Adaptive reversible image watermarking algorithm based on DE

Zhengwei Zhang[1, 2], Lifa Wu[2], Yunyang Yan[1], Shaozhang Xiao[1], Shangbing Gao[1]
1. Faculty of Computer and Software Engineering, Huaiyin Institute of Technology,
Huai'an, Jiangsu 223003, China
[e-mail: zzw49010650@sina.com]
2. College of Command Information System, PLA University of Science and Technology,
Nanjing, Jiangsu 210007, China
[e-mail: yunyang@hyit.edu.cn]
*Corresponding author: Zhengwei Zhang

Received June 19, 2016; revised November 29, 2016; accepted December 28, 2016;
published March 31, 2017

Abstract

In order to improve the embedding rate of reversible watermarking algorithm for digital image and enhance the imperceptibility of the watermarked image, an adaptive reversible image watermarking algorithm based on DE is proposed. By analyzing the traditional DE algorithm and the generalized DE algorithm, an improved difference expansion algorithm is proposed. Through the analysis of image texture features, the improved algorithm is used for embedding and extracting the watermark. At the same time, in order to improve the embedding capacity and visual quality, the improved algorithm is optimized in this paper. Simulation results show that the proposed algorithm can not only achieve the blind extraction, but also significantly heighten the embedded capacity and non-perception. Moreover, compared with similar algorithms, it is easy to implement, and the quality of the watermarked images is high.

Keywords: reversible watermarking; difference expansion (DE); texture complexity; Adaptive; IWT

This work is supported by the Jiangsu province Natural Science Foundation of China (No.BK20131069). At the same time this work is also supported by the National Natural Science Foundation of China (NSFC) (No.61402192).

https://doi.org/10.3837/tiis.2017.03.028 **ISSN : 1976-7277**

1. Introduction

Reversible image watermarking is the premise of ensuring the visual quality of the original image. The watermark is embedded into the original image, and when the watermark is extracted, the original image can be recovered without damage. Compared with traditional digital image watermarking [1], reversible image watermarking raises higher requirements for watermark embedding, which also enables it to have a wider research and application significance in judicial, military and medical fields that have a high demand for image authenticity and integrity. One of the basic objectives of the reversible image watermarking algorithm is to obtain the maximum effective information embedding quantity with small distortion.

Reversible data hiding techniques can be classified into two approaches: spatial domain and transform domain. In spatial domain techniques, the pixel values of some bits in the image are changed to hide data. Transform domain techniques, on the other hand, conceal information in the image area that is less exposed to cropping, compression, and image processing, and that is considered as an advantage over spatial domain techniques [2,3]. Currently, the majority of digital image data hiding techniques are spatial-domain ones. Difference expansion (DE) is a digital image revisable data hiding technique introduced by Tian *et al.* [4]. In Tian's scheme, the average value and the difference between two adjacent pixels were calculated. The difference is then doubled to generate an even value. Then, the secret bit is added into the even value to produce an expansion difference. The expansion difference is then equally shared with the adjacent pixels. It is motivated by its high payload capacity and low distortion rate.

Tzu *et al.* [5] proposed a difference expansion-based lossless data embedding technique. This technique divides every pixel in the original cover image into two nibbles, and each nibble pair between any two adjacent pixels can be utilized to hide secret information. The results demonstrated that this technique can quickly and correctly extract hidden information, hide a large payload capacity without causing any visual deformation, and retrieve original cover image from the stego one. Literature [6-8] constructed a prediction operator which could remove pixel relativity to improve the concentration of prediction difference, and then respectively adopted the difference expansion embedding method or histogram translation embedding method to embed the watermark into the prediction difference. Among the algorithms, Luo et al.'s algorithm [6] had the best interpolation effect with highest interpolation concentration. So the algorithm also has the best integral performance. Chrysochos *et al.* [9] put forward a reversible data hiding technique based on DE in triplets. It can embed two bits of secret information in a triplet of coefficients. Liu et al. [10] proposed a reversible data hiding method based on DE and bilinear interpolation. In their scheme, two secret bits could be embedded into one cover pixel. This approach divides the cover image into several blocks that are sized 2×2. Next, the prediction value is produced by the bilinear interpolation method, and the difference between the current embedding pixel and the prediction value is calculated for each block. The maximum absolute difference for each block is extracted to determine whether each pixel in a block can embed two secret bits or not. If the maximum absolute difference is smaller than the predetermined threshold, then the pixel can embed two secret bits. Therefore, the ideal hiding capacity of this method is better than that of the other three methods [11, 12, 13]. Combined difference expansion with LSB algorithm proposed in Literature [14], the reversible image watermarking algorithm obtains larger embedding capacity with better visual quality. It is unavoidable that distortion from the original image will happen when embedding watermarking information. Therefore, in order to further reduce the image distortion, Li et al. [15] proposed a reversible information hiding algorithm based on pixel sequence and prediction difference. Regardless of its improvement in the hiding image quality, it has low embedding capacity. The bigger the image block is, the lower the capacity is. Lin et al. [16] proposed the algorithm of combining histogram translation with prediction difference to enlarge the embedding capacity of reversible watermarking and have a good effect. Lu *et al.* [17] proposed a reversible hiding method based on the DE, histogram and bilinear interpolation. The proposed method does not need to search for the peak points, nor does extra data need to be compressed. In the proposed method, the pixel (except for the reference pixel) in a smooth or complex region can embed secret data. After some secret data are embedded, the remaining secret data are embedded into the reference pixel. Consequently, the proposed method can embed a large amount of secret data. Hala *et al.* [18] proposed an adaptive difference expansion-based reversible data hiding technique with the capability of embedding either three bits or two bits or one bit for each pixel. To embed and extract data, the proposed technique has some salient features such as its capability to control the embedding capacity by using three global embedding parameters. These parameters are computed using the statistics of the embedded pixel surrounding pixels. The proposed technique has a high payload capacity and a good PSNR.

For the difference expansion embedding method, the overflow location map is one of the important factors that affect the embedding capacity, and it is very important to improve the performance of the algorithm. A reversible watermarking algorithm based on difference expansion and reversible contrast graph was proposed in literature [19] to divide the image into 2×2 image blocks. In each image block, the first two pixels are reversible contrast image pixel pairs, and the other two pixels are the difference extended pixel pairs, and the two pixel pairs are used for embedding information. Reversible contrast image pixel pairs are mainly used to embed small amount of additional information to replace location image, which can improve the embedding capacity. However, half of the pixel pairs in the algorithm adopt reversible contrast image translation, and the image quality decreases a lot. Literature [20] raised a reversible embedding method based on differential histogram translation. In order to avoid the pixel overflow, the method adjusts the pixel values to a certain range before translation, and records the positions of the pixels in the position map. This embedding method is quite distinctive in dealing with pixel overflow. But the compressed position map needs to be embedded.

Based on the analysis of the traditional difference expansion watermarking algorithm, an improved reversible image watermarking algorithm based on adaptive difference expansion is proposed in this paper. This algorithm can not only improve the embedding capacity and reduce the image distortion effectively, but also avoid the pixel overflow.

The rest of this paper is organized as follows: Section 2 presents related work of difference expansion watermarking. Section 3 introduces the details of the proposed watermarking technique. In Section 4, the performance evaluation and experimental results of proposed watermarking technique and other similar techniques are presented. Finally, Section 5 summarizes the paper.

2. Correlation Algorithm

2.1 DE algorithm

Tian [4] proposed that the expansion algorithm based on the adjacent pixels was to conduct integer transform of any one of the image pixels $P=(x, y)$ and get the mean value l and the difference value h. Accordingly, the mean value l and the difference value h can lossless restore the original image pixel values x and y after the inverse transform.

Positive transform: $l=\left\lfloor \frac{x+y}{2} \right\rfloor$, $h=x-y$

inverse transform： $x=l+\left\lfloor \frac{h+1}{2} \right\rfloor$, $y=l-\left\lfloor \frac{h}{2} \right\rfloor$

The resulting difference h is shifted left 1 bit, and the watermark is embedded into its least significant bit, and this is the difference expansion. Its mathematical expression is: $h'=2h+b$.

Meanwhile, the pixel values obtained after embedding the watermark by the difference expansion may cause the pixel overflow, so x, y acquired by inverse transform should be limited to the range of [0, 255], or it will no longer be reversible in watermark extraction and image restoration. Therefore, h' should be restricted: $|h'| \le min(2(255-l), 2l+1)$

The difference expansion watermarking algorithm is based on the difference between the pixels in the image for watermark embedding, and its embedding capacity is limited. To this end, a watermarking algorithm based on generalized difference expansion arises spontaneously.

2.2 Generalized DE method

Compared to the adjacent pixel expansion algorithm presented by *Tian*, generalized difference expansion algorithm makes more fully use of the redundant information between adjacent pixels. It picks up a plurality of adjacent pixels for processing, and which can be used to embed more watermark information. This paper uses this method to embed information into the selected original image pixel blocks. Suppose X=(x_0, x_1, x_2, x_3,…, $x_{n\text{-}1}$) is a set of pixel values, the direct transform of the generalized integer transform is:

$$\bar{x} = \left\lfloor \sum_{i=0}^{n-1} a_i x_i \,/\, \sum_{i=0}^{n-1} a_i \right\rfloor \tag{1}$$

$$d_1 = x_1 - x_0$$
$$d_2 = x_2 - x_0$$
$$\vdots$$
$$d_{n-1} = x_{n-1} - x_0$$

For a group of pixel interpolations d_1, d_2, …, d_{n-1}, Equation (2) can be used separately to hide 1 bit watermark information b:

$$d_i^{'} = 2 \times d_i + b \tag{2}$$

Where d_i' is the pixel difference after embedding the watermark. It requires the watermark embedding process not to cause the overflow of image pixel values. The corresponding inverse transform is:

$$x_0^{'} = \bar{x} - \left\lfloor \sum_{i=1}^{n-1} d_i^{'} \,/\, \sum_{i=0}^{n-1} a_i \right\rfloor \tag{3}$$

$$x_1^{'} = x_0^{'} + d_1^{'}$$
$$x_2^{'} = x_0^{'} + d_2^{'}$$
$$\vdots$$
$$x_{n-1}^{'} = x_0^{'} + d_{n-1}^{'}$$

A set of pixel values $\mathrm{X}^{'} = (x_0^{'}, x_1^{'}, x_2^{'}, x3, \ldots, x_{n-1}^{'})$ is generated by the generalized difference expansion algorithm, in which the mean of the group pixels is:

$$\bar{x}^{'} = \left\lfloor \sum_{i=0}^{n-1} a_i x_i^{'} \,/\, \sum_{i=0}^{n-1} a_i \right\rfloor \tag{4}$$

After deduction and calculation, $\bar{x}' - \bar{x} = 0$. Visibly a group of pixels have the same group mean value after the generalized difference expansion transform. After using the generalized difference expansion algorithm to embed information into the pixel blocks, the mean value of the block pixels should be unchanged, and that is a fairly strict requirement.

2.3 Image texture complexity analysis

Generally, the smoother the image in the block, the smaller the differences between the internal pixels will be, and the objective distortion of the original load caused by the embedding of watermark information with the differential expansion method will be relatively small. Then watermark embedding can be preferentially conducted; the more complicated the image texture is, the larger the differences between the pixels will be, and the hiding information will trigger larger distortion. Hence, a pixel block smoothness measuring function needs to be constructed to select appropriate pixel blocks. In this paper we construct a pixel block smoothness approximate measuring function to calculate the approximate smoothness value of each pixel block.

In **Fig. 1**, X_0 represents the current pixel block, X_1, X_2, X_3, X_4 are the neighboring pixel blocks. The approximate smoothness degree function of X_0 will be:

$$\rho(X_0)=[\sum_{i=0}^{4}(\bar{x}_i-\bar{X})^2]/5 \tag{5}$$

In Formula (5), $\bar{x}_0$, $\bar{x}_1$, $\bar{x}_2$, $\bar{x}_3$ and $\bar{x}_4$ are the pixel mean values of X_0, X_1, X_2, X_3 and X_4, respectively, which can be calculated by Formula (6).

$$\bar{x}_i=\sum_{j=0}^{3}x_{i,j}/4 \tag{6}$$

In Formula (5): $\bar{X}$ is the mean value of $\bar{x}_0$, $\bar{x}_1$, $\bar{x}_2$, $\bar{x}_3$ and $\bar{x}_4$ as shown in Formula (7).

$$\bar{X}=\sum_{i=0}^{4}\bar{x}_i/5 \tag{7}$$

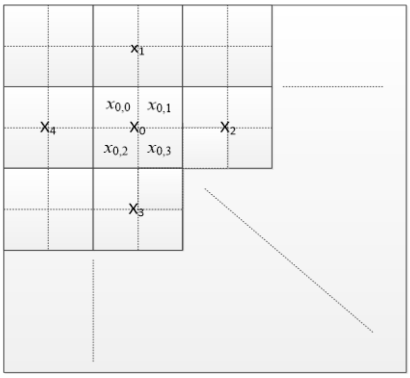


Fig. 1. The original carrier image is divided into 2×2 pixel blocks

This paper first uses Formula (5) to calculate the approximate smoothness values of all pixel blocks. Then we sort the smoothness values from small to large. At last we conduct watermark embedding for the front pixel blocks according to the information amount to be embedded. Experiments have proven that this method can effectively alleviate the problem of over-occupation of the position map, and improve the embedding capacity. Meanwhile, from Formula (5) we can see the approximate smoothness degree value of each pixel block is invariant whenever before or after embedding watermarks, which can be used to identify the information embedding location and thus replace location marking figures that occupy a lot of space.

2.4 Improved DE watermarking algorithm

The difference expansion watermarking algorithm presented in the previous paper uses the difference between the pixels in the image to embed the watermark. If the difference is large, it is easy to generate overflow or a sharp decline in the visual quality of the watermarked image. In order to improve the visual quality of the image without reducing the embedding capacity, the difference expansion algorithm is improved in this paper.

Partitioning of the non-overlapped block was conducted on the original carrier image, and three continual pixels in the block were selected successively. Then a pixel pair can be created by the mean value of the three selected pixels and the mean pixel values of the remaining pixels. One digit watermark is embedded into the pixel pair through the differential expansion method, and the first of the original three pixels was replaced by the new created three pixels, and so on and so forth.

As shown in **Fig. 2**, partitioning of a 4×4 non-overlapped block was conducted on the original carrier image. First, calculate the mean value of the three pixels marked as 1, 2 and 3, get value A_1, and calculate the mean value of the rest of the pixels in the sub-block, get value B_1. Then use the difference expansion method to embed a one-digit watermark into A_1 and B_1. Replace the original pixel marked with 1 with the mean value of the newly generated three pixels. Similarly, calculate the mean value of the three pixels in the sub-block marked with 2, 3, and 4, get value A_2, and calculate the mean value of the rest of the pixels in the sub-block, get value B_2. Then use the differential expansion method to embed a one-digit watermark into A_2 and B_2. Replace the original pixel marked with 2 with the mean value of the newly generated three pixels. By this analogy, select the order according to the three pixels as shown in Figure 1 until the three pixels marked with 13, 1 and 2. In this way, the process of the watermark embedding of a sub-block is accomplished.

This paper calculates the mean value of three continual pixels. If there are only two pixels in the sub-block, the two pixels will be combined with the first one to form a three-pixel sequence; if there is only one pixel left in the sub-block, then the pixel will be combined with the first and the second pixels to form the three-pixel sequence.

This method can avoid some pixels from being salient points in the smooth block or large difference with the mean value of the block pixel of the whole image, which can effectively improve the visual quality. In addition, the maximum embedding capacity of this method in a 4×4 sub-block is 16-digit watermark, which can embed one-digit more watermark information than the generalized differential expansion watermark algorithm.

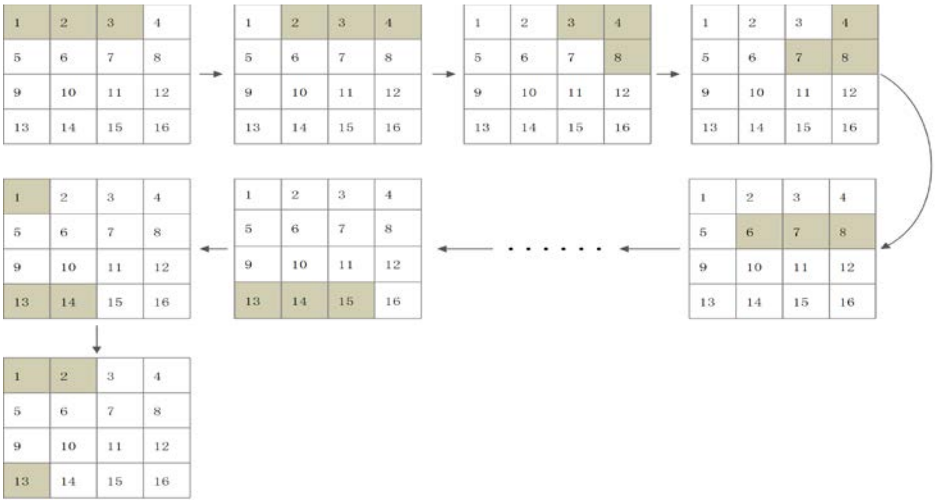


Fig. 2. Flow chart of 3 pixel selection in the block

Assume $X = (a_1, a_2, a_3, a_4, \ldots, a_{16})$ is a group of pixel values in a 4×4 sub-block，then the improved differential expansion will be transformed as:

For i=1 to 16

$$\bar{X} = \lfloor (a_1 + a_2 + \cdots + a_{16}) / 16 \rfloor$$

$$\bar{a}_i = \lfloor (a_{i\%16} + a_{(i+1)\%16} + a_{(i+2)\%16}) / 3 \rfloor$$

$$\bar{b}_i = \lfloor (a_{(i+3)\%16} + a_{(i+4)\%16} + \cdots + a_{(i+15)\%16}) / 13 \rfloor$$

$$h_i = \bar{a}_i - \bar{b}_i$$

$$h_i' = 2 \times h_i + w$$

$a_i = \bar{a}_i'$ (Use reverse transformation to get $\bar{a}_i'$, which is used to replace a_i pixel in the original image block)

$$i = i + 1$$

End

In the current sub-block, the operation is performed 16 times. That is to say the watermark embedding operation of the image block is completed. If the overflow of the image pixel value is encountered in the watermark embedding process, the value of the pixel is processed.

3. Algorithm Design

In this paper, the improved difference expansion watermarking algorithm is mainly used to achieve watermark embedding and extraction. It mainly includes 2 modules: watermark embedding and watermark extraction.

3.1 Watermark embedding

The watermark embedding process proposed in this paper is shown in Fig. 3. The concrete steps of embedding are as follows:

STEP1: Divide the original image *L* (size: *M* × *N*) into non-overlapped blocks $\mathrm{L_i}$ (size: a × a), $i \in (0, \frac{M \times N}{a \times a})$;

STEP2: Conduct Arnold scrambling on the watermark information *W* to be embedded, and transform the scrambled image into a one-dimensional vector;

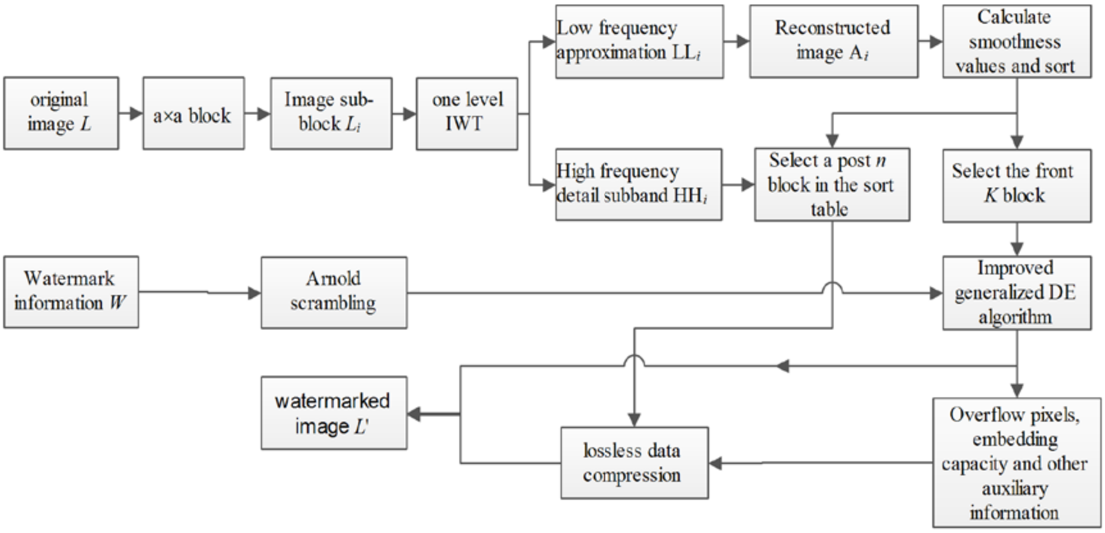

original image L
a×a block
Image sub-block L_i
one level IWT
Low frequency approximation LL_i
Reconstructed image A_i
Calculate smoothness values and sort
High frequency detail subband HH_i
Select a post n block in the sort table
Select the front K block
Watermark information W
Arnold scrambling
Improved generalized DE algorithm
watermarked image L'
lossless data compression
Overflow pixels, embedding capacity and other auxiliary information

Fig. 3. Watermark embedding process

STEP3: Conduct 1 level of IWT on each L_i block and get the low-frequency approximate part LL_i and high-frequency detailed parts HL_i, LH_i, HH_i of each block; for the only low-frequency approximate part LL_i of each block, conduct the image reconstruction to get the image A_i, $i \in (0, \frac{M \times N}{a \times a})$, which is of the same size as the original image, as shown in Fig. 4;

(a) (b)

Fig. 4. (a) Lena original image (b) Low frequency reconstructed imag

STEP4: Use Formula (5) to calculate the approximate smoothness values of all pixel blocks A$_i$, sort the smoothness values from small to large and establish a sequence index information table. Then, based on the watermark information to be hided, properly select the first pixel blocks in the front of the sequence (assume the first k blocks) and embed the watermarks successively;

STEP5: Based on the above method, use the improved generalized differential expansion algorithm to embed the watermark for any selected smooth block A$_i$ ($0\leq i\leq$k). Regarding the pixels that exceeded gray value range after the information is embedded by the difference expansion method, mark in the overflow map; compress the overflow map and hide it in the original carrier image together with the auxiliary information as watermark embedding capacity and watermark scrambling times *B*.

STEP6: Regarding the image blocks (original pixel block with complicated texture) not used to embed watermark information, conduct IWT and select the high-frequency detailed sub-band HH_i to embed auxiliary information (use a lossless data compression method to embed auxiliary information for the high-frequency part). Here we choose the last *n* blocks in the sequence to embed auxiliary information and save the selected sub-block *n* for watermark extraction;

STEP7: Generate watermarked image L' through a reverse operation.

3.2 Watermark extraction

If we embed the watermark information of any pixel pair (*x*, *y*) in the original image by the differential expansion method and the embedded watermark information is 1, then the new generated value of the pixel pair (*a*, *b*) will be:

$$a=\frac{\boldsymbol{x}+\boldsymbol{y}}{2}+\frac{2(x-y)+1+1}{2}=\frac{x+y}{2}+x-y+1$$

$$b=\frac{x+y}{2}-\frac{2(x-y)+1}{2}=\frac{x+y}{2}-x+y$$

Accordingly, $a-b=2x-2y+1$。

So in any pixel pair, when the embedded watermark information is 1, the difference of new pixel pair is odd. In the same way, if the embedded watermark information is 0, the difference of new pixel pair is even. By this method, when we restore the original carrier image, if the difference of pixel pair (a, b) in the watermarked image is odd, the watermark information is 1, otherwise 0. Watermark information can be extracted by this method.

The processes of the watermark extraction and carrier image recovery are shown in **Fig. 5**. The detailed steps are as follows:

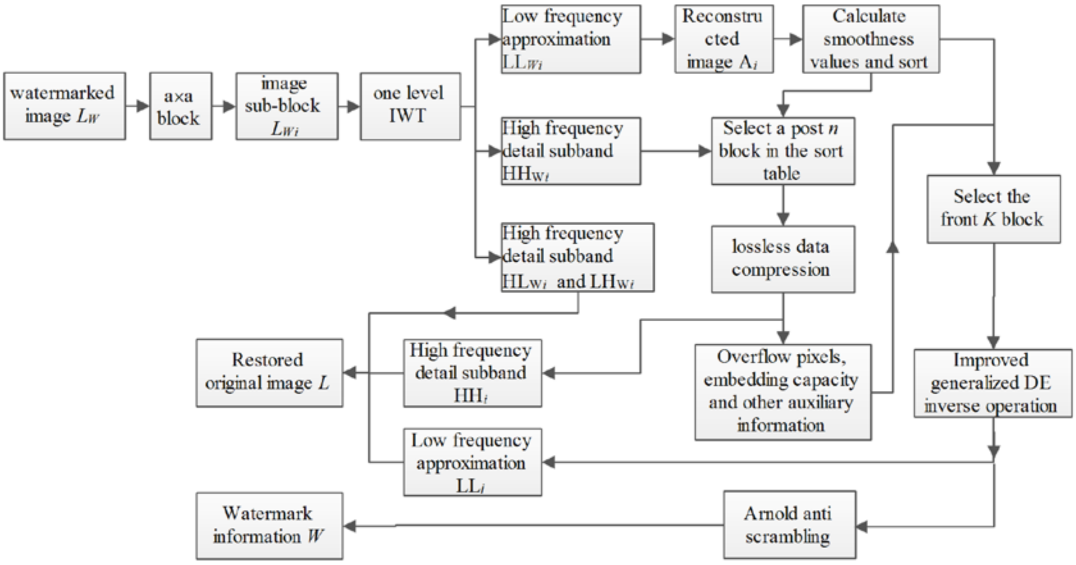

watermarked image Lw
axa block
image sub-block Lwi
one level IWT
Low frequency approximation LLwi
Reconstructed image Ai
Calculate smoothness values and sort
High frequency detail subband HHwi
Select a post n block in the sort table
Select the front K block
High frequency detail subband HLwi and LHwi
lossless data compression
Restored original image L
High frequency detail subband HHi
Overflow pixels, embedding capacity and other auxiliary information
Improved generalized DE inverse operation
Low frequency approximation LLi
Watermark information W
Arnold anti scrambling

Fig. 5. Watermark extraction process

STEP1: Divide the watermarked image L_W (size: $M \times N$) into non-overlapped blocks L_{Wi} (size: $a \times a$), $i \in (0, \frac{M \times N}{a \times a})$ 。

STEP2: Conduct one level of IWT on each L_{Wi} block and get the low-frequency approximate part LL_{Wi} and high-frequency detailed parts HL_{Wi}, LH_{Wi} and HH_{Wi} of each block; for the only low-frequency approximate part LL_{Wi} of each block, conduct image reconstruction to get the image A_i, $i \in (0, \frac{M \times N}{a \times a})$, which is of the same size as the original image;

STEP3: Use Formula (5) to calculate the approximate smoothness values of all pixel blocks A_i, sort the smoothness values from small to large and establish a sequence index information table. The pixel mean values in the block keep the same after embedding the information into the pixel block with the generalized differential expansion method. So before and after watermark embedding, the sort order for blocks as per the smoothness value is also the same;

STEP4: Conduct IWT on each block $\mathrm{L_{Wi}}$ generated by the stable sorting of the smoothness values from small to large, and get the low-frequency part $\mathrm{LL_{Wi}}$ and high-frequency part $\mathrm{HH_{Wi}}$ (here refers to the diagonal high-frequency part) of each block;

STEP5: Extract the auxiliary information embedded from the high-frequency part $\mathrm{HH_{Wi}}$ (the last n sub-blocks in the smoothness value sequence), get the watermark capacity, image overflow position and watermark scrambling times B, and restore the high-frequency part $\mathrm{HH_i}$ of the original image.

STEP6: Based on the auxiliary information extracted from the high-frequency part $\mathrm{HH_{Wi}}$, the capacity of the embedded watermark can be concluded. As the approximate smoothness values of each pixel block A_i calculated by the generalized differential expansion method are constant before and after watermarking embedding and based on the watermark capacity, the pixel blocks in the front of the sequence are chosen as the smoothness block set and the rest are taken as the complicated part.

STEP7: The newly generated pixel from the watermark information embedding through the differential expansion method has an odd-even feature against the difference value. By taking advantage of this feature, the watermark information of each pixel block A_i in the smoothness block set is extracted and the low-frequency part $\mathrm{LL_i}$ of the original image is restored.

STEP8: Conduct IWT on the high-frequency part HH_i and low-frequency part LL_i of the restored original image with two mid-frequency parts $\mathrm{HL_{Wi}}$ and $\mathrm{LH_{Wi}}$ which do no change in the sub-block of the watermark image, and then get the pixel block $\mathrm{L_i}$ of the original image.

STEP9: Reconstruct the obtained pixel blocks L_i according to the sequence index information table and restore the original image L.

3.3 Algorithm optimization

The watermark is embedded through the above algorithm. If the watermark information capacity is smaller or the original image texture is smoother, the watermarked image may show higher visual quality. However, when the capacity of the embedded watermark is larger or the original image texture is more complex, the watermark information is embedded in the same way, and the visual quality of the watermarked image is poor. Therefore, the algorithm optimized in this paper can allow watermarked images to have higher visual quality.

Grounded upon the above embedding algorithm, the watermark information part to be embedded is selected and embedded to get the hidden image with part of watermark information. Then, with the same embedding algorithm as above, the rest of the watermark information to be embedded is embedded into the hidden image and the final hidden image is acquired. Compared with the hidden image obtained by the previous watermark embedding method that embeds watermark information at one time, it has better visual quality, especially for images with complicated texture.

By calculating the smoothness value of each pixel block, set a threshold value $\mathrm{T_1}$, select the block as the watermark embedding block with the smoothness value less than the threshold value $\mathrm{T_1}$.

(1) Based on the above watermark embedding algorithm, embed the watermark in the selected block. If all of the watermarks are embedded, embedding will be accomplished and the final watermark image can be obtained.

(2) If only part of the watermark is embedded in the selected block, then divide the embedded watermarked image and calculate the smoothness value of each pixel block of the watermark image, and embed the rest of the watermark into the blocks whose smoothness values are less than the threshold value T_1.

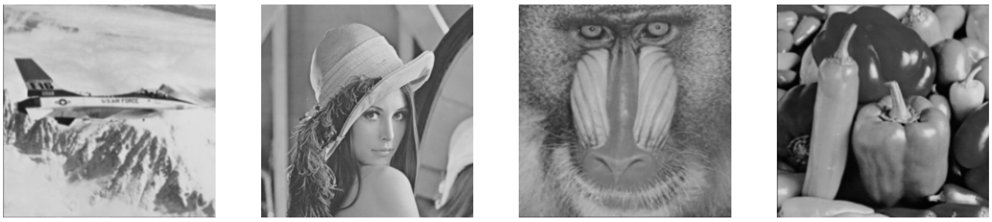
(3) If the watermark is still not embedded, continue doing it following Step 2 until it is done.

(4) After the watermark is embedded, the watermark image can be obtained.

For different types of images, the set threshold value T_1 is not the same. If the image texture is smooth, we can set T_1 larger; if the image texture is complicated, we can set T_1 smaller.

4. Experimental Results and Performance Analysis

Plane, Lena, Baboon and Pepper are all 512×512 eight-bit standard grey images (all images are derived from http://sipi.suc.edu/database) that are selected in the experiment as the original carrier images (as shown in **Fig. 6**), and the watermark image is 32×32 binary image (as shown in **Fig. 7**). In the experiment, information embedding rate (ER), image objective distortion (PSNR) and structure similarity (SSIM) are used to evaluate the performance of the algorithm.



Plane *Lena* *Baboon* *Pepper*

Fig. 6. Original carrier images

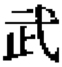


Fig. 7. Watermark image

According to **Fig. 6**: the image Plane has a simple texture, good smoothness, small differences between the pixel blocks, high embedding rate, and good watermark image quality; while the image Baboon has a complicated texture, poor smoothness and large differences between the pixel blocks. So its image distortion caused is big by embedding watermark information and it also has a relatively low embedding rate. Details will be illustrated by the following experimental results.

4.1 Integrity assessment

Generally, the reversible watermarking algorithm requires carrier images to completely recover extracted watermark images. Therefore, it is measured by the NC (Normalized Correlation) of the original carrier image and the recovered carrier image after the watermark extraction. The calculation formula is shown in Eq. (8):

$$NC=\frac{\sum_{i=0}^{L-1}\sum_{j=0}^{K-1}I(i,j)I'(i,j)}{\sum_{i=0}^{L-1}\sum_{j=0}^{K-1}\left[I(i,j)\right]^2} \qquad (8)$$

Where $I(i,j)$, $I'(i,j)$ respectively denote the pixel values at (i,j) of the original image and the recovered carrier image after the watermark extraction. L and K denote respectively the rows and columns of the image. For the original carrier image and the recovered carrier image, NC is required to be 1, which means the carrier images are generally completely recovered.

Table 1. Integrity assessment table without attack

Image (512×512)	*Lena*	*Plane*	*Baboon*	*Peppers*
NC	1	1	1	1

Table 1 denotes the integrity of the results of the 4 different types of watermarked images without any attack based on this algorithm. It shows that the original image can be recovered completely without the attack. This indicates that the algorithm is reversible.

4.2 Visual quality assessment

Currently, the PSNR (peak signal-to-noise ratio) is one of the main indicators that evaluate the visual quality of reversible watermarking. The greater the PSNR value is, the less the representative image is distorted, and the better visual quality the watermarked image has. The smaller the PSNR is, the more the representative image is distorted, and the more serious visual quality loss the watermark image suffers. The calculation formula of PSNR is shown in Eq. (9):

$$PSNR=10\log\left(\frac{255^2}{\frac{1}{MN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\left[I'(i,j)-I(i,j)\right]^2}\right) \qquad (9)$$

Where $I(i,j)$ and $I'(i,j)$ respectively denote the pixel values at (i,j) of the original image and the watermarked image. M and N respectively represent the rows and columns of the image. It is generally believed that if the lost image quality is partial and the PSNR value on the visual quality of reconstruction is no less than 30 decibels, its visual quality is still acceptable.

4.3 Embedding capacity assessment

The reversible watermarking has a valid embedding capacity. That is to say, an important evaluation indicator of embedding capacity is that the number of bits can be embedded into per pixel. The embedding quantity can be divided into actual embedding quantity and effective embedding quantity. The actual embedding quantity is the number of bits embedded in the information hiding method by itself. The effective embedding quantity expresses that the actual embedding quantity subtracts additional information (in order to restore images to be transmitted to the receiver or with secret information embedded into the image information). Commonly the effective embedding quantity index is used as an embedding rate ER (Embedding Rate), and is expressed as:

$$ER = \frac{Num_{sec} - Num_{extra}}{Num_{pixel}} \tag{10}$$

In Eq. (10), Num_{sec} denotes the number of bits in the secret embedded information; Num_{extra} denotes the number of bits in the additional information; Num_{pixel} denotes the number of pixels in the image carrier. The embedding rate can be more intuitionistic to reflect the embedding capacity size of a scheme.

4.4 Structural similarity

Although the PSNR has been widely used in stego image quality assessment, its mathematical definition has some limitations [21-22], and the pixel correlation and human visual system characteristics are not taken into account. Under this background, a method of image quality assessment, which is based on the similarity of image structure, was proposed by Wang [23] *et al.* in 2004. The structural similarity (SSIM) has been used to evaluate the quality of the stego images. It is defined as (11):

$$SSIM(x, y|w) = \frac{(2\overline{w}_x\overline{w}_y + C_1)(2\sigma_{w_x w_y} + C_2)}{(\overline{w}_x^2 + \overline{w}_y^2 + C_1)(\sigma_{w_x}^2 + \sigma_{w_y}^2 + C_2)} \tag{11}$$

where C_1 and C_2 are small constants, $\overline{w}_x$ is the mean value of the region w_x, and $\overline{w}_y$ is the mean value of the region w_y. $\sigma_{w_x}^2$ is the variance of w_x, and $\sigma_{w_x w_y}$ is the covariance between the two regions. The higher value of the SSIM implies a higher quality level for the stego image.

The value range of SSIM is [0, 1]. SSIM= 0 indicates that the corresponding two images are completely different. SSIM=1 indicates that the corresponding two images are exactly the same. In the evaluation of the reversible information hiding algorithm, the SSIM value of the hidden image is close to 1 as much as possible.

All things considered, whether to embed watermark information by using the optimized method mainly depends on the watermark capacity to be embedded. It means whether to choose a single-layer, two-layer or multiple-layer method to embed watermark information. Generally, if the embedded watermark is too small, it is recommended that the single-layer self-adaptive difference expansion reversible watermark method is used for embedding. This is because under a specific threshold, the small watermark can be embedded into the smooth part of the carrier image once for all, without any further layer-dividing embedment.

If the watermark information to be embedded has a large capacity, by using single-layer embedding the watermark information may be embedded into the complicated texture area of the carrier image, thus affecting the visual effect of the watermark image. So this kind of watermark needs layer-dividing embedment with an optimized method. When conducting layer-dividing embedment, a reasonable T_1 should be set as the threshold value. Different image types require different T_1 values. Considering both the visual quality and the embedding quality, this paper uses the smooth blocks to be embedded, which occupy 40% of the total sub-blocks of the image as the alternate value of T_1 to layer-dividing selects the smooth block to embed the watermark information.

This paper embeds 32×32 binary images into the 512×512 carrier image. Limited by the watermark information capacity, the paper directly adopts the single-layer self-adaptive differential expansion reversible watermark method for embedding.

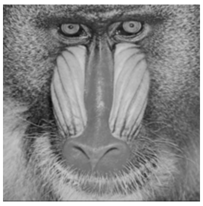
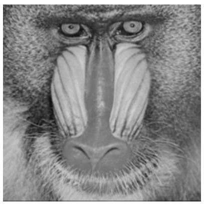
The algorithms in this paper and in literature [18] based on the original carrier image after using 4×4 blocks, the PSNR and SSIM are shown in **Table 2** (the average of data taken 20 tests).

Table 2. Comparison of PSNR (dB) and SSIM

Image Name	**Proposed Algorithm**		**Literature [18] algorithm**	
	PSNR	**SSIM**	**PSNR**	**SSIM**
Lena	59.36	0.992	45.72	0.982
Baboon	58.21	0.989	47.37	0.985
Plane	61.31	0.996	48.26	0.987
Peppers	58.32	0.990	45.81	0.981

Compared with the algorithm in literature [18], the highest PSNR of the above 4 original carrier images in this algorithm is 61.31dB. This means that this algorithm has better invisibility. At the same time, SSIM is also higher than that in literature [18]. Form **Table 2**, it is easy to note that the proposed algorithm outperforms that in Literature [18] in terms of the same payload capacity with good SSIM and PSNR values. The results presented here demonstrates that the proposed algorithm significantly increases the quality of stego images. Specific effects of visual and extraction are shown in **Table 3**:

Table 3. Algorithm experiments visual effects

Image name	**Original image**	**Watermarked image**	**Original watermark**	**Extracted watermark**	***PSNR***
Lena					59.36
Baboon					58.21
Plane					61.31
Peppers					58.32

As observed from the figures, we find that human eyes do not feel the existence of watermark information in watermarked images. The corresponding PSNR value shows that images have better imperceptibility by using different types of algorithms. When the watermarked image has a good visual effect, the average PSNR value can be as high as 59.3dB. From Tables 2 and 3, we can see that this algorithm has a good imperceptibility for different texture images, and can meet the requirements of visual perception.

In calculating the maximum embedding capacity of the original image, due to the large capacity of the embedded watermark, the optimization method is needed to embed the watermark. In order to calculate conveniently and better analyze

the performance of the algorithm, the two-layer embedded algorithm is used to embed the watermark.

Table 4. Comparison between this algorithm, literature [18] method, and literature [24] method in terms of payload capacity, SSIM, and PSNR for original images

Image	method	Payload in bytes	SSIM	PSNR (%)				
				10	30	70	90	100
Lena	literature [24]	276599	0.7331	54.37	42.92	36.83	32.91	31.95
	literature [18]	9767	0.9188	45.78	43.48	41.58	39.78	38.64
	this algorithm	184652	0.8331	47.86	45.89	39.64	38.10	37.65
Baboon	literature [24]	41519	0.8232	51.25	42.43	34.94	33.48	31.02
	literature [18]	11056	0.9047	45.86	43.84	41.95	40.56	39.76
	this algorithm	134614	0.8547	49.86	46.84	40.75	39.36	38.46
Plane	literature [24]	392740	0.6724	50.12	48.54	46.80	45.11	31.65
	literature [18]	13232	0.8931	45.66	42.89	39.54	37.10	36.65
	this algorithm	186922	0.8412	48.76	45.76	40.38	38.78	37.90
Peppers	literature [24]	138695	0.8154	52.62	42.98	35.12	33.58	31.53
	literature [18]	8562	0.9369	46.24	44.77	41.82	39.54	38.26
	this algorithm	169704	0.8183	48.89	45.86	39.72	37.84	37.21

The *PSNR* is utilized to estimate the deformation between the original cover image and resulted stego image when embedding 10, 30, 70, 90 and 100% from the allowed cover image capacity. Form **Table 4**, it is easy to note that the proposed adaptive difference expansion-based reversible watermarking technique outperforms the techniques in literature [18] and literature [24] in terms of payload capacity with good SSIM and PSNR values. The results presented here demonstrated that the proposed adaptive difference expansion-based reversible watermarking technique significantly increases the payload capacity while still keeping the quality of stego images.

Table 5. The comparison of PSNR and maximum payload in different sizes

original image	2×2		3×3		4×4		6×6	
	PSNR	Payload	PSNR	Payload	PSNR	Payload	PSNR	Payload
Lena	40.62	102646	41.34	106514	42.44	110748	41.75	111648
Baboon	41.87	76489	42.19	78682	44.15	81457	42.91	816745
Plane	41.25	108461	42.22	111245	43.51	117466	42.73	119432
Peppers	41.06	90152	41.88	95260	43.26	99362	42.39	100277

As is shown, the visual quality of the watermarked image is influenced by the size of the image sub-block; whether the sub-block is small or big, the image quality will become worse (taking one-layer embedding as the example). The maximum load of the original image is also slightly changed with the sub-block size, mainly due to the influence from the pixel point overflow.

Taking Lena as an example, when the sub-block size is 2×2 or 3×3, the image is imperceptible and has poor performance. When it increases to 4×4, the performance is better. When the sub-block size is 6×6, the PSNR value is a little higher than 3×3 but lower than 4×4. For different sub-blocks, the maximum embedding capacity will be slightly changed mainly due to the impact of the pixel point overflow.

1) Performance comparison of single-layer embedding and other algorithms

Table 6. The comparison of PSNR and maximum payload from different algorithms

original image	**Kim algorithm**		**Hong algorithm**		**Li algorithm**		**This algorithm**	
	PSNR	**Payload**	**PSNR**	**Payload**	**PSNR**	**Payload**	**PSNR**	**Payload**
Lena	49.77	34123	49.98	47549	50.23	30678	42.44	110748
Baboon	49.50	11279	51.24	13024	50.37	14139	44.15	81457
Plane	49.98	51142	50.31	64923	51.16	39187	43.51	117466
Peppers	49.68	27045	49.87	34758	50.41	28961	43.26	99362

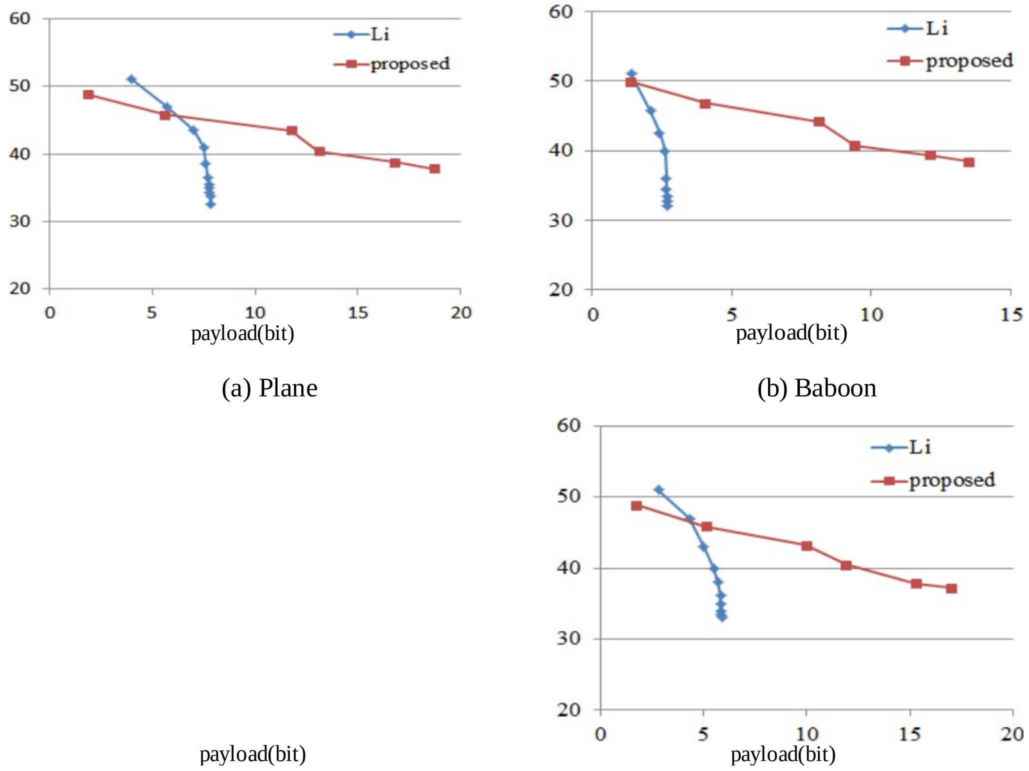
The performance of this algorithm is compared with that of the Kim algorithm [26], Hong algorithm [27] and Li algorithm [28], and the results are shown in **Table 6**. Among them, the Kim algorithm embedded data by selecting four sub graphs, and the embedded parameter L was set to 0; the Hong algorithm selected the bilinear interpolation method, and the values of T_1 and T_0 parameters were set to 8 and 60; the Li algorithm embedded watermark in light of the prediction-error expansion algorithm; In this paper, the DE algorithm is used to embed the watermark, and the block size is set to 4×4. Experimental results show that the performance of this algorithm is better than that of the Kim algorithm, Hong algorithm and Li algorithm. Compared to Kim and Hong algorithms, for simple texture images, both the quality and maximum payload of stego images obtained by this algorithm increase. And for texture complex images, the performance of this algorithm is slightly decreased, but the performance of this algorithm can be improved by adjusting the parameters. Compared with the Li algorithm, the proposed algorithm produces more predictive error bits, which can be used more fully and can be embedded in a larger capacity. For example, for the test image Lena, if the block size is set to 4×4, the stego image quality will be 42.44dB, the maximum payload will be 110748, which are slightly better than the other three algorithms.

2) Performance comparison of two-layer embedding and other algorithms

Table 7. The PSNR and maximum payload for multi-level embedding

original image	This algorithm				literature [25]	
	k=1		K=2			
	PSNR	**Payload**	**PSNR**	**Payload**	**PSNR**	**Payload**
Lena	42.44	110748	37.65	184652	33.92	215849
Baboon	44.15	81457	38.46	134614	27.82	136754
Plane	43.51	117466	37.90	186922	36.85	221475
Peppers	43.26	99362	37.21	169704	31.43	178912

Table 7 shows the experimental data generated when this algorithm is used in two-layer embedding. It can be seen that with the increase of the number of the embedding layer, the effective payload will substantially increase, and the visual quality of the stego image has declined. With respect to all watermark information embedded through single-layer, the visual quality of the stego image through two-layer is much higher. In this algorithm, the block size is set to 4×4.



(a) Plane

(b) Baboon

(c) Lena

(d) Peppers

Fig. 8. The comparison of various PSNR and payload from different algorithms

Fig. 8 (a)-(d) show the effective payload and the change trend of PSNR in different test images and using different algorithms for two-layer embedding. Because the algorithm adopts the generalized difference expansion, the embedding rate is high. In the same effective payload conditions, the least number of pixels is modified by this algorithm, so the PSNR values of stego images are generally higher than that of the other algorithms and hidden better. For the Baboon and Peppers images, the performance of the algorithm is decreased obviously when the number of embedding layers is large. As a result, the algorithm is more suitable for simple texture images with respect to complex texture images.

Fig. 8 shows the performance comparison between our proposed method and other related difference expansion-based data hiding methods. From the experimental results, we can see that our proposed method achieves the better performance than Li method. At the initial stage, the algorithm and Li algorithm yield similar performance, but with the increase of the embedding capacity, this algorithm has significantly better performance than Li algorithm.

5. Conclusion

The paper presents an improved difference expansion reversible watermarking scheme that is capable of providing a high payload for original images. The proposed method can restore the original image and extract the watermark from the stego image. The major contribution is that the proposed difference expansion-based reversible data hiding method can improve the embedding capacity while still maintaining good quality of stego image, while other techniques cannot. The performance of the proposed method is investigated. The test results ensure that the proposed method outperforms the existing schemes in terms of payload while preserving good stego image quality level.

Reference

[1] M. J. Sahraee, S. Ghofrani, “A robust blind watermarking method using quantization of distance between wavelet coefficients,” *Signal, Image and Video Processing*, 7:799-807, 2013. Article (CrossRef Link)

[2] Li, B., He, J., Huang, J., Shi, Y.Q., “A survey on image steganography and steganalysis,” *J. Inf. Hiding Multimed. Signal Process*, 2(2), 142-172, 2011. Article (CrossRef Link)

[3] Hussain, M., Hussain, M., “Survey of image steganography techniques,” *Int. J. Adv. Sci. Technol*. 54, 113-124, 2013. Article (CrossRef Link)

[4] Tian, J., “Reversible data embedding using a difference expansion,” *IEEE Trans. Circuits Syst. Video Technol*. 13(8), 890-896, 2003. Article (CrossRef Link)

[5] Tzu-Chuen, L., Chin-Chen, C., “Lossless nibbled data embedding scheme based on difference expansion,” *Image Vis. Comput*. 26, 632-638, 2008. Article (CrossRef Link)

[6] Luo L X, Chen Z Y, Chen M, "Reversible image watermarking using interpolation technique," *IEEE Transactions on information forensics and security*, 15(1): 187-193, 2010. Article (CrossRef Link)

[7] Lee C F, Chen H L, Tso H K, "Embedding capacity raising in reversible data hiding based on prediction of difference expansion," *The Journal of System*, 83(10):1864-1872, 2010. Article (CrossRef Link)

[8] Hong W, Chen T S, "A local variance-controlled reversible data hiding method using prediction and histogram-shifting," *The Journal of System and Software*, 83(12): 2653-2663, 2010. Article (CrossRef Link)

[9] Chrysochos, E., Fotopoulos, V., Skodras, A.N., "A new difference expansion transform in triplets approach for reversible data hiding," *Int. J. Comput. Math*. 88(10), 2016-2025, 2011. Article (CrossRef Link)

[10] Liu YC, Wu HC, Yu SS, "Adaptive DE-based reversible steganographic technique using bilinear interpolation and simplified location map," *Multimed Tools Appl*, 52(2-3):263-276, 2011. Article (CrossRef Link)

[11] Tian J, "Reversible data embedding using a difference expansion," *IEEE Trans Circ Syst Video Technol*, 13(8):890-896, 2011. Article (CrossRef Link)

[12] Tseng HW, Hsieh CP, "Prediction-based reversible data hiding," *Inform Sci*, 179(14): 2460- 2469, 2009. Article (CrossRef Link)

[13] Alattar AM, "Reversible watermark using the difference expansion of a generalized integer transform," *IEEE Trans Image Process*, 13(8):1147-1156, 2004. Article (CrossRef Link)

[14] Maity H K, Maity S P, "Reversible image watermarking using modified difference expansion," in *Proc. of Emerging Applications of Information Technology (EAIT), 2012 Third International Conference on. IEEE*, 17(3): 320-323, 2012. Article (CrossRef Link)

[15] Li X.L, Li J, Li B, et al., "High-fidelity reversible data hiding scheme based on pixel-value ordering and prediction-error expansion," *Signal Processing*, 93(1): 198-205, 2012. Article (CrossRef Link)

[16] LIN S L, HUANG C- F. LI0U M H, et al., "Improving histogram based reversible information hiding by an optimal weight-based prediction scheme," *Journal of Information hiding and Multimedia Signal Processing*, 1(1): 19-33, 2013. Article (CrossRef Link)

[17] Tzu-Chuen Lu, Chin-Chen Chang, Ying-Hsuan Huang, "High capacity reversible hiding scheme based on interpolation, difference expansion and histogram shifting," *Multimedia Tools and Applications*, 72: 417-435, 2014. Article (CrossRef Link)

[18] Hala S. El-sayed, S. F. El-Zoghdy, Osama S. Faragallah, "Adaptive Difference Expansion-Based Reversible Data Hiding Scheme for Digital Images," *Arabian Journal for Science and Engineering*, 41: 1091-1107, 2016. Article (CrossRef Link)

[19] Ma Kun, Niu Xin-Xin, "An improved reversible watermarking scheme," in *Proc. of International Conference on Signal Processing. Beijing: [s. n.]*, 2229-2232, 2008. Article (CrossRef Link)

[20] Li Zhuo, Chen Xiao-Ping, Pan Xue-Zeng, et al., “Lossless data hiding scheme based on adjacent pixel difference,” in *Proc. of the 8th International Conference on Computer Engineering and Technology. Washington, DC: IEEE Computer Society*, 588-592, 2009. Article (CrossRef Link)

[21] WANG Z, BOVIK A C, “A universal image quality index,” *Signal Processing Letters, IEEE*, 9(3): 81-84, 2002. Article (CrossRef Link)

[22] WANG Z, BOVIK A C, LU L, “Why is image quality assessment so difficult,” in *Proc. of ICASSP2002: proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*: Vol 4, IV-3316, 2002. Article (CrossRef Link)

[23] WANG Z, BOVIK A C, SHEIKH H R, et al., “Image quality assessment: from error visibility to structural similarity,” *IEEE Transactions on Image Processing*, 13(4): 600-612, 2004. Article (CrossRef Link)

[24] Shahidan M. Abdullah, Azizah A. Manaf, “Multiple Layer Reversible Images Watermarking Using Enhancement of Difference Expansion Techniques,” *Networked Digital Technologies*, Part I, CCIS 87, pp. 333-342, 2010. Article (CrossRef Link)

[25] Shaowei Weng, Jeng Shyang Pan, “Adaptive reversible data hiding based on a local smoothness estimator,” *Multimedia Tools and Applications*, 74: 10657-10678, 2015. Article (CrossRef Link)

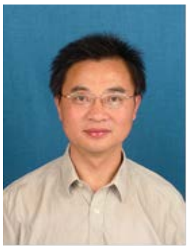
[26] Kim K.S, Lee M.J, Lee H.Y, et al., “Reversible data hiding exploiting spatial correlation between sub-sampled images,” *Pattern Recognition*, 42(11):3083-3096, 2009. Article (CrossRef Link)

[27] Hong W, “Reversible data embedding for high quality images using interpolation and reference pixel distribution mechanism,” *Journal of Visual Communication and Image Representation*, (22):131-140, 2011. Article (CrossRef Link)

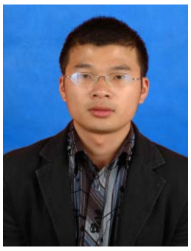
[28] Li X.L, Li J, Li B, et al., “High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion,” *Signal Processing*, 93(1):198-205, 2012. Article (CrossRef Link)

Zhengwei Zhang received the BS degree in Computer Science and Technology from the Jiangsu University of Science and Technology in 2004. He received the MS degree in Computer Science and Technology from the Jiangnan University in 2011. He is pursuing the Ph.D. degree with College of Command Information System, PLA University of Science and Technology. He is on the subject of information security and image processing. His current research interests include information hiding and digital watermarking.

Lifa Wu is an associate professor in the Department of College of Command Information System at PLA University of Science and Technology, China. He received the BS degree in Computer Science and Technology from the PLA College of Communication Engineering in 1991. He received the MS degree in Computer Science and Technology from the PLA College of Communication Engineering in 1995. He received the Ph.D. degree with College of Computer, Nanjing University. He is on the subject of information security and network security. His current research interests include information security and satellite communication.



Yunyang Yan received the BS degree in computer application from the Nanjing University of Aeronautics and Astronautics in 1988. He received the MS degree in computer application from the Southeast University in 2002 and the PhD degree from the Nanjing University of Science and Technology (NUST), on the subject of pattern recognition and intelligence systems in 2008. His current research interests include pattern recognition, computer vision.



Shaozhang Xiao received the BS degree in communication engineering from the Jiangnan University in 2004. He received the MS degree in Computer Science and Technology from the Jiangnan University in 2011. His current research interests include signal processing, numerical analysis.

Shangbin Gao received the BS degree in mathematics from the Northwestern Polytechnical University in 2003. He received the MS degree in applied mathematics from the Nanjing University of Information and Science and Technology in 2006. He received the Ph.D. degree with School of Computer Science and Technology, Nanjing University of Science and Technology (NUST). His current research interests include pattern recognition and computer vision.