A reversible data hiding scheme using image interpolation

Xiang-Guang Xiong
School of Big Data and Computer Science, Guizhou Normal University, Baoshan North Road, Guiyang 550001, P. R. China
E-mail: xxg0851@163.com

Reversible data hiding scheme can completely recover the original cover image after the embedded secret data have been correctly extracted, which has become a hot topic of data hiding. However, the embedding capacities of these existing reversible data hiding schemes are usually rather limited. In this paper, an improved reversible data hiding scheme based on image interpolation technique is proposed, which can embed a large amount of secret bits into a cover image. The experimental results show that the proposed scheme can obtain better performance in terms of embedding capacity and image quality compared with other similar schemes based on image interpolation.

Keywords: Image Interpolation; Reversible Data Hiding; High-Capacity.

1.Introduction

In recent years, many reversible data hiding schemes have been proposed. In general, these reversible schemes can be classified into four types: reversible data hiding using lossless compression [1-2], reversible data hiding using difference expansion (DE) [3-4], reversible data hiding using histogram shifting (HS) [5], and reversible data hiding using image interpolation [6-7].

In 2003, Tian [3] proposed a reversible data hiding scheme based on DE technique, which can provide a higher capacity while keeping a lower distortion compared with previous schemes [1-2]. Afterwards, Coltuc [4] improved it by using context embedding. Besides DE technique, the HS technique was another effective method for reversible data hiding. The first reversible data hiding scheme based on HS was proposed by Ni et al. [5]. In this scheme, the peak point of histogram was used to embed data and can achieve a very good image quality. However, its embedding capacity was rather low.

Unlike DE and HS schemes, Jung et al. [6] proposed a scheme based on interpolation technique. Later on, this scheme has been improved by Lee et al. [7]. In [7], in order to enhance the capacity, it improved the interpolation method, using 3×3 overlapping block to embed data instead of 2×2 non-overlapping block

and selected the maximum pixel among four corner pixels in each block as a pivot pixel to compute the difference of each non-pivot pixel. In this paper, a new scheme is proposed. Experimental results show that it has better performance.

2.Proposed Scheme

2.1.*Proposed neighbor mean interpolation*

(1) The low-resolution original image O is directly down-sampled from the input image I by Eq. (1) as below.

$$O(i,j) = I(2i-1, 2j-1), 1 \le i \le w/2, 1 \le j \le h/2 \tag{1}$$

(2) Except the last row and the last column of image O, the image interpolation process can be described in Eq. (2) as follows.

$$\begin{cases} C(2i-1, 2j-1) = O(i,j) \\ C(2i-1, 2j) = \lfloor (O(i,j) + O(i, j+1))/2 \rfloor \\ C(2i, 2j-1) = \lfloor (O(i,j) + O(i+1, j))/2 \rfloor \\ C(2i, 2j) = \lfloor (O(i+1, j) + O(i, j+1))/2 \rfloor \end{cases} \tag{2}$$

where $\lfloor \bullet \rfloor$ represents the floor function.

(3) For the last column of image O (except the last pixel), the interpolation process can be computed as

$$\begin{cases} C(2i-1, 2j-1) = O(i,j) \\ C(2i-1, 2j) = O(i,j) \\ C(2i, 2j-1) = \lfloor (O(i,j) + O(i+1, j))/2 \rfloor \\ C(2i, 2j) = \lfloor (O(i,j) + O(i+1, j))/2 \rfloor \end{cases} \tag{3}$$

(4) For the last row of image O (except the last pixel), the interpolation process can be computed as

$$\begin{cases} C(2i-1, 2j-1) = O(i,j) \\ C(2i-1, 2j) = O(i,j) \\ C(2i, 2j-1) = \lfloor (O(i,j) + O(i, j+1))/2 \rfloor \\ C(2i, 2j) = \lfloor (O(i,j) + O(i, j+1))/2 \rfloor \end{cases} \tag{4}$$

(5) For the last row and the last column pixel of image O, the interpolation process can be computed as

$$\begin{cases} C(2i-1, 2j-1) = O(i,j) \\ C(2i-1, 2j) = O(i,j) \\ C(2i, 2j-1) = O(i,j) \\ C(2i, 2j) = O(i,j) \end{cases} \tag{5}$$

Note that, the pixel $O(i,j) = I(2i-1, 2j-1)$, $1 \le i \le w/2$, $1 \le j \le h/2$ is defined as a pivot pixel which will never be changed when the secret data embedding is performed.

2.2.*Data embedding procedure*

(1) Compute the maximum value *Max* and the minimum value *Min* among four corner (pivot) pixels of each 3×3 pixel block as follows.

$$\begin{cases} Max = \max\{C(i,\mathrm{j}), C(i,\mathrm{j}+2), C(i+2,\mathrm{j}), C(i+2,\mathrm{j}+2)\} \\ Min = \min\{C(i,\mathrm{j}), C(i,\mathrm{j}+2), C(i+2,\mathrm{j}), C(i+2,\mathrm{j}+2)\} \end{cases} \tag{6}$$

(2) After the maximum value *Max* and the minimum value *Min* are obtained, the difference values between each non-pivot pixel and *Max* and *Min* are computed, respectively. These difference values are computed as below.

$$\begin{cases} d_{11} = \mathrm{abs}(C(i,j+1) - Max) \\ d_{12} = \mathrm{abs}(C(i+1,j) - Max) \\ d_{13} = \mathrm{abs}(C(i+1,j+1) - Max) \end{cases} \tag{7}$$

$$\begin{cases} d_{21} = \mathrm{abs}(C(i,j+1) - Min) \\ d_{22} = \mathrm{abs}(C(i+1,j) - Min) \\ d_{23} = \mathrm{abs}(C(i+1,j+1) - Min) \end{cases} \tag{8}$$

where $\mathrm{abs}(x)$ represents a function which returns the absolute value of *x*.

(3) Compute the summation D_1 and D_2 of these difference values, respectively. The values D_1 and D_2 can be obtained as follows.

$$\begin{cases} D_1 = d_{11} + d_{12} + d_{13} \\ D_2 = d_{21} + d_{22} + d_{23} \end{cases} \tag{9}$$

(4) If D_1 larger than D_2, d_{11}, d_{12} and d_{13} are selected as the final difference values. Otherwise, d_{21}, d_{22} and d_{23} are selected as the final difference values. That is

$$d_k = \begin{cases} d_{1k}, if\ D_1 \ge D_2 \\ d_{2k}, else \end{cases}, k = 1,2,3 \tag{10}$$

(5) After the difference values d_k are obtained, the amount of secret data in bits can be computed, and denoted as n_k which defined as

$$n_k = \begin{cases} 1 & , if\ d_k = 0 \| d_k = 1 \\ \lfloor \log_2 d_k \rfloor, else \end{cases}, k = 1,2,3 \tag{11}$$

(6) Assume b_k represent a bit string of n_k secret data bits to be embedded. For each non-pivot pixel $C(i, j+1)$, $C(i+1, j)$ and $C(i+1, j+1)$, the corresponding watermarked pixels can be obtained via

$$\begin{cases} \begin{cases} S(i, j+1) = C(i, j+1) + b_1 & , \ if \ \ C(i, j+1) + b_1 \le 255 \\ S(i, j+1) = C(i, j+1) - b_1 & , else \end{cases} \\ \begin{cases} S(i+1, j) = C(i+1, j) + b_2 & , \ if \ \ C(i+1, j) + b_2 \le 255 \\ S(i+1, j) = C(i+1, j) - b_2 & , else \end{cases} \\ \begin{cases} S(i+1, j+1) = C(i+1, j+1) + b_3 , & if \ \ C(i+1, j+1) + b_3 \le 255 \\ S(i+1, j+1) = C(i+1, j+1) - b_3 , & else \end{cases} \end{cases} \tag{12}$$

(7) Repeat Steps 1-6 until the last block has been performed or the secret data has been embedded.

2.3.*Data extracting procedure*

(1) The original image *O* can be recovered directly from the watermarked image *S* by Eq. (13) as follows.

$$O(i, j) = S(2i-1, 2j-1), 1 \le i \le w/2, 1 \le j \le h/2 \tag{13}$$

(2) Using image *O*, the cover image *C* is obtained by proposed image interpolation method defined in Section 2.1.

(3) Similar with Steps 1-5 of data embedding process, the amount of embedded secret data in bits for each pivot pixel $C(i, j+1)$, $C(i+1, j)$ and $C(i+1, j+1)$ can be computed, and denoted as n_k .

(4) Compute the difference values between the watermarked pixel and original pixel. These difference values are computed as follows.

$$\begin{cases} b_1 = abs(S(i, j+1) - C(i, j+1)) \\ b_2 = abs(S(i+1, j) - C(i+1, j)) \\ b_3 = abs(S(i+1, j+1) - C(i+1, j+1)) \end{cases} \tag{14}$$

(5) According to n_k and b_k , the embedded sub-secret data s_k can be generated as below.

$$s_k = \mathrm{dec2bin}(b_k, n_k), k = 1, 2, 3 \tag{15}$$

where $\mathrm{dec2bin}(x, y)$ represents a function which returns a binary representation of *x* with at least *y* bits.

(6) Merge the bits in s_k to form the extracted secret data and repeat Steps 1-6 until the last block has been performed or the secret data has been extracted.

3.Experimental Results

In this section, the experimental results are given. Two 512×512 standard test images were used as input image as shown in Fig. 1. The random generated bit stream was used as the to-be-embedded watermarking signal.

(a) Original Barbara (b) Watermarked Barbara (c) Original Goldhill (d) Watermarked Goldhill

Fig. 1 Original image and watermarked image

In the proposed data hiding scheme, each input gray-scale image *I* was firstly scaled down to 1/4 of its original size and then scaled up to original size by proposed image interpolation method to generate an enlarged image *C* for hiding secret data. The watermarked image is shown in Fig.1. As shown in Fig.1, the visual qualities of these watermarked images are satisfactory. In addition, Table 1 shows the comparison results of visual quality and embedding capacity using the proposed data hiding scheme, Jung et al.'s scheme [6] and Lee et al.'s scheme [7] for two test images. According to these simulation results, the proposed data hiding scheme generates higher embedded capacity than other two schemes.

Tab. 1. Comparison of PSNR and payload for three schemes

Images	Proposed		Jung et al.		Lee et al.	
	PSNR	Capacity	PSNR	Capacity	PSNR	Capacity
Barbara	23.4665	566770	24.1577	297153	23.5648	476502
Goldhill	28.2130	532994	29.4227	249933	28.3503	441762

The performance comparison results in terms of image quality and embedding capacity for three data hiding schemes are shown in Fig.2. As shown in Fig.2, for these images, the top curve is the proposed scheme. Obviously, the performances of the proposed scheme are better than other two schemes. For Jung et al.' scheme, at the same embedding capacity, the PSNR value is averagely about 0.3-0.5 dB lower than that of the proposed data hiding scheme for two test images. For Lee et al.' scheme, at the same embedding capacity, the PSNR value is averagely about 0.5-1 dB lower than that of the this scheme for two test images.

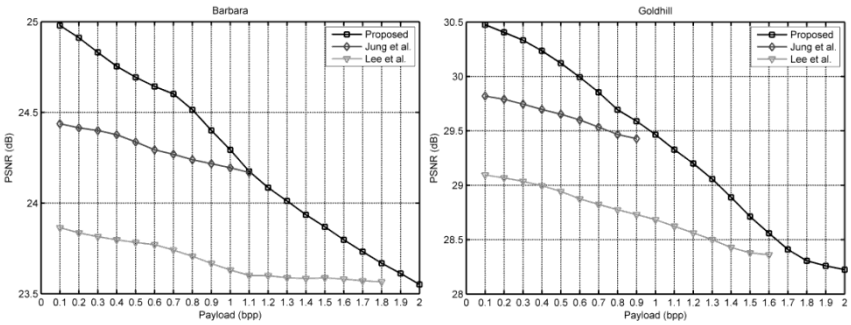

Fig. 2 Performance comparisons of different schemes

4.Conclusion

In this paper, an improved reversible data hiding scheme based on image interpolation is proposed. Similar with Jung et al.'s and Lee et al.'s interpolation methods, the proposed method has also a low-time complexity. Compared to the similar schemes, the experimental results show it has higher embedding capacity and better image quality for watermarked image at the same payload.

Acknowledgements

Supported by the Natural Science Foundation of Department of Education of Guizhou Province (Qian-Jiao-He KY Zi [2015]434).

References

1. J. Fridrich, M. Goljan, R. Du, Lossless data embedding: new paradigm in digital watermarking, EURASIP Journal on Applied Signal Processing, 2, 185 (2002).
2. M. U. Celik, G. Sharma, A. M. Tekalp, et al., Lossless generalized-LSB data embedding, IEEE Transactions on Image Processing, 14, 253 (2005).
3. J. Tian, Reversible data embedding using a difference expansion, IEEE Transactions on Circuits and Systems for Video Technology, 13, 890 (2003).
4. D. Coltuc, Improved embedding for prediction-based reversible watermarking, IEEE Transaction Information Forensics and Security, 6, 873 (2011).
5. Z. Ni, Y. Q. Shi, N. Ansari, et al., Reversible data hiding, IEEE Transactions on Circuits and Systems for Video Technology, 16, 354 (2006).
6. K. H. Jung, K. Y. Yoo, Data hiding method using image interpolation, Computer Standards & Interfaces, 31, 465 (2009).

7. C. F. Lee, Y, L. Huang, An efficient image interpolation increasing payload in reversible data hiding, Expert Systems with Applications, 39, 6712 (2012).