A reversible data hiding scheme based on the Sudoku technique

Thai-Son Nguyen \(^{a,b}\), Chin-Chen Chang \(^a\)

\(^a\) Department of Information Engineering and Computer Science, Feng Chia University, Taichung 40724, Taiwan, ROC
\(^b\) Department of Information Technology, Tra Vinh University, Tra Vinh Province, Viet Nam

**Abstract**
Data hiding, also known as information hiding, plays an important role in information security for various purposes. Reversible data hiding is a technique that allows distortion-free recovery of both the cover image and the secret information. In this paper, we propose a new, reversible data hiding scheme that is based on the Sudoku technique and can achieve higher embedding capacity. The proposed scheme allows embedding more secret bits into a pair of pixels while guaranteeing the good quality of the stego-image. The experimental results showed that the proposed scheme obtained higher embedding capacity than some other previous schemes. In addition, our proposed scheme maintained the good visual quality of the stego-image (i.e., PSNR > 46 dB), which outperforms some existing schemes.

1. Introduction
Data hiding, also known as information hiding, is important in information security for various purposes, such as copyright protection and content authentication. The main idea of data hiding is to embed the secret data into the cover media, i.e., images, audio, video or text, to avoid attracting the attention of malicious attackers in the Internet, which is a public channel. In recent years, many data hiding schemes have been proposed, and most of them have been irreversible data hiding schemes [1–3]. This means that after the secret data have been extracted from the stego-image, the cover image is distorted permanently and cannot be restored correctly. However, some special fields, such as the military and medical fields, require that the cover image be restored to its original condition after the secret data have been extracted. As a result, various reversible data hiding schemes [4–7,9–13,16,20] have been proposed to solve this issue. These schemes allow the correct extraction of the secret information and the reconstruction of the cover image without any distortion.

Most reversible data hiding schemes are based on difference expansion (DE) [4,5,9,12,13,18] and histogram shifting [6,7,10,11,19]. In 2003, Tian introduced a reversible DE-based data hiding scheme [4]. In this scheme, the difference between two neighboring pixels is calculated. Then, the difference value is doubled to embed one secret bit. In 2007, Thodi and Rodriguez [5] proposed an expansion scheme based on error prediction (PE) to hide the secret data. In this scheme, a prediction technique is used to predict the pixel value. Then, the difference between the pixel value and its predicted value is calculated to embed the secret data. The above two schemes [4,5] are based on the DE technique to obtain high embedding capacity and good image quality. However, in DE-based schemes, the pixels may have an overflow or underflow problem. To achieve better visual quality, many researchers have proposed histogram-shifting schemes. In 2006, Ni et al. proposed the first histogram-shifting scheme [6]. In Ni et al.’s scheme, most of the pixels are modified by one grayscale value to embed secret data. Their scheme achieved high visual quality of the stego-image, but the embedding capacity is limited. In 2009, Tai et al. [11] proposed a new reversible data hiding scheme based on histogram modification. In Tai et al.’s scheme, a binary tree structure is designed to determine the peak point, which is used for embedding data. To extract the secret data, the level of the binary tree must be sent to the receiver. Also in 2009, Kim et al. [7] introduced a reversible data hiding scheme based on the different histogram shifting to obtain high capacity and imperceptible embedding by dividing the cover image into several sub-images. The difference values between the sub-sampled images are calculated. Then, the difference values are shifted to embed more secret data. To improve Kim et al.’s scheme, Luo et al. [9] proposed a reversible data hiding scheme based on selecting the median pixel of each block to structure the reference sub-image. They separated the image blocks into four categories, and, for each category, one corresponding embedding strategy is used to hide the secret message. However, the embedding capacity of their scheme also is limited because the reference pixels are not used for embedding secret data. To obtain better embedding...
capacity, Lou et al. [13] designed a novel technique by using a multiple-layer embedding technique and a logarithm transformation function to hide secret data into the reduced difference expansion. In 2010, Li et al. [10] proposed a reversible data hiding scheme based on the difference between adjacent pixels to embed the secret data. In Li et al.'s scheme, the difference sequence of pixels is explored for concealing the secret data. In order to increase the hiding capacity and maintain good image quality, Qin et al. introduced a new scheme in 2012 [12] that exploited the relationship between the prediction error and the threshold to decide whether the current pixel is embeddable or not. To provide a general framework for histogram shifting-based reversible data hiding, Li et al. [19] designed two simple functions, i.e., shifting and embedding functions, for reversible data hiding. Then, based on these two functions, two different reversible data hiding schemes are introduced. These two schemes are expected to further improve the performance of the previous works. Their scheme I obtained high embedding capacity. However, their scheme II only worked well for low embedding capacity. Instead of using histogram shifting for embedding data as was done in [19], Gui et al. [18] used prediction-error expansion for reversible data hiding. In Gui et al.'s scheme, the complexity measurement is divided into various levels, and the size of the embedded data is calculated. Then, the smoother regions are used to embed more secret bits. However, the quality of the stego images in this scheme is unsatisfactory. The average PSNR of their scheme is always less than 40 dB for the embedding rate of 1 bit per pixel (bpp).

In this paper, to achieve high embedding capacity and good quality of the stego-image, we proposed a new, reversible data hiding scheme based on a reference matrix, which is constructed by using the Sudoku technique. According to Sudoku's properties, the proposed scheme can embed the secret digit in the base-9 numeral system into each pixel pair of the grayscale cover image. Hence, every digit in the Sudoku grid is reduced by one to generate the sub-matrix $S$, which contains digits from 0 to 8, as shown in Fig. 2. Then, the sub-matrix $S$ is used to construct the reference matrix $RM$. Fig. 3 shows an example of the reference matrix $RM$, which is used for embedding and extracting secret data.

3. The proposed scheme

After carefully observing Chang et al.'s scheme [15], we found that it was based on Sudoku's properties to obtain high embedding capacity and good image quality. The properties of the Sudoku technique are the key features that allow the achievement of high embedding capacity and good image quality at the same time. However, the disadvantage of Chang et al.'s scheme is that the cover image is damaged permanently and cannot be recovered after the secret information has been extracted from it. In order to solve this problem, in this section, we present our design of a new, reversible data embedding scheme based on the properties of Sudoku technique to obtain the goals of improving security, increasing embedding capacity, and, especially, obtaining reversibility while guaranteeing good visual quality of the stego-image. Our reversible data hiding scheme is divided into two phases, i.e., the data embedding phase and the data extracting phase, which are described in Subsections 3.1 and 3.2, respectively.

3.1. Data embedding phase

In this section, the pixels of the cover image are first paired by using a pairing technique, and each pair of pixels is used to embed a secret digit. To extract the embedded secret digits and restore the original cover image, the decoder must determine which pair of pixels was selected for embedding the secret digit. To facilitate this process, location information must be embedded so the decoder can use it for extracting the secret data. To accomplish this, we generated a 2-D binary location map $LM$, which contains the location information of all embedded pairs. The location map $LM$ is initialized with "0."

The cover image $I$ is divided into two areas, i.e., the embeddable area and the non-embeddable area, which are shown in Fig. 4. The least significant bits (LSBs) of the pixels in the non-embeddable area are used to record the information of the location map $LM$. Therefore, they must be extracted and concatenated into the secret data $B$ to generate the embedded data, $S$.

For use in our proposed scheme, the reference matrix $RM$ must be prepared as shown in Fig. 3, and the embedded data $S$ must be converted into based-9 numeral system digits. Let us denote the converted secret digits $D = \{d_1, d_2, \ldots, d_N\}$, where $N$ is the total number of secret digits.

![Fig. 1. Example Sudoku solution.](Image 1)

![Fig. 2. Sub-matrix $S$.](Image 2)
number of secret digits, and \( d_i \in [0, 8], 1 \leq i \leq N \). Fig. 5 shows the flowchart of the data embedding phase.

The embedding algorithm is described in detail below:

Step 1: The cover image \( I \) is divided into two areas, i.e., the embeddable area and the non-embeddable area as shown in Fig. 4. The non-embeddable area consists of the lowest four rows of the cover image. The LSBs of the pixels in the non-embeddable area are used to record the information of the location map \( LM \). Therefore, the LSBs of the pixels in the non-embeddable area must be extracted and concatenated into the secret data \( B \) to generate the embedded data \( S \). The pixels in the embedding area are divided into pairs of pixels \((I_{2i}, I_{2i+1})\) for data embedding.

Step 2: The reference matrix \( RM \) is constructed and the embedded data \( S \) are converted into base-9 numeral system digits \( D \). In addition, the pixels in the embeddable area are partitioned into pairs of pixels.

Step 3: For each pixel pair consisting of two consecutive pixels \((I_{2i}, I_{2i+1})\) in the embedding area that is located in the reference matrix \( RM \) at the row \( I_{2i} \) and the column \( I_{2i+1} \), meaning that indices of the reference matrix \( RM \) are pixel values. From the location of the pair of pixels \((I_{2i}, I_{2i+1})\) in the reference matrix \( RM \), two sets of referred elements, \( VE \) and \( HE \), are constructed in Steps 3.1 and 3.2, respectively. Then, go to Step 4 to continue.

---

**Step 3.1:** (Select eight referred elements for \( VE \))

- If \( I_{2i} > 7 \), then \( VE = \{RM(I_{2i}/8, I_{2i+1}), RM(I_{2i}/7, I_{2i+1}), RM(I_{2i}/6, I_{2i+1}), RM(I_{2i}/5, I_{2i+1}), RM(I_{2i}/4, I_{2i+1}), RM(I_{2i}/3, I_{2i+1}), RM(I_{2i}/2, I_{2i+1}), RM(I_{2i}/1, I_{2i+1}), RM(I_{2i}, I_{2i+1})\} \).

**Step 3.2:** (Select eight referred elements for \( HE \))

- If \( I_{2i} < 248 \), then \( HE = \{RM(I_{2i}, I_{2i+1} + 8), RM(I_{2i}, I_{2i+1} + 7), RM(I_{2i}, I_{2i+1} + 6), RM(I_{2i}, I_{2i+1} + 5), RM(I_{2i}, I_{2i+1} + 4), RM(I_{2i}, I_{2i+1} + 3), RM(I_{2i}, I_{2i+1} + 2), RM(I_{2i}, I_{2i+1} + 1), RM(I_{2i}, I_{2i+1})\} \).

**Step 3.3:** If \( I_{2i} \leq 7 \) or \( I_{2i+1} > 248 \), then the corresponding location of the current pair of pixels in the location map \( LM \) is modified to “1.”

Step 4: The secret digit \( d_i \) is read from the secret digits \( D \). According to \( d_i \), two referred elements, \( RM(X_{VE}, Y_{VE}) \) and \( RM(X_{HE}, Y_{HE}) \), are found from \( VE \) and \( HE \), respectively, such that \( RM(X_{VE}, Y_{VE}) = RM(X_{HE}, Y_{HE}) = d_i \).

**Step 5:** The candidate element \((X, Y)\) is calculated by using Eq. (1).

\[
X = \frac{X_{VE} + X_{HE}}{2}, \quad Y = \frac{Y_{VE} + Y_{HE}}{2}.
\]

**Step 6:** If \( X \) or \( Y \) is a decimal fraction, the pair of pixels cannot be used to embed the secret digit \( d_i \). Therefore, the corresponding location in the location map \( LM \) is changed to “1.”

**Step 7:** Otherwise, the original pixel pair is modified to the candidate element \((X, Y)\) to embed the secret digit \( d_i \).
Step 8: Repeat Steps 3 through 7 until all secret digits have been embedded.

After all steps have been completed, the location map \(LM\) will be compressed losslessly by using JBIG-kit in [8]. The compressed bitstream is denoted as \(L\), which is embedded into the lowest four rows of the cover image, known as the non-embeddable area, by using LSBs substitution. Finally, the stego-image is sent to the receiver.

Notice that the size of the location map is based on the content of different images. Since, for six test images, the location map is determined during embedding process. Then, this location map is compressed losslessly by using JBIG-kit to construct the compressed bitstream \(L\). According to the experiment, the size of \(L\) is calculated and it is always smaller than 5000 bits. Therefore, LSBs of pixels of the lowest four rows are experimentally used to record the location map in the proposed scheme.

We now discuss how to improve the security of the reference matrix \(RM\) which is helpful for practical applications. Actually, the reference matrix is generated from the selected Sudoku solution [15]. To the best of our knowledge, Russell and Jarvis [14] found that the number of different Sudoku solutions would be 5,472,730,538. Accordingly, many different reference matrices can be obtained for embedding data. Therefore, instead of one reference matrix \(RM\) or multiple reference matrices are shared in the public channel for embedding data, a secret key \(K\) is needed in the system. The secret key is used to select the suitable Sudoku solution to construct the corresponding reference matrix, and this key is shared between the sender and the receiver in advance. To do so, the security of the proposed scheme is guaranteed.

To better explain the data embedding phase of the proposed scheme, an example is provided in Fig. 6. Assume that the original pair of pixels \((I_{2i}, I_{2i+1})\) of the cover image is \((60, 55)\), and that the secret digit \(d_i\) is 3. According to Step 3, the pair of pixels is located onto reference matrix \(RM\), and the two referred sets, \(VE\) and \(HE\), are generated. Then, the referred elements, \(RM(VE, Y_{VE})\) and \(RM(HE, Y_{HE})\), are determined from two sets, \(VE\) and \(HE\), such that \(RM(54, 55) = RM(60, 63) = 3\). Then, the candidate element \((X, Y)\) is calculated by using Eq. (1), i.e., \((X, Y) = (57, 59)\). To embed the secret digit \(d_i = 3\), the original pair of pixels \((60, 55)\) is modified to the candidate element \((57, 59)\).

3.2. Data extracting phase

After receiving the stego-image \(I\) from the sender, a receiver can extract the secret digits \(D\) and restores the cover image \(I\) by following the steps in the data extracting phase. The flowchart of the data extracting phase is shown in Fig. 7.

Step 1: The reference matrix \(RM\) is constructed, and the stego-image \(I\) is divided into two areas, i.e., the embeddable area and the non-embeddable area, in the same manner used in the data embedding phase. Then, the LSBs of the pixels in the non-embeddable area are extracted to retrieve the compressed
bitstream \( L \), which is decoded by using JBIG-kit in [8] to reconstruct the location map \( LM \).

Step 2: The pixels in the embeddable area are partitioned into pairs of pixels in the same pairing technique used in the data embedding phase. For each pair of stego pixels \((I_{2i}, I_{2i+1})\), the corresponding location map value is checked.

Step 3: If the corresponding location map value is “0,” the secret digit \( d_i \) has been embedded in the current pair of stego pixels \((I_{2i}, I_{2i+1})\). Then, the current pair of stego pixels is located onto the reference matrix \( RM \) at the row \( I_{2i} \) and the column \( I_{2i+1} \).

From the location of the pair of stego pixels \((I_{2i}, I_{2i+1})\) in the reference matrix \( RM \), two sets, \( U \) and \( L \), are generated as shown in Fig. 8. Then, for each location \( U_j \) in set \( U \), the corresponding location in set \( L \) is searched to find location \( L_j \), where \( L_j \) is the symmetric location of \( U_j \) with respect to \((I_{2i}, I_{2i+1})\), and to verify that the value at location \( L_j \) is equal to the value at location \( U_j \) in the reference matrix \( RM \). After the locations \( U_j \) and \( L_j \) are found, the secret digit \( d_i \) is extracted as the value at location \( U_j \) or the value at location \( L_j \) in the reference matrix \( RM \), two referred sets, \( VE \) and \( HE \), are generated. Then, the original pair of pixels \((I_i, I_{i+1})\) is reconstructed as the location of the intersection of the two sets, \( VE \) and \( HE \), as shown in Fig. 9.

Step 4: If the corresponding location map value is “1,” no secret digit is extracted in this case. Therefore, the pair of original pixels is recovered as \((I_{2i}, I_{2i+1})\).
Step 5: Repeat Steps 2 through 4 until all secret digits have been extracted.

After the secret digits \( D \) have been extracted completely, the embedded data \( S \) are regenerated by converting the secret digits \( D \) into binary information. Then, to reconstruct the original cover image \( I \), the LSBs of the pixels in the non-embeddable area are recovered from the embedded data \( S \) by substituting the LSBs.

To illustrate the data extracting phase clearly, an example is given as shown in Fig. 9. Assume that the pair of stego pixels, \((I_{2i}, I_{2i+1}) = (57, 59)\), is located onto the reference matrix \( RM \). Then, two sets, \( U \) and \( L \), are generated. It is easy to determine the location \( U_i \) (54, 55) and the location \( L_i \) (60, 63) from the sets, \( U \) and \( L \), respectively. Here, the location \( L_i \) is the symmetric location of \( U_i \) with respect to the location (57, 59), and the value at location \( L_i \) is equal to the value at location \( L_i \) such that \( RM(54, 55) = RM(60, 63) = 3 \). The secret digit \( d_i \) is extracted as 3. From the locations of \( U_i \) and \( L_i \), i.e. (54, 55) and (60, 63), the two referred sets, \( VE \) and \( HE \), are constructed in the reference matrix \( RM \). Then, the pair of original pixels is recovered as the intersection location (60, 55) of the two sets, \( VE \) and \( HE \), in the reference matrix \( RM \).

4. Experimental results

This section describes the experiments that were conducted to demonstrate the embedding capacity and visual quality of our proposed scheme as compared to those of some previous schemes. Six 512 \( \times \) 512 grayscale test images, from smooth to complex, were used as input test images in our simulation, as shown in Fig. 10. In our experiments, we used a pseudo-random bit generator to yield the secret data \( B \). All computing was performed on a PC with a 2.1 GHz Intel\textsuperscript{®} Core\textsuperscript{™}, 2 CPU, and a 1 GB of RAM. The operating system was Windows 7 Professional, and the proposed algorithm was programmed with Microsoft Visual Studio 2005 C#.

The peak signal-to-noise ratio (PSNR) was used to measure the quality of the stego-image compared with the original cover image. A large PSNR value indicates that the scheme obtained good visual quality of the stego-image, i.e., there was only a small amount of distortion. Conversely, a small PSNR value indicates that the stego-image had poor visual quality (large distortion). The PSNR is defined in Eq. (2).

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

where the mean square error (MSE) for a \( W \times H \) grayscale image is defined as shown in Eq. (3).

\[
\text{MSE} = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} (X_{ij} - Y_{ij})^2,
\]

where \( X_{ij} \) and \( Y_{ij} \) are the pixel values of the test image and the stego-image, respectively.

In our scheme, to embed the secret digit \( d_i \in [0, 8] \), the pair of pixels \((I_{2i}, I_{2i+1})\) is mapped into the reference matrix. Then, two points, \( RM(X_{VE}, Y_{VE}) \) and \( RM(X_{HE}, Y_{HE}) \), are determined in two sets of referred elements, \( VE \) and \( HE \), respectively, such that \( RM(X_{VE}, Y_{VE}) = RM(X_{HE}, Y_{HE}) = d_i \). A right triangle is constructed in the reference matrix according to three points \( RM(I_{2i}, I_{2i+1}), RM(X_{VE}, Y_{VE}), \) and \( RM(X_{HE}, Y_{HE}) \). Then, the midpoint \((X, Y)\) of the subtense \( [RM(X_{VE}, Y_{VE}), RM(X_{HE}, Y_{HE})] \) is determined in the right triangle. Eventually, the pair of pixels \((I_{2i}, I_{2i+1})\) is modified to the pair of \((X, Y)\) to
Table 1
Comparison of results for the six schemes in terms of capacity and PSNR value.

<table>
<thead>
<tr>
<th>Schemes</th>
<th>Metric</th>
<th>Lena (bits)</th>
<th>F16 (bits)</th>
<th>Baboon (bits)</th>
<th>Peppers (bits)</th>
<th>Zelda (bits)</th>
<th>Barbara (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kim et al.’s scheme [7]</td>
<td>Capacity (bits)</td>
<td>61,832</td>
<td>81,173</td>
<td>20,965</td>
<td>59,707</td>
<td>63,399</td>
<td>38,289</td>
</tr>
<tr>
<td></td>
<td>PSNR (dB)</td>
<td>44.2</td>
<td>44.55</td>
<td>43.63</td>
<td>44.16</td>
<td>44.2</td>
<td>43.85</td>
</tr>
<tr>
<td></td>
<td>SSIM</td>
<td>0.9412</td>
<td>0.9046</td>
<td>0.9903</td>
<td>0.9481</td>
<td>0.954</td>
<td>0.9673</td>
</tr>
<tr>
<td></td>
<td>PSNR (dB)</td>
<td>48.68</td>
<td>48.83</td>
<td>48.5</td>
<td>48.67</td>
<td>48.85</td>
<td>48.52</td>
</tr>
<tr>
<td></td>
<td>SSIM</td>
<td>0.9746</td>
<td>0.952</td>
<td>0.9966</td>
<td>0.9779</td>
<td>0.98</td>
<td>0.9863</td>
</tr>
<tr>
<td>Li et al.’s scheme [10]</td>
<td>Capacity (bits)</td>
<td>60,817</td>
<td>79,295</td>
<td>23,946</td>
<td>62,255</td>
<td>54,460</td>
<td>33,890</td>
</tr>
<tr>
<td></td>
<td>PSNR (dB)</td>
<td>48.67</td>
<td>48.75</td>
<td>48.33</td>
<td>48.68</td>
<td>48.6</td>
<td>48.42</td>
</tr>
<tr>
<td></td>
<td>SSIM</td>
<td>0.9739</td>
<td>0.9509</td>
<td>0.9960</td>
<td>0.9776</td>
<td>0.9795</td>
<td>0.9858</td>
</tr>
<tr>
<td>Tai et al.’s scheme [11]</td>
<td>Capacity (bits)</td>
<td>22,377</td>
<td>45,472</td>
<td>9,818</td>
<td>33,393</td>
<td>38,546</td>
<td>26,324</td>
</tr>
<tr>
<td></td>
<td>PSNR (dB)</td>
<td>48.32</td>
<td>48.53</td>
<td>48.21</td>
<td>48.42</td>
<td>48.26</td>
<td>48.34</td>
</tr>
<tr>
<td></td>
<td>SSIM</td>
<td>0.9762</td>
<td>0.9543</td>
<td>0.9966</td>
<td>0.9795</td>
<td>0.9819</td>
<td>0.9872</td>
</tr>
<tr>
<td>Qin et al.’s scheme [12]</td>
<td>Capacity (bits)</td>
<td>104,807</td>
<td>117,586</td>
<td>31,457</td>
<td>117,954</td>
<td>109,482</td>
<td>116,284</td>
</tr>
<tr>
<td></td>
<td>PSNR (dB)</td>
<td>45.75</td>
<td>46.31</td>
<td>44.21</td>
<td>46.42</td>
<td>45.89</td>
<td>46.25</td>
</tr>
<tr>
<td></td>
<td>SSIM</td>
<td>0.9503</td>
<td>0.9218</td>
<td>0.9912</td>
<td>0.9572</td>
<td>0.959</td>
<td>0.9861</td>
</tr>
<tr>
<td>Proposed scheme</td>
<td>Capacity (bits)</td>
<td>119,633</td>
<td>118,665</td>
<td>117,228</td>
<td>119,457</td>
<td>120,567</td>
<td>122,292</td>
</tr>
<tr>
<td></td>
<td>PSNR (dB)</td>
<td>48.67</td>
<td>48.75</td>
<td>48.81</td>
<td>48.7</td>
<td>48.65</td>
<td>48.59</td>
</tr>
<tr>
<td></td>
<td>SSIM</td>
<td>0.9742</td>
<td>0.9513</td>
<td>0.9954</td>
<td>0.9784</td>
<td>0.9792</td>
<td>0.98564</td>
</tr>
</tbody>
</table>

Fig. 11. Comparison of multilayer embedding of the four test images.
embed the secret digit $d_i$ (see Fig. 6). The modification of embedding the secret digit $d_i$ can be calculated by

\[
m = \frac{\text{side}(\text{RM}(X_{\text{HE}}, Y_{\text{VE}}), \text{RM}(X_{\text{HE}}, Y_{\text{HE}}))}{2} - \frac{\sqrt{\text{side}(\text{RM}(I_{2i}, I_{2i+1}), \text{RM}(X_{\text{HE}}, Y_{\text{HE}}))^2 + \text{side}(\text{RM}(I_{2i}, I_{2i+1}), \text{RM}(X_{\text{HE}}, Y_{\text{VE}}))^2}}{2},
\]

(4)

where $m$ is the modification value of embedding the digit $d_i$ and side ($A_1, A_2$) is the distance function that is used to calculate the distance of two points $A_1$ and $A_2$. Since two sets, $\text{VE}$ and $\text{HE}$, contain only eight elements, thus the value of $\text{side}(\text{RM}(I_{2i}, I_{2i+1}), \text{RM}(X_{\text{HE}}, Y_{\text{HE}}))$ and $\text{side}(\text{RM}(I_{2i}, I_{2i+1}), \text{RM}(X_{\text{HE}}, Y_{\text{VE}}))$ is always smaller than 8. This is because $\text{RM}(I_{2i}, I_{2i+1})$ and $\text{RM}(X_{\text{HE}}, Y_{\text{VE}})$ are two elements of $\text{VE}$ and $\text{RM}(I_{2i}, I_{2i+1})$ and $\text{RM}(X_{\text{HE}}, Y_{\text{HE}})$ are two elements of $\text{HE}$, respectively. Therefore, the modification of $m$ obtained by Eq. (4) is less than 5.66. According to statistically analysis on six test images, the average number of pixels is used for embedding data which is around 28% of the pixels of the image. As a consequence, the PSNR value of stego-images will be greater than 46 dB for single layer embedding. Table 1 shows the comparison of the results of embedding capacity and the corresponding PSNRs of our proposed scheme and Kim et al.’s scheme [7], Luo et al.’s scheme [9], Li et al.’s scheme [10], Tai et al.’s scheme [11], and Qin et al.’s scheme [12]. As Table 1 shows, the embedding capacity of our proposed scheme was greater than 110,000 bits, which is significantly better than the other five schemes. This is because the proposed scheme is based on the reference matrix, so, when a suitable pair of pixels is identified, the secret digit is embedded in it. In addition, the embedding capacity and the visual quality of the proposed scheme outperformed those of the other five schemes in all cases, regardless of whether the images were smooth or complex. Furthermore, we also used structural similarity index (SSIM) [17] to further evaluate the image quality of the stego images obtained by the proposed scheme and previous schemes [7,9–12]. As can be seen in Table 1, for SSIM, two schemes [9,11] achieved slightly better than the proposed scheme in average. However, the proposed scheme offered four times embedding capacity as large as in [9,11]. In addition, the SSIM value obtained by the proposed scheme is also greater than that of other three schemes [7,10,12].

Fig. 11(a)–(d) shows that the multi-layer embedding capacity and the PSNR value of the proposed scheme were greater than those of the other schemes. Basically, for the test images, “Lena”, “F16”, “Baboon”, and “Peppers”, the top curve is the curve for our proposed scheme. At the same PSNR value, the proposed scheme achieved higher embedding capacity than the other seven schemes [7,9–12,18,19]. The proposed scheme also had better image quality than the other seven schemes. Gui et al.’s scheme [18] and Li et al.’s scheme [19] provided better visual quality of the stego-image and larger embedding capacity for all of images than other five schemes [7,9–12]. For the Lena, F16, and Baboon images, Li et al.’s scheme [19] always had better results than Gui et al.’s scheme [18], as shown in Fig. 11(a)–(c). However, for the Peppers image, Gui et al.’s scheme had a greater embedding capacity than Li et al.’s scheme, when the embedding layer was increased, as shown in Fig. 11(d). However, the proposed scheme was always superior to the schemes of Gui et al. [18] and Li et al. [19] in both embedding capacity and visual quality of the stego-image. Fig. 11 also shows that the proposed scheme achieved more consistent results than the other seven schemes for the different images. In a complex image, i.e., Baboon, the embedding capacity of the proposed scheme was significantly greater than that of Gui et al.’s scheme, as shown in Fig. 11(c), because our proposed scheme was based on a reference matrix. Therefore, the pixel values in the proposed scheme were modified by small amounts to embed one secret digit. In contrast, Gui et al.’s scheme doubled the prediction error to embed only one secret bit.

5. Conclusions

In this paper, a new, reversible data hiding scheme based on the Sudoku technique is proposed to obtain high embedding capacity and good image quality. Our experimental results showed that, the proposed scheme had larger embedding capacity than the other seven schemes, when the average embedding capacity was greater than 110,000 bits. The proposed scheme also achieved good visual quality of the stego-image, i.e., the PSNR value of the proposed scheme was greater than 46 dB for all of the test images. In addition, the proposed scheme achieved more consistent results for the different test images than the other five schemes.

References


