



Research article

Large-capacity reversible image watermarking based on improved DE

Shaozhang Xiao, Xingyuan Zuo, Zhengwei Zhang* and Fenfen Li

Faculty of Computer and Software Engineering, Huaiyin Institute of Technology, Huai'an, Jiangsu 223003, China

* **Correspondence:** Email: zzw49010650@sina.com; Tel: +8613952399281.

Abstract: Aiming at solving the problems of bad imperceptibility and low embedding rate of existing algorithms, a novel large-capacity reversible image watermarking based on improved difference expansion (DE) is proposed. Firstly, the smoothness calculation algorithm is used to calculate and sort the smoothness values of the divided image sub-blocks; then, the scrambled watermark is embedded into the sub-blocks with less smoothness after removing the abrupt point by using the generalized difference expansion (GDE); finally, the absolute difference operation is applied to the generated overflow pixels to make their pixel values within a reasonable range for embedding watermark information. Under the premise of ensuring a certain visual quality, multiple watermark embedding can effectively improve the embedding rate. The simulation results show that this algorithm not only realizes blind extraction, but also recovers the original images without loss. At the same time, this algorithm achieves a high embedding rate (the average embedding rate is as high as 77.91 dB) without decreasing the visual quality.

Keywords: difference expansion (DE); reversible image watermarking; overflow processing; Arnold transform

1. Introduction

Reversible image watermarking technology requires that the embedding of the watermark information into the original images should be conducted on the premise of ensuring the visual quality of the images; meanwhile the original images can be restored nondestructively after extracting the watermark. Compared with the digital image watermarking 1, reversible image watermarking has higher requirements on watermark embedding, which also makes it more widely studied and applied

in military, medical, and other fields with high image authenticity and integrity. One of the fundamental goals of watermarking is to increase the embedding rate while reducing the certain visual quality of the image.

Tian 2 proposed a reversible watermarking technology with high embedding rate, which used the difference expansion (DE) to embed watermark based on adjacent pixel pairs. This algorithm has attracted great attentions. Consequently, a number of improved embedding algorithms have been developed based on the DE.

However, using the DE method, an overflow will be generated, which will degrade the watermarking embedding quality. Therefore, removing the overflow location map is greatly valuable to enhance the performance of the algorithms. A reversible watermarking algorithm that combined the DE and the reversible contrast map was proposed in 3. In this algorithm, the contrast pixel pair was mainly used to embed a small amount of auxiliary information instead of the location map, and the embedding capacity was greatly enhanced. However, the deterioration in image quality was more serious. Wang et al. 4 proposed a reversible watermarking method based on difference histogram. It was more unique in terms of overflow processing, but it still needed to embed a compressed positioning map. A reversible image watermarking scheme based on the DE effectively tackled the issues of overflow and underflow encountered during embedding in [5]. A quadratic DE based reversible technique was developed by Zhang et al. [6] where the DE was applied twice to improve the security and embedding capacity while ensuring the quality. The location map was not generated, which further allowed hiding large watermark.

The reversible image watermarking algorithm combining the DE with the least significant bit (LSB) was proposed in 7. The algorithm realized greater embedding capacity while maintaining better imperceptibility. By combining histogram transform and prediction DE, Lin et al. 8 put forward a reversible watermarking technology, which effectively expanded the embedding capacity and achieved promising results.

Muhammad et al. [9] introduced a reversible image watermarking algorithm based on Genetic Programming (GP). Firstly, the algorithm resolved the image pixels, then executed integer wavelet transform (IWT) on the adjusted image, and blocked the sub-band coefficients without overlapping. The GP algorithm was implemented to achieve the optimal load for coefficient compression, and subsequently, the LSB was utilized to embed the watermark into the compressed coefficients. The algorithm guaranteed that the image embedded with the watermark had better visibility and higher payload. A reversible image watermarking algorithm based on local prediction was put forward by Vinoth et al. [10]. This algorithm divided image into many blocks with no superposition. Pixels on the edge of the block were detected by Median Edge Detection predictor (MED), and other pixels were predicted by local prediction. It had better visual quality and lower embedding rate. Jia et al. [11] proposed a reversible data hiding algorithm to lessen an invalid pixel shift in histogram shift. Based on image texture, the algorithm reduced the invalid shift of pixels in the histogram shift. The algorithm had distinguished embedding capacity and visual quality.

Literature [12] proposed an enhanced reduced DE based reversible image watermarking aiming to reduce the loss of image quality during embedding. Likewise, Yu et al. 12 also proposed a reversible watermarking algorithm based on multidimensional prediction error expansion, which can reduce the embedding distortion by discarding the embedding map that may produce high distortion. The algorithm was more suitable for images with smooth and simple textures. In addition, a reversible watermarking based on double-layer difference expansion was proposed in 14. In this algorithm,

smooth pixel pairs were preferred to embed watermark, and the distortion can be well controlled. However, the embedding rate was not high.

Based on the analysis of the traditional DE watermarking, an improved large-capacity reversible image watermarking algorithm based on the DE is proposed in this paper, which can effectively improve the embedding capacity and reduce the image distortion. It provides an effective solution to avoiding pixel overflow.

2. Research background

The reversible image watermarking studied mainly uses multi-scale decomposition, interpolation expansion, generalized difference expansion (GDE) and overflow processing to realize the embedding and extraction of watermarking algorithm.

2.1. GDE method

Compared to the adjacent pixel expansion algorithm presented by Tian [2], 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 $X = (x_0, x_1, x_2, x_3, \dots, x_{n-1})$ is a set of pixel values, the direct transform of the generalized integer transform is:

$$\begin{aligned} \bar{x} &= \left[\frac{\sum_{i=0}^{n-1} a_i x_i}{\sum_{i=0}^{n-1} a_i} \right] & (1) \\ d_1 &= x_1 - x_0 \\ d_2 &= x_2 - x_0 \\ &\vdots \\ d_{n-1} &= x_{n-1} - x_0 \end{aligned}$$

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

$$d'_i = 2 \times d_i + b \quad (2)$$

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:

$$\begin{aligned} x'_0 &= \bar{x} - \left[\frac{\sum_{i=1}^{n-1} d'_i}{\sum_{i=0}^{n-1} a_i} \right] & (3) \\ x'_1 &= x'_0 + d'_1 \\ x'_2 &= x'_0 + d'_2 \\ &\vdots \\ x'_{n-1} &= x'_0 + d'_{n-1} \end{aligned}$$

A set of pixel values $X' = (x'_0, x'_1, x'_2, x'_3, \dots, x'_{n-1})$ is generated by the generalized difference expansion algorithm, in which the mean of the group pixels is:

$$\bar{x}' = \left\lfloor \frac{\sum_{i=0}^{n-1} a_i x'_i}{\sum_{i=0}^{n-1} a_i} \right\rfloor \quad (4)$$

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.

2.2. Multi-scale decomposition

An image F is divided into multiple scales from a whole to parts, thus obtaining sub-image blocks $F_{i,j}$ at different scales, where $i = 1, 2, \dots, d$ is denoted as the segmentation scale; $j = 1, \dots, 4^{i-1}$ is denoted as the number of sub-blocks at each scale. Due to the limited size of the image, the segmentation scale is also limited.

The segmentation method is displayed in Figure 1. From left to right there is segmentation from the 1st to the 3rd scales, respectively. The whole image is the sub-blocks $F_{1,1}$ obtained after being decomposed at the 1st scale, and then the whole image is divided into four sub-blocks $F_{2,1}, F_{2,2}, F_{2,3}$ and $F_{2,4}$; each sub-block is called a second scale sub-block, and then they are divided until they reach the maximum scale set.

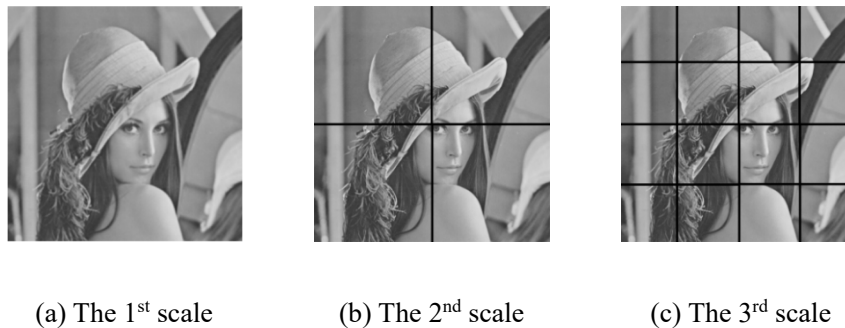


Figure 1. Multi-scale segmentation of *Lena* image.

As the traditional multi-scale decomposition also belongs to fixed-sized block decomposition method, this paper makes an improvement of the multi-scale decomposition.

The improved image multi-scale decomposition is to divide a rectangular image into 4 equal-sized square blocks, and then determine whether or not these four square blocks meet the homogeneity criterion. If it is satisfied, the current block maintains unchanged; otherwise it continues to be decomposed into four square blocks, and determines whether or not they meet the criterion, until all the blocks meet the given criterion. The decomposition criterion can be expressed as:

$$|p_i - p_{ave}| > (g_l - 1) \times \gamma \quad (5)$$

In Formula (5), P_i and P_{ave} are the gray value of any pixel and the average gray value of all

pixels in a square block, respectively; g_l is the gray level of a pixel; γ is a decimal in the range of $[0,1]$. The criterion is, when the guidelines when the absolute value of the difference between the gray value of either pixel and the average gray value of all pixels in the square block is greater than $(g_l - 1) \times \gamma$, the block needs to be further divided, as shown in Figures 2 and 3.

As per the block division method, it divides images into unfixed sizes. Also from the perspective of the division results, the image blocks decomposed have pixels with high homogeneity, which are suitable for lossless watermark embedding. As specified in the algorithm, the minimum block size is 4×4 .



Figure 2. Original image.

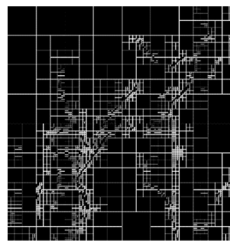


Figure 3. The generated block map after modified multi-scale decomposition.

After multi-scale decomposition, the size of each image block is $2^{2+n} \times 2^{2+n}$, $n \in \{0,1,2 \dots,7\}$. The size information of each block can be converted into binary form. Each of the decomposed sub-blocks is coded by the sub-block size, as shown in Table 1.

Table 1. Scale coding of image block after multi-scale decomposition.

Size / pixels	code	Size / pixels	code
4×4	000	64×64	100
8×8	001	128×128	101
16×16	010	256×256	110
32×32	011	512×512	111

After multi-scale decomposition, the image sub-blocks obtained are sorted (from top to bottom, from left to right), and the scale information of each sub-block is recorded sequentially as per the ordering result, thereby constituting the decomposition information q of the original image. Given there are a small number of large-sized blocks decomposed, Huffman encoding [15] can be used to further reduce the length of the image decomposition information, denoted as $\text{Huf}(q)$.

For security of the algorithm, the length and encode table of the parameter Huf (q) are sent to the recipient in form of a secret key.

2.3. Overflow processing

The GDE algorithm may produce an overflow after embedding watermark. The overflow processing will have a significant impact on the watermark embedding capacity. In this paper, the overflow points generated by embedding the watermark into the original image are labeled in a location map having the same image size. The generated location map is a binary image marked by 0 or 1 for each pixel. At the same time, for the overflow point, the original pixel value is replaced by the generated value by using the GDE. The difference between the generated value and the maximum pixel value of the image is taken into consideration by the inverse DE to calculate the new pixel value.

The 2×2 image block contains 4 pixel values (a, b, c, d). After embedding the watermark through the GDE, it becomes 4 new pixel values (A, B, C, D). If the generated second pixel overflows, it will be overflowed to make its gray value within the range of 0–255 in order to complete the watermark embedding. The specific adjustment is carried out as follows:

If the generated pixel value B is positive overflow, the transformed pixel value is $255 - |B - 255|$.

If the generated pixel value B is negative overflow, the transformed pixel value is $0 + |B - 0|$.

For instance, if the pixel value generated by the DE is 266, the new image pixel value will be 244. The transform is detailed as follows:

$$266 - 255 = 11$$

$$255 - 11 = 244$$

The original image is divided into sub-blocks with a size of 2×2 pixels. As shown in Figure 4, if the pixel values in one of the sub-blocks are (244, 254, 250, 248), the amount of the watermark embedded by the GDE is 3 bits. Suppose that the embedded watermark is (1, 1, 1), then, the pixel values of the generated image sub-block are (239, 260, 252, 248) by using a generalized difference transform processing. As a result, the value of the second pixel in the image sub-block will overflow and the overflow processing is needed to better embed the watermark.

Through performing inverse difference transform, the pixel values of the image sub-block are (239, 250, 252, 248). After embedding the watermark information (1, 1, 1) into the transformed sub-block by using the generalized difference transform again, the pixel values of the new sub-block are (230, 253, 257, 249). The sub-block overflows by using the GDE, and the pixel values of the generated sub-block are (230, 253, 253, 249) by performing inverse transformation.

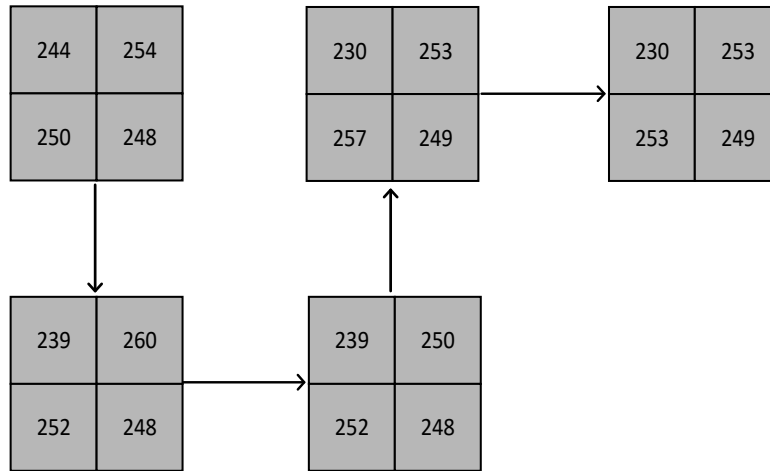


Figure 4. Overflow processing flow chart.

After overflow processing, the watermark can be embedded into the generated overflow pixels by using the GDE. The overflow point generated by the last DE is expressed by using overflow location. Keeping certain visual quality, the overflow algorithm can carry out multiple times for watermark embedding.

3. The proposed scheme

As mentioned in Section 1, the shortcomings in terms of embedding capacity and visual quality of the watermarked images in previous schemes is identified. At the same time, the existing algorithms do not deal with overflow very well. In order to solve the problems of bad imperceptibility and low embedding rate of the existing algorithms, a novel large-capacity reversible image watermarking based on the DE is proposed. The proposed scheme effectively controls the overflow of pixel values when embedding watermark with the DE, and uses multi-scale decomposition technology to remove the abrupt points in each image-block. After the information is embedded into the sub-block by the GDE, the average value of the pixels in each sub-block remains unchanged to extract the watermark information effectively. The scheme is made of two procedures: watermark embedding and watermark extraction.

3.1. Watermark embedding

The embedding process of the reversible watermarking algorithm is shown in Figure 5, and the specific operation flow is described as follows:

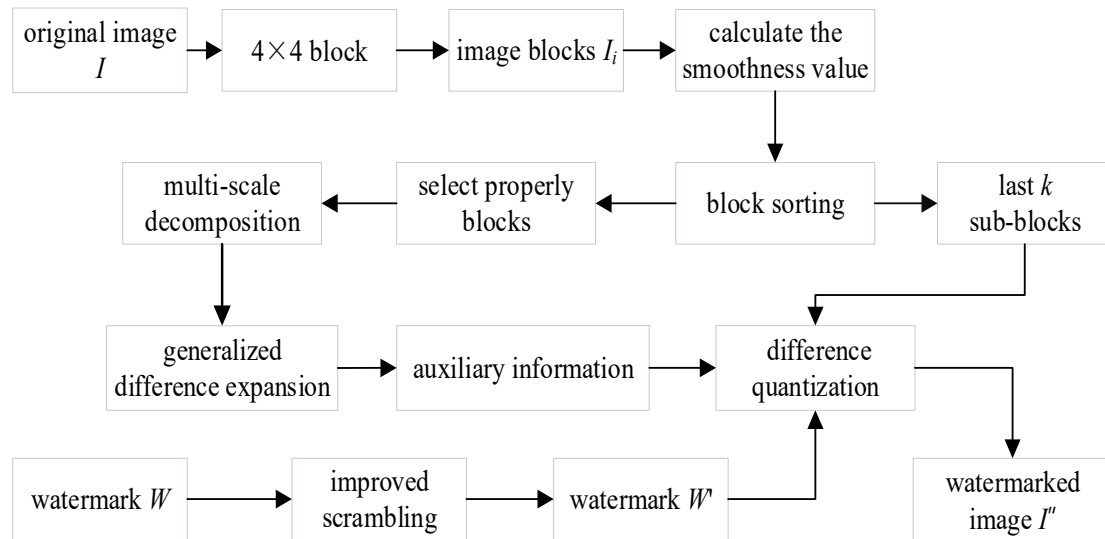


Figure 5. Watermark embedding flow chart.

STEP1: The watermark W' is obtained when the watermark W is scrambled by using Arnold transform 16. The W' is converted into a one-dimensional binary sequence. To enhance robustness and security, the scrambling method is improved as follows:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}^c \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}^d \begin{pmatrix} x \\ y \end{pmatrix} \bmod M \quad x', y' \in \{0, 1, 2, \dots, N-1\} \quad (6)$$

where (x, y) is the coordinate of the original image pixel, (x', y') is the transformed coordinate, M refers to the size of the image matrix, and c and d are the randomly generated scrambled numbers.

STEP2: The original image $I (M \times N, M$ and N are all integer multiples of 4) is divided into non-overlapping image sub-blocks $I_i, i \in \left(0, \frac{M \times N}{4 \times 4} - 1\right)$;

STEP3: The smoothness value of each block I_i is calculated 18, then all sub-blocks are sorted in an ascending order with respect to their smoothness values, along with the establishment of a sort index table according to the sorting results. Finally, the blocks in front of the sequence are selected properly to embed the watermark in turn;

STEP4: There may be abrupt points in the selected image sub-block I_i . In this regard, the multi-scale decomposition algorithm is applied to delete the image sub-blocks with abrupt points. The effective watermark information will not be embedded into these blocks. Finally, according to the number of the embedded watermarks, the pixel blocks in front of the sequence (image sub-blocks containing abrupt points have been removed) are selected to embed watermarks properly;

STEP5: Considering the requirement of watermark embedding capacity, the pixel sub-blocks (assuming the first n blocks) in front of the sequence are selected to embed the watermark in turn. Note that, the algorithm removes the selected image sub-blocks with abrupt points (if there are m blocks). Therefore, the number of image blocks used for embedding watermark information is $n-m$. At the same time, the sorting sequence numbers of the image sub-blocks containing abrupt points are recorded for watermark extraction.

STEP6: The watermark is embedded into any selected sub-block A_i ($0 \leq i \leq n$) by using the GDE (For the block size of 4×4 pixels, the 15-bit binary watermark can be embedded). When using the DE to embed watermark, the pixels beyond the range of the gray values will be marked in the binary location map having the same size as the original image. Specifically, if an overflow occurs, a value of 1 is marked at the corresponding location, and if no overflow occurs, a value of 0 is marked at the corresponding location, thereby generating an overflow location map.

STEP7: For identifying the corresponding pixel points of overflows, a transform algorithm (Section 2) is adopted to avoid the pixel overflow. The pixel value transform is carried out for the identified pixels, so that the watermark can be embedded with the GDE again. The overflowed pixel value is beyond the range of the image gray values, and the absolute difference is calculated to make the pixel value within a reasonable range.

If the pixel value generated by the DE is positive overflow, that is a , the transformed pixel value is $255 - |a - 255|$.

If the pixel value generated by the DE is negative overflow, that is b , the transformed pixel value is $0 + |b - 0|$.

The auxiliary information (the compressed overflow graph, embedding capacity and scrambling times c and d , etc.) is hidden into the complex texture blocks.

STEP8: For the corresponding sub-blocks not used to embed the watermark in the original image I , those are the original sub-blocks with high texture complexities. Then, the last k sub-blocks are selected to realize the embedding of the auxiliary information by the difference quantization method and the selected k is saved for watermark extraction.

STEP9: The auxiliary information is embedded into each pixel in each selected pixel sub-block by difference quantization.

STEP9.1: Calculate the average pixel value of each original image pixel block corresponding to the last k sub-blocks in the sorting sequence as follows:

$$\overline{avg} = \left\lfloor \frac{x_1 + x_2 + \dots + x_{m \times n}}{m \times n} \right\rfloor \quad (7)$$

where m and n refer to the row and column of the divided sub-blocks, respectively. $x_1, x_2, \dots, x_m, x_n$ are the pixels contained in the sub-blocks.

STEP9.2: The maximum pixel value and the minimum pixel value in each sub-block are obtained, and the auxiliary information is embedded using difference quantization. Each image sub-block can embed two bits of auxiliary information using difference quantization. In each sub-block, one bit of the auxiliary information is embedded by comparing the minimum pixel value and the average value, and then the next bit of the auxiliary information is embedded by comparing the maximum pixel value and the average value.

1) Embed the auxiliary information by comparing the minimum pixel value and the average value.

$$a' = \begin{cases} a-1 & ((\lfloor \overline{avg} - a \rfloor) \% 2 = 1, w=1) \parallel ((\lfloor \overline{avg} - a \rfloor) \% 2 = 0, w=0) \\ a & ((\lfloor \overline{avg} - a \rfloor) \% 2 = 1, w=0) \parallel ((\lfloor \overline{avg} - a \rfloor) \% 2 = 0, w=1) \end{cases} \quad (8)$$

where, a represents the pixel value to be embedded; \overline{avg} refers to the average value of the pixels in the sub-block where the embedded pixel is located; w represents the embedded binary watermark information, and $\%$ represents the operation of model.

When embedding one bit of the auxiliary information by comparing the minimum pixel value and the average value, if $(\lfloor \overline{avg} - a \rfloor \% 2 = 1, w = 1)$ and $(\lfloor \overline{avg} - a \rfloor \% 2 = 0, w = 0)$ are established at the same time, the new minimum pixel value a' is equal to $a - 1$; if $(\lfloor \overline{avg} - a \rfloor \% 2 = 1, w = 0)$ and $(\lfloor \overline{avg} - a \rfloor \% 2 = 0, w = 1)$ are established at the same time, the new minimum pixel value a' is equal to a .

2) Embed the auxiliary information by comparing the maximum pixel value and the average value.

$$a' = \begin{cases} a+1 & ((\lfloor \overline{avg} - a \rfloor \% 2 = 1, w = 1) \parallel ((\lfloor \overline{avg} - a \rfloor \% 2 = 0, w = 0)) \\ a & ((\lfloor \overline{avg} - a \rfloor \% 2 = 1, w = 0) \parallel ((\lfloor \overline{avg} - a \rfloor \% 2 = 0, w = 1)) \end{cases} \quad (9)$$

When embedding one bit of the auxiliary information by comparing the maximum pixel value and the average value, if $(\lfloor \overline{avg} - a \rfloor \% 2 = 1, w = 1)$ and $(\lfloor \overline{avg} - a \rfloor \% 2 = 0, w = 0)$ are established at the same time, the new maximum pixel value a' is equal to $a + 1$; if $(\lfloor \overline{avg} - a \rfloor \% 2 = 1, w = 0)$ and $(\lfloor \overline{avg} - a \rfloor \% 2 = 0, w = 1)$ are established at the same time, the new maximum pixel value a' is equal to a .

STEP10: The watermarked image I_i is obtained by embedding the watermark information with the GDE and the differential quantization technology.

3.2. Watermark extraction

Suppose that the watermark is embedded into any pixel pair $\langle x, y \rangle$ by the DE. If the embedded watermark is 1, the value of the new pixel pair $\langle a, b \rangle$ will be:

$$a = \left\lfloor \frac{x+y}{2} \right\rfloor + \left\lfloor \frac{(x-y) \times 2 + 1}{2} \right\rfloor = \left\lfloor \frac{x+y}{2} \right\rfloor + (x-y) + 1 \quad (10)$$

$$b = \left\lfloor \frac{x+y}{2} \right\rfloor - \left\lfloor \frac{(x-y) \times 2 + 1}{2} \right\rfloor = \left\lfloor \frac{x+y}{2} \right\rfloor - (x-y) \quad (11)$$

Accordingly, $a - b = 2x - 2y + 1$

Therefore, for all pixel pairs, if the one-bit embedded watermark value is 1, the difference between the new pixel pair is odd; otherwise, if the one-bit embedded watermark value is 0, the difference between the new pixel pair is even. Accordingly, when resorting the original image, if the difference between the generated pixel pair $\langle a, b \rangle$ is odd, the one-bit watermark value is 1, otherwise it is 0. Based on this, the watermark can be extracted in order.

The watermark extraction flow is shown in Figure 6. The specific operation flow is detailed as follows:

STEP1: The watermarked image I'' ($M \times N$, M and N are all integer multiples of 4) is divided

into non-overlapping image sub-blocks I_i'' , $i \in \left(0, \frac{M \times N}{4 \times 4} - 1\right)$;

STEP2: Calculate the smoothness value of each block I_i'' , then sort all sub-blocks in an ascending order with respect to their smoothness values and create the sort index table accordingly. Since the average value of the pixels in each sub-block remains unchanged after the information is embedded into the sub-block with the GDE, the smoothness value of the image block stays unchanged before and after the watermark is embedded, and the two sort orders are the same [18].

STEP3: The posterior k sub-blocks in the sequence are selected to extract the auxiliary information by differential quantization.

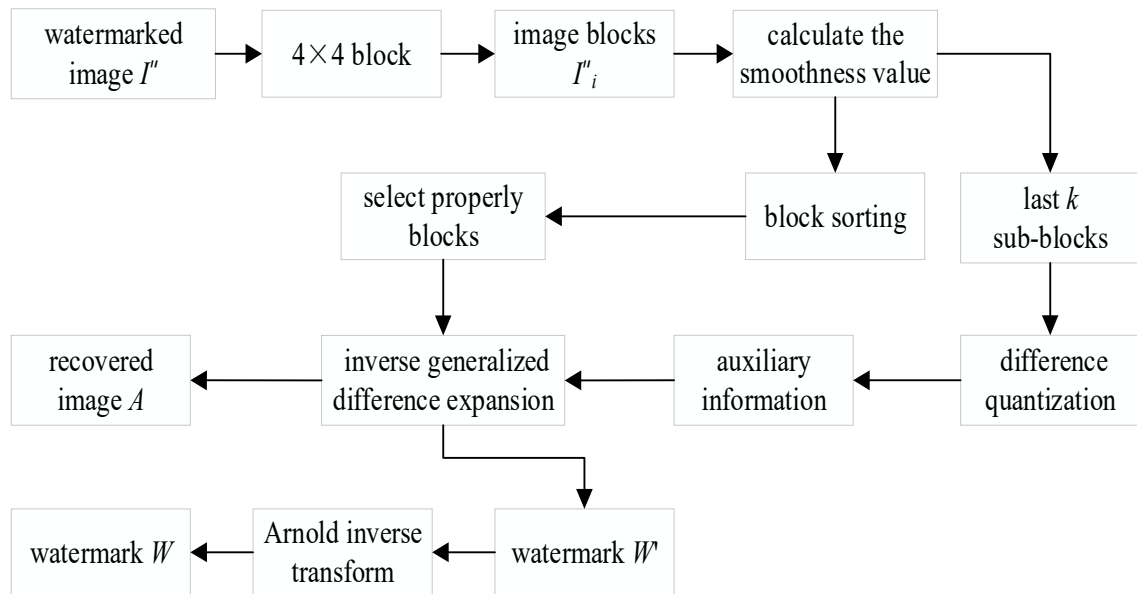


Figure 6. The flow chart of watermark extraction and image recovery.

The auxiliary information is extracted by the difference quantization method. The order of extracting information is opposite to that of embedding information. Recall that one bit of the auxiliary information is first embedded with a minimum pixel value, and then the next bit of the auxiliary information is embedded with a maximum pixel value. Therefore, in order to keep the parity relation between the minimum value and the average value, as well as between the maximum value and the average value, before and after embedding the auxiliary information, the maximum pixel value is first used to extract one bit of the auxiliary information, and then the minimum pixel value is used to extract another bit of the auxiliary information as well.

$$w = \begin{cases} 1 & (\lfloor \overline{avg} - a \rfloor) \% 2 = 0 \\ 0 & (\lfloor \overline{avg} - a \rfloor) \% 2 = 1 \end{cases} \quad (12)$$

where, a represents the pixel value used to extract the watermark, \overline{avg} refers to the average value of the pixels in the sub-block where the embedded pixel is located, w represents the embedded binary watermark information, and $\%$ represents the operation of model.

STEP4: According to the extracted auxiliary information, an inverse GDE is used to extract the

watermark from the first n blocks that are well-ordered (remove the image sub-blocks containing the abrupt points). The watermark information W is recovered by Arnold inverse transform.

STEP5: After extracting the watermark information, the recovered image I'' is combined with the corresponding sub-blocks containing the abrupt points to obtain the image I . The image I is then combined with the image obtained after extracting the auxiliary information to obtain the final image A .

4. Experimental simulation and result analysis

In this experiment, eight 8-bit standard gray images with the size of 512×512 pixels, including *Lena*, *Barbara*, *Baboon*, *Pepper*, *Girl*, *Cameraman*, *Man* and *Couple* were selected as the host images for watermark embedding (Figure 7) (all images were derived from <http://sipi.suc.edu/database>). A binary image with the size of 32×32 pixels was selected as the watermark image (Figure 8).

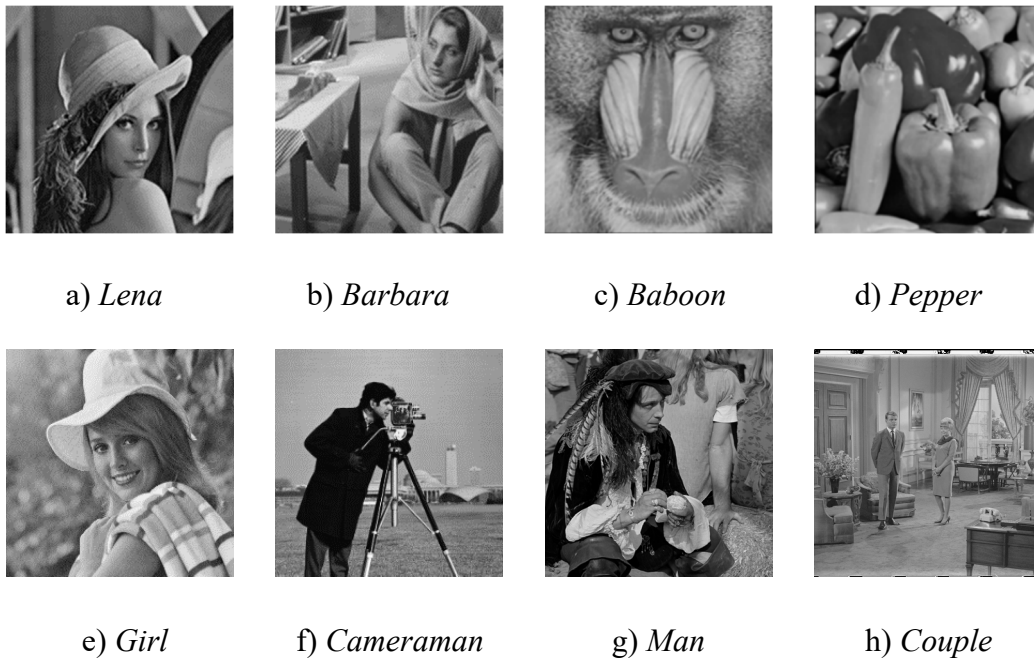


Figure 7. Experimental test images.



Figure 8. Watermark image.

The integrity of the watermark embedding algorithm is generally measured by the NC (Normalized Correlation), whose analytical form is given in Eq (13):

$$NC = \frac{\sum_{i=0}^{L-1} \sum_{j=0}^{K-1} I(i, j)I'(i, j)}{\sum_{i=0}^{L-1} \sum_{j=0}^{K-1} [I(i, j)]^2} \quad (13)$$

where, $I(i, j)$ and $I'(i, j)$ respectively denote the pixel values at (i, j) in the original image and the carrier image of recovery after the watermark is extracted. L and K denote the numbers of rows and columns of the image respectively.

Table 2. Integrity assessment of this algorithm without any attacks.

Image (512 × 512 pixels)	<i>Lena</i>	<i>Barbara</i>	<i>Baboon</i>	<i>Peppers</i>	<i>Girl</i>	<i>Cameraman</i>	<i>Man</i>	<i>Couple</i>
<i>NC</i>	1	1	1	1	1	1	1	1

The above Table 2 indicates the integrity assessment results of the watermarked images without attacks. It indicates that the NC values of these images are the same with the value of 1 when suffering from no attacks. In the case of no attacks, if the NC value is 1, the original image is completely restored. This shows that the carrier image restored by this algorithm has high visual quality.

PSNR (Peak Signal to Noise Ratio) is currently one of the main indicators used to evaluate the visual quality of the reversible watermarking. The PSNR is formally formulated as shown in Eq (14):

$$PSNR = 10 \log \left(\frac{255^2}{\frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [I'(i, j) - I(i, j)]^2} \right) \quad (14)$$

where $I(i, j)$ and $I'(i, j)$ respectively denote the pixel values at position (i, j) in the original image and the watermarked image. Here, M and N represent the numbers of rows and columns of the image respectively.

The SSIM (structural similarity) is utilized to evaluate the quality of a stego image and the mathematical form of the SSIM is stated in Eq (15):

$$SSIM(x, y|w) = \frac{(2\bar{w}_x \bar{w}_y + C_1)(2\sigma_{w_x w_y} + C_2)}{(\bar{w}_x^2 + \bar{w}_y^2 + C_1)(\sigma_{w_x}^2 + \sigma_{w_y}^2 + C_2)} \quad (15)$$

where C_1 and C_2 are small constants, \bar{w}_x is a mean value of the region w_x , while \bar{w}_y represents the average 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. A higher value of the SSIM indicates a higher quality level for the stego image. The numeric value of the SSIM lies in the range of [0, 1]. In the evaluation of the reversible information hiding algorithm, the SSIM value of the hidden image is the value closest to 1.

The values of the PSNR and SSIM are shown in Table 3 by using the algorithm proposed in this paper and those proposed in [7] and [10] after dividing sub-blocks with 4×4 pixels.

After the watermark is embedded into the aforementioned eight images, the PSNR value of the proposed algorithm can reach as high as 54.63 dB, which is better than the algorithms in [17] and [17]. At the same time, compared with the algorithms in [17] and [17], the SSIM is also higher. As reflected from Table 3, it's not hard to see that this algorithm is better than those in [17] and [10] and achieves excellent SSIM and PSNR under the same payload capacity. It means that the proposed algorithm has better visual quality. The details are shown in Table 4.

Table 3. Visual quality analysis of the proposed algorithm.

Image name	Proposed algorithm		literature [17]		literature [17]	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
<i>Lena</i>	54.63	0.989	46.72	0.982	52.18	0.984
<i>Baboon</i>	52.36	0.987	48.37	0.985	51.37	0.988
<i>Barbara</i>	53.51	0.993	47.54	0.983	51.93	0.986
<i>Peppers</i>	53.14	0.991	46.81	0.981	51.73	0.987
<i>Girl</i>	54.27	0.989	47.13	0.986	52.43	0.987
<i>Cameraman</i>	53.69	0.991	46.92	0.982	51.52	0.985
<i>Man</i>	52.73	0.988	47.98	0.984	51.49	0.987
<i>Couple</i>	53.12	0.990	46.85	0.983	52.17	0.986

It can be witnessed from these watermarked images that the naked eye is unable to sense the presence of the watermark. The watermarked images possess good visual effects, and their corresponding PSNR values illustrate the proposed algorithm with good imperceptibility for various image types. The average PSNR value of the above eight images shown in Figure 7 after embedding the watermark is up to 53.43 dB.

It can be seen from Tables 3 and 4, the proposed algorithm has good visual invisibility for different texture types of images. In order to estimate the maximum watermark embedding capacity, the watermark needs to be embedded into all sub-blocks (except for the block containing the abrupt points). In this paper, the algorithm can be embedded many times under the premise of guaranteeing a certain visual quality. Here, an example of one-time embedding is provided.

As shown in Table 5, 10, 30, 70, 90 and 100% refer to the proportion of the watermark capacity to be embedded with regard to the maximum embedding capacity. The PSNR and SSIM are used to evaluate the visual quality of the watermarked images when embedding 10, 30, 70, 90 and 100% of the maximum embedding capacity. In Table 5, it is obvious that the proposed reversible watermarking technique outperforms the techniques in [17] and [18] with competitive SSIM and PSNR values in terms of payload capacity. The results show that the proposed technology not only guarantees the visual quality of the watermarked images, but also significantly improves the payload capacity.

Table 6 shows the experimental results of the calculated PSNR values, the SSIM values of the original image and the restored image after extracting the watermark, and the SSIM values of the original watermark and the extracted watermark by the three algorithms. It can be seen that the PSNR value of the watermarked image obtained by this algorithm is higher, especially much higher than that of the methods in [19]. Furthermore, the original image can be recovered after extracting the watermark, and the similarity between the extracted watermark and the original watermark is same. Compared with the methods in [19–21], the proposed algorithm has improved significantly.

Table 4. Visual effect of algorithm experiment.


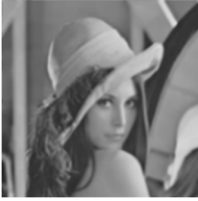
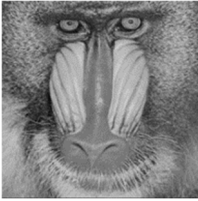
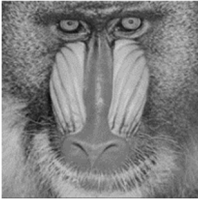












Image name	Original images	Watermarked images	Original watermark	Extracted watermark	PSNR
<i>Lena</i>			武	武	54.63
<i>Baboon</i>			武	武	52.36
<i>Barbara</i>			武	武	53.51
<i>Peppers</i>			武	武	53.14
<i>Girl</i>			武	武	54.27
<i>Cameraman</i>			武	武	53.69
<i>Man</i>			武	武	52.73
<i>Couple</i>			武	武	53.12

Table 5. Performance comparison of watermarking algorithms.

Image name	Algorithms	Embedding capacity (bit)	SSIM	PSNR				
				10%	30%	70%	90%	100%
<i>Lena</i>	Literature 18	276,599	0.7331	54.37	42.92	36.83	32.91	31.95
	Literature 17	9767	0.9188	45.78	43.48	41.58	39.78	38.64
	Proposed	236,432	0.7935	48.65	45.24	38.91	37.05	36.34
<i>Baboon</i>	Literature 18	41,519	0.8232	51.25	42.43	34.94	33.48	31.02
	Literature 17	11,056	0.9047	45.86	43.84	41.95	40.56	39.76
	Proposed	224,512	0.8217	47.44	44.65	38.77	37.21	36.42
<i>Barbara</i>	Literature 18	392,740	0.6724	50.12	48.54	46.80	45.11	31.65
	Literature 17	13,232	0.8931	45.66	42.89	39.54	37.10	36.65
	Proposed	232,864	0.8011	47.33	44.14	38.47	37.28	36.17
<i>Peppers</i>	Literature 18	138,695	0.8154	52.62	42.98	35.12	33.58	31.53
	Literature 17	8562	0.9369	46.24	44.77	41.82	39.54	38.26
	Proposed	232,434	0.8115	48.13	44.35	38.54	36.79	35.84
<i>Girl</i>	Literature 18	293,674	0.7323	53.14	42.17	36.32	32.33	31.35
	Literature 17	14,876	0.9157	45.31	42.86	41.03	39.25	38.12
	Proposed	224,796	0.7938	48.09	44.71	38.20	36.51	35.96
<i>Cameraman</i>	Literature 18	179,643	0.8123	52.32	42.36	34.42	33.97	31.02
	Literature 17	12,392	0.9219	46.03	44.10	41.25	39.05	37.72
	Proposed	236,562	0.8112	47.67	43.82	38.09	36.29	35.33
<i>Man</i>	Literature 18	159,427	0.8032	51.47	41.49	33.35	32.99	30.42
	Literature 17	10,074	0.9182	45.91	43.85	41.07	38.79	37.41
	Proposed	224,862	0.8112	47.67	43.82	38.09	36.29	35.33
<i>Couple</i>	Literature 18	233,912	0.7638	52.38	42.67	34.84	33.37	31.22
	Literature 17	12,128	0.9298	46.11	44.23	41.54	39.25	37.83
	Proposed	235,956	0.8231	47.98	44.09	38.26	36.48	35.53

In this paper, 3 fixed embedding rates of 0.5, 1.0 and 1.2 bpp were selected for comparing the corresponding PSNRs, and the results are shown in Table 7. The “—” in Table 7 indicates that the corresponding experimental measurement cannot be carried out because of the characteristics of the image itself. It can be seen from Table 7, in the case of the same selected embedding rate, when the embedding rate is 0.5 bpp, the average PSNR of the proposed algorithm is 2.67 dB higher than that in 22, which is 1.93 dB higher than that in 23. With the increase of the embedding rate, the average PSNR of the proposed algorithm is 2.6 dB higher than that in 22 when reaching 1.0 bpp, which is 1.6 dB higher than that in 23. When the same embedding rate is 1.2 bpp, the average PSNR of the proposed algorithm is 2.5 dB higher than that in 22, which is 0.7 dB higher than that in literature 23. The main reason for this difference is that the algorithm in 22 uses a reversible image watermarking based on the histogram shift interpolation technique. The algorithm obtains a better performance in the small embedding amount because of the more concentrated prediction error histogram. The host image exhibits a vast proportion of pixels in the lightest or darkest end. Due to the lack of a gray overflow control scheme, the auxiliary information for the host images cannot be completely saved, therefore leading to an embedding failure. In 23, the watermarks were mostly distributed in the local area with strong watermark embedding ability, and the watermarks were rarely distributed in the local area with weak watermark embedding ability. It inevitably produces distortion after iterative processing, thus affecting the image quality. Therefore, the proposed algorithm achieves a better image visual quality and is

suitable for embedding high-capacity watermark information.

Table 6. Performance comparison of watermarking algorithms.

Evaluating indicators	Images	Algorithms			
		Literature 19	Literature 20	Literature 21	Proposed
PSNR (watermarked images)	<i>Lena</i>	46.03	50.60	48.05	54.63
	<i>Baboon</i>	41.22	48.64	49.22	52.36
	<i>Barbara</i>	46.15	47.22	49.01	53.51
	<i>Peppers</i>	42.05	41.31	48.20	53.14
	<i>Girl</i>	45.71	49.63	48.41	54.27
	<i>Cameraman</i>	43.35	43.29	48.77	53.69
	<i>Man</i>	43.28	43.25	48.59	52.73
	<i>Couple</i>	42.11	41.25	48.08	53.12
	SSIM (original image and recovered image after extracting)	<i>Lena</i>	1	0.9975	0.9945
<i>Baboon</i>		0.9999	0.9995	0.9994	1
<i>Barbara</i>		0.9999	0.9995	0.9982	1
<i>Peppers</i>		0.9999	0.9954	0.9964	1
<i>Girl</i>		1	0.9987	0.9954	1
<i>Cameraman</i>		0.9999	0.9969	0.9968	1
<i>Man</i>		0.9999	0.9973	0.9965	1
<i>Couple</i>		0.9999	0.9964	0.9962	1
SSIM (original watermark and extracted watermark)		<i>Lena</i>	1	0.9905	0.9835
	<i>Baboon</i>	1	0.9738	0.9978	1
	<i>Barbara</i>	1	0.9676	0.9845	1
	<i>Peppers</i>	1	0.9655	0.9878	1
	<i>Girl</i>	1	0.9836	0.9803	1
	<i>Cameraman</i>	1	0.9752	0.9845	1
	<i>Man</i>	1	0.9749	0.9842	1
	<i>Couple</i>	1	0.9652	0.9875	1

5. Conclusions

This paper presents an improved large-capacity reversible image watermarking based on the DE. The algorithm can effectively improve the embedding rate and visual quality. The proposed scheme effectively controls the overflow of pixel values when embedding watermark with the DE, and uses multi-scale decomposition technology to remove the abrupt points in each image-block. On the one hand, when embedding watermark, the GDE has higher embedding amount than the DE. On the other hand, the overflow processing method proposed in this paper can improve the embedding capacity without locating the overflow pixels. There is no need to eliminate the overflow pixels which may be generated when using the GDE to embed the watermark. The GDE is used to embed the watermark directly without considering the overflow of pixels. Keeping certain visual quality, the overflow algorithm can carry out multiple times for watermark embedding. After extracting the embedded

watermark, the algorithm can recover the original image without any loss. Compared with other algorithms, the main contribution of this proposed algorithm is to enhance the embedding rate while maintaining promising visual quality. The simulation results showed that the effective watermark capacity embedded in this algorithm was superior to other algorithms under the premise of maintaining good visual quality.

Table 7. PSNR value comparison under the same embedding rate (dB).

Embedding rate	Original images	<i>Lena</i>	<i>Baboon</i>	<i>Barbara</i>	<i>Peppers</i>	<i>Girl</i>	<i>Cameraman</i>	<i>Man</i>	<i>Couple</i>
0.5 bpp	Literature 22	42.5	33.9	42.8	--	--	--	--	--
	Literature 23	42.7	38.3	42.9	--	41.5	41.1	40.9	--
	Proposed	41.5	40.9	41.8	41.4	41.4	41.2	41.1	41.4
1.0 bpp	Literature 22	33.2	--	34.3	--	--	33.7	--	--
	Literature 23	34.7	--	35.2	--	34.9	--	--	--
	Proposed	36.3	36.2	36.4	35.8	36.3	35.9	35.8	35.9
1.2 bpp	Literature 22	--	--	30.9	--	--	--	--	--
	Literature 23	32.1	--	32.8	--	31.8	32.5	32.4	--
	Proposed	33.2	32.8	33.4	32.9	33.2	32.8	32.9	32.8

Acknowledgments

This work is supported by the National Statistical Science Research Project (2018LY12), and Six Talent Peaks Project in Jiangsu Province (XYDXXJS-011).

Conflict of interest

All authors declare no conflicts of interest in this paper.

References

1. Z. Zhang, L. Wu, S. Xiao, S. Gao, Adaptive reversible image watermarking algorithm based on IWT and level set, *EURASIP J. Adv. Signal Process.*, **15** (2017), 1–16. doi: 10.1186/s13634-017-0450-7.
2. J. Tian, Reversible data embedding using a difference expansion, *IEEE Trans. Circuits Syst. Video Tech.*, **8** (2003), 890–896. doi: 10.1109/TCSVT.2003.815962.
3. X. L. Li, J. Li, B. Li, High-fidelity reversible data hiding scheme based on pixel-value ordering and prediction-error expansion, *Signal Process.*, **1** (2012), 198–205. doi: 10.1016/j.sigpro.2012.07.025.
4. J. X. Wang, J. Q. Ni, J. W. Pan, A high capacity reversible data hiding scheme based on generalized prediction-error expansion and adaptive embedding, *Signal Process.*, **5** (2014), 370–380. doi: 10.1016/j.sigpro.2013.12.005.

5. Z. W. Zhang, L. F. Wu, Y. Y. Yan, S. Z. Xiao, An improved reversible image watermarking algorithm based on difference expansion, *Int. J. Distr. Sens. Net.*, **1** (2017), 1–15. doi: 10.1177/1550147716686577.
6. Z. W. Zhang, M. J. Zhang, L. Y. Wang, Reversible image watermarking algorithm based on quadratic difference expansion, *Math. Prob. Eng.*, **1** (2020), 1–8. doi: 10.1155/2020/1806024.
7. H. K. Maity, S. P. Maity, Reversible image watermarking using modified difference expansion, Emerging Applications of Information Technology (EAIT), in *2012 Third International Conference on IEEE*, **3** (2012), 320–323. doi: 10.1109/EAIT.2012.6407936.
8. S. L. Lin, C. F. Huang, M. H. Liou, C. Y. Chen, Improving histogram based reversible information hiding by an optimal weight-based prediction scheme, *J. Inf. Hid. Multimed. Signal Process.*, **1** (2013), 19–33.
9. A. Muhammad, S. Q. Aqsa, K. Asifullah, Protection of medical images and patient related information in healthcare: Using an intelligent and reversible watermarking technique, *Appl. Soft Comput.*, **51** (2017), 168–179. doi: 10.1016/j.asoc.2016.11.044.
10. K. C. Vinoth, V. Natarajan, Hybrid local prediction error based difference expansion reversible watermarking for medical images, *Comput. Electr. Eng.*, **53** (2016), 333–345. doi: 10.1016/j.compeleceng.2015.11.033.
11. Y. J. Jia, Z. X. Yin, X. P. Zhang, Reversible data hiding based on reducing invalid shifting of pixels in histogram shifting, *Signal Process.*, **163** (2019), 238–246. doi: 10.1016/j.sigpro.2019.05.020.
12. M. Ntahobari, T. Ahmad, Protecting data by improving quality of stego image based on enhanced reduced difference expansion, *Int. J. Electr. Comput. Eng.*, **4** (2018), 2468–2476. doi: 10.11591/ijece.v8i4.
13. X. Yu, X. Wang, Q. Q. Pei, Reversible watermarking based on multi-dimensional Prediction-error expansion, *Multimed. Tools Appl.*, **14** (2018), 18085–18104. doi: 10.1007/s11042-018-5794-y.
14. W. G. Su, Y. L. Shen, X. Wang, Two-layer reversible watermarking algorithm using difference expansion, *J. Comput. Res. Develop.*, **7** (2019), 1498–1505.
15. Y. Q. Li, J. X. Li, Y. H. Liang, Image watermarking algorithm based on image characteristics and Huffman coding, *Comput. Appl. Soft.*, **30** (2013), 128–130.
16. N. K. Anil, M. Haribabu, B. C. Hima, Novel image watermarking algorithm with DWT-SVD, *Int. J. Comput. Appl.*, **1** (2014), 12–17.
17. S. E. Hala, S. F. Elzoghdy, S. F. Osama, Adaptive difference expansion-based reversible data hiding scheme for digital images, *Arabian J. Sci. Eng.*, **41** (2016), 1091–1107. doi: 10.1007/s13369-015-1956-7.
18. Z. W. Zhang, L. F. Wu, Y. Y. Yan, An improved reversible image watermarking algorithm based on difference expansion, *Int. J. Distr. Sens. Net.*, **1** (2017), 1–15. doi: 10.1177/1550147716686577.
19. M. Arsalan, S. A. Malik, A. Khan, Intelligent reversible watermarking in integer wavelet domain for medical images, *J. Syst. Soft.*, **4** (2014), 883–894. doi: 10.1016/j.jss.2011.11.005.
20. O. M. Al-Osamah, E. K. Bee, Two-dimensional difference expansion (2D-DE) scheme with a characteristics-based threshold, *Signal Process.*, **1** (2015), 154–162. doi: 10.1016/j.sigpro.2013.12.005.
21. B. Lei, E. L. Tan, S. Chen, N. Dong, T. Wang, H. Lei, Reversible watermarking scheme for medical image based on differential evolution, *Expert Syst. Appl.*, **7** (2016), 3178–3188. doi: 10.1016/j.eswa.2013.11.019

22. K. Swaraja, Medical image region based watermarking for secured telemedicine, *Multimed. Tools Appl.*, **21** (2018), 28249–28280. doi: 10.1007/s11042-018-6020-7.
23. H. A. Zheng, C. A. Wang, J. B. Wang, S. C. Xiang, A new reversible watermarking scheme using the content-adaptive block size for prediction, *Signal Process.*, **164** (2019), 74–83. doi: 10.1016/j.sigpro.2019.05.035.



AIMS Press

©2022 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)