smooth for the prediction error $e_{i,j}$ when $p_{i,j} \geq T$, as shown in Fig. 6(b). As the result, most of the data are embedding into higher pixel value.

Because high spatial correlation among predicted value $p_{i,j}$ and original pixel value $I_{i,j}$, embedding watermarking according to $p_{i,j}$ and $T$ can solve overflow and underflow effectively.

![Histogram](image)

(a) $p_{i,j} < T$
(b) $p_{i,j} \geq T$

Fig. 7. The bottom histogram of prediction errors of Cameraman image.

Combination of the above section, see Fig.7, the amount of large value prediction errors in the embedded direction is less, and more prediction errors are on the other side of the zero point, they don’t change value through embedding process. As the result, image’s quality is improved when those pixels shifting to solve the overlapping problem.

E. Modification of predicted value in half plane2

Firstly proposed method embed watermarking into half plane1, when half plane1 capacity is fully used, the half plane2 continue to embed watermarking. Because the Histogram shifting scheme (10), (11) shifts pixels value $[k/2]$ and $k + [k/2]$ , the high spatial correlation among neighboring pixels are influenced by the histogram shift scheme in half plane1. Lead to reduced image capacity and increased image distortion in half plane2. In order weaken the influence. The predicted value $p_{i,j}$ in half plane2 is computed as follow:

1) Compute predicted value $p_{i,j}$ using (1).

$$p_{i,j} = \begin{cases} p_{i,j} + \frac{k}{2}, & \text{if } p_{i,j} \geq T \\ p_{i,j} - \frac{k}{2}, & \text{if } p_{i,j} < T \end{cases}$$

See Fig.8, when using equation (12) in half plane2, the prediction error distribution is better.

![Histogram](image)

Compute using (12)  
Compute using (1) 
Fig. 8. $k=10$, the histogram of prediction errors of Lena image

F. Maximum of threshold value $k$

To ensure there is no overflow and underflow during imbedding process, limit threshold value $k$ is necessary. We can use (13) in half plane1 and half plane2 separately. The maximum threshold value $k$ is computed as follow:

$$k + [k / 2] + 1 \leq \min \begin{cases} 255 - I_{i,j}, & \text{if } p_{i,j} < T \\ I_{i,j} - 0, & \text{if } p_{i,j} \geq T \end{cases}$$

Table2 show the experimental results of maximum threshold value $k$, as long as the value $k$ don’t exceed those values for each image, the overflow and underflow problem not occur during embedding process.

<table>
<thead>
<tr>
<th>Image</th>
<th>maximum threshold $k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airplane</td>
<td>$k &lt; 20$</td>
</tr>
<tr>
<td>Lena</td>
<td>$k &lt; 20$</td>
</tr>
<tr>
<td>Barbara</td>
<td>$k &lt; 13$</td>
</tr>
<tr>
<td>Cameraman</td>
<td>$k &lt; 20$</td>
</tr>
</tbody>
</table>

G. Embedding and Extraction

The proposed method is used prediction error and histogram shift method to embedding data. For extracting watermarking
bits and recovering original image, threshold value $k$ and payload size $P_{l1}$ in half plane1 or payload size $P_{l2}$ in half plane2 should be known in extraction process. They can be transmitted through a covert channel or it may also be included in the image.

The LSB value of the first 47 image pixels value of final raw are replaced with threshold value $k$ (7 bits), payload size $P_{l1}$ (20bits) and payload size $P_{l2}$ (20bits). Original 47 LSB values are collected as payload embedding into image. The final raw of image is not used for embedding data. In the section, embedding steps and extraction steps illustrate in detailed.

I. Embedding process

In proposed method, prediction errors that are close to zero are underflow and overflow problems causing by histogram shifting scheme, we adopt an threshold value $k$ to control the embedding range of prediction errors, combined with predicted threshold value $T$, we use histogram shifting to avoid underflow and overflow problems. Firstly, the half plane1 is used for embed date, then the half plane2.

The embedding scheme is designed as follows:

1) The original LSB values of the first 47 image pixels value of final raw are collected as payload embedding into image.

2) Compute predicted threshold value $T$ using (9) in half plane2;

3) Compute the following value:
   - predicted value $p_{i,j}$ using (1);
   - prediction errors $e_{i,j}$ using (2);

4) The embedding histogram shifting scheme is designed as follow:

   $$E_{i,j} = \begin{cases} 2e_{i,j} + b + \lfloor k/2 \rfloor, & \text{if } e_{i,j} \in \left[ -\left\lfloor k/2 \right\rfloor, k \right] \text{ and } p_{i,j} < T \\ e_{i,j} + k + \left\lfloor k/2 \right\rfloor + 1, & \text{if } e_{i,j} > k \text{ and } p_{i,j} < T \end{cases}$$ (14)

   $$E_{i,j} = \begin{cases} 2e_{i,j} - k - \left\lfloor k/2 \right\rfloor - 1, & \text{if } e_{i,j} \in \left[ -k, \left\lfloor k/2 \right\rfloor \right] \text{ and } p_{i,j} \geq T \\ e_{i,j}, & \text{if } e_{i,j} < -k \text{ and } p_{i,j} \geq T \end{cases}$$ (15)

The watermarked image pixel $W_{i,j}$ is computed using (4)

5) Repeat step3 to step4 until the final pixel of half plane1;

6) For half plane2 repeat step2 to step4 again, but the predicted threshold value $T$ is computed using (9) in half plane1, predicted value using (12);

7) Threshold value $k$ and payload size $P_{l2}$ in half plane2 are embedded into the LSB value of the first 47 image pixels value of final raw.

After all message bits are embedded, the watermarked image is received

II. Extracting process

In the extracting process, we need threshold value $k$, predicted value $p_{i,j}$, predicted threshold value $T$, payload size $P_{l1}$ in half plane1 and payload size $P_{l2}$ in half plane2.

The prediction value $p_{i,j}$ and prediction threshold value $T$ can be computed from watermarked image. Threshold value $k$ and payload length $P_{l1}$ and $P_{l2}$ are extracted from the LSB value of the first 47 image pixels value of final raw

The extraction scheme is designed as follows: start with half plane2

1) Find threshold value $k$, payload length $P_{l1}$ and $P_{l2}$;

2) Compute predicted threshold value $T$ using (9) in half plane1;

3) Compute the following value:
   - prediction value $p_{i,j}$ using (12);

Expansion of prediction error $E_{i,j}$ using (5);

4) The watermarked bits can be extracted using (16) and $E_{i,j}$ must move $\lfloor k/2 \rfloor$ according to $p_{i,j}$ and $T$ before extracted;

   $$b = \begin{cases} (E_{i,j} - \lfloor k/2 \rfloor) \mod 2, & \text{if } E_{i,j} \in \left[ -\lfloor k/2 \rfloor, \lfloor k/2 \rfloor \right] \text{ and } p_{i,j} < T \\ (E_{i,j} + \lfloor k/2 \rfloor + 1) \mod 2, & \text{if } E_{i,j} \in \left[ -\lfloor k/2 \rfloor - 1, \lfloor k/2 \rfloor \right] \text{ and } p_{i,j} \geq T \end{cases}$$ (16)

5) The extraction histogram shifting scheme is designed as follow:

   $$e_{i,j} = \begin{cases} (E_{i,j} - k - \lfloor k/2 \rfloor)/2, & \text{if } E_{i,j} \in \left[ -\lfloor k/2 \rfloor, \lfloor k/2 \rfloor + 1 \right] \text{ and } p_{i,j} < T \\ E_{i,j} - k - \lfloor k/2 \rfloor - 1, & \text{if } E_{i,j} > 2k + \lfloor k/2 \rfloor + 1 \text{ and } p_{i,j} < T \end{cases}$$

   $$e_{i,j} = \begin{cases} (E_{i,j} - b - \lfloor k/2 \rfloor)/2, & \text{if } E_{i,j} \in \left[ -\lfloor k/2 \rfloor, \lfloor k/2 \rfloor + 1 \right] \text{ and } p_{i,j} > T \\ E_{i,j} - k - \lfloor k/2 \rfloor - 1, & \text{if } E_{i,j} < -2k - \lfloor k/2 \rfloor - 1 \text{ and } p_{i,j} > T \end{cases}$$ (17)

6) The original image pixel $I_{i,j}$ can be recovered using (8);

7) Repeat step3 to step6 until the final pixel of half plane2;
8) For half plane 1 repeat step 2 to step 6 again, but the predicted threshold value \( T \) is computed using (9) in half plane 2, predicted value \( p_{ij} \), using (1):

9) Recover the LSB value of the first 47 image pixels value of final raw;

IV. EXPERIMENTAL RESULTS

Four size of 512 \( \times \) 512 grayscale images are used for simulation, as shown in Fig. 9. The test watermark is a random binary string. We simulate the algorithm proposed by FUJIYOSHI et al. [9] to compare the performance with our scheme.

The definition of the PSNR (peak signal-to-noise ratio) is shown as follow:

\[
PSNR = 10 \times \log_{10} \left( \frac{255^2 \times m \times n}{\sum_{i=1}^{m} \sum_{j=1}^{n} \left[ I(i, j) - W(i, j) \right]^2} \right)
\]

(19)

Where \( m \), \( n \) are the size of the image.

Table 3 and Table 4 show the maximum capacity and corresponding PSNR(dB) in different threshold value \( k \) for four images in Fig. 9. See Table 5, the simulation result show that the proposed method is perform better than FUJIYOSHI et al. [9] in image capacity.

Table 3: Experimental Results of Different Value for Images Airplane and Lena

<table>
<thead>
<tr>
<th>Threshold ( k )</th>
<th>Airplane</th>
<th>Lena</th>
</tr>
</thead>
<tbody>
<tr>
<td>( k = 0 )</td>
<td>46377</td>
<td>33449</td>
</tr>
<tr>
<td>( k = 1 )</td>
<td>70453</td>
<td>59607</td>
</tr>
<tr>
<td>( k = 2 )</td>
<td>138364</td>
<td>111925</td>
</tr>
<tr>
<td>( k = 3 )</td>
<td>146688</td>
<td>125580</td>
</tr>
<tr>
<td>( k = 4 )</td>
<td>184610</td>
<td>162027</td>
</tr>
</tbody>
</table>

Table 4: Experimental Results of Different Value for Images Barbara and Cameraman

<table>
<thead>
<tr>
<th>Threshold ( k )</th>
<th>Barbara</th>
<th>Cameraman</th>
</tr>
</thead>
<tbody>
<tr>
<td>( k = 0 )</td>
<td>25571</td>
<td>93725</td>
</tr>
<tr>
<td>( k = 1 )</td>
<td>46157</td>
<td>114907</td>
</tr>
<tr>
<td>( k = 2 )</td>
<td>87689</td>
<td>198843</td>
</tr>
<tr>
<td>( k = 3 )</td>
<td>99106</td>
<td>201689</td>
</tr>
<tr>
<td>( k = 4 )</td>
<td>128908</td>
<td>227356</td>
</tr>
</tbody>
</table>

Table 5: Experimental Results of Capacity Comparision

<table>
<thead>
<tr>
<th></th>
<th>Proposed</th>
<th>FUJIYOSHI [9]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airplane</td>
<td>63932</td>
<td>223578</td>
</tr>
<tr>
<td>Lena</td>
<td>62366</td>
<td>212230</td>
</tr>
<tr>
<td>Barbara</td>
<td>32980</td>
<td>174432</td>
</tr>
<tr>
<td>Cameraman</td>
<td>10353</td>
<td>244733</td>
</tr>
</tbody>
</table>

With the same embedding capacity, the PSNR of our scheme is higher than FUJIYOSHI et al. [9]. See Fig. 10, the dramatic decline caused by the threshold value \( k \), since when \( k = 0 \), the maximum capacity in Airplane is 46377, see Table 3, in order to get more embedding capacity, the threshold value is changed to \( k = 1 \), the maximum capacity is reach to 70453, this is same as Fig. 11, Fig. 12, Fig. 13.
In this paper, we have proposed a reversible watermarking method without using a location map to solve overflow and underflow problems during embedding process and extracting process. The concept is to find predicted threshold value and histogram shifting scheme exploit the predicted threshold to embedding data. We find a relationship between predicted threshold value and prediction error. This feature helps us achieve the paper goal of location map free watermarking method with better image quality. Capacity control achieves better image quality with different payload length that other location free method is failed to do. The image capacity and quality are great improved in the location map free scheme.

ACKNOWLEDGEMENT
This paper is the partial result of project NSC100-2632-E-366-001-MY3. We would like to thank the supporting of Ministry of Science and Technology, R.O.C., also thank the value advices of Miss Yu-Hsiu Huang and Prof. Chun-Hung Richard Lin.

References