High Capacity Data Hiding based on Predictor and Histogram Modification  
Hui-Yu Huang, Shih-Hsu Chang

Abstract—In this paper, we propose a high capacity image hiding technology based on pixel prediction and the difference of modified histogram. This approach is used the pixel prediction and the difference of modified histogram to calculate the best embedding point. This approach can improve the predictive accuracy and increase the pixel difference to advance the hiding capacity. We also use the histogram modification to prevent the overflow and underflow. Experimental results demonstrate that our proposed method within the same average hiding capacity can still keep high quality of image and low distortion.

Keywords—data hiding, predictor

I. INTRODUCTION

Data hiding can serve as the process of embedding secret data into a cover media (host signal). The image for carrying data is called a cover image, and the image carrying the embedded information is called a stego-image. Reversible data hiding scheme not only embeds data into cover images, but also restores the original image from the stego-image after the embedded data have been extracted. If the original image cannot recover, this will cause permanent distortion of original image after the secret messages have been extracted, and thus the image quality is degraded. However, distortion for some applications is undesirable, such as medical image. For data hiding, many studies have been reported. The most common methods are least-significant bits (LSB) [1, 2], difference expansion [3,4,5], and histogram-based techniques [6-11] in the spatial domain. In addition, combining the prediction mechanism to work a reversible data hiding have been presented [12,13]. Celik et al. [1] presented a generalized-LSB modification method to improve the lossless data hiding capacity. Hu et al. [3] proposed a difference expansion (DE) based on integer Harr wavelet transform, which utilized the horizontal as well as vertical difference images for data hiding. Tain [4] presented a high capacity, low distortion reversible data embedding algorithm used DE, authors explored the redundancies in digital image to achieve high capacity and keep the distortion low.

Ni et al. [6] proposed a histogram-based method to achieve data hiding, which used the zero and peak points of an image histogram to embed message and recovered the original image after extracting the embedded data. Based on the histogram modification of pixel differences to design the reversible data hiding technique presented by [7,8], these schemes used pixel differences between original gray value and prediction value to construct the histogram of difference. Tsai et al. [9] proposed a data hiding based histogram shifting for medical images.

The prediction technique was used to estimate the similarity of neighboring pixels and the residual histogram of the predicted errors of the host image was used to hide the secret data. They used the overlapping between peak and zero pairs to increase the capacity. Wu et al. [12] proposed an embedding secret data method into compression codes during the lossless image compression based on predictive coding. In this study, it is mainly using the error values via predictive coding stage to hide the secret data into a host image and extract those of data by referring to a hiding-tree. In this paper, we propose a high capacity data hiding method based on pixel prediction and modification of prediction errors (PPMPE) which can outperform the prior works not only in terms of payload but also in terms of stego-image quality. The rest of this paper is organized as follows. Section 2 presents the proposed method. Experimental results are shown in Section 3. Finally, conclusions are given in Section 4.

II. PROPOSED METHOD

A. Median Edge Detection Predictor

The embedding process of proposed PPMPE based on pixel prediction and modification of prediction errors. JPEG-LS uses the median edge detection predictor (MED) to obtain good decorrelation [14]. The MED is used to predict the gray value of pixel in this paper. It is based on the causal neighboring pixels shown in Figure 1, where x denotes the current pixel, a, b, and c are neighboring pixels in the relative positions. The pixel x is predicted by the MED predictor represented as:

\[
x = MED(x) = \begin{cases} 
\min(a,b), & \text{if } c \geq \max(a,b), \\
\max(a,b), & \text{if } c \leq \min(a,b), \\
\frac{a+b-c}{2}, & \text{otherwise}
\end{cases}
\]  (1)

where \( \hat{x} \) is the predicted value of x. If a, b, or c lies outside of image, here, it is set to be zero.

The prediction error \( e \) between pixel x and its prediction value \( \hat{x} \) is defined as \( e = x - \hat{x} \). For good predictor, the error histogram sharply distributes around zero point with two-sided exponential decay for most natural images. The prediction difference \( d \) between pixel x and its prediction value \( \hat{x} \) is defined as \( d = |e| = |x - \hat{x}| \). The difference histogram should highly and sharply distribute around zero point with one-sided exponential decay for most natural images. The proposed PPMPE embeds the message into the pixel that its difference is the peak point of difference histogram. This operation significantly increases the number of embeddable pixel such that PPMPE has high hiding capacity.

![Fig. 1 The a, b, and c are the neighboring pixels of current pixel x in JPEG-LS](image-url)
B. Embedding Process

Assume that the secret hiding message $M$ has $J$ symbols and $M = \{m_1, m_2, \ldots, m_J\}$, where $m_j \in [0, T]$ is the data hiding in the pixel. The value $m_j$ can be represented by $\log_2(T+1)$ bits. The amount of hiding message is $J \times \log_2(T+1)$ bits. The $T$ is a predefined threshold value which determines the amount of hiding data and the image quality of stego-image $Y$. For example, if $T=1$, the pixel can be embedded one bit of $m_j$. Larger $T$ resulted in high amount of hidden data but lower image quality of stego-image. For an $N$-pixel 8-bit grayscale cover image $X$ with a pixel value $x_i$, where $x_i$ is the grayscale value of the $i$th pixels, $0 \leq x_i \leq N-1$, $N = W \times H$, and $W \times H$ is the size of image. Details of the embedding procedures are described as follows.

Step 1: Calculate the histogram differences. Let $His(i), i = 1, \ldots, N-1$, be scan the cover image $X$ by raster-scan., next to compute the pixel differences $d_i$ between pixel $x_i$ and its prediction value $\hat{x}_i$, where $d_i = |x_i - \hat{x}_i|$ and $\hat{x}_i = MED(x_i)$ in Eq. (1).

Step 2: Determine the peak position $p$ from $His$: The peak point $p$ is the argument for which the value attains its maximum value of histogram $His$, where

$$p = \arg \max_{d=0\text{ to } 255} His(d). \tag{2}$$

The corresponding maximum value of $His$ is defined as $N_p$, where

$$N_p = His(p). \tag{3}$$

Step 3: Data hiding: variables $j$ and $k$ are initialized to be zero. Scan the whole image in the same raster-scan as in Step 1. For each $i$th pixel $x_i$ does the following steps, where $i=0$ to $N-1$.  

(i) If $d_i < p$, let $y_i = x_i$.

(ii) If $d_i > p$, shift $x_i$ by $T$ units according to the rule:

$$y_i = \begin{cases} x_i + T, & \text{if } x_i \geq \hat{x}_i, \\ x_i - T, & \text{if } x_i < \hat{x}_i, \end{cases} \tag{4}$$

(iii) If $d_i = p$, the current secret data $m_j$ is embedding into $x_i$ according to the rule:

$$y_i = \begin{cases} x_i + m_j, & \text{if } x_i \geq \hat{x}_i, \\ x_i - m_j, & \text{if } x_i < \hat{x}_i. \end{cases} \tag{5}$$

Step 4: Prevent overflow/underflow process: After modification of a pixel, the pixel value maybe beyond the grayscale level. To prevent overflow and underflow, we use additional array $R$ to record the overflow and underflow. The current pixel $y_i$ is modified by the following rule

$$y_i = \begin{cases} 0, & \text{if } y_i \leq 0, r_k = y_i, \\ 255, & \text{if } y_i \geq 255, r_k = y_i - 255. \end{cases} \tag{6}$$

The $r_k$ recorded the overflow or underflow, $r_k \in [0, T]$.

Step 5: Go to Step 3 until the hiding message is completely embedded.

Based on the above steps, the secret data can be successfully hidden into the host image. Next, let $K=k$ and output the stego-image $Y$ and the overflow/underflow information record as $R = \{r_0, r_1, \ldots, r_{K-1}\}$. Because of $r_k \in [0, T]$, it needs $K \times \log_2(T+1)$ bits to store the overhead information. Let $|O|$ be the total bits to store the overflow/underflow information, where $|O| = K \times \log_2(T+1)$. If the pixel difference is equal to $p$ in Step (iii), the secret data can be embedded. Hence, the maximum hiding capacity is $N_p \times \log_2(T+1)$. The additional information $T$, $p$, and $J$ are needed to extract the embedded data from the stego-image. Assume it needs 8 bits to store the value of $T$, 8 bits to store the value of $p$, and 20 bits to store the value of $J$. These 36 bits can be stored in the header of the image file. Hence, the real hiding capacity (pure payload) can be defined as

$$mPay_{\text{pur}} = N_p \times \log_2(T+1) - |O| - 36, \tag{7}$$

where $J$ is smaller than or equal to $N_p$. The maximum value of pure payload is

$$mPay_{\text{pur}} = N_p \times \log_2(T+1) - |O| - 36. \tag{8}$$

C. Extraction Process

After the receiver receives a stego-image $Y$, $T$, $p$, and the overflow/underflow information $R$, the receiver can extract the hiding message from the stego-image $Y$ and losslessly recover the cover image $X$. Details of the extraction procedures are described in the following.

Assume that $R$ is the overflow/underflow information, where $R = \{r_0, r_1, \ldots, r_{K-1}\}$. The variables $j$ and $k$ are initialized to be zero. Let $T_p = T + p$. The constant $T_p$ is used in algorithm. Scan the whole image in the same raster-scan to get $\{y_i | 0 \leq i < N\}$. For each $i$th pixel $y_i$ does the following steps, where $i=0$ to $N-1$, $N=WH$, and $WH$ is the size of image.

Step 1: Recover overflow/underflow process: The pixel $y_i$ may be modified by the process in Eq. (6). It can be recovered by the following rule.

$$y_i = \begin{cases} -r_k, & \text{if } y_i = 0, \\ 255 + r_k, & \text{if } y_i = 255, \end{cases} \tag{9}$$

where $r_k$ is the current overflow or underflow value and $r_k \in [0, T]$. 

For each test, the prediction difference $d_j^p$ between pixel $x_j$ and the prediction value $\hat{x}_j$ is $d_j^p = |y_j - \hat{x}_j|$, where $\hat{x}_j = MED(x_j)$ in Eq. (1).

**Step 3:** Extract message $m_j$ and recover the pixel $x_j$ of cover image $X$:

(i) If $d_j^p < p$, let $x_j = y_j$.

(ii) If $d_j^p > T_p$, shift $y_j$ by $T$ units according to the rule:

$$x_j = \begin{cases} y_j - T, & \text{if } y_j \geq \hat{x}_j, \\ y_j + T, & \text{if } y_j < \hat{x}_j. \end{cases}$$

(iii) If $p \leq d_j^p \leq T_p$, the secret data $m_j$ is extracted and $x_j$ is recovered defined as

$$M_j = d_j^p - p,$$

$$x_j = \begin{cases} y_j - m_j, & \text{if } y_j \geq \hat{x}_j, \\ y_j + m_j, & \text{if } y_j < \hat{x}_j. \end{cases}$$

**Step 4:** Go to Step 1 until the hidden messages are completely extracted.

### III. RESULTS

In our experiments, the test image is of size 512×512. Given a predefined value $T$, the hidden data $\{m_j\}$ is generated by a random number generator such that $0 \leq m_j \leq T$. In addition, in order to evaluate our system performance, we use the peak signal-to-noise ratio (PSNR) to measure the quality of stego-image expressed as:

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

$$MSE = \left( \frac{1}{H \times W} \right) \sum_{i=0}^{W-1} \sum_{j=0}^{H-1} (x_{ij} - \hat{x}_{ij})^2,$$

where $MSE$ is mean-square error, and variable $W$ and $H$ denote the width and height of an image. $x_{ij}$ and $y_{ij}$ present the cover image and stego-image at $i$th pixel, respectively.

In our experiments, the results will present the average pure-payload and PSNR of 100 runs for data hiding in each cover image.

The $N_p$ is the corresponding number of peak point $p$. The maximum hiding capacity is $N_p \times \log_2(T + 1)$. For each test image, the peak point $p$ is 1 and the $N_p$ with the different $T$ is shown in Tables I and II. In Table I, all stego-images except “Pepper” have no overflow or underflow condition, i.e., $|O| = 0$. The length $|O|$ of “Pepper” is 15 bits. The average pure payload $\text{mPurPay}$ and average hiding bit rate are about 51013 bits and 0.1946 (bpp), respectively. For smooth image, like as “Lena”, there are strong spatial correlations among pixels. Hence, their values of $N_p$ are higher than those of non-smooth images. Images with high textured areas, such as “Mandrill”, there are low spatial correlation and low values of $N_p$. For Tables I and II, it is clear that the lower $T$ resulted the larger image quality of stego-image. Hence, large hiding capacities can be obtained by repeating the proposed data hiding process for the smaller $T$. Table III shows the pure payload versus the PSNR of the proposed algorithm for $T=1$.

The comparison of PSNRs of Ni et al. method [6], MPE [13], and the proposed PPMPE under the same pure payloads for test images is shown in Table IV. When the average pure payload $\text{PurPay}$ is about 9772 bits, the average PSNR of PPMPE is 56.89dB which is higher than Ni et al. method and MPE method, respectively. Hence, our proposed method is superior to those of Ni et al. method and MPE method. Figure 2 shows the stego-images with the different pure payload for “Lena” image.

### IV. CONCLUSIONS

In this paper, we have presented a high capacity image hiding based on the difference of modified histogram and prediction. Experimental results demonstrate that our approach can gain the capacity of data embedding and preserve high quality of images.

### REFERENCES


### TABLE I

| Test images | \( p \) | \( N_p \) | \(|O|\) (bits) | \( m\text{PurPay} \) (bits) | Bit rate (bpp) | PSNR |
|-------------|-------|-------|-------|----------------|-------------|------|
| Lena        | 1     | 67411 | 0     | 67375          | 0.2570      | 49.63 |
| Mandrill    | 1     | 23949 | 0     | 23913          | 0.0912      | 48.60 |
| Pepper      | 1     | 61803 | 15    | 61752          | 0.2356      | 49.43 |
| Average     |       | 51054 | 51013 | 0.1946         | 49.22       |

### TABLE II

| Test images | \( p \) | \( N_p \) | \(|O|\) (bits) | \( m\text{PurPay} \) (bits) | Bit rate (bpp) | PSNR |
|-------------|-------|-------|-------|----------------|-------------|------|
| Lena        | 1     | 67411 | 0     | 134786         | 0.5142      | 49.26 |
| Mandrill    | 1     | 23949 | 0     | 47862          | 0.1826      | 39.11 |
| Pepper      | 1     | 61803 | 214   | 123356         | 0.4706      | 40.05 |
| Average     |       | 51054 | 51013 | 0.1946         | 42.81       |

### TABLE III

**PURE PAYLOAD VERSUS THE PSNR (dB) OF THE PROPOSED ALGORITHM BY REPEATING THE PROPOSED DATA HIDING PROCESS FOR T=1.**

<table>
<thead>
<tr>
<th>( m\text{PurPay} ) (bits)</th>
<th>bpp</th>
<th>Lenna</th>
<th>Mandrill</th>
<th>Pepper</th>
</tr>
</thead>
<tbody>
<tr>
<td>78645</td>
<td>0.30</td>
<td>47.54</td>
<td>37.54</td>
<td>46.15</td>
</tr>
<tr>
<td>104859</td>
<td>0.40</td>
<td>44.34</td>
<td>34.23</td>
<td>43.37</td>
</tr>
<tr>
<td>131074</td>
<td>0.50</td>
<td>41.77</td>
<td>31.46</td>
<td>38.40</td>
</tr>
<tr>
<td>157288</td>
<td>0.60</td>
<td>39.70</td>
<td>29.07</td>
<td>35.68</td>
</tr>
<tr>
<td>209717</td>
<td>0.80</td>
<td>36.12</td>
<td>24.93</td>
<td>31.80</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Test images</th>
<th>PSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lenna</td>
<td>43.74</td>
</tr>
<tr>
<td>Mandrill</td>
<td>40.65</td>
</tr>
<tr>
<td>Pepper</td>
<td>38.58</td>
</tr>
<tr>
<td>Average</td>
<td>37.88</td>
</tr>
</tbody>
</table>

### TABLE IV

**COMPARISON OF PSNRS (dB) OF Ni’s METHOD [6], MPE [13], AND THE PROPOSED PPMPE UNDER THE SAME PURE PAYLOAD FOR T=1**

<table>
<thead>
<tr>
<th>Test images</th>
<th>( m\text{PurPay} ) (bits)</th>
<th>Bit rate</th>
<th>Ni’s PSNR</th>
<th>MPE PSNR</th>
<th>PPMPE PSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>6657</td>
<td>0.0254</td>
<td>48.19</td>
<td>61.80</td>
<td>61.15</td>
</tr>
<tr>
<td>Mandrill</td>
<td>5685</td>
<td>0.0217</td>
<td>48.22</td>
<td>51.79</td>
<td>53.01</td>
</tr>
<tr>
<td>Jet</td>
<td>16974</td>
<td>0.0648</td>
<td>48.27</td>
<td>55.14</td>
<td>56.52</td>
</tr>
<tr>
<td>Average</td>
<td>9772</td>
<td>0.0373</td>
<td>48.23</td>
<td>56.24</td>
<td>56.89</td>
</tr>
</tbody>
</table>

Fig. 2 The stego-images of “Lena” at the different pure payload. (a) 41.77 dB embedded with 0.5 bpp, (b) 37.92 dB embedded with 0.70 bpp, (c) 32.61 dB embedded with 1.0 bpp, and (d) 25.22 dB embedded with 1.50 bpp.