Interpolation-based hiding scheme using the modulus function and re-encoding strategy

Tzu-Chuen Lu

Department of Information Management, Chaoyang University of Technology, Taichung 41349, Taiwan, ROC

A R T I C L E   I N F O

Article history:
Received 23 February 2017
Revised 20 July 2017
Accepted 22 July 2017
Available online 24 July 2017

Keywords:
Image interpolation
Reversible information hiding
Re-encoding
RS analysis

A B S T R A C T

Biswapati et al. proposed a interpolation-based hiding scheme. The scheme directly conceals the information, which records the position of the modified pixel to generate the stego-image. The position value is very large, thus creating a large image distortion. This study reduces the value range of the position values and re-encodes the values to reduce the distortion. The proposed scheme examines the probabilities for the position values and re-encodes the value according to its occurrence number. A re-encode function is used to obtain the rank of the position value in descending order. The most frequent position value is re-encoded to zero. The re-encoded codes are ciphered to generate mapping codes with negative and positive numbers. A mapping function is proposed to map the re-encoded code to the mapping code. The mapping code is half of the re-encoded code such that the image distortion becomes small. The proposed scheme uses different sizes of embedding blocks to control the hiding rate and image quality. Compared with other state-of-the-art methods, the proposed scheme is better in terms of hiding payload and image quality.

© 2017 Elsevier B.V. All rights reserved.

1. Introduction

The information hiding technique is used to share secret messages, detect tampered data, verify ownership, track piracy, and augment data. In an information hiding scheme, cover media, such as image, video, text, execution file, and audio, could be used to carry the secret message. The media that carries the message is called a stego-media. Unauthorized persons cannot detect any difference between the cover media and the stego-media. This study uses an image as the cover media to conceal the secret message for generating the stego-image [8,10].

Information hiding schemes can be categorized into two types, namely, reversible and non-reversible, according to whether the stego-image can be reversed or not. The reversible data-hiding (RDH) scheme can recover the original image after the concealed message is extracted. Conversely, the non-reversible hiding scheme cannot recover the stego-image to its original state. Research has proposed many related schemes given that the RDH technique can be used in many applications, such as military use, medical purposes, and digital archiving. Recent RDH schemes include histogram shifting, difference expansion, dual images, and image interpolation.

The difference expansion (DE) technique computes the distance between two pixels (or prediction value and pixel) and conceals secret bits into any two-distance times. Tian [10], Alattar [1], and Li et al. [7] proposed DE-based RDH methods to generate the stego-image. DE-based RDH schemes can effectively embed secret information in the cover image. However, this technique can cause great distortion, which diminishes the image quality of the stego-image.

Histogram-shifting hiding techniques are proposed to improve the quality of the stego-image using the DE method. The histogram technique computes the probability of pixels to generate a histogram and points out the peak pixel in the histogram to embed the information. The other pixels between the peak pixel and a zero pixel are shifted to create a space for hiding the secret message. For example, Ni, and Lee et al. proposed histogram-shifting-based hiding schemes [8,6]. The image quality of the histogram-shifting-based scheme is high, but the embedding payload is low.

Dual-image-based techniques were proposed in 2014 to enhance the embedding payload. The dual-image-based RDH scheme replicates the original image to generate two copy images and conceals information in two images. For example, Qin et al. applied the modulus function and exploited the modification direction and three embedding rules to generate two stego-images [17]. Lu et al.

E-mail address: tceu@cyut.edu.tw
1 URL: http://www.cyut.edu.tw/~tceu

http://dx.doi.org/10.1016/j.sigpro.2017.07.025
0165-1684/© 2017 Elsevier B.V. All rights reserved.
utilized the center-folding strategy to fold the secret message before concealing it in the stego-image to enhance the image quality [13]. Nevertheless, the major drawback of this technique is that it requires two images to extract the message.

One technique to solve this problem is image interpolation, which extends an extra pixel between two neighboring pixels to embed the secret message instead of generating another image. Many researchers proposed interpolation-based RDH schemes to increase the embedding capacity [2,11,12,14,15,16]. For example, Malik et al. proposed an image interpolation-based RDH scheme using pixel value adjusting feature [14]. Lee et al. proposed a data-hiding method based on reutilized exploiting modification direction, image interpolation, and canny edge detection [11]. Lu applied the center-folding strategy and interpolation technique with neighboring pixels (INP) to propose an adaptive interpolation-based hiding scheme. In this scheme, the secret message is folded by the center value to reduce the value range and decrease the image distortion [12]. Biswapati et al. used a weighted matrix to compute the modulus summation to determine which pixel should be modified or not. The position value is added to the interpolated pixel [2].

In Biswapati et al.’s scheme, an original image is divided into several parts with a size of $3 \times 3$. Then, Biswapati et al. used the interpolation method to generate a cover block size of $5 \times 5$ for each original part. Each cover block has 12 interpolated pixels, which can be used to conceal 48 secret bits. To enhance the security of the scheme, Biswapati et al. updated the weighted matrix by using a shared secret key. Only authorized personnel who know the secret key can extract the correct message from the stego-image.

In Biswapati et al.’s scheme, the secret message is not directly concealed in the interpolated pixels. A weighted matrix is used to compute the modulus sum and compare the sum with the secret message to make sure the sum is equal to the secret message. If the sum is not equal to the secret message, then the sum is subtracted from the secret message to obtain a modified position value. The position value is then hidden into the interpolated pixel to generate the stego-pixel.

Biswapati et al.’s scheme can hide numerous secret messages in the cover image. However, the image quality of Biswapati et al.’s scheme can still be improved.

The key factor that influences the image quality of Biswapati et al.’s scheme is the modified position values. The value is usually very large, thus creating a large image distortion between the interpolated image and the stego-image.

This study reduces the value range of the position values and re-encodes the values to reduce the distortion. The proposed scheme examines the probabilities for the position values and re-encodes the value according to its occurrence number. For the position value with a high occurrence number, the proposed scheme encodes it with a small number close to zero. Conversely, the proposed scheme encodes the rare value with a large number. Frequent position value with a small code can effectively reduce the distortion between the interpolated image and the stego-image.

Furthermore, the proposed scheme uses different sizes of embedding blocks to control the hiding rate and image quality.

### 2. Related works

Many types of interpolation techniques, such as neighbor mean interpolation (NMI) and INP, have been proposed. Lu applied INP to expand an image, and Biswapati et al. used NMI to generate virtual pixels.

This study describes NMI (Section 2.1), INP (Section 2.2), Lu’s scheme (Section 2.3), and Biswapati et al.’s scheme (Section 2.4) in Section 2.

#### 2.1. NMI

Image interpolation is the process of enlarging the size of an original image by inserting virtual pixels between two neighboring pixels [15]. Jung and Yoo used these pixels to embed a secret message [5]. In their scheme, the virtual pixel is computed by the average value of the neighboring pixels. Fig. 1 shows a diagram of the scheme. Fig. 1(a) is an original image $I$. The virtual pixels are computed to generate a cover image using the following equation:

$$C_{NMI}^{(i,j)} = \begin{cases} \left( \frac{(i-1,j) + (i+1,j)}{2} \right), & \text{if } i = 2h, \ j = 2w + 1, \\ \left( \frac{(i-1,j) + (i+1,j)}{2} \right), & \text{if } i = 2h + 1, \ j = 2w. \\ \left( \frac{(i-1,j-1) + (i+1,j-1)}{2} \right), & \text{otherwise}. \end{cases}$$

In Eq. (1), $h$ and $w$ are the height and width, respectively, of a cover image, and $(i,j)$ is the coordinate of the pixel. The interpolated image is called cover image $C$, Jung and Yoo concealed secret bits $b$ in the virtual pixels $C_{NMI}^{(i,j)}$ to generate the stego-pixel $C_{NMI}^{(i,j)}$, as shown in Fig. 1(c). When the receiver receives the stego-image $C$, the secret bits can be extracted from the stego-pixels $C_{NMI}^{(i,j)}$, and the original image can be restored by reducing the stego-image. Fig. 2 shows how the original image is enlarged and how secret bits are hidden using Jung and Yoo’s scheme. Fig. 2(a) is the original image $I = [84, 86, 88, 81]$. Suppose that the secret bit is $b = (1011011)_2$. The interpolated pixels are calculated by $C_{NMI}^{(1,2)} = \{84,86,88\} = 85$, $C_{NMI}^{(2,1)} = \{84,88\} = 86$, and $C_{NMI}^{(2,2)} = \{84,85,86\} = 85$. The virtual pixels are shown in Fig. 2(b).

Next, they divide the cover image into several block sizes of $2 \times 2$ to embed the secret bits. The first pixel $l_{(1,1)}$ is the base pixel used to compute the differences with other interpolated pixels. Let $d_{NMI}^{b_1}$, $d_{NMI}^{b_2}$, and $d_{NMI}^{b_3}$ be the differences between $l_{(1,1)}$ and three virtual pixels $C_{NMI}^{(1,2)}, C_{NMI}^{(2,1)}$, and $C_{NMI}^{(2,2)}$. The equation is expressed as follows:

$$\begin{align*}
&d_{NMI}^{b_1} = |C_{NMI}^{(1,2)} - l_{(1,1)}|, \\
&d_{NMI}^{b_2} = |C_{NMI}^{(2,1)} - l_{(1,1)}|, \\
&d_{NMI}^{b_3} = |C_{NMI}^{(2,2)} - l_{(1,1)}|.
\end{align*}$$

The differences in Fig. 2(b) are $d_{NMI}^{b_1} = |85 - 84| = 1$, $d_{NMI}^{b_2} = |86 - 84| = 2$, and $d_{NMI}^{b_3} = |85 - 84| = 1$. The difference $d_{NMI}^{b_2}$ is a key factor for judging the length of the secret bits $L_{NMI}$, which could be concealed in the interpolated pixel. The length is computed by the following:

$$\begin{align*}
&L_{NMI}^{b_1} = \lfloor \log_2(d_{NMI}^{b_1}) \rfloor, \\
&L_{NMI}^{b_2} = \lfloor \log_2(d_{NMI}^{b_2}) \rfloor, \\
&L_{NMI}^{b_3} = \lfloor \log_2(d_{NMI}^{b_3}) \rfloor.
\end{align*}$$

In the example in Fig. 2(b), the lengths of the secret bits are $L_{NMI}^{b_1} = \lfloor \log_2(1) \rfloor = 0$, $L_{NMI}^{b_2} = \lfloor \log_2(2) \rfloor = 1$, and $L_{NMI}^{b_3} = \lfloor \log_2(1) \rfloor = 0$. In this example, $L_{NMI}^{b_1}$ and $L_{NMI}^{b_3}$ are both 0, which indicates that the virtual pixels $C_{NMI}^{(1,2)}$ and $C_{NMI}^{(2,2)}$ are non-embeddable. The value of $L_{NMI}^{b_2}$ is 1, which indicates that one secret bit can be embedded into the virtual pixel $C_{NMI}^{(2,1)}$. The scheme transforms the secret bit of $b$ into a decimal-based symbol $\beta$ and concealed into $C_{NMI}^{(2,1)}$. The first secret bit of $b$ is $1_2$, and the transformed symbol is $\beta = (1)_0$. The secret symbol $\beta$ is then hidden in $C_{NMI}^{(2,1)}$ to obtain
the stego-pixel \( C_{(1,2)}^{\text{INP}} = C_{(1,2)}^{\text{INP}} + \beta = 86 + 1 = 87 \). The final results are illustrated in Fig. 2(c).

2.2. INP

Lee and Huang proposed an interpolation using the INP scheme [6]. In Lee and Huang’s scheme, the virtual pixel \( C_{(i,j)}^{\text{INP}} \) is the average of two neighboring pixels, and the weight of the pixel on the right-hand side is higher than that of the pixel on the left side. The interpolated pixel is computed by

\[
C_{(i,j)}^{\text{INP}} = \begin{cases} 
\left( \frac{I_{(i-1,j)} + I_{(i,j+1)}}{2} \right), & \text{if } i = 2h, \ j = 2w + 1, \\
\left( \frac{I_{(i-1,j)} + I_{(i,j+1)}}{2} \right), & \text{if } i = 2h + 1, \ j = 2w, \\
\left( \frac{C_{(i-1,j)}^{\text{INP}} + C_{(i-1,j+1)}^{\text{INP}}}{2} \right), & \text{otherwise}.
\end{cases}
\]

Using the sample example, the original image is \( I = [84, 86, 88, 81] \). The virtual pixels are \( C_{(1,2)}^{\text{INP}}(1,1) = \left( \frac{I_{(1,1)} + I_{(1,1)} + I_{(1,1)}}{2} \right) = \frac{(84 + 86 + 88)}{2} \) = 84. The lengths of the differences are \( L_1^{\text{INP}} = \log_2(d_1^{\text{INP}}) = \log_2(2) = 1 \), and \( L_2^{\text{INP}} = \log_2(d_2^{\text{INP}}) = \log_2(4) = 2 \). The transformed decimal secret symbol of \( b = 10_{10} \) can be embedded in the first interpolated pixel \( C_{(1,2)}^{\text{INP}} \) to obtain the stego-pixel \( C_{(1,2)}^{\text{INP}} = C_{(1,2)}^{\text{INP}} + \beta = 84 + 2 = 86 \). The final stego-pixels are shown in Fig. 3.

2.3 Lu’s \((t, F_t)\) scheme

Lu [12] proposed an adaptive interpolation-based hiding scheme to improve Lee and Huang’s scheme. In the scheme, a pa-
The value $pos$ is then added to the interpolated pixel to generate the stego-pixel. For example, Fig. 7(a) shows the original image. The interpolated pixel $C^\text{BIS \_1}_{(2, 2)}$ is computed by $C^\text{BIS \_1}_{(2, 2)} = \left[\frac{84 + 86}{2}\right] = 85$. The cover image is shown in Fig. 7(b). The original image block is used to compute the modulus value with the weighted matrix, as shown in Fig. 6(c). The modulus value is $\text{Val} = \left[\begin{array}{c} 84 \times 1 + 86 \times 2 + \cdots + 86 \times 8 \end{array}\right]$ mod 16 = 9. Assume that the secret bit is $b = (1011)_2$ and that the transformed secret symbol is $\beta = (11)_4$. The symbol is not equal to the modulus value. The modified position is $pos = \beta - \text{Val} = 11 - 9 = 2$. The first stego-pixel $C^\text{BIS \_1}_{(1, 2)}$ is then computed by $C^\text{BIS \_1}_{(1, 2)} = C^\text{BIS \_1}_{(2, 2)} + pos = 85 + 2 = 87$. The final stego-image is shown in Fig. 7(c).

In Biswapati et al.'s scheme, each block size of $5 \times 5$ can conceal 48 secret bits. The hiding capacity of the scheme is large. However, the scheme directly adds the position value to the interpolated pixel, thereby causing a large distortion.

To improve the image quality of Biswapati et al.'s scheme, this study re-encodes these position values by using a pre-processing procedure. The frequent position values are mapped to a small value close to zero to reduce the distortion between the interpolated pixel and the stego-pixel.

### 3. Proposed scheme

The diagram of the proposed scheme is illustrated in Fig. 8. In the diagram, Fig. 8(a) is the original image. The proposed scheme applies Lee and Huang's INP interpolation method to enlarge the original image for generating the cover image. Fig. 8(b) presents the cover image, which is divided into several blocks. A weighted matrix shown in Fig. 8(c) is used to compute the modulus value. The scheme computes the modulus value for each block and conceals the secret message in the interpolated pixels. Each interpolated pixel in the block can carry $L$ secret bits. The scheme extracts $L$ secret bits from the secret message $b$ and transforms it into a decimal-based secret symbol $\beta$. The differences between the modulus value and the symbol $\beta$ are counted to generate a histogram. The differences are re-encoded according to their occurrence number. To further decrease the image distortion, the proposed scheme narrows down the re-encoded code to generate the mapping code. The mapping code is half of the re-encoded code. The mapping codes are then added to the interpolated pixels to generate the stego-pixels. The detailed embedding process is illustrated in Fig. 8.

#### 3.1. Embedding process

After generating the cover image using Eq. (4), the scheme divides the image into several blocks with two different sizes $n \times n, n \in [3, 4]$. Fig. 9 shows the blocks with different sizes. The proposed scheme uses block size to control the hiding capacity. For a block size of $3 \times 3$, five interpolated pixels can be used to conceal messages. For a block size of $4 \times 4$, 12 interpolated pixels can be used to embed secret bits.

Each block has four original pixels $I = \{I_{(1, 1)}, I_{(1, 2)}, I_{(2, 1)}, I_{(2, 2)}\}$ that are used to compute a modulus value with a weighted matrix. Let $wm = \{wm_{(1, 1)}, wm_{(1, 2)}, wm_{(2, 1)}, wm_{(2, 2)}\}$ be the weighted matrix. The matrix can be generated using a secret key to increase the security of the scheme. The modulus value is computed by

$$\text{Val} = \left[\sum \sum I_{(i,j)} \times wm_{(i,j)}\right] \text{mod } (2 \times k + 1),$$

where $k$ is a constant used to control the length of the secret bit concealed in each interpolated pixel. The length of the secret bit is calculated by

$$L = \left[\log_2(2 \times k + 1)\right].$$
Fig. 6. Diagram of Biswapati et al.’s hiding scheme.

(a) Original image $I$

<table>
<thead>
<tr>
<th>$I_{1,1}$</th>
<th>$I_{1,2}$</th>
<th>$I_{1,3}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I_{2,1}$</td>
<td>$I_{2,2}$</td>
<td>$I_{2,3}$</td>
</tr>
<tr>
<td>$I_{3,1}$</td>
<td>$I_{3,2}$</td>
<td>$I_{3,3}$</td>
</tr>
</tbody>
</table>

(c) Weighted matrix $wm$

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
</tbody>
</table>

(b) Cover image $C$

<table>
<thead>
<tr>
<th>$I_{1,1}$</th>
<th>$C_{BIS}^{(1,1)}$</th>
<th>$I_{1,2}$</th>
<th>$C_{BIS}^{(1,2)}$</th>
<th>$I_{1,3}$</th>
<th>$C_{BIS}^{(1,3)}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{BIS}^{(2,1)}$</td>
<td>$C_{BIS}^{(2,2)}$</td>
<td>$C_{BIS}^{(2,3)}$</td>
<td>$C_{BIS}^{(2,4)}$</td>
<td>$C_{BIS}^{(2,5)}$</td>
<td></td>
</tr>
<tr>
<td>$C_{BIS}^{(3,1)}$</td>
<td>$C_{BIS}^{(3,2)}$</td>
<td>$C_{BIS}^{(3,3)}$</td>
<td>$C_{BIS}^{(3,4)}$</td>
<td>$C_{BIS}^{(3,5)}$</td>
<td></td>
</tr>
<tr>
<td>$C_{BIS}^{(4,1)}$</td>
<td>$C_{BIS}^{(4,2)}$</td>
<td>$C_{BIS}^{(4,3)}$</td>
<td>$C_{BIS}^{(4,4)}$</td>
<td>$C_{BIS}^{(4,5)}$</td>
<td></td>
</tr>
</tbody>
</table>

(d) Stego-image $C'$

<table>
<thead>
<tr>
<th>$I_{1,1}$</th>
<th>$C_{BIS}^{(1,1)}$</th>
<th>$I_{1,2}$</th>
<th>$C_{BIS}^{(1,2)}$</th>
<th>$I_{1,3}$</th>
<th>$C_{BIS}^{(1,3)}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{BIS}^{(2,1)}$</td>
<td>$C_{BIS}^{(2,2)}$</td>
<td>$C_{BIS}^{(2,3)}$</td>
<td>$C_{BIS}^{(2,4)}$</td>
<td>$C_{BIS}^{(2,5)}$</td>
<td></td>
</tr>
<tr>
<td>$C_{BIS}^{(3,1)}$</td>
<td>$C_{BIS}^{(3,2)}$</td>
<td>$C_{BIS}^{(3,3)}$</td>
<td>$C_{BIS}^{(3,4)}$</td>
<td>$C_{BIS}^{(3,5)}$</td>
<td></td>
</tr>
<tr>
<td>$C_{BIS}^{(4,1)}$</td>
<td>$C_{BIS}^{(4,2)}$</td>
<td>$C_{BIS}^{(4,3)}$</td>
<td>$C_{BIS}^{(4,4)}$</td>
<td>$C_{BIS}^{(4,5)}$</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 7. Example of Biswapati’s scheme.

(a) Original image

| 84 | 86 | 86 |
| 88 | 81 | 81 |
| 84 | 86 | 86 |

(b) Cover image

| 84 | 85 | 86 | 86 | 86 | 84 | 87 | 86 | 90 | 86 |
| 86 | 84 | 83 | 83 | 83 | 80 | 87 | 86 | 85 | 80 |
| 88 | 84 | 81 | 81 | 81 | 88 | 88 | 87 | 85 | 81 |
| 86 | 84 | 83 | 83 | 83 | 84 | 80 | 80 | 81 | 89 |
| 84 | 85 | 86 | 86 | 86 | 84 | 87 | 86 | 81 | 86 |

(c) Stego-image

Fig. 8. Diagram of the proposed scheme.

(a) Original image $I$

(b) Cover image $C$

(c) Weighted matrix ($wm$)

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

(d) Difference re-encoding

(e) Stego-image $I'$
When the constant $k$ is set to four, the length of the secret bits is $L = \log_2(2 \times 4 + 1) = 3$. Each interpolated pixel can conceal three secret bits. When $k$ is set to eight, then the length is $L = \log_2(2 \times 8 + 1) = 4$. Four secret bits can be embedded into each interpolated pixel.

The scheme extracts $L$ secret bits from the secret message and transforms it into a decimal-based secret symbol $\beta$. Then, the modulus value is subtracted from the secret symbol to obtain the difference
diff = (\beta - \text{Val}) \% (2 \times k + 1). \tag{12}

The scheme counts the occurrence number of each diff to generate a histogram. Let $H = [h_0, h_1, \cdots, h_{2k+1}]$ be the occurrence histogram, where $h_i$ is the occurrence number of $\text{diff}_i$. The differences are then ordered by the occurrence number in decreasing order to obtain $H' = [h_0', h_1', \cdots, h_{2k+1}']$. Let $\text{Idx}(\cdot)$ be a function that returns the location of $\text{diff}$ in $H'$ and $\text{ldx} (\text{diff})$ be the re-encoded code of diff.

Fig. 10 shows an example. Fig. 10(a) is the cover block size of $3 \times 3$. The four original pixels are $I = [84, 86, 88, 81]$, and the weighted matrix is $w_m = \{2, 3, 4, 1\}$. In this example, we want to hide four bits for each interpolated pixel; thus, the constant $k$ is set to 8. The length of the secret bit is $L = \log_2(2 \times 8 + 1) = 4$. The modulus value is $\text{Val} = \{(84 \times 2) + (86 \times 3) + (88 \times 4) + (81 \times 1) \mod (2 \times 8 + 1) = 9\}$. Assume that the secret bits are $b = (0000)_2$ and the transformed secret symbol is $\beta = (0)_{10}$. The difference is $\text{diff} = (0 - 9) \% (2 \times 8 + 1) = 8$.

Suppose that the occurrence number of each diff is shown in $H$ of Fig. 11. The occurrence number of $\text{diff} = 8$ is $h_8 = 7$. The difference after ordering is denoted as $H'$. The occurrence number $h_8 = 7$ is ranked in the second position, and the index of $\text{diff} = 8$ is $\text{ldx} (\text{diff}) = 1$.

The re-encoding processing is to set the most frequent position value to zero. However, the re-encoded codes $\text{ldx} (\text{diff})$ are positive numbers. The values of the other codes are still very large. To narrow down the value range, the re-encoded codes are ciphered to generate mapping codes with negative and positive numbers. The closest codes to zero are $-1$ and $1$; thus, the re-encoded codes 1 and 2 are mapped to $-1$ and $1$, respectively. The next close codes are $-2$ and $2$; hence, the re-encoded codes 3 and 4 are mapped to $-2$ and $2$, respectively, and so on. Fig. 12 shows the diagram of the mapping process. The largest re-encoded codes $2 \times k$ and $2 \times k + 1$ are mapped to $-k$ and $k$, respectively.

The formal equation used to cipher the re-encoded code to generate a mapping code is $\text{diff} = \text{sign} (\text{ldx} (\text{diff})) \times \left[ \frac{\text{ldx} (\text{diff}) + 1}{2} \right]$, and $\text{diff} = \text{sign} (\text{ldx} (\text{diff})) \times \left[ \frac{\text{ldx} (\text{diff}) + 1}{2} \right]$.

In the equation, $\text{ldx} (\text{diff})$ is the mapping code, and $\text{sign}(x)$ returns the sign of $x$. If $x$ is an even number, then $\text{sign}(x) = -1$. Otherwise, $\text{sign}(x) = 1$. The re-encoding map among $\text{diff}$, $H$, $H'$, and $\text{diff}$ is shown in Fig. 13.

The mapping code $\text{diff}$ instead of the difference $\text{diff}$ is concealed in the interpolated pixel to generate the stego-pixel. The equation is $C_{\text{ref}}^{\text{diff}} (i,j) = C_{\text{ref}}^{\text{diff}} (i,j) + \text{diff}$. \tag{14}

Following the same example, the re-encoded code $\text{ldx} (\text{diff}) = 1$ is computed by $\text{diff} = \text{sign} (1) \times \left[ \frac{1 + 1}{2} \right] = 1$. The first stego-pixel is calculated by $C_{\text{ref}}^{\text{diff}} (i,j) = C_{\text{ref}}^{\text{diff}} (i,j) + \text{diff}$. The final stego-image is shown in Fig. 10(c).

3.2. Overflow/Underflow problem

The proposed scheme may suffer from an overflow or underflow problem when the interpolated pixel is smaller than $k - 1$ (or higher than $255-k$) and the re-encoded code is smaller than 0 (or higher than k). For example, assume that the interpolated pixel is $C_{\text{ref}}^{\text{diff}} (i,j) = 254$ and the re-encoded code is $\text{diff} = 4$. The stego-pixel is $C_{\text{ref}}^{\text{diff}} (i,j) + \text{diff} = 254 + 4 = 258$. The pixel has an overflow problem. To solve this problem, the scheme modifies Eq. (14) as follows:

$C_{\text{ref}}^{\text{diff}} (i,j) \times \text{diff} \begin{cases} \text{C}_{\text{ref}}^{\text{diff}} (i,j) + \text{diff}, & \text{if } 0 \leq C_{\text{ref}}^{\text{diff}} (i,j) \leq 255, \\ \text{C}_{\text{ref}}^{\text{diff}} (i,j) + 2 \times (2 \times k + 1), & \text{if } C_{\text{ref}}^{\text{diff}} (i,j) < 0; \leq 255. \end{cases}$ \tag{16}

Following the same example, the temporary pixel $C_{\text{ref}}^{\text{diff}} (i,j) = 254 + \text{diff}$ is greater than 255. The stego-pixel is modified as $C_{\text{ref}}^{\text{diff}} (i,j) = 2 \times (2 \times k + 1) = 258 - 2 \times (2 \times 8 + 1) = 224$. The proposed scheme contains several procedures, which include dividing blocks, computing the modulus value, transforming the secret message, calculating the difference $\text{diff}$ to generate the histogram $H$, sorting the histogram to obtain $H'$, re-encoding $\text{diff}$ to obtain $\text{ldx} (\text{diff})$, mapping $\text{ldx} (\text{diff})$ to obtain $\text{diff}$, and adding $\text{diff}$ to the interpolated pixel to generate the stego-pixel. The pseudocode of the proposed scheme is shown below.

---

Input: an interpolated image of size $h \times w$
Output: the stego-image
Pre-process:
(1) Divide the image into server blocks with size $n \times n$.
(2) For each block
(3) Compute the modulus value by using Eq. 10.
(4) For each pixel in the block
(5) Extract the secret message and transform it to a secret symbol $\beta$.
(6) Compute the difference $\text{diff}$ by using Eq. 12.
(7) Generate a histogram $H$ to record the occurrence number of each $\text{diff}$.
(8) End for
(9) End for
(10) Sort $H$ in descending order to obtain $H'$.

Embedding process:
(1) Divide the image into server blocks with size $n \times n$.
(2) For each block
(3) Compute the modulus value by using Eq. 10.
(4) For each pixel in the block
(5) Extract the secret message and transform it to a secret symbol $\beta$.
(6) Compute the difference $\text{ldx} (\text{diff})$ by using Eq. 12.
(7) Re-encode $\text{diff}$ to obtain $\text{ldx} (\text{diff})$.
(8) Compute the mapping code $\text{diff}$ by using Eq. 13.
(9) Generate the stego-pixel by using Eq. 14.
(10) End for
(11) End for
Fig. 10. Example of the proposed scheme.

<table>
<thead>
<tr>
<th>$diff$</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>...</th>
<th>8</th>
<th>...</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H$</td>
<td>1</td>
<td>9</td>
<td>4</td>
<td>...</td>
<td>7</td>
<td>...</td>
<td>6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$H'$</th>
<th>9</th>
<th>7</th>
<th>6</th>
<th>...</th>
<th>1</th>
<th>...</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>$diff$</td>
<td>1</td>
<td>8</td>
<td>16</td>
<td>2</td>
<td>0</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$idx(diff)$</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Fig. 11. Re-encoding process of the example.

Fig. 12. Diagram of the mapping code.

$diff$ | 0 | 1 | 2 | ... | $2 \times k$ |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$H$</td>
<td>$h_0$</td>
<td>$h_1$</td>
<td>$h_2$</td>
<td>...</td>
<td>$h_{2\times k}$</td>
</tr>
</tbody>
</table>

$H'$ | $h'_0$ | $h'_1$ | $h'_2$ | ... | $h'_{2\times k}$ |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$\overline{diff}$</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>...</td>
<td>-$k$</td>
</tr>
</tbody>
</table>

Fig. 13. Re-encoding map of the proposed scheme.
The algorithm complexity of the proposed scheme includes two parts, namely, pre-process and the embedding process. Steps (2)–(9) of the pre-process include two ‘for’ loops, but it processes each pixel only once. Hence, the algorithm complexity is $O(h \times w)$. Step (10) sorts the histogram. Only $2 \times k$ elements are present in the histogram. Thus, the sorting complexity is $O((2 \times k)^2) = O(k^2)$. The value of $k$ is no more than 8. Hence, the sorting complexity can be ignored.

The complexity of each step in the embedding process is $O(1)$. Thus, the algorithm complexity of the embedding process is the same as that of the pre-process. The total algorithm complexity of the proposed scheme is $O(2 \times (h \times w))$.

3.3. Extraction and recovery

The stego-image $I'$, the numbers of $n$ and $k$, the re-encoding map, and the weighted matrix $wm$ are sent to the receiver. The receiver divides the stego-image into several blocks with sizes of $n \times n$. Each block has four original pixels $I' = \{I'_{1,1}, I'_{1,2}, I'_{2,1}, I'_{2,2}\}$ that are the same as the original image $I = \{I_{1,1}, I_{1,2}, I_{2,1}, I_{2,2}\}$. The original pixels are used to compute the interpolated pixels $C_{i,j}^{ref}$ using Eq. (4) and the modulus value $Val$ with the weighted matrix $wm$ using Eq. (10).

Then, the interpolated pixel is subtracted from the stego-pixel to obtain the concealed message. The equation is as follows:

$$diff = \left| C_{i,j}^{ref} - C_{i,j}^{ref} \right|.$$  \hfill (17)

If $diff$ is larger than $2 \times k + 1$, then the pixel will have an underflow problem and will have added $2 \times (2 \times k + 1)$ previously. Therefore, the value $2 \times (2 \times k + 1)$ needs to be subtracted from the value $diff$ to obtain the original $diff$. Another condition is that if $diff$ is smaller than zero, then the pixel will have an overflow problem and will have subtracted $2 \times (2 \times k + 1)$ previously. There-
Fig. 17. Stego-images of the proposed scheme and the original scheme.
Fig. 18. Image quality comparisons among the proposed scheme and other methods.
fore, the value $2 \times (2 \times k + 1)$ needs to be added to the value $\text{diff}$ to obtain the original $\text{diff}$. Otherwise, $\text{diff}$ is equal to $\text{diff}$. The equation is expressed as follows:

$$
\text{diff} = \begin{cases} 
\text{diff}' - 2 \times (2 \times k + 1), & \text{if } \text{diff}' > 2 \times (2 \times k + 1) \\
\text{diff}' + 2 \times (2 \times k + 1), & \text{if } \text{diff}' < 0 \\
\text{diff}', & \text{otherwise}
\end{cases}
$$

(18)

The re-encoding map is then used to find the corresponding value $\text{diff}$ of $\text{diff}$. The proposed scheme computes the secret symbol using the following equation:

$$
\beta = (\text{diff} + \text{Val}) \% (2 \times k + 1).
$$

(19)

Continuing the example above, in Fig. 10(c), the stego-pixels are $I' = \{84, 86, 88, 81\}$. The interpolated pixel is $C_{(1,2)}^{\text{ref}} = 84$ by using Eq. (4). The modulus value is $\text{Val} = [(84 \times 2) + (86 \times 3) + (88 \times 4) + (81 \times 1)] \mod (2 \times 8 + 1) = 9$ by using Eq. (10). The extracted difference is $\text{diff}' = |C_{(1,2)}^{\text{ref}} - C_{(1,2)}^{\text{ref}}| = 85 - 84 = 1$. As $0 < \text{diff}' < 2 \times (2 \times k + 1)$, the value $\text{diff}$ is equal to $\text{diff}$, where $\text{diff} = 1$. According to the re-encoding map shown in Fig. 11, the corresponding value of $\text{diff}$ is $\text{diff} = 8$. The secret symbol is computed by $\beta = (8 + 9) \% (2 \times 8 + 1) = 0$. The length of the secret bits is $L = \log_2(2 \times 8 + 1) = 4$. Therefore, the secret symbol is transformed to the binary bit string $(0000)_2$ with a length of 4 to obtain the original secret bits.
(a) The original scheme

<table>
<thead>
<tr>
<th>diff</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>...</th>
<th>7</th>
<th>8</th>
<th>sum</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H$</td>
<td>17,953</td>
<td>18,444</td>
<td>17,993</td>
<td>18,069</td>
<td>18,009</td>
<td>...</td>
<td>18,090</td>
<td>17,963</td>
<td></td>
<td></td>
</tr>
<tr>
<td>order</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>...</td>
<td>7</td>
<td>8</td>
<td>3,224,467</td>
<td>12.300</td>
</tr>
<tr>
<td>distortion</td>
<td>0</td>
<td>18,444</td>
<td>71,972</td>
<td>162,621</td>
<td>288,144</td>
<td>...</td>
<td>886,410</td>
<td>1,149,632</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(b) The reEncode scheme

<table>
<thead>
<tr>
<th>$H'$</th>
<th>18,444</th>
<th>18,090</th>
<th>18,069</th>
<th>18,009</th>
<th>17,993</th>
<th>...</th>
<th>17,953</th>
<th>0</th>
<th>sum</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{Idx}(\text{diff})$</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>...</td>
<td>7</td>
<td>8</td>
<td>790,893</td>
<td>3.017</td>
</tr>
<tr>
<td>diff</td>
<td>0</td>
<td>-1</td>
<td>1</td>
<td>-2</td>
<td>2</td>
<td>...</td>
<td>4</td>
<td>-4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>distortion</td>
<td>0</td>
<td>18,090</td>
<td>18,069</td>
<td>72,036</td>
<td>71,972</td>
<td>...</td>
<td>287,248</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig. 19. Re-encoding map among diff, $H$, $H'$, and $\text{diff}$ of Lena with $n=3$ and $k=4$.

Fig. 20. Comparison results of NMI, INP, CRS, Lu, and the proposed method.
where **TSB** is the total number of secret bits embedded in the cover image, and **bpp** means bits per pixel.

### 4.3. Determining the number of \( n \) and \( k \)

In the proposed scheme, the constants \( n \) and \( k \) are used to control the image quality and hiding payload. The first experiment aims to test the performance of the proposed scheme with different \( n \) and \( k \). To examine the efficiency of the re-encoding strategy, the study implements the proposed method without the re-encoding strategy to generate another scheme called the original scheme. The proposed scheme is called reEncode. In the original scheme, the difference \( \text{diff} \) is added directly to the interpolated pixel without re-encoding. The stego-pixel is computed by \( C'_{i,j} = C_{i,j} + \text{diff} \).

**Table 1** compares the original scheme and the reEncode scheme with difference parameters \( n \) and \( k \). The image qualities of the original scheme and the reEncode scheme are 37.23 and 43.31 dBs, respectively, when the block size is set to \( n \times n = 3 \times 3 \). The hiding capacity is 433,500 bits when the hiding bit is set to \( k = 4 \). When the block size is set to \( n \times n = 4 \times 4 \), the image quality is decreased to 35.90 and 41.91 dBs by using the original scheme and the reEncode scheme, respectively. However, the hiding capacity can increase to 589,824 bits with \( k = 4 \). However, this condition

\[
\text{PSNR}(I) = 10 \times \log_{10} \left( \frac{255^2}{\text{MSE}} \right) \text{ (dB)}, \quad \text{and} \quad \text{MSE} = \frac{1}{h \times w} \sum_{i=0}^{h-1} \sum_{j=0}^{w-1} (I_{i,j} - I'_{i,j})^2, \quad (21)
\]

where \( h \) and \( w \) are the image height and width, respectively; \( I \) is the cover image \( \{I_{1,1}, I_{1,2}, \ldots, I_{h,w}\} \); and \( \text{MSE} \) is the mean square errors. A high PSNR value means a low difference between the cover image and the stego-image. Conversely, a low PSNR value means a high difference.

The hiding payload is calculated by

\[
\text{bpp} = \frac{\text{TSB}}{h \times w} \text{ (bpp)}, \quad (22)
\]
Table 1

<table>
<thead>
<tr>
<th>n</th>
<th>k</th>
<th>Original</th>
<th>reEncode</th>
<th>Capacity</th>
<th>bpp</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>4</td>
<td>37.23</td>
<td>41.31</td>
<td>433,500</td>
<td>1.65</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>35.90</td>
<td>41.97</td>
<td>589,824</td>
<td>2.25</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>35.44</td>
<td>41.51</td>
<td>156,060</td>
<td>0.60</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>35.19</td>
<td>41.26</td>
<td>108,375</td>
<td>0.41</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>35.03</td>
<td>41.10</td>
<td>79,935</td>
<td>0.30</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>34.93</td>
<td>41.00</td>
<td>61,440</td>
<td>0.23</td>
</tr>
<tr>
<td>9</td>
<td>4</td>
<td>35.00</td>
<td>41.07</td>
<td>47,040</td>
<td>0.18</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>34.86</td>
<td>40.93</td>
<td>39,015</td>
<td>0.15</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>31.82</td>
<td>37.38</td>
<td>578,000</td>
<td>2.20</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>30.48</td>
<td>36.03</td>
<td>786,432</td>
<td>3.00</td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>30.03</td>
<td>35.58</td>
<td>208,080</td>
<td>0.79</td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>29.78</td>
<td>35.34</td>
<td>144,500</td>
<td>0.55</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>29.62</td>
<td>35.18</td>
<td>106,580</td>
<td>0.41</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>29.51</td>
<td>35.08</td>
<td>81,920</td>
<td>0.31</td>
</tr>
<tr>
<td>9</td>
<td>8</td>
<td>29.59</td>
<td>35.15</td>
<td>62,720</td>
<td>0.24</td>
</tr>
<tr>
<td>10</td>
<td>8</td>
<td>29.45</td>
<td>35.01</td>
<td>52,020</td>
<td>0.20</td>
</tr>
<tr>
<td>3</td>
<td>16</td>
<td>25.20</td>
<td>31.38</td>
<td>722,500</td>
<td>2.76</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>23.87</td>
<td>30.04</td>
<td>983,040</td>
<td>3.75</td>
</tr>
</tbody>
</table>

Lena image. Furthermore, the color of the image is the closest to the original one.

Fig. 19 shows the re-encoding map example among diff, H, H′, and diff of Lena with n = 3 and k = 4. The total numbers of the most frequent occurrence differences diff = 1 and diff = 4 are 18,444 and 18,009, respectively. The original scheme directly adds the values to the interpolated pixels. The distortions made by the differences will be $\text{diff} \times \text{diff} \times H(\text{diff}) = 1 \times 1 \times 18444$ of $\text{diff} = 1$, and $4 \times 4 \times 18090 = 288144$ of $\text{diff} = 4$. The total distortion of the original scheme is 3224467, and the MSE is computed as $\frac{3224467}{18090} = 12.3$.

Considering the same image, the re-Encode scheme sorts H in descending order to obtain H′ and the re-encoded code diff. Hence, $\text{diff} = 1$ and $\text{diff} = 4$ are re-encoded to 0 and 1. Furthermore, the codes are mapped to obtain the mapping codes 0 and 1. The distortions made by the differences are 0 and 18,090. The total distortion of the re-Encoded scheme is 790,893, and MSE is 3.017. We can see that the distortion is greatly decreased such that the image quality of the stego-image is increased effectively.

4.2. Comparison Results

Table 2 shows the comparison between the proposed method with $n = 4$ and $k = 8$ and the four state-of-the-art methods in terms of PSNR and bpu. The PSNR of the proposed scheme is 4 db better than those of the other three methods on the other images and 5 db better than that of Lu’s scheme. However, the PSNR is 2.75 db less than that of NMI on Tiffany.

The comparison figures with different hiding payloads are shown in Fig. 20. The image quality and the hiding payload of the proposed scheme with $n = 4$ and $k = 8$ are better than those of the other methods.

Table 3 shows the execution time comparisons of NMI, INP, CRS, Lu, and the proposed method. The proposed scheme statistics the histogram of the differences and ranking the values. Hence, the execution time (approximately 1.14 s) is worse than that of the other methods. However, the total execution time is still acceptable.

Table 4 shows the comparison between Biswashati’s scheme and the proposed scheme with $n = 4$ and $k = 8$, and $n = 4$ and $k = 4$, respectively. In the high hiding capacity, the PSNR value of the proposed scheme is higher than that of Biswashati’s scheme by approximately 0.23 db. In the low hiding capacity, the PSNR value of the proposed scheme is higher than that of Biswashati’s scheme by approximately 4.73 db.

4.3. Steganalysis

To prove the security of the proposed scheme, steganalysis tests such as histogram steganalysis, RS steganalysis, primary sets, Chi square, sample pairs, RS analysis, and fusion detection are conducted.

Histogram steganalysis is used to compare the shapes of the histograms of the cover image and the stego-image to determine if a message has been concealed in the image. Figs. 21(a) and (b) show the histogram comparison between Lena and Mandrill, respectively. The curve starting with the symbol “*” is the histogram of the stego-image. In the figure, the shape of the stego-image is almost the same as that of the cover image.

RS steganalysis is also conducted. Fridrich et al. [4] proposed RS steganalysis in 2001. In their scheme, pixels are categorized into different groups by using a judgment function and a flipping function. The judgment function determines the smoothness or regularity of each group. The flipping function defines the groups into three different categories: regular (R), singular (S), and unusable (U). The group percentages of regular, singular, and unusable with mask $M = [0 \ 0 \ 0 \ 0]$ and $\neg M = [1 \ 0 \ 0 \ 0]$ are indicated by $R \_M \_G,$
Table 2
PSNR and bpp comparison of NMI, INP, CRS, Lu, and the proposed method for stego-image and cover image.

<table>
<thead>
<tr>
<th>Method</th>
<th>File Name</th>
<th>NMI</th>
<th>INP</th>
<th>CRS</th>
<th>Lu</th>
<th>Original reEncode</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PSNR</td>
<td>bpp</td>
<td>PSNR</td>
<td>bpp</td>
<td>PSNR</td>
<td>bpp</td>
</tr>
<tr>
<td>Airplane</td>
<td>33.05</td>
<td>1.05</td>
<td>32.64</td>
<td>1.19</td>
<td>31.54</td>
<td>1.51</td>
</tr>
<tr>
<td>Tiffany</td>
<td>37.77</td>
<td>0.93</td>
<td>37.15</td>
<td>1.09</td>
<td>36.00</td>
<td>1.37</td>
</tr>
<tr>
<td>Lake</td>
<td>32.48</td>
<td>1.10</td>
<td>31.76</td>
<td>1.25</td>
<td>30.75</td>
<td>1.60</td>
</tr>
<tr>
<td>Lena</td>
<td>34.89</td>
<td>1.10</td>
<td>34.32</td>
<td>1.25</td>
<td>33.19</td>
<td>1.57</td>
</tr>
<tr>
<td>Mandrill</td>
<td>32.40</td>
<td>1.10</td>
<td>31.85</td>
<td>1.25</td>
<td>31.36</td>
<td>1.60</td>
</tr>
<tr>
<td>Pepper</td>
<td>34.27</td>
<td>1.10</td>
<td>33.72</td>
<td>1.25</td>
<td>33.39</td>
<td>1.53</td>
</tr>
</tbody>
</table>

Table 3
Execution time comparisons of NMI, INP, CRS, Lu, and the proposed method.

<table>
<thead>
<tr>
<th>Method</th>
<th>Execution time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NMI</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4
PSNR and bpp comparison between Biswapati’s scheme and the proposed method.

<table>
<thead>
<tr>
<th>Method</th>
<th>File name</th>
<th>Biswapati Capacity</th>
<th>PSNR</th>
<th>reEncode Capacity</th>
<th>PSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Lena</td>
<td>385.80</td>
<td>36.03</td>
<td>560.000</td>
<td>41.97</td>
</tr>
<tr>
<td></td>
<td>Airplane</td>
<td>385.80</td>
<td>36.03</td>
<td>560.000</td>
<td>41.97</td>
</tr>
</tbody>
</table>

R_FM_G, S_M_G, S_FM_G, U_M_G, and U_FM_G have the same situations. Therefore, the proposed scheme cannot be detected by the RS steganalysis.

The experimental results of the primary sets, chi square, sample pairs, RS analysis, and fusion detection [3] are shown in Table 5. All experimental numbers are small, thus indicating that the proposed scheme is secure and robust.

5. Conclusion

This study proposes an interpolation-based reversible hiding scheme by using the difference in the re-encoding strategy. The scheme applies the INP interpolation method to enlarge the original image for generating the cover image. The secret message is then concealed in the interpolated pixel. The proposed scheme uses a modulus function and a weighted matrix to compute a modulus value for hiding secret bits. The difference between the modulus value and the secret message is embedded in the virtual pixel. Before embedding, the differences are re-encoded according to their occurrence number.

The proposed scheme re-encodes the frequency difference with a small value. Conversely, the rare difference is re-encoded with a large value. Furthermore, the proposed scheme uses two parameters n and k to control the image quality and hiding payload, where n×n is the size of a block, and k is the constant used to decide the length of the secret bits. To obtain a high image quality, small n and k are recommended, for example, n = 3 and k = 4. By contrast, to obtain a high hiding payload, large n and k are suggested, for example, n = 4 and k = 8.

Table 5
Steganalysis results of StegExpose.

<table>
<thead>
<tr>
<th>File name</th>
<th>n</th>
<th>k</th>
<th>Primary sets</th>
<th>Chi square</th>
<th>Sample pairs</th>
<th>RS analysis</th>
<th>Fusion (mean)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airplane</td>
<td>3</td>
<td>4</td>
<td>0.00640</td>
<td>0.00329</td>
<td>0.00775</td>
<td>0.00378</td>
<td>0.00530</td>
</tr>
<tr>
<td>Tiffany</td>
<td>3</td>
<td>8</td>
<td>0.00561</td>
<td>0.00581</td>
<td>0.00214</td>
<td>0.00670</td>
<td>0.00506</td>
</tr>
<tr>
<td>Lake</td>
<td>3</td>
<td>8</td>
<td>0.00792</td>
<td>0.00305</td>
<td>0.01047</td>
<td>0.01122</td>
<td>0.00816</td>
</tr>
<tr>
<td>Lena</td>
<td>3</td>
<td>8</td>
<td>0.00865</td>
<td>0.00388</td>
<td>0.00482</td>
<td>0.00844</td>
<td>0.00645</td>
</tr>
<tr>
<td>Mandrill</td>
<td>3</td>
<td>8</td>
<td>0.00765</td>
<td>0.00561</td>
<td>0.00083</td>
<td>0.00890</td>
<td>0.00428</td>
</tr>
<tr>
<td>Pepper</td>
<td>3</td>
<td>8</td>
<td>0.002910</td>
<td>0.00757</td>
<td>0.02454</td>
<td>0.03490</td>
<td>0.02403</td>
</tr>
</tbody>
</table>
Experimental results show that the embedding payload of the proposed method with \( n = 4 \) and \( k = 8 \) is 4 db better than those of INP, NMI, CRS, and more than 5 db better than that of Lu’s scheme for a similar image quality. The image quality of the proposed method with \( n = 3 \) and \( k = 4 \) is 4.73 db better than that of Biswapati’s scheme for a similar hiding payload. The steganalysis results show that the proposed scheme is secure and robust against many attacks.

Acknowledgments

This study was financially supported by a research grant from Taiwan’s Ministry of Science and Technology (MOST 105-2221-E-324-020).

References