J. Vis. Commun. Image R. 22 (2011) 131–140

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

Reversible data embedding for high quality images using interpolation and reference pixel distribution mechanism

Wien Hong [a,*], Tung-Shou Chen [b]

[a] Yu Da University, Department of Information Management, Miaoli 361, Taiwan
[b] National Taichung Institute of Technology, Department of Computer Science and Information Engineering, Taichung, Taiwan

ARTICLE INFO

Article history:
Received 16 April 2010
Accepted 17 November 2010
Available online 19 November 2010

Keywords:
Reversible data hiding
Histogram-shifting
Image interpolation
Reference pixel
Steganography
Interpolation error
Expansion embedding
Embedding capacity

ABSTRACT

This paper proposes a reversible data hiding method based on image interpolation and the detection of smooth and complex regions in the cover images. A binary image that represents the locations of reference pixels is constructed according the local image activity. In complex regions, more reference pixels are chosen and, thus, fewer pixels are used for embedding, which reduces the image degradation. On the other hand, in smooth regions, less reference pixels are chosen, which increases the embedding capacity without introducing significant distortion. Pixels are interpolated according to the constructed binary image, and the interpolation errors are then used to embed data through histogram shifting. The pixel values in the cover image are modified one grayscale unit at most to ensure that a high quality stego image can be produced. The experimental results show that the proposed method provides better image quality and embedding capacity compared with prior works.

1. Introduction

Data hiding is a technique that conceals secret data into a carrier to convey messages [1]. Digital images are one of the media that is suitable to convey messages due to several reasons. Firstly, digital images are often transmitted over the Internet which would arouse little suspicion. Secondly, the high correlation between pixels provides rich space for data embedding [2]. When a digital image is used as a carrier, the image used to embed data is called the cover image, and the image with data embedded is called the stego image. In the embedding process, the pixels of the cover image are modified and thus, distortion occurs.

In general, the more the cover image is distorted, the more vulnerable is the stego image to steganalytic attempts [3]. To prevent the stego image from being suspicious and detected, either visually or statistically, the distortions caused by data embedding should be as small as possible, which imply that a high quality embedded image is demanded [4]. For most of the existing data hiding techniques, the distortions caused by data embedding are permanent, i.e., the stego image cannot be restored to its original state [3,5,6]. However, for some applications, such as medical or military images, it is desired that the original cover image can be completely recovered because of the requirements for legal considerations or high-precision nature [7]. To fulfill these requirements, the reversible data hiding scheme for high quality images is introduced.

In 2003, Tian [8] proposed a reversible data hiding method using the difference expansion technique. In his method, one bit can be embedded into two consecutive pixels; therefore, the maximum embedding capacity is 0.5 bpp. Alattar [9] generalized the difference expansion technique so that $n-1$ bits can be embedded into n pixels, resulting the maximum embedding capacity $(n-1)/n$ bpp. However, the difference expansion based reversible data hiding methods have to double the differences between pixels; therefore, a larger distortion occurs and may not be suitable for applications where high quality images are demanded. In 2006, Ni et al. [7] proposed a novel histogram-shifting reversible data hiding technique. In Ni et al.'s method, pixel values are modified one grayscale value at most and thus, a high quality stego image can be achieved. However, the maximum payload is limited by the peak height of the image histogram; therefore, the payload of their method is relatively low. Hwang et al. [10] also proposed a reversible data hiding method based on histogram-shifting and had better embedding efficiency than Ni et al.'s work. In 2007, Thodi and Rodriguez [11] proposed a very different method by expanding the prediction errors. Because the prediction error is usually smaller than the difference between two consecutive pixel values, the stego image quality obtained by their method is better than that of Tian's method. However, Thodi and Rodriguez's method is also based on expansion-embedding technique, a larger distortion may occur; therefore, their method is not suitable for applications requiring high quality images.

* Corresponding author. Fax: +886 37 652143.
E-mail addresses: wienhong@ydu.edu.tw, hongwien@ms53.hinet.net (W. Hong), tschen@ntit.edu.tw (T.-S. Chen).

1047-3203/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.jvcir.2010.11.004

Lin et al. [12] proposed a reversible data hiding method in 2008. They partition the cover image into blocks, and calculate the difference values between the first column and the rest of the columns. Data are embedded by modifying these difference values. Later, Hu et al. [13], Hong et al. [14], and Fallahpour et al. [15] independently presented methods based on modification of prediction errors to achieve the goal of reversible data hiding. Tai et al. [16] also presented a similar method in which the differences of two consecutive pixels are calculated. Data embedding is done by modifying the histogram of the absolute value of the differences.

In 2009, Tsai et al. [17] proposed another method based on modifying the difference values between pre-selected basic pixels and their neighbors. Their method partitions the cover image into blocks. The center pixel of each block is selected as the basic pixel for referencing, and these pixels will not be modified in the embedding stage. To conceal data, the partitioned blocks are visited and the differences between the center pixel and other pixels in the same block are calculated. Data are embedded in the difference value domain by using the histogram-shifting technique. In the same year, Kim et al. [18] also proposed a similar embedding technique. They sub-sample the cover image and choose a sub-image as the reference image. The differences between the reference image and other sub-images are calculated. Data are then concealed into the difference values by using the histogram-shifting technique.

In Tsai et al.'s and Kim et al.'s methods, the difference values used for concealing data are obtained by subtracting the reference pixel values from the corresponding neighboring pixel values. This is equivalent to employ the nearest neighbor interpolation technique to interpolate the pixels other than the reference pixels. It is known that the nearest neighbor interpolation method produces larger interpolation errors, resulting in a less peaked interpolation error histogram, which implies that some embeddable spaces are sacrificed. In 2010, Luo et al. [19] proposed a new reversible data hiding method based on interpolation technique and concealed data into interpolation errors. Instead of using the nearest neighbor interpolation technique, they offered a feasible image interpolation algorithm [20] to obtain the interpolation errors. Their method achieves better image quality for watermarked images.

In Kim et al.'s, Tsai et al.'s, and Luo et al.'s methods, the number of reference pixels is fixed and is not adjustable according to the image characteristics. However, in most cases, the number of reference pixels will affect the payload and the stego image quality. Because the reference pixels cannot be used to carry data, excessive reference pixels may occupy some embeddable space, resulting in a decrease in payload. On the contrary, a small number of reference pixels may lead to a less accurate prediction result, and subsequently reduce the payload. Besides, these methods do not consider the local image characteristics. In general, since larger interpolation errors are usually obtained in complex regions, these regions often embed less but distort more. In this paper, we propose a novel method to offer a reference pixel distribution mechanism (RPDM) such that the number of reference pixels is self-adjustable according to the local image characteristics. Besides, more sophisticated interpolation techniques are employed and local image characteristics are considered to provide a better interpolation result to enhance the stego image quality and payload.

The rest of this paper is organized as follows. In Section 2, the influence of the distribution of reference pixels on payload and distortion is investigated. The RPDM is presented, and the embedding and extraction algorithms are followed. Experimental results are presented in Section 3, and concluding remarks are given in Section 4.

2. Proposed methods

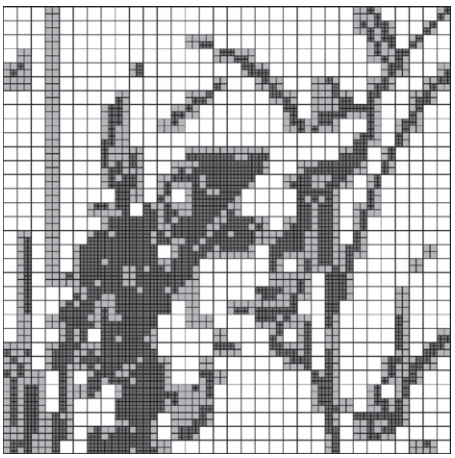
The basic idea of the proposed method is to embed data bits into interpolation errors using the histogram-shifting technique. The reference pixels are adaptively selected in the cover image and pixels other than the reference pixels are interpolated. Interpolation errors are obtained by subtracting the interpolated pixels from the original image. Pixels with larger interpolation errors will be excluded in the embedding process and only those interpolation errors smaller than a pre-defined threshold are employed to conceal data. Data bits are then concealed by modifying the interpolation errors. Because reference pixel values are not changed in the embedding process, the same set of interpolated pixels can be obtained in the decoding process and thus, the embedded data bits can be extracted and the original image can be restored.

When the difference values between the reference pixels and their neighbors are exploited to conceal data, two approaches can be employed to enhance the stego image quality and payload: (1) a better interpolation method and (2) a better distribution of reference pixels. Generally, the prediction result obtained by the nearest neighbor interpolation is less accurate. The proposed method employs the bi-linear and bi-cubic interpolations to obtain a more accurate prediction value. Besides, we have noticed that the benefits are different resulting from placing equal amount of reference pixels in smooth regions and complex regions. For example, Fig. 1 shows the distribution of smooth and complex regions of the Lena image. If Tsai et al.'s method is applied to the Lena image with a 3×3 block size to embed data, there are 145,920 pixels in smooth regions, 58,496 pixels in moderate regions, and 57,728 pixels in complex regions. The payload in smooth regions is 0.18 bpp with MSE 0.874, whereas the payload in complex regions is 0.06 bpp with MSE 0.938, which is only 1/3 of the payload in smooth regions, but with higher MSE.

The analysis shows that the payload in smooth regions is not only much higher than that of complex regions, but the distortion caused by data embedding is also smaller than that of the complex regions. Therefore, the distribution of the reference pixels should take the smoothness of the cover image into account. In the proposed method, we reduce the number of reference pixels in smooth regions and increase the number of reference pixels in complex regions. Since the prediction result is generally more accurate in smooth regions, slightly reducing the number of reference pixels in smooth regions often affects the prediction accuracy insignificantly, but increases the number of pixels that are likely to be embeddable. On the contrary, the prediction result is less accurate in complex regions. The pixels with larger prediction errors not only provide no payload, but also cause distortion since they have to be shifted. Therefore, the pixels in complex regions shall be marked as reference pixels to prevent unnecessary pixel shifting.

2.1. Reference pixel distribution mechanism (RPDM)

To achieve a better embedding efficiency, the distribution of reference pixels is determined by the local complexity of the cover image. We propose a reference pixel distribution mechanism (RPDM) and employ a range function to evaluate the smoothness where pixels are located. The range function $\mathrm{Range}(x_1, x_2, \cdots, x_n)$ is defined as the absolute difference between the maximum and minimum values of the given values $x_1, x_2, \cdots, x_n$. If the range of the upper, lower, left and right reference pixels of the current reference pixel is smaller than a pre-defined threshold T_0, the current reference pixel is likely within a smooth region; therefore, the current reference pixel is set to be a non-reference pixel. In this case, we have one more pixel that is likely to carry data. Besides, for a square region with four reference pixels exactly located in four corners, if the range of these four reference pixels is larger than a pre-defined threshold T_1, this region is like to be a complex one. Therefore, all pixels in this region are marked as reference pixels to prevent them from being modified in the embedding process.



	Pixels	Payload (bpp)	MSE
smooth	145,920	0.180	0.874
moderate	58,496	0.118	0.907
complex	57,728	0.060	0.938

Fig. 1. The distribution of smooth and complex regions of the Lena image. The white blocks represent smooth regions, gray blocks represent moderate regions, and black blocks represent complex regions.

Let the size of the cover image I be $M \times M$, the procedure to construct the reference pixels using RPDM is listed below:

Step 1: Initialize a binary image B with size $M \times M$ to record the positions of reference pixels. The pixels valued 0 in B represent the corresponding positions in the cover image I that are reference pixels, and valued 1, otherwise.

Step 2: For $0 \leqslant i, j \leqslant M-1$, sequentially scan the pixels in the binary image B, and the value of $B_{i,j}$ is set according to the following rule:

$$B_{i,j} = \begin{cases} 0, & \mathrm{mod}(i,\Delta) = 0 \quad \text{and} \quad \mathrm{mod}(j,\Delta) = 0, \\ 1, & \text{otherwise}, \end{cases}$$

where Δ is a pre-defined integer and mod(x,y) is a function that returns the reminder of x/y. This step is equivalent to place the reference pixels with distance Δ along the vertical and horizontal directions.

Step 3: For pixels $I_{i,j}$ satisfying $0 \leqslant i-\Delta, j-\Delta, i+\Delta, j+\Delta < M$, if $B_{i,j} = 0$, $\mathrm{mod}(i/\Delta + j/\Delta, 2) = 0$, and

$$\mathrm{Range}(I_{i-\Delta j}, I_{ij-\Delta}, I_{i+\Delta j}, I_{ij+\Delta}) < T_0,$$

where T_0 is a pre-defined threshold, then $I_{i,j}$ is likely located in a smooth region. In this case, $B_{i,j}$ is set to 1, which is equivalent to set $I_{i,j}$ to be a non-reference pixel.

Step 4: For pixels $I_{i,j}$ satisfying $0 \leqslant i+\Delta, j+\Delta \leqslant M$, if $B_{i,j}$, $B_{i+\Delta,j}$, $B_{i,j+\Delta}$ and $B_{i+\Delta,j+\Delta}$ are all 0, this means that these pixels are reference pixels. If

$$\mathrm{Range}(I_{ij}, I_{i+\Delta j}, I_{ij+\Delta}, I_{i+\Delta j+\Delta}) > T_1$$

where T_1 is a pre-defined threshold, these four reference pixels are likely located in a complex region. In this case, $B_{i'j'}$, $i \leqslant i' \leqslant i+\Delta$, $j \leqslant j' \leqslant j+\Delta$ are set to 0. This is equivalent to mark all the pixels enclosed by these four reference pixels as reference pixels.

A binary image B which records the positions of the reference pixels can be generated from the steps listed above. In the embedding process, the binary image B can be served as a reference to embed data. Note that the same binary image B can be constructed from the stego image directly by performing the same steps listed above and thus, it is not necessary to keep B. We use a simple example to demonstrate how to distribute the reference pixels. Suppose there is a cover image I sized 10×10, as shown in Fig. 2(a). Let $T_0 = 5$, $T_1 = 50$ and $\Delta = 3$. Fig. 2(b) shows the binary image B after finishing Step 2, where the positions of reference pixels are shaded. In the binary image B, reference pixels $I_{3,3}$ and $I_{6,6}$ satisfy $\mathrm{mod}(i/\Delta + j/\Delta, 2) = 0$. Note that $\mathrm{Range}(I_{0,3}, I_{3,0}, I_{6,3}, I_{3,6}) = |37-34| = 3 < T_0$; therefore, $I_{3,3}$ is marked as non-reference pixel, as shown in Fig. 2(c). Finally, $\mathrm{Range}(I_{6,6}, I_{6,9}, I_{9,6}, I_{9,9}) = 155$, which is larger than T_1; therefore, all the pixels enclosed by these four reference pixels are marked as reference pixels, as shown in Fig. 2(d).

Fig. 3(a) and (b) shows two binary images constructed by different Δ. As can be seen in Fig. 3, the complex regions, such as edges, are darker than other regions. This indicates that these regions are set as the reference pixels. On the other hand, the black dots are relatively sparser in the smooth regions such as on Lena's shoulder. The sparser dots indicate that some reference pixels in the smooth region have been removed.

2.2. Embedding procedure

After the binary image B is constructed, the embedding procedure can begin. During embedding, pixels values will be modified one grayscale unit at most according to the corresponding interpolation errors. In this case, the overflow or underflow problems might occur because pixels valued 0 or 255 might be modified to -1 or 256, respectively. To prevent this problem, pixels that may cause overflow or underflow are pre-processed as follows [13]. For pixels valued 0 and 255 in the cover image, their positions are recorded in a location map and the pixel values are set to 1 and 254, respectively. Because the pixel values are modified one grayscale unit at most in the embedding process, the pre-processed cover image will never have pixels overflow or underflow. The embedding procedure is shown in Fig. 4, and the detailed procedure is listed as follows:

Input: Cover image I of size $M \times M$, encrypted secret data S, two thresholds (T_0, T_1) and the distance between reference pixels Δ.

Output: Stego image Is, peaks p_e^- and p_e^+, $|L_M|$ and $|S|$.

Step 1: Scan the cover image I and record the positions of those pixels valued 0 and 255. The recorded positions are served as a location map and are compressed using a run-length coder. The result is denoted by L_M. Concatenate L_M and S to obtain the bit stream L_S. Pixels valued 0 and 255 are then modified to 1 and 254, respectively. Let the modified cover image be I'.

Step 2: Use the modified cover image I' with the parameters Δ, T_0, T_1 and employ RPDM to generate the binary image B, as described in Section 2.1. Let $\mathbf{B}_0$ and $\mathbf{B}_1$ be the sets of positions (i,j) satisfying $B_{i,j} = 0$ and $B_{i,j} = 1$, respectively.

	0	1	2	3	4	5	6	7	8	9
0	31	32	32	34	32	36	36	34	32	34
1	31	33	32	34	31	32	40	34	36	40
2	33	34	32	34	31	32	40	32	28	32
3	35	35	34	33	32	28	36	40	32	32
4	28	34	28	32	32	32	32	35	32	37
5	27	32	24	32	32	32	33	32	35	32
6	33	32	18	37	21	42	40	42	80	70
7	42	26	26	64	72	98	32	32	82	110
8	62	27	34	32	44	90	32	100	50	90
9	80	82	77	70	85	72	22	32	32	177

(a)

	0	1	2	3	4	5	6	7	8	9
0	0	1	1	0	1	1	0	1	1	0
1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1
3	0	1	1	0	1	1	0	1	1	0
4	1	1	1	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1	1	1
6	0	1	1	0	1	1	0	1	1	0
7	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1	1
9	0	1	1	0	1	1	0	1	1	0

(b)

	0	1	2	3	4	5	6	7	8	9
0	0	1	1	0	1	1	0	1	1	0
1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1
3	0	1	1	1	1	1	0	1	1	0
4	1	1	1	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1	1	1
6	0	1	1	0	1	1	0	1	1	0
7	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1	1
9	0	1	1	0	1	1	0	1	1	0

(c)

	0	1	2	3	4	5	6	7	8	9
0	0	1	1	0	1	1	0	1	1	0
1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1
3	0	1	1	1	1	1	0	1	1	0
4	1	1	1	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1	1	1
6	0	1	1	0	1	1	0	0	0	0
7	1	1	1	1	1	1	0	0	0	0
8	1	1	1	1	1	1	0	0	0	0
9	0	1	1	0	1	1	0	0	0	0

(d)

Fig. 2. (a) Original image. (b) Binary image B after finishing Step 2. (c) Binary image B after finishing Step 3. (d) Binary image B after finishing Step 4.

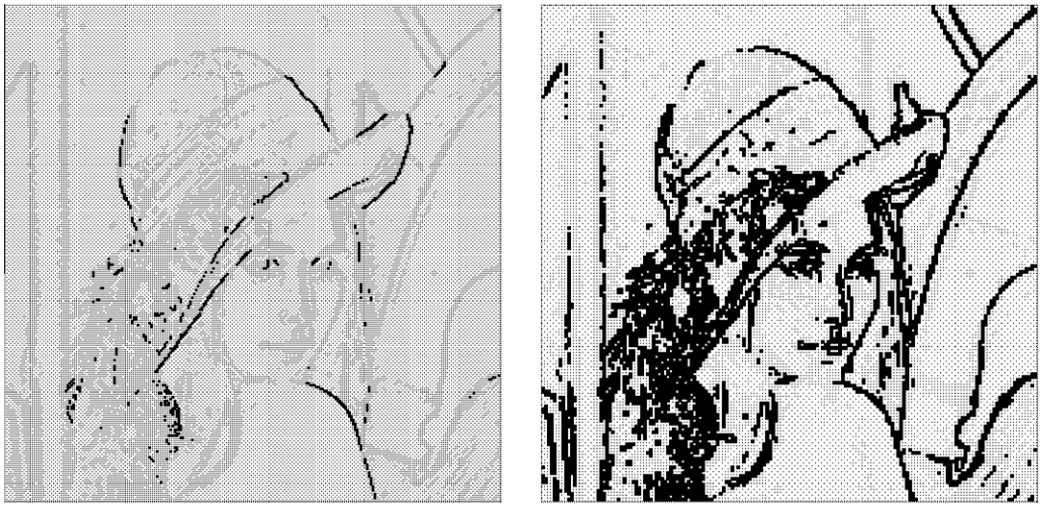


Fig. 3. Binary image B. The black dots represent the locations of reference pixels. (a) $\Delta = 2$, $T_0 = 20$, $T_1 = 100$. (b) $\Delta = 3$, $T_0 = 20$, $T_1 = 100$.

Step 3: Scan the image I'. If the position of the scanned pixel $I'_{i,j}$ satisfies $(i,j) \in \mathbf{B}_1$, then the visited pixel $I'_{i,j}$ is a non-reference pixel. In this case, the neighboring pixels $I'_{k,\ell}$'s of the current pixel $I'_{i,j}$ satisfying $(k,\ell) \in \mathbf{B}_0$ are selected. An interpolation method is then applied to these selected pixels to interpolate the pixel $I'_{i,j}$. Let the interpolation result be $P_{i,j}$.

Step 4: For all the positions satisfying $(i,j) \in \mathbf{B}_1$, subtract $P_{i,j}$ from $I'_{i,j}$ to obtain the interpolation error $E_{i,j}$.

Step 5: For all the interpolation errors, the value that occurs the most frequent is denoted by p_e^+, and the value that occurs the second most frequent is denoted by p_e^-. Without loss of generality, we assume $p_e^- < p_e^+$.

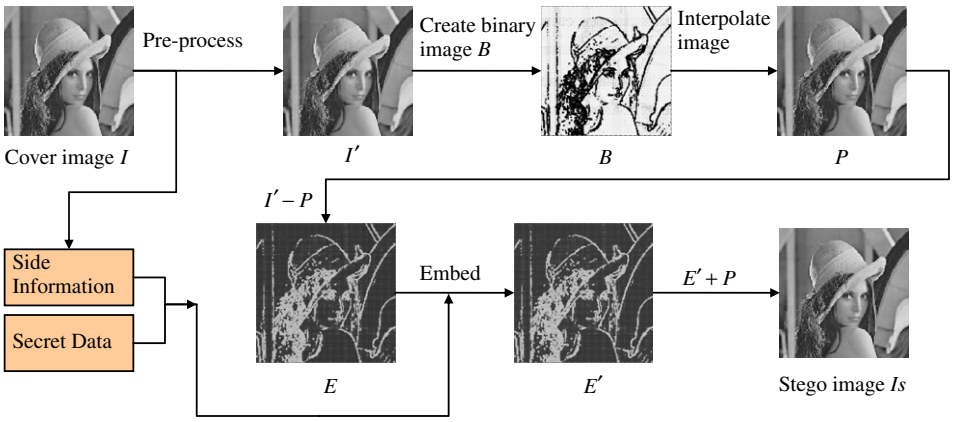


Fig. 4. Data embedding procedure.

Step 6: Scan the interpolation errors. If $E_{i,j} = p_e^+$ or $E_{i,j} = p_e^-$, then the scanned interpolation error is capable of embedding one bit. In this case, one bit b is extracted from L_S. If $b = 0$, set $E'_{i,j} = E_{i,j}$. If $b = 1$ and $E_{i,j} = p_e^+$, set $E'_{i,j} = p_e^+ + 1$; if $b = 1$ and $E_{i,j} = p_e^-$, set $E'_{i,j} = p_e^- - 1$. If $E_{i,j} \neq p_e^+$ and $E_{i,j} \neq p_e^-$, then this interpolation error cannot be used to embed data. In this case, if $E_{i,j} > p_e^+$, set $E'_{i,j} = E_{i,j} + 1$; if $E_{i,j} < p_e^-$, set $E'_{i,j} = E_{i,j} - 1$.

Step 7: The stego image Is can be obtained by adding up the interpolation value P and the modified interpolation errors E'. The parameter Δ, peaks p_e^- and p_e^+, thresholds T_0 and T_1, and the sizes of L_M and S (denoted by $|L_M|$ and $|S|$, respectively) are recorded and served as a key for decoding.

For most natural images, p_e^- and p_e^+ are very close to 0 or simply 0; therefore, three bits are sufficient to record them. We use 16 bits to record T_0 and T_1, two bits to record Δ, and use $\lceil \log_2 |L_M| \rceil$ and $\lceil \log_2 |S| \rceil$ bits to record the length of L_M and S, respectively. Therefore, the key size is

$$2 \times 3 + 16 + 2 + \lceil \log_2 |L_M| \rceil + \lceil \log_2 |S| \rceil = 24 + \lceil \log_2 |L_M| \rceil + \lceil \log_2 |S| \rceil \text{ bits}.$$

For a 512×512 image, the size of L_M is usually less than 10 kbits, and the size of S is less than 100 kbits. Therefore, the key size for most natural images will less than $24 + \lceil \log_2 10^4 \rceil + \lceil \log_2 10^5 \rceil = 54$ bits.

2.3. Extraction and restoration procedures

When the receiver has the stego image and the key, the embedded secret data can be extracted and the stego image can be restored to its original state. The detailed procedure is listed below:

Input: Stego image Is of size $M \times M$, two peaks p_e^- and p_e^+, $|L_M|$, $|S|$, Δ, T_0 and T_1.

Output: Cover image I and secret data S.

Step 1: Construct the binary image B of the stego image Is with the parameters Δ, T_0 and T_1 using RPDM described in Section 2.1.

Step 2: Scan the stego image Is and interpolate the value of non-reference pixels using the corresponding neighboring reference pixels. The interpolation result is denoted by P. Note that the binary image B constructed in Step 1 and interpolation result P obtained in this step are exactly the same as B and P obtained in the embedding phase.

Step 3: For the positions satisfying $(i,j) \in \mathbf{B}_1$, subtract $P_{i,j}$ from $Is_{i,j}$. The result is the modified interpolation error $E'_{i,j}$.

Step 4: Scan the modified interpolation error E'. If the scanned position $(i,j) \in \mathbf{B}_1$, $E'_{i,j}$ has been probably modified in the embedding phase and possibly one bit is embedded. To extract the embedded bit, if $E'_{i,j} = p_e^+$ or $E'_{i,j} = p_e^-$, one bit 0 is extracted; if $E'_{i,j} = p_e^+ + 1$ or $E'_{i,j} = p_e^- - 1$, one bit 1 is extracted. Repeat this step until $|L_S| = |L_M| + |S|$ bits have been extracted.

Step 5: To restore the stego image to the original cover image, E' has to be recovered to its original state in advance. To do this, the elements of E' are scanned. If $E'_{i,j} > p_e^+$, set $E_{i,j} = E'_{i,j} - 1$; if $E'_{i,j} < p_e^-$, set $E_{i,j} = E'_{i,j} + 1$; otherwise, set $E_{i,j} = E'_{i,j}$.

Step 6: Adding up the interpolation value P and the original interpolation error E, the pre-processed image I' can be obtained.

Step 7: In the extracted bit stream, the first $|L_M|$ bits record the compressed location map, and other $|S|$ bits record the embedded secret data. The position information recorded in L_M is extracted. Those pixels valued 1 or 254 in I' with positions recorded in L_M are then set back to 0 or 255, respectively. The resulting image is the original cover image I.

3. Experimental results

To demonstrate the superiority of the proposed method, six common test images, Lena, Jet, Elaine, Boat, Peppers and Baboon, of size 512×512 obtained from USC image database [21] were employed as the cover images, as shown in Fig. 5. Note that these color images were transformed into 8-bit grayscale images if they were originally in color format. The secret bits were generated by using a pseudo-random number generator. Matlab 7.0 was used for the program writing and running under an Intel E7500 CPU with 2 GB RAM.

3.1. Comparison of interpolation methods and threshold values

To compare the effect of the interpolation methods to the embedding efficiency, we used the test image Lena as the cover image, set $T_0 = 0$, $T_1 = 255$, $\Delta = 2$ and used different interpolation methods-nearest neighbor, bi-linear and bi-cubic to perform the experiments. The results are shown in Fig. 6.

As shown in Fig. 6, the results obtained from bi-linear and bi-cubic method achieve better image quality than that of the nearest neighbor interpolation method at all embedding rates. Note that

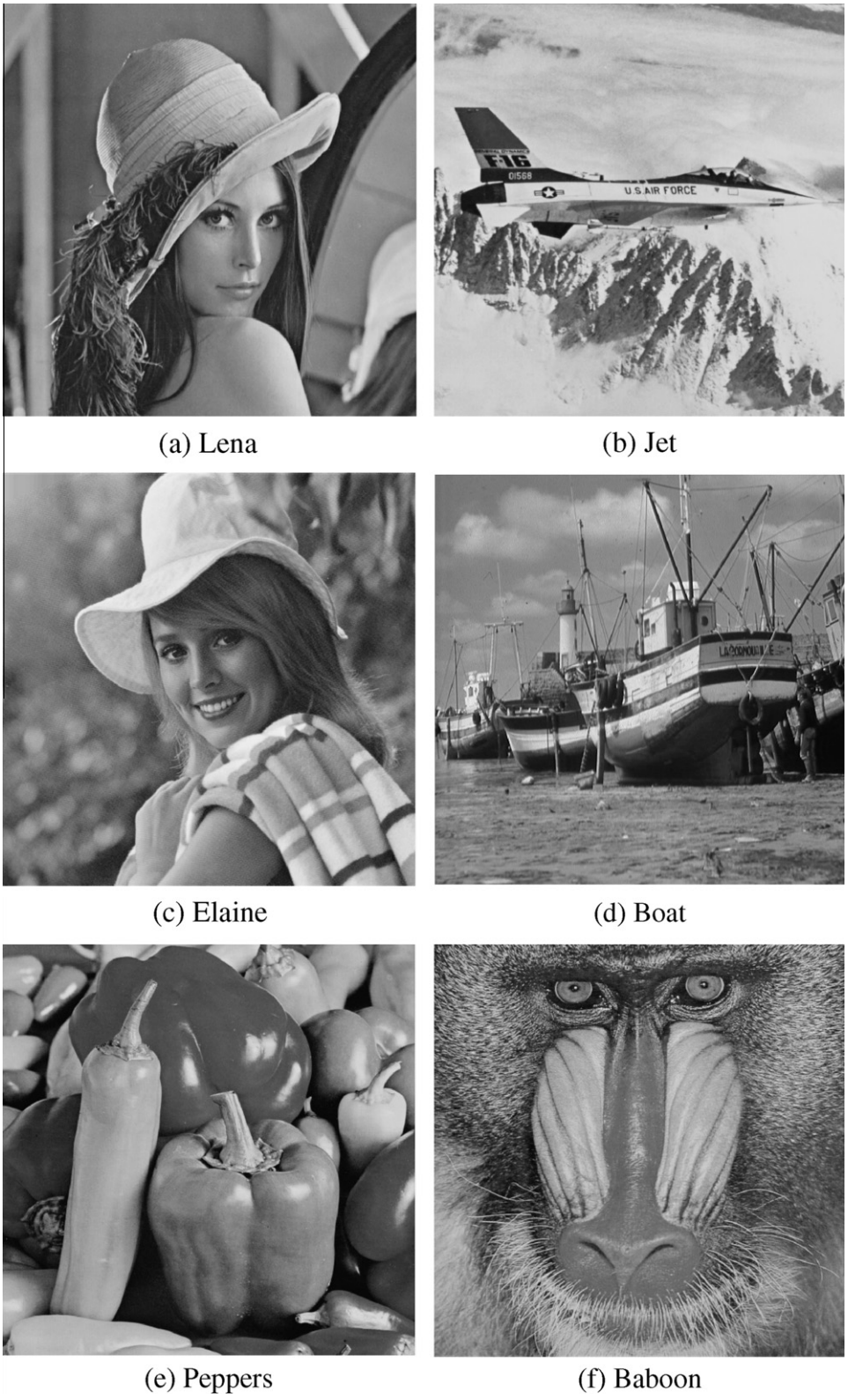


Fig. 5. Six test images.

the result obtained by using bi-linear interpolation method is comparable with that of bi-cubic method but requires less computational cost. The same results are also obtained in other test images; therefore, we recommend using the bi-linear interpolation method in the embedding process.

Fig. 7 shows the results obtained by setting $T_0 = 0$ and varying T_1 to demonstrate the effect caused by the threshold T_1. In this figure, the test image was Lena, Δ was set to 2 and the bi-linear interpolation is used. Fig. 7 reveals that when no threshold mechanism was used, i.e., setting $T_0 = 0$ and $T_1 = 255$, the stego image quality

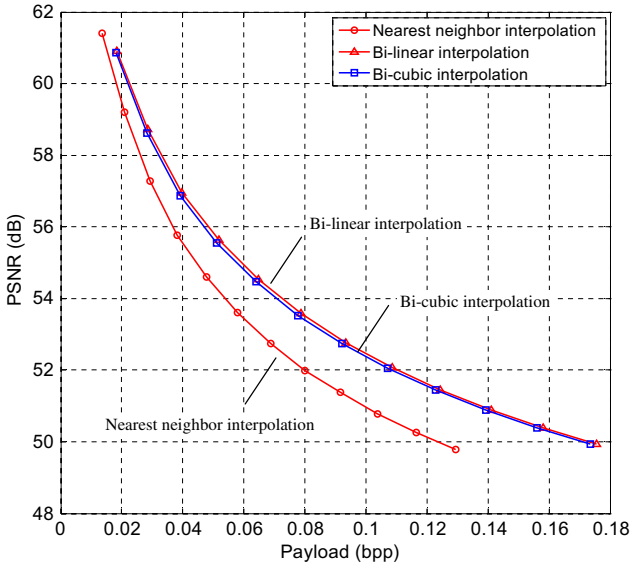


Fig. 6. Results obtained from different interpolation methods for test image Lena and $\Delta = 2$.

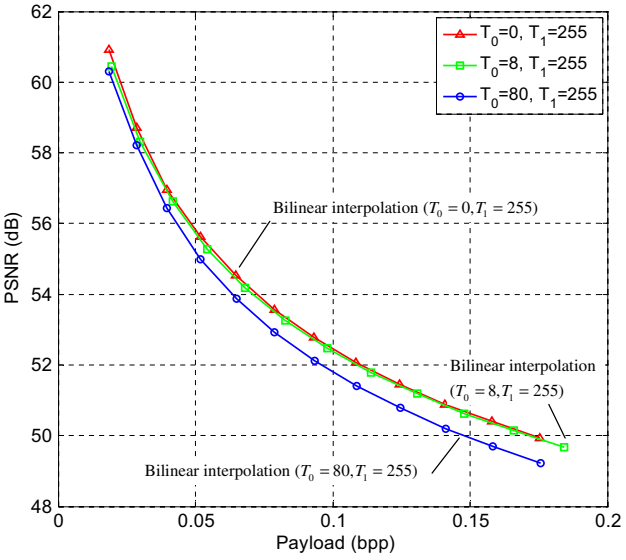


Fig. 8. Comparison for various T_0 for test image Lena and $\Delta = 2$.

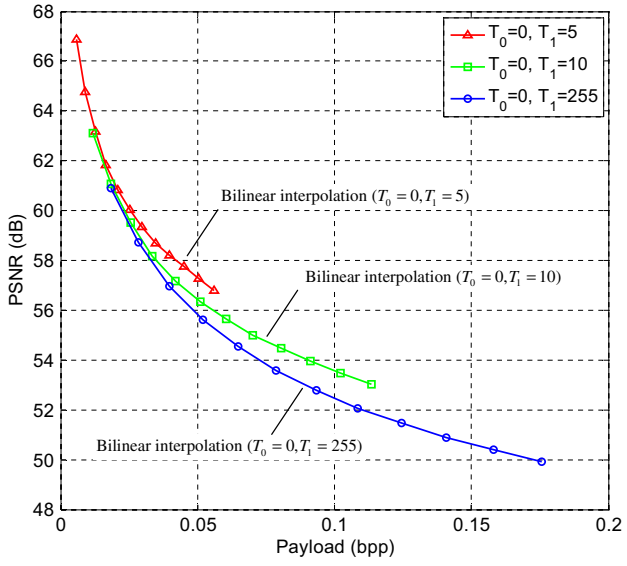


Fig. 7. Comparison for various T_1 for test image Lena ($\Delta = 2$ and $T_0 = 0$).

is the lowest. However, decreasing T_1 will increase the stego image quality but will lead to lower embedding capacity. This is because a decrease in T_1 turns more non-reference pixels into reference pixels. These reference pixels cannot be used to embed data but are effectively prevented from being modified; therefore, the stego image quality is increased.

Fig. 8 gives the comparison results when setting $T_1 = 255$ and varies T_0 to show the effects caused by T_0. In this figure, the test image Lena was used and Δ was set to 2. As shown in the figure, when $T_0 = 8$, the payload is slightly higher than that of $T_0 = 0$. This is because setting $T_0 = 8$ slightly decreases the number of reference pixels in smooth regions and thus the payload increased. However, if T_0 was raised to 80, excessive reference pixels will be changed to non-reference pixels. This may cause the interpolation less accurate and, subsequently, result in a slightly decrease in image quality.

As can be seen from the above experiments, the payload can be increased by selecting a larger T_0, and the quality can be enhanced by decreasing T_1. Therefore, the parameters T_0 and T_1 can be chosen to maximize the image quality for a given payload.

3.2. Comparison of the maximum payload and PSNR values

The aim of the proposed method is to produce high quality stego images, i.e., pixel values in the cover images are confined

Table 1
Comparison of maximum payload (bits) and PSNR (dB).

Method	Lena		Jet		Elaine		Boat		Peppers		Baboon	
	P_ℓ	PSNR	P_ℓ	PSNR	P_ℓ	PSNR	P_ℓ	PSNR	P_ℓ	PSNR	P_ℓ	PSNR
Ni et al.	5462	48.20	16,008	48.18	4878	48.20	11,441	48.20	5415	48.20	5432	48.22
Kim et al.	34,123	49.77	51,142	49.98	21,965	49.63	22,480	49.63	27,045	49.68	11,279	49.50
Tsai et al.	38,950	49.05	58,105	49.26	25,462	48.92	25,788	48.92	32,186	48.99	12,983	48.80
Luo et al.	45,279	49.95	62,714	50.17	27,687	49.73	28,041	49.74	33,783	49.80	14,544	49.59
Proposed	47,549	49.98	64,923	50.31	27,194	49.74	28,739	50.11	34,758	49.87	13,024	51.24

Table 2
Comparison of averaged maximum payload (bits) and PSNR (dB).

Method	Ni et al.	Kim et al.	Tsai et al.	Luo et al.	Proposed
Averaged maximum payload (bits)	8611	71,408	81,565	85,476	86,068
Averaged PSNR (dB)	48.23	50.26	49.53	50.49	50.48

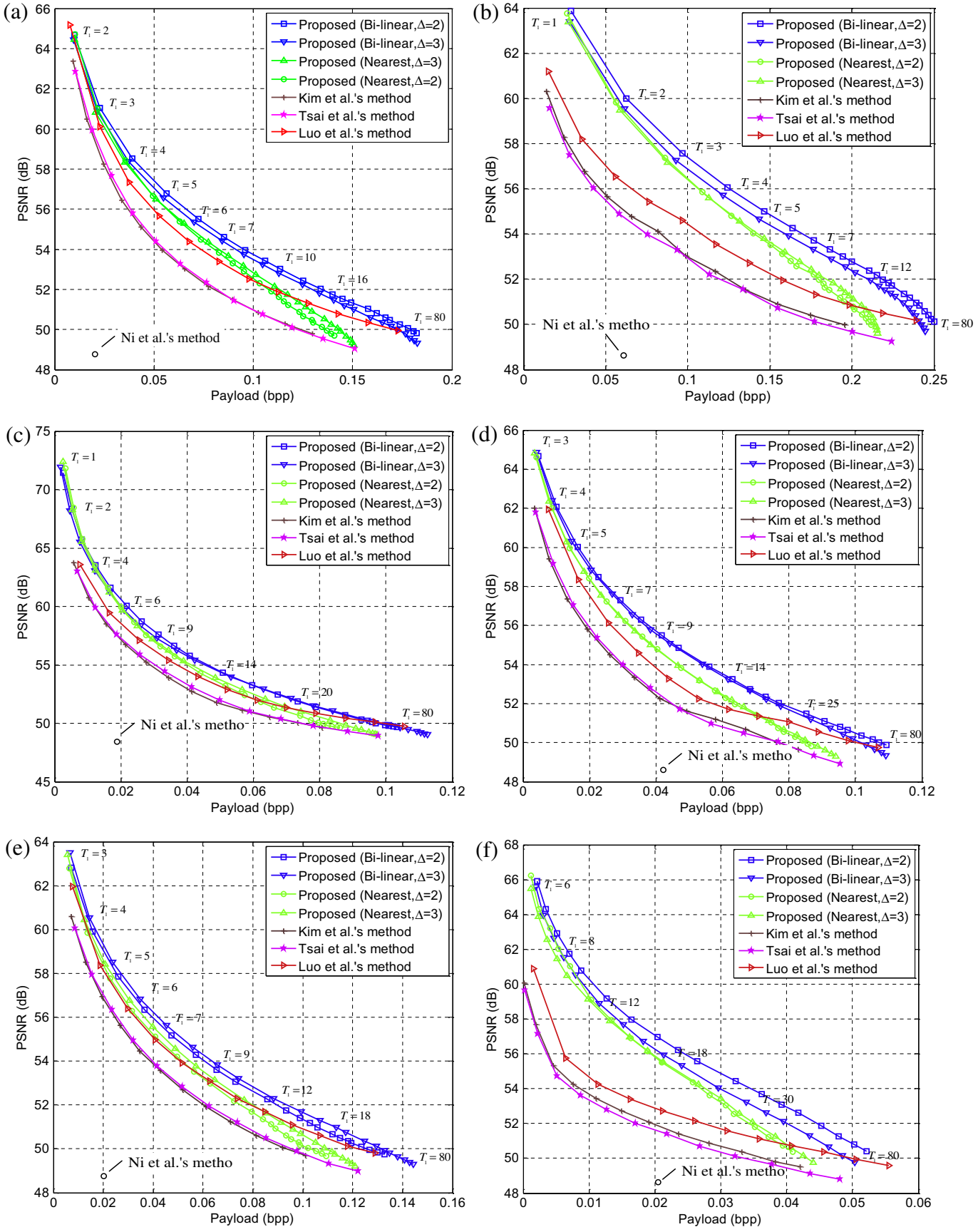


Fig. 9. (a) Test image Lena. (b) Test image Jet. (c) Test image Elaine. (d) Test image Boat. (e) Test image Peppers. (f) Test image Baboon.

to be modified one grayscale value at most; therefore, only single layer embedding is considered in the experiments. For single layer embedding, the PSNR value is guaranteed to be higher than $10\log_{10}(255^2/1) = 48.13$ dB.

Table 1 shows the comparisons of maximum payload P_ℓ and the corresponding PSNR with Ni et al.'s, Kim et al.'s, Tsai et al.'s, and Luo et al.'s methods for six test images. In Ni et al.'s method, two pairs of peak and zero points were selected for data embedding. In Kim et al.'s method, four sub-sampled images were used and embedding level L was set to 0. In Tsai et al.'s method, the block size was set to 3×3, as suggested in their paper. In the proposed method, we set $T_0 = 8$, $T_1 = 60$ and bi-linear interpolation is employed. Because in Kim et al.'s, Tsai et al.'s, and the proposed method, the reference pixels are not designed to carry data bits, to make a fair comparison, the reference pixels in Luo et al.'s method are implemented carrying no data. These settings ensured that pixels in the cover image in each method were modified one grayscale unit at most, so that a high quality stego image can be obtained.

Table 1 reveals that the proposed method performs better than Ni et al.'s, Kim et al.'s, Tsai et al.'s methods and is comparable to Luo et al.'s method. For example, for the Lena, Jet, Boat, and Peppers images, the proposed method achieves the highest payload and PSNR. For the test images Elaine and Baboon, the maximum payload of the proposed method is slightly lower than that of Luo et al.'s method but with higher PSNR. However, the performance of the proposed method can be enhanced by adjusting the parameters T_0 and T_1. For example, if we set $T_0 = 8$ and $T_1 = 120$, the maximum payload of the Baboon image is 15,003 at 49.67 dB, which is better than that of Luo et al.'s method.

We also test the performance of the proposed method and Ni et al.'s, Kim et al.'s, Tsai et al.'s, and Luo et al.'s method for 1000 images of size 512×512 randomly selected from a large image database [22]. Many researches [23,24] adopt images in this database as their standard test images. The results are listed in Table 2. Note that the images in [22] are taken from modern digital cameras which produce impressive image quality with low image noise; therefore, a higher prediction accuracy is obtained in theses test images and subsequently, results in a larger maximum payload.

The results shown in Table 2 revealed that the proposed method performs better than Ni et al.'s, Kim et al.'s, and Tsai et al.'s methods and is comparable to Luo et al.'s method. The maximum payload of the proposed method is slightly higher than that of Luo et al.'s method under almost equivalent image quality. However, the maximum payload and PSNR are both higher than those of Ni et al.'s, Kim et al.'s, and Tsai et al.'s methods.

3.3. Performance comparison for various payloads

Fig. 9(a)–(f) are the comparisons of the proposed method with Ni et al.'s, Kim et al.'s, Tsai et al.'s, and Luo et al.'s methods for different test images under various payloads. In the proposed method, we varied T_1 and set $T_0 = \lceil T_1/10 \rceil$ to obtain the payload–PSNR curves.

Fig. 9(a)–(f) reveals that the stego image quality of the proposed method outperforms other methods for various payloads. Luo et al.'s method performs better than Ni et al.'s, Kim et al.'s, and Tsai et al.'s methods, however, these methods provide no solution to select better regions for data embedding. On the other hand, the proposed method offers a mechanism to select pixels that are likely embeddable and exclude those pixels that are likely unembeddable. Therefore, even if the nearest neighbor interpolation method is used, the quality of stego image is still higher than those of other methods. Take the smooth image Lena and complex image Baboon as examples, when the nearest neighbor interpolation method is used and set $\Delta = 3$, the PSNR of Lena image of the proposed method is 52.5 dB at 0.1 bpp; whereas Luo et al.'s method is comparable to the proposed method and other methods achieve only around 51 dB. For the Baboon image, the PSNR of the proposed method is 55.9 dB at 0.02 bpp; whereas only around 52–53 dB can be obtained in other methods.

When the interpolation is set to bi-linear, the proposed method achieves the best performance. It is interesting to note that setting $\Delta = 2$ will generally obtain better capacity–PSNR curves for most images than setting $\Delta = 3$. This is because more pixels will be selected as the reference pixels and thus provides the bi-linear interpolation method a more accurate interpolation result.

4. Conclusions

In this paper, we propose a reversible data hiding method based on image interpolation technique. In previous works, the reference pixels are distributed equally spaced and provide no mechanism to distribute these pixels based on local image characteristics. Besides, the pixels are interpolated or predicted by their nearest neighbors, resulting in a decrease in payload. The proposed method employs better prediction techniques and offers a mechanism to add or remove reference pixels based on local image characteristics, i.e., reducing the number of reference pixels in smooth regions to increase the payload and increasing the number of reference pixels in complex regions to prevent excessive image degradation. The experimental results revealed that the proposed method achieves better PSNR for all embedding rates than those of prior works.

Acknowledgment

This research was supported by the National Science Council of the Republic of China under the Grant No. NSC99-2622-E-412-006-CC3.

References

[1] N. Provos, P. Honeyman, Hide and seek: an introduction to steganography, IEEE Security and Privacy Magazine 1 (3) (2003) 32–44.
[2] A. Cheddad, J. Condell, K. Curran, P.M. Kevitt, Digital image steganography: survey and analysis of current methods, Signal Processing 90 (3) (2010) 727–752.
[3] X. Zhang, S. Wang, Efficient steganographic embedding by exploiting modification direction, IEEE Communications Letters 10 (11) (2006) 781–783.
[4] H. Wang, S. Wang, Cyber warfare—steganography vs. steganalysis, Communications of the ACM 47 (10) (2004) 76–82.
[5] J. Mielikainen, LSB matching revisited, IEEE Signal Processing Letters 13 (5) (2006) 285–287.
[6] J.M. Guo, Improved data hiding in halftone images with cooperating pair toggling human visual system, International Journal of Imaging Systems and Technology 17 (6) (2008) 328–332.
[7] Z. Ni, Y.Q. Shi, N. Ansari, W. Su, Reversible data hiding, IEEE Transactions on Circuits and Systems for Video Technology 16 (3) (2006) 354–362.
[8] J. Tian, Reversible data embedding using a difference expansion, IEEE Transactions on Circuits and Systems for Video Technology 13 (8) (2003) 890–896.
[9] A.M. Alattar, Reversible watermark using the difference expansion of a generalized integer transform, IEEE Transactions on Image Processing 13 (8) (2004) 1147–1156.
[10] J. Hwang, J.W. Kim, J.U. Choi, A reversible watermarking based on histogram shifting, Lecture Notes in Computer Science 4283 (2006) 348–361.
[11] D.M. Thodi, J.J. Rodríguez, Expansion embedding techniques for reversible watermarking, IEEE Transactions on Image Processing 16 (3) (2007) 721–730.
[12] C.C. Lin, W.L. Tai, C.C. Chang, Multilevel reversible data hiding based on histogram modification of difference images, Pattern Recognition 41 (12) (2008) 3582–3591.
[13] Y. Hu, H.K. Lee, J. Li, DE-based reversible data hiding with improved overflow location map, IEEE Transactions on Circuits System and Video Technology 19 (2) (2009) 250–260.
[14] W. Hong, T.S. Chen, C.W. Shiu, Reversible data hiding for high quality images using modification of prediction errors, Journal of Systems and Software 82 (11) (2009) 1833–1842.
[15] M. Fallahpour, D. Megias, M. Ghanbari, Subjectively adapted high capacity lossless image data hiding based on prediction errors, Multimedia Tools and Applications, Springer, 2010. doi:10.1007/s11042-010-0486-2.

[16] W.L. Tai, C.M. Yeh, C.C. Chang, Reversible data hiding based on histogram modification of pixel differences, IEEE Transaction on Circuits and Systems for Video Technology 19 (6) (2009) 906–910.
[17] P.Y. Tsai, Y.C. Hu, H.L. Yeh, Reversible image hiding scheme using predictive coding and histogram shifting, Signal Processing 89 (6) (2009) 1129–1143.
[18] K. Kim, M. Lee, H. Lee, H. Lee, Reversible data hiding exploiting spatial correlation between sub-sampled images, Pattern Recognition 42 (11) (2009) 3083–3096.
[19] L. Luo, Z. Chen, M. Chen, X. Zeng, Z. Xiong, Reversible image watermarking using interpolation technique, IEEE Transactions on Information Forensics and Security 5 (1) (2010) 187–193.
[20] L. Zhang, X. Wu, An edge-guided image interpolation algorithm via directional filtering and data fusion, IEEE Transactions on Image Processing 15 (8) (2006) 2226–2238.
[21] Image database. Available from: <http://sipi.usc.edu/database/>.
[22] Image database. Available from: <http://dud.inf.tu-dresden.de/~westfeld/rsp/rsp.html/>.
[23] L. Rossi, F. Garzia, R. Cusani, Peak-shaped-based steganographic technique for JPEG images, EURASIP Journal on Information Security 2009 (2009), doi:10.1155/2009/382310. 8p, Article ID 382310.
[24] R.M. Chao, H.C. Wu, C.C. Lee, Y.P. Chu, A novel image data hiding scheme with diamond encoding, EURASIP Journal on Information Security 20 (2009), doi:10.1155/2009/658047. 9p, Article ID 658047.