Reversible data hiding for high quality images using modification of prediction errors

Wien Hong\textsuperscript{a,}\textsuperscript{*}, Tung-Shou Chent\textsuperscript{b}, Chih-Wei Shiu\textsuperscript{a}

\textsuperscript{a}Department of Information Management, Yu Da College of Business, Miaoli 361, Taiwan
\textsuperscript{b}Graduate School of Computer Science and Information Technology, National Taichung Institute of Technology, Taichung 404, Taiwan

A R T I C L E   I N F O

Article history:
Received 24 September 2008
Received in revised form 19 March 2009
Accepted 16 May 2009
Available online xxxx

Keywords:
Reversible data hiding
Histogram-shifting
Error modification

A B S T R A C T

In this paper, a reversible data hiding scheme based on modification of prediction errors (MPE) is proposed. For the existing histogram-shifting based reversible data hiding techniques, though the distortion caused by embedding is low, the embedding capacity is limited by the frequency of the most frequent pixel. To remedy this problem, the proposed method modifies the histogram of prediction errors to prepare vacant positions for data embedding. The PSNR of the stego image produced by MPE is guaranteed to be above 48 dB, while the embedding capacity is, on average, almost five times higher than that of the well-known Ni et al. techniques with the same PSNR. Besides, MPE not only has the capability to control the capacity-PSNR, where fewer data bits need less error modification, but also can be applied to images with flat histogram. Experimental results indicate that MPE, which innovatively exploits the modification of prediction errors, outperforms the prior works not only in terms of larger payload, but also in terms of stego image quality.

© 2009 Elsevier Inc. All rights reserved.

1. Introduction

During transmission, if the digital media itself shows artifact of hiding effect, the intentional opponent may doubt the digital media carries secret messages. Therefore, an embedded digital should maintain an imperceptible quality to keep the embedded media from drawing attention (Wang et al., 2008). In general, the major concerns of data hiding techniques are the embedding capacity and imperceptibility. These two concerns are different from the techniques of digital watermarking. The purpose of digital watermarking is to protect the ownership of a digital media. Since a digital media is easily tampered or modified, a watermarking technique must be designed to have the capability against some common signal processing operations such as noise or lossy compression. The retrieved watermarks may not be exactly the same as the original ones; however, the ownership can still be verified according to the retrieved watermarks. On the other hand, a data hiding technique must extract embedded data without losing any bit. Therefore, the requirements of the robustness against common signal processing operations, or the ability to prevent bit errors occurring during storage and transmission, are not as emphasized as in digital watermarking technique (Yu et al., 2007; Wang and Wang, 2004).

Any existing digital media such as audios, videos, and digital images can be used as carriers. The digital image is often used as carrier since it is delivered the most over the Internet. It is important that the image with embedded data should not arouse any suspicious. The image for carrying data is called a cover image, and the image carrying the embedded information is called a stego image. When the information is embedded into images, the pixel values in the image will be changed, and thus the image quality is degraded. Since the altered pixels cannot be recovered into their original state after the secret messages has been extracted, permanent distortion will occur. Distortion for some applications is unacceptable. For example, a distorted chest X-ray image could result in an incorrect medical diagnosis. In these applications, techniques for reversible data hiding are necessary.

Reversible data hiding is a technique that not only embeds data into cover images, but also restores the original images from the stego images after the embedded data have been extracted (Alattar, 2004). Since the original cover image must be recovered after extracting the secret message, this requirement imposed on reversible data hiding technique a penalty of lower payloads, larger distortion and higher computational cost-in comparison to those non-reversible techniques. Despite these drawbacks, the number of reversible data hiding techniques proposed in literature has increased recently; suggesting the increasing needs in this field. The reversible data hiding technique developed at the early stage mainly relies on lossless compression technique. In 2002, Fridrich et al. proposed an R-S scheme for which compressed message bits were reversibly embedded in the status of group of pixels (Fridrich...
et al., 2002). Gelik et al. devised a generalized least significant bit (G-LSB) technique to increase the payload of Fridrich et al. method (Celik et al., 2002). Awrangjib and Kankanhalli proposed a reversible scheme that embedded compressed data into the original image with the consideration of the human visual system to minimize the perceptible artifacts (Awrangjib et al., 2005). The methods mentioned above involved approaches that embedded losslessly compressed features extracted from the original cover image.

The other type of reversible data hiding methods can be categorized as expansion–embedding based techniques. A common feature of these techniques is using a decorrelation operator to create features with small magnitudes. Data can be embedded by expanding these features to create vacant space into which message bits are embedded. The first approach under this category was proposed by Tian (Tian, 2003), and extended by many recent researches (Alattar, 2004; Kamstra and Heijmans, 2005; Thodi and Rodriguez, 2007; Kim et al., 2008). Expansion embedding based approaches usually suffer from undesirable distortion when the values of features are large. Therefore, this method might not be suitable for applications where higher image quality is demanded. Another category of reversible data hiding can be classified as histogram-shifting based techniques. In these techniques, a histogram of feature elements is created, and data can be embedded by shifting histogram bins. The well-known technique proposed by Ni et al. in 2006 is of this category (Ni et al., 2006). Some other histogram-shifting based reversible data hiding techniques can be found in Hwang et al. (2006) and Fallahpour and Sedaaghi (2007). Unfortunately, the capacity of histogram-shifting based techniques is low and highly dependent on the histogram distribution of the cover image. In general, the higher the peak of image histogram, the more the embedding capacity is. In addition to the techniques mentioned above, Coltuc introduced a very different approach to reversibly embedded data based on simple transforms with low mathematical complexity (Coltuc, 2007). These emerging remarkable reversible techniques suggested the increasing attention of the reversible data hiding techniques have been receiving.

Many applications demand high quality images, such as medical or military images. The well-known reversible data hiding method proposed by Ni et al. can produce relatively high quality stego image ($\geq 48.13$ dB); however, the embedding capacity is low and is limited by the distribution of image histogram. Several studies were based on Ni et al.’s method and were either trying to enhance the image quality further or to increase the embedding capacity (Hwang et al., 2006; Fallahpour and Sedaaghi, 2007; Xuan et al., 2007). For example, Xuan et al. proposed a novel optimum histogram pair based reversible data hiding technique using integer wavelet transform and adaptive histogram modification with excellent performance (Xuan et al., 2007). However, their method involved integer wavelet transform (IWT) and optimal parameters selection, the computational cost is higher than Ni et al. method.

In this paper, we proposed a new reversible data hiding technique based on modification of prediction errors (MPE). Since the histograms in the domain of prediction errors are sharply distributed, the embedding capacity is higher than that of traditional histogram-shifting method for the same image quality. Besides, MPE only modifies less error values for embedding fewer data bits; therefore, a high quality stego image can be obtained.

Detailed advantages of MPE over Ni et al. method will be discussed in Section 4. The rest of this paper is organized as follows. In Section 2, Ni et al.’s histogram-shifting technique for reversible data hiding will be described. The MED predictor and the proposed method are presented in Section 3, followed by the experimental results in Section 4. Conclusions from the experimental results are addressed in Section 5.

2. Reviews of the histogram-shifting technique

The histogram-shifting technique proposed by Ni et al. (2006) reversibly embeds data into images by shifting the histogram bins. In their method, one grayscale level will be changed at most in every pixel; therefore, an acceptable stego image quality can be obtained ($\text{PSNR} \geq 48.13$ dB). The embedding procedure of Ni et al. technique is listed below:

1. Obtain the histogram $h(x)$, $x \in [0, 255]$ of the 8-bit cover image.
2. Find the maximum value $h(\alpha)$ and the minimum value $h(\beta)$ of $h(x)$, where $\alpha, \beta \in [0, 255]$. The parameter $\alpha$ is called the peak point and $\beta$ is called the minimum point. If $h(\beta) = 0$, then $\beta$ is called the zero point. Without loss of generality, we assume $\alpha < \beta$.
3. If the minimum value $h(\beta) > 0$, then record all the positions of the pixels with gray level $\beta$ in array $L_{\beta}$, and then set $h(\beta) = 0$.
4. The whole image is scanned in sequential order, that is row-by-row and from top to bottom. Then, shift histogram $h(x)$ with $x \in (\alpha, \beta)$ to the right by one unit. This procedure indicates that all pixel values satisfying $x \in (\alpha, \beta)$ are added by one.
5. Scan the whole image again and then embed secret bits as well as $L_{\beta}$ (if any). If the pixel value is $x$, and the bit to-be-embedded is 1, then the pixel value is set to $x + 1$. If the to-be-embedded bit is 0, the pixel value remains unchanged.

In Ni et al. method, if $h(\beta) > 0$, we should embed additional recovery information $L_{\beta}$ into the cover image. Therefore, the pure embedding capacity $EC$ of this method is given by

$$EC = h(\alpha) - |L_{\beta}|,$$

where $|L_{\beta}|$ is the number of bits required to store $L_{\beta}$. The following steps are the procedures for extracting the secret data and restoring the stego image to its original:

1. Scan the stego image in the same sequential order as we used in the embedding phase. If a pixel value $x + 1$ is encountered, then a bit 1 is extracted. If a pixel value $x$ is encountered, then a bit 0 is extracted.
2. Scan the stego image again. If the pixel value $x \in (\alpha, \beta]$, then the pixel value $x$ is subtracted by 1.
3. If there is recovery information $L_{\beta}$ found in the extracted data, set the pixel values to $\beta$ if the location of this pixel is found in $L_{\beta}$. In this way, the original image can be recovered without any distortion.

The algorithm described above selects only a pair $\alpha$ and $\beta$ for hiding information. In fact, one can choose more pairs of maximum and minimum points to increase the embedding capacity.

The main emphasis of Ni et al. method is on histogram generation, maximum and minimum point determination, pixel scanning, and pixel value addition or subtraction. The scheme is in spatial domain and computational cost is low. In Ni et al. method, the whole image is scanned three times in the embedding phase, thus the total computational complexity is $O(3k \times M \times N)$ for cover image of size $M \times N$ and $k$ represents number of pair.

3. The proposed method

Histogram-shifting is a technique for embedding data into histogram bins by shifting the histograms of the feature elements to prepare vacant positions for embedding. The occurrence of the
most frequent feature elements determines the embedding capacity. These most frequent feature elements are called the embeddable elements. The distortion resulting from histogram-shifting embedding primarily depends on the number of feature elements that are shifted. Therefore, it is desirable to increase the number of embeddable elements while minimizing the number of feature elements that has to be shifted.

Ni et al. method can be regarded as a classical histogram-shifting technique, where the features are the pixel values in the spatial domain. The key issue of their method is that whether enough embeddable elements exist in the histogram for data embedding or not. An extreme example in which Ni et al. method does not work is for an image with exactly flat histogram. Besides, no matter how small the amount of to-be-embedded data is, their method has to evacuate histogram bins beforehand, and thus the stego image quality is decreased significantly when embedding small amount of data. The performance of Ni et al. histogram-shifting technique can be improved by generating features that has a sharply distributed histogram to maximize the embeddable elements. Besides, the embedding algorithm should have the ability to control capacity–PSNR relationship, since embedding fewer data bits does not need to evacuate histogram bins.

In this section, we propose a novel reversible data hiding technique based on modification of prediction errors (MPE). Instead of using feature elements in the spatial domain, we use a predictor to create feature elements in the domain of prediction errors. MPE adopts the idea of histogram-shifting technique to embed data in the domain of prediction errors. In fact, MPE needs not to actually generate the error histogram because some characteristics of the error histogram for most natural images, for example, sharply distributed at zero with two-sided exponential decay, are well-known, as shown in Fig. 1. Note that though the distribution of the histogram of prediction errors for the image Baboon is not as sharp as the image Lena, the histogram still shows a peak at zero with two-sided decay.

One of the advantages of this approach is that the prediction operation significantly increases the number of embeddable features in the feature set, and subsequently increases the embedding capacity. Second, because we now have a peaked histogram centered at prediction value equal to zero, the position of the peak point that maximizing the embedding capacity is already known. Therefore, there is no need to scan the image again for searching the peak and minimum point. Third, no evacuation of histogram bins is needed beforehand, as Ni et al. technique has to do. MPE only modifies fewer prediction errors for embedding fewer amounts of data; therefore, rough capacity–PSNR control can be achieved. And fourth, the embedding procedure only involves simple calculations such as addition and subtraction, and the whole image is scanned just one time for the embedding process. Therefore, for two pairs of peak and zero, the computational complexity is $O(2 \times M \times N)$, which is slightly efficient than Ni et al. algorithm. The pairing rules will be addressed in the following subsection.

### 3.1. Embedding algorithm

The embedding process of MPE involves calculating the prediction errors from the neighborhood of a given pixel, and then embedding the message bits in the modified prediction errors. The median edge detection (MED) predictor (Gonzalez and Woods, 2002; Salomon, 2000) is used in MPE to predict pixel values. MED predictor is based on the values of context pixels $a$, $b$, and $c$ to predict pixel $x$, as shown in Fig. 2:

$$
\hat{x} = \begin{cases} 
\min(a, b) & \text{if } c \geq \max(a, b), \\
\max(a, b) & \text{if } c \leq \min(a, b), \\
a + b - c & \text{otherwise},
\end{cases}
$$

where $\min(a,b)$ and $\max(a,b)$ represent the larger and smaller function in calculating $a$ and $b$, respectively. Suppose the predicted result of pixel $x$ is $\hat{x}$, then the prediction error $e$ can be obtained by subtracting the prediction from $x$, i.e., $e = x - \hat{x}$. In the following, we will illustrate how MPE embeds messages by modifying the values of prediction errors.

The basic idea of MPE is to represent message bits 0 and 1 if prediction error $e = 0$ and $e = 1$ is encountered, respectively, because we already know $e = 0$ is the most frequent prediction error. To
 embed more message bits, MPE uses additional pair \( e = -1 \) and \( e = -2 \) to represent message bits 0 and 1, respectively. One may freely choose \( e = 0 \) and \( e = 1 \) to represent message bit 0, \( e = -1 \) and \( e = 2 \) to represent message bit 1 also. The choice depends on the user. To embed secret messages, let \( I \) be an 8-bit grayscale image with size \( M \times N \), \( I_{ij} \) be the pixel located on row \( i \) and column \( j \) in \( I \). \( 0 \leq i \leq M - 1, \; 0 \leq j \leq N - 1 \). \( I \) be the stego image and the size is the same as \( I \). The secret messages are encrypted by RSA or DES in advance, and the result is denoted by \( S \). The pixel values in the corresponding positions of \( S \) and \( I \) are listed as follows:

Input: An 8-bit grayscale image \( I \) of size \( M \times N \), and the encrypted secret messages \( S \).

Output: Stego image \( I' \) and auxiliary information \( A \).

Step 1. Prepare an empty matrix \( I' \) and initialize the pixel values in the first row and first column of \( I' \) respectively to the pixel values in the corresponding position of \( I \). Since the predicted results of pixel values in first row and first column are usually less accurate, these pixels are not used to embed data.

Step 2. For \( 1 \leq i \leq M - 1, \; 1 \leq j \leq N - 1 \), scan each pixel \( I_{ij} \) by using raster scan order. If \( I_{ij} = 0 \) or \( I_{ij} = 255 \), then set \( I_{ij} = I_{ij} \) and record the position \((i, j)\) in array \( A \) as the auxiliary information for recovering, then proceed to next pixel.

Step 3. Use Eq. (2) to predict the value of \( I_{ij} \) by setting \( a = I_{i-1, j}, \; b = I_{i, j-1}, \; c = I_{i-1, j-1} \). Let the predicted result to be \( \hat{I}_{ij} \).

Step 4. Calculate prediction error \( e \). Prediction error \( e \) is the difference between pixel \( I_{ij} \) and its predicted result \( \hat{I}_{ij} \), i.e., \( e = I_{ij} - \hat{I}_{ij} \).

Step 5. If all the secret bits in \( S \) have been embedded, then set \( L = (i, j) \), and go to Step 9. The variable \( L \) is used to mark the scanned position when embedding is finished.

Step 6. If \( e = 0 \) or \( e = -1 \), then go to Step 7 for data embedding. Otherwise, go to Step 8.

Step 7. If the to-be-embedded bit is 0, then prediction error \( e \) remains unchanged. If the to-be-embedded bit is 1 and \( e = 0 \), then modify the value of \( e \) to \( e + 1 \); if \( e = -1 \), then modify the value of \( e \) to \( e - 1 \). After embedding, go to Step 8.

Step 8. If \( e > 0 \), then modify the value of \( e \) to \( e + 1 \); if \( e < -1 \), then modify the value of \( e \) to \( e - 1 \).

Step 9. Set \( I_{ij} = \hat{I}_{ij} + e \).

Step 10. If \( i \neq M - 1 \) or \( j 
eq N - 1 \), then update the index \( i \) and \( j \) to go to the next pixel, and go to Step 2. Otherwise, the embedding procedure is completed.

The output of the above algorithm is a stego image \( I' \). Fig. 3 shows the flowchart of embedding procedure. In the process of embedding, each pixel value is changed one grayscale level at most, so the image quality is ensured to be above 48.13 dB (Ni et al., 2006).

Now we use a simple example to illustrate the procedure of embedding information. Suppose \( S = \text{SoS1S2} = 101_2 \) is the message to be embedded. The original image \( I \) has \( 4 \times 4 \) pixels as shown in Fig. 2. We prepare a \( 4 \times 4 \) empty matrix \( I' \) first, and then initialize the top row and the left column of \( I' \), the result is shown in Fig. 4b. Let \( a = I_{00}' = 154b = I_{10}' = 156c = I_{01}' = 154 \), the prediction \( \hat{I}_{11}' = 156 \) of \( I_{11}' \) can be obtained by using Eq. (2). Now, let \( e = h_{11} - h_{11}' = 0 \), so this pixel is embeddable. Extract secret bit \( s_0 = 1 \) from \( S \), then modify the prediction error \( e \) by setting \( e = e + 1 = 0 + 1 = 1 \), and then set \( h_{11}' = h_{11} + e = 157 \). The stego image after embedding is shown in Fig. 4c.

Next, we proceed to predict \( h_{12}' \). Let \( a = I_{11}' = 157b = I_{21}' = 153c = I_{12}' = 156 \), the predicted value of \( I_{12}' \) is \( \hat{I}_{12}' = 154 \). Because \( e = l_{12} - l_{12}' = -4 \), no data can be embedded; therefore, we set \( e = e - 1 = -5 \), and \( l_{12}' = l_{12} - e = 149 \). Following the same procedure, we obtain \( l_{13}' = 148 \) (the embedded value is \( s_1 = 0 \)), and \( l_{13}' = 158 \) (the embedded value is \( s_2 = 1 \)). The stego image is shown in Fig. 4d.

Now we go on to predict \( l_{22}' \). The predicted value of \( l_{22}' \) is \( \hat{I}_{22}' = 150 \); therefore \( e = l_{22} - l_{22}' = 7 \). Since all bits in \( S \) are embedded, we set \( L = (2, 2) \). After setting \( L \), there is no need to modify prediction error \( e \), so we can directly set \( l_{22}' = l_{22} + e = 157 \). The stego image is shown in Fig. 4e. Finally, compute the pixel values for the remainder pixels in the stego image \( I' \), and the final stego image is shown in Fig. 4f.

3.2. Extraction and image restoration algorithm

After receiving the stego image \( I' \), we use the same scan order as in the embedding phase to predict pixel values again, and calculate the prediction error \( e \). We know that, if the value \( e \) is 0 or \(-1 \), the embedded secret bit is 0. If the value \( e \) is 1 or \(-2 \), then the embedded secret bit is 1. If the value \( e \) is not one of the four numbers \(-2, -1, 0, 1 \), then there is no bit embedded. Since we have modified the prediction errors during embedding, the original image can be restored by modifying the prediction errors back to their original value. The detailed steps for extracting secret message and recovering the original image are listed below, and the corresponding flowchart is shown in Fig. 5:

Input: Stego image \( I' \), the end of embedding position \( L \) and auxiliary information \( A \).

Output: The bit stream \( S \) and the original image \( I' \).

Step 1. Prepare a matrix \( I' \) to store the recovered image. The size of \( I' \) is the same as stego image \( I' \). Initialize the pixel values in the first row and the first column of \( I' \) to the pixel values in the corresponding positions of \( I \). For \( 1 \leq i \leq M - 1 \) and \( 1 \leq j \leq N - 1 \), scan each pixel \( I_{ij} \) in the stego image by using the raster scan order.

Step 2. If \( (i, j) \) was recorded as the auxiliary information in \( A \), then set \( I_{ij}' = I_{ij} \) and proceed to next pixel.

Step 3. Used Eq. (2) to predict the value of \( I_{ij} \) by using \( a = I_{i-1, j}, \; b = I_{i, j-1}, \; c = I_{i-1, j-1} \). Suppose the predicted value is \( \hat{I}_{ij} \).

Step 4. Calculate the prediction error \( e = I_{ij} - \hat{I}_{ij} \).

Step 5. According to the parameter \( L \), decide whether all the embedded information has been extracted or not. If they are, then go to Step 11.

Step 6. If \( e = 0 \), then the embedded secret bit is 0, and the prediction error \( e \) remains unchanged.

Step 7. If \( e = 1 \), then the embedded secret bit is 1, and the prediction error \( e \) is modified to \( e - 1 \).

Step 8. If \( e = -1 \), then the embedded secret bit is 0, and the prediction error \( e \) remains unchanged.

Step 9. If \( e = -2 \), then the embedded secret bit is 1, and the prediction error \( e \) is modified to \( e + 1 \).

Step 10. If \( e > 1 \), then prediction error \( e \) is modified to \( e - 1 \). If \( e < -2 \), then the prediction error \( e \) is modified to \( e + 1 \).

Step 11. Set \( I_{ij}' = I_{ij} + e \).
Predict $I'_{ij}$ using Eq. (2) to obtain predicted value $\hat{I}'_{ij}$.

$e = I_{ij} - \hat{I}'_{ij}$

All bits of $S$ embedded?

Record stopping location $L$.

$e = 0$ or $e = -1$

$e = -1$ or $e = 0$

$e = e - 1$

$e = e + 1$

$e = e + 1$

$e = e - 1$

$e = e$

$I'_{ij} = \hat{I}'_{ij} + e$

Update $i$ and $j$

$y$

$y$

$y$

$y$

$y$

$y$

Output $I'$, $L$ and $A$

End

Input $I$, initialize $I'$ and set $i = j = 1$

$I'_{ij} = 0$ or $I'_{ij} = 255$

Set $I'_j = I_{ij}$

Record $(i, j)$ in $A$, and update $i$ and $j$

Print stopping location $L$.

Embedded bits $s_k = 0$

$e = e$

$e = e - 1$

$e = e + 1$

$e = e + 1$

$e = e - 1$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$

$e = e$
Fig. 5. Flowchart of extraction and recovering procedure.

Step 12. If \( i \neq M - 1 \) or \( j \neq N - 1 \), then update the value of \( i \) and \( j \) to go to the next pixel, and go to Step 2; otherwise, we have finished the secret messages extraction and original image recovery.

We continue the example described in the last subsection, and illustrate how to extract the embedded information and restore the image to its original. The stego image from the last sub-section is shown in Fig. 6a. To extract the embedded information, we prepare a \( 4 \times 4 \) matrix \( I' \) first, and then initialize it, the result is shown in Fig. 6b. Next, we predict \( I'_{1} \) based on \( a = I'_{0, 0} = 154 \), \( b = I'_{1, 0} = 156 \), and \( c = I'_{0, 0} = 154 \). The predicted result is \( I'_{1} = 156 \), and the prediction error is \( e = I'_{1} - I'_{1} = 157 - 156 = 1 \); therefore, the embedded secret bits is 1. Modify the prediction error \( e \) to \( e - 1 + 1 = 0 \), then the original pixel value can be found to be \( I = I'_{1} + e = 156 + 0 = 156 \). Now the recovered image \( I \) is shown in Fig. 6c.

The predicted value of \( I_{1} \) is \( I_{1} = 154 \) (\( a = I_{11} = 1570 \), \( b = I_{12} = 153 \), \( c = I_{13} = 156 \), and \( e = I'_{1} - I_{1} = -5 \)); therefore, there is no secret value embedded. After modifying \( e \) to \( 4 \), we obtain \( I_{1} = I_{1} + e = 154 + 4 = 150 \). The prediction value \( I_{1} \) of pixel \( I_{1} \) is 149, \( e = I_{1} - I_{1} = 1 \); therefore, the embedded bit is 0. At this time, \( e \) remains unchanged, and \( I_{1} = I_{1} + e = 148 \). The predicted value \( I_{2} \) of pixel \( I_{2} \) is 157 and obtains \( e = 1 \); therefore, the
embedded bit is 1. After modifying $e$ to 0, $I_{2,1} = \hat{I}_{2,1} + e = 157$. The recovered image $r'$ is shown in Fig. 6d.

Finally, the predictive value of pixel $I_{2,2}$ is $\hat{I}_{2,2} = 150$, and prediction error is $e = I_{2,2} - \hat{I}_{2,2} = 7$. Because $L = (2,2)$, the decoder knows that all the embedded information has been extracted, and therefore, we set $I'_{2,2} = I_{2,2} + e = 157$. The recovered image $r'$ is shown in Fig. 6e. The rest of pixels are processed in the same manner until all pixels have been done. The final recovered image is shown in Fig. 6f.

### 4. Experimental results

In this section, we will show the feasibility and the performance of MPE in terms of pure payload and image quality over the relevant techniques proposed by Ni et al. and other researchers. The algorithms were implemented in Matlab, and the experiments were performed by embedding and extracting random generated bit streams. In all experiments, two pairs of peak and minimum points were used for data embedding. Note that the prediction error $e$ is obtained by subtracting the predicted value $\hat{I}_j$ from $I_j$, i.e., $e = I_j - \hat{I}_j$. Therefore, the original image can be reconstructed by calculating $I_j = \hat{I}_j + e$. The MPE algorithm embeds one message bit by shifting prediction error $e$ one unit at most, and the stego pixel value $I_j$ is obtained by performing the calculation $I_j = \hat{I}_j + e$. Thus, $I_j$ will differ from the original pixel value $I_j$ by at most one. The prediction error will never be accumulated or propagated. Due to this reason, pixel values between peak and minimum points will be either added or subtracted by one at most. Therefore, the worst case for the experiments is that all the pixels are shifted by one gray scale value, implying that the mean square error MSE is almost equal to one. This leads to the PSNR of the stego image being

$$\text{PSNR} = 10 \times \log_{10} \frac{255^2}{\text{MSE}} \geq 48.13 \text{ dB}$$

Note that modification of errors may not be allowed if the pixel value after modified is overflowed or underflowed. At this time, the location of this pixel has to be recorded as the auxiliary information for data extraction. Fortunately, the size of auxiliary information is often zero or negligibly small for most natural images since the overflow/underflow problem rarely occurs.

To compare MPE with Ni et al. method, five 8-bit gray images sized $512 \times 512$ were selected as test images. The results are shown in Table 1. Note that the results may depend on the distribution of the prediction error. With a better predictor for a given test image, MPE may perform better. In our experiments, MED predictor was used. The pure embedding capacity EC, which is also referred to pure payload in this paper, is measured in bits. Since none of these test images possesses overflow or underflow problem, and thus no auxiliary information is needed to record them.

In the Table above, the percentage of the increased payload is calculated by

$$\text{Increased payload} = \frac{\text{pure payload of MPE} - \text{pure payload of Ni et al. method}}{\text{pure payload of Ni et al. method}} \times 100\%.$$

Table 1 shows that the stego image quality for both methods is slightly higher than the theoretical value 48.13 dB; however, the pure payload of the proposed method is, on average, 4.74 times higher than that of Ni et al. method. The embedding capacity of Ni et al. method is limited by the peak value of an image histogram in the spatial domain, and the image Airplane owns the highest peak value; therefore, the embedding capacity is the largest compared to other capacities using Ni et al. technique. On the other hand, the embedding capacity of MPE mainly relies on the accuracy of pixel value prediction. The experiments revealed that the images Lena and Airplane preserve higher embedding capacity. This can be attributed to the smoother texture of these two images. The overall prediction accuracy for smoother images such as Lena and Airplane is higher than for complex images such as Baboon, resulting in larger payload for Lena and Airplane, and smaller payload for Baboon.

Besides, in Ni et al. method, once the peak and minimum points have been chosen, no matter how small the amount of message to be embedded, the pixels with gray value between the peak and minimum points will be changed one grayscale unit. Therefore, the embedding capacity remaining is also referred to pure payload in this paper, is measured in bits. Since none of these test images possesses overflow or underflow problem, and thus no auxiliary information is needed to record them.

In the Table above, the percentage of the increased payload is calculated by

$$\text{Increased payload} = \frac{\text{pure payload of MPE} - \text{pure payload of Ni et al. method}}{\text{pure payload of Ni et al. method}} \times 100\%.$$
fore, when the embedding rates are small, MPE carries much better stego image quality because embedding fewer data bits only needs to modify smaller amount of prediction errors. Table 2 lists the comparison of PSNRs of the two methods under the maximum capacity provided by Ni et al. method. The gain in PSNR for the image Baboon is the smallest and for Lena is the largest. Again, the image Baboon has lower gain because less accurate prediction occurs at complex textures. The experiments show that, on average, the stego image quality of the proposed method gains 7.99 dB higher than that of Ni et al. method.

It is worth mentioning that there are a small number of special images, the amount of bits required to store the overheads for Ni et al. method is larger than their maximum embedding capacity. In this case, the images cannot be embedded with any information. For example, a linear or circular gradient image of size 512 × 512 shown in Fig. 7a and b, respectively, cannot be used to embed data. On the contrary, according to our experiments, 138,327 bits and 100,610 bits can be embedded into these two images, respectively, by using MPE.

In addition, we applied MPE to 23 natural photographic test images sized 768 × 512. Many researches used these images as the standard test images to evaluate their proposed methods (Malvar, 2000; Sharma and Reilly, 2003). The experimental results using these test images are shown in Table 3. The results demonstrated that the embedding capacity of MPE is not only several times higher than Ni et al. method, but also gains slightly higher image quality.

![Gradient images. (a) Linear gradient and (b) circular gradient.](image)

**Table 2**
Comparison of PSNR with same embedding capacity.

<table>
<thead>
<tr>
<th>Image</th>
<th>Pure payload (bits)</th>
<th>Ni et al. method</th>
<th>MPE</th>
<th>Gain (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>6657</td>
<td>48.19</td>
<td>61.80</td>
<td>13.61</td>
</tr>
<tr>
<td>Airplane</td>
<td>16,974</td>
<td>48.27</td>
<td>55.14</td>
<td>6.87</td>
</tr>
<tr>
<td>Tiffany</td>
<td>10,085</td>
<td>48.21</td>
<td>55.59</td>
<td>7.38</td>
</tr>
<tr>
<td>Boat</td>
<td>8932</td>
<td>48.27</td>
<td>56.79</td>
<td>8.52</td>
</tr>
<tr>
<td>Baboon</td>
<td>5685</td>
<td>48.22</td>
<td>51.79</td>
<td>3.57</td>
</tr>
<tr>
<td>Avg.</td>
<td>9667</td>
<td>48.23</td>
<td>56.22</td>
<td>7.99</td>
</tr>
</tbody>
</table>

**Table 3**
Experimental results for various test images (payload is measured in bits).

<table>
<thead>
<tr>
<th>Image</th>
<th>Ni et al. method</th>
<th>MPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pure payload</td>
<td>PSNR(db)</td>
<td>Pure payload</td>
</tr>
<tr>
<td>1</td>
<td>13,788</td>
<td>48.30</td>
</tr>
<tr>
<td>2</td>
<td>76,619</td>
<td>48.57</td>
</tr>
<tr>
<td>3</td>
<td>14,898</td>
<td>48.43</td>
</tr>
<tr>
<td>4</td>
<td>12,207</td>
<td>50.04</td>
</tr>
<tr>
<td>5</td>
<td>7466</td>
<td>48.92</td>
</tr>
<tr>
<td>6</td>
<td>16,664</td>
<td>48.24</td>
</tr>
<tr>
<td>7</td>
<td>23,121</td>
<td>48.42</td>
</tr>
<tr>
<td>8</td>
<td>6705</td>
<td>48.17</td>
</tr>
<tr>
<td>9</td>
<td>17,669</td>
<td>48.34</td>
</tr>
<tr>
<td>10</td>
<td>15,565</td>
<td>48.47</td>
</tr>
<tr>
<td>11</td>
<td>36,315</td>
<td>48.34</td>
</tr>
<tr>
<td>12</td>
<td>22,108</td>
<td>48.41</td>
</tr>
<tr>
<td>13</td>
<td>8506</td>
<td>48.51</td>
</tr>
<tr>
<td>14</td>
<td>9443</td>
<td>48.23</td>
</tr>
<tr>
<td>15</td>
<td>9375</td>
<td>48.82</td>
</tr>
<tr>
<td>16</td>
<td>13,106</td>
<td>48.84</td>
</tr>
<tr>
<td>17</td>
<td>18,938</td>
<td>48.25</td>
</tr>
<tr>
<td>18</td>
<td>16,188</td>
<td>48.67</td>
</tr>
<tr>
<td>19</td>
<td>12,381</td>
<td>48.34</td>
</tr>
<tr>
<td>20</td>
<td>49,448</td>
<td>48.41</td>
</tr>
<tr>
<td>21</td>
<td>20,165</td>
<td>48.48</td>
</tr>
<tr>
<td>22</td>
<td>11,977</td>
<td>48.36</td>
</tr>
<tr>
<td>23</td>
<td>12,910</td>
<td>48.58</td>
</tr>
<tr>
<td>Avg.</td>
<td>19,372</td>
<td>48.57</td>
</tr>
</tbody>
</table>
It is interesting to note that, in Ni et al. method, the capacity of the second image is much higher than that of others. This is because the pixel values in this image concentrate on certain range, as shown in Fig. 8a, causing a peaked histogram, thus it is greatly suitable for using Ni et al. method to hide data. Even though, the capacity MPE still has 18% higher than that of Ni et al. method. Besides, for the fourth image, its PSNR is 50.04 dB by using the method of Ni et al., which is higher than other images. This is because the range between the maximal and minimal points selected is small, and thus the number of the pixels to be shifted is relatively less which leads to the high PSNR. If MPE is applied to the same capacity (12,207 bits), the stego image quality is 58.52 dB, which is much higher than 50.04 dB obtained by using Ni et al. method.

Two texture images and two medical images, shown in Fig. 9, are also tested using the proposed MPE technique. The results are shown in Table 4. Note that for the texture images, the increased payload is lower than those of the medical images, due to the intrinsic lack of predictability of complex pixel activities. Nevertheless, MPE also outperforms Ni et al. method in terms of payload under the same PSNR.

We also compare the performance of MPE with some other promising reversible data hiding methods that have been proposed recently (Tian, 2003; Kamstra and Heijmans, 2005; Thodi and Rodriguez, 2007; Xuan et al., 2007). These methods exploit either expansion embedding technique (Tian, 2003; Kamstra and Heijmans, 2005; Thodi and Rodriguez, 2007), or histogram-shifting technique (Xuan et al., 2007) to achieve an acceptable image quality. Since the goal of MPE is to produce high stego image quality with minimal PSNR 48.13, we compare the embedding capacity of MPE with these newly proposed algorithms, including Tian’s, Kamstra et al., Thodi et al. and Xuan et al. methods, for image quality higher than 48.13 dB. The test images are Lena and Baboon, which often lead in different data embedding performance. The results are as shown in Fig. 10.

It is clear that, for the test image Lena, MPE achieves higher embedding capacity than the other four methods under the same image quality. This is because MPE is capable of keeping the distortion low while embedding small amounts of messages and can automatically create just enough space to embed the desired payload. Tian’s method is not capable of embedding small payloads at low distortion, because of larger overhead to keep a location map of pairs that are used for expansion. Kamstra et al., Thodi et al. and Xuan et al. methods perform better than Tian’s method at low distortion in that much less overhead is need for their techniques. Nevertheless, MPE is typically at least 3 dB better than their method under the same capacity. For the image Baboon, the PSNR improvements are varied from 0.5 dB to 1 dB, compared to Kamstra et al. and Thodi et al. method. Although Xuan et al. method performs slightly better than MPE in terms of capacity when PSNR higher than 50 dB, MPE is still comparable to Xuan et al. method since their method involves integer wavelet transform and optimal

![Fig. 8. Image histograms. (a) Histogram of image 2 and (b) histogram of image 4.](image)

![Fig. 9. (a) and (b) are texture images. Originally obtained form http://www.cgtextures.com. (c) and (d) are medical images. Originally obtained form http://www.rad.jhmi.edu/mri/MRI_Info_SubDivisions.htm.](image)

![Table 4 Experimental results for test images shown in Fig. 9.](image)
parameters selection stage, the computational cost is higher than that of MPE. To sum up, our method has the best or comparable capacity–distortion curve in virtually all cases. As an additional advantage, MPE runs much faster because no compression or transformation technique is used, while others are essential.

5. Conclusions

In this paper, we proposed a novel reversible data hiding technique based on modification of prediction errors. Pixel values are first predicted, and then error values are obtained. Message bits are embedded reversibly by modifying the values of prediction errors. MPE remedies the major drawbacks of Ni et al. method – low embedding capacity and inability to control the capacity, by embedding secret message bits into errors values. MPE has the capability to keep the distortion low when embedding fewer messages and automatically vacant enough space for embedding desired payload. Besides, the PSNR of the stego image produced by MPE is guaranteed to be above 48 dB. A variety of test results indicates the superior performance of MPE over Ni et al. technique. According to our experiment, the averaged embedding capacity of MPE is several times higher than that of Ni et al. technique. MPE also outperforms recently proposed reversible data hiding methods such as Tian’s, Kamstra et al., Xuan et al. and Thodi et al. techniques at low distortion rates.

References


Wien Hong received his M.S. and Ph.D. degree from State University of New Your at Buffalo in 1994 and 1997, respectively. Since 1999, he is an assistant professor in the Department of Information Management at Yu-Da College of Business, Taiwan. His research interests include digital watermarking, data hiding and data compression.

Tung-Shou Chen received the B.S. and Ph.D. degrees from National Chiao Tung University in 1986 and 1992, respectively, both in Computer Science and Information Engineering. From 1994 through 1997, he was the faculty of the Department of Information Management at National Chin-Yi Institute of Technology in Taiwan. From 1998 through 2000, he was both the dean of Student Affairs and a professor of the Department of Computer Science and Information Management at Providence University in Taiwan. Since August 2000 he has been a professor of Graduate School of Computer Science and Information Technology at National Taichung Institute Technology in Taiwan. From 2004 through 2007, he was also the dean of the Graduate School of Computer Science and Information Technology at National Taichung Institute of Technology. His current research interests include data mining, image cryptosystems, and image compression.

Chih-Wei Shiu received his M.S. degree from Yu-Da College of Business, Department of Information Management in 2009. Currently, he is working towards his Ph.D. degree at National Chung-Hsing University, Graduate school of Computer Information Engineering, Taiwan. His current research interests include data compression and steganography.