

Image Scrambling based on Spiral Filling of Bits

Haohao Yuan^{1,2} and Lianyuan Jiang^{1,3*}

¹*School of Computer, Guangxi University of Science and Technology, Liuzhou 545006, Guangxi, China*

²*School of Information Engineering, Wuhan University of Technology, Wuhan 430070, Hubei, China*

³*College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, Jiangsu, China*

**jly_jly_jly@163.com*

Abstract

To enhance the security of the digital image information transmitted through Internet, this paper proposes an image scrambling algorithm based on spiral filling of bits. This algorithm first reads the pixel values of all the odd-number lines in the images successively from top to bottom and then does the same to the even-number lines. All results will be saved in a one-dimensional array. All the elements in the array are transformed into binary digits and saved in a one-dimensional array that is 8 times longer than the previous one. In this new array every element only saves one binary digit. The bit sequences corresponding to the image odd-number lines in this array are rearranged in an inverted order. The arrays disposed into an inverted order successively fill a matrix whose line number is the same as the old image and column number is 8 times of the old image in a direction of heliciform. Then a new matrix is generated. Lastly every 8 elements in this new matrix are considered as 8 binary digits and are transformed into decimal digits. That is how image scrambling is conducted. This algorithm is simple and easy to be realized. Abundant experimental results have shown that the algorithm has well scrambling effect and is able to recover image without distortion.

Keywords: *Image scrambling; Spiral filling; Image encryption; Pixel value*

1. Introduction

Since 1980s, the computer science technology develops rapidly. People are nowadays enjoying the convenience brought by the information technology development. Meanwhile they are also facing the crisis of information security. For example, much digital media content is stolen, copied illegally and spread while being uploaded and transmitted on the Internet. Thus the problem of information security attracts many researchers' attentions globally [1, 2]. The encryption of the digital media information is an effective way to protect and image scrambling is an important method to conduct encryption.

Digital image scrambling dates back to the encryption of cable television signals. To date digital television technology develops and is popularized rapidly, making digital image scrambling technology an important subject to be studied. Digital image scrambling is to turn an image into a disorganized and unrecognizable image through scrambling algorithm. For those who know about inverse scrambling algorithm, they can turn the scrambling image back into the original one. But it is impossible for those who do not know how to do it to acquire the original image information [3, 4]. Image scrambling technology as a common method to conduct image encryption can also be used to do pretreatment of image hiding and digital watermarking [5-8].

The common image scrambling algorithm mainly concludes the scrambling of pixel position, pixel gray value and that of both. The image scrambling based on position switching conducts scrambling by intersecting image pixels. Its security relies mainly on the scrambling matrix [3, 5, 9-11]. And the image scrambling based on pixel gray value switching conducts scrambling by changing the gray values of image pixel points. This algorithm possesses a higher security comparatively [12-14]. For example, Lei Zhong-kui *et al.* [9] proposed three improved algorithms to settle the existing problem appeared in the process of traditional knight-tour transformation. The improved scrambling algorithms expanded the key database, strengthening the scrambling effect. Fan Tie-sheng *et al.* [10] proposed an image scrambling algorithm matched by Soma Cube which changes the image pixel gray values by altering the bit planes of original images first and then partitions the altered images according to Soma Cube choosing two random joint methods of the Cube and switching them correspondingly. Thus the image scrambling is realized. He Bing [11] proposed a digital image scrambling algorithm based on chaos theory. The algorithm first ranks the chaos sequences from small to large and then makes a record of all the serial numbers of elements ranked in forms of set sequence, generating a key stream. At the end the key stream generates relevant index values according to which position scrambling is conducted to the pixels of corresponding positions. And exclusive OR calculation is done to the adjacent pixel values to get the final scrambling image. Shao Li-ping *et al.* [14] constructed the snow slide image scrambling switching which conducts the scrambling through inverse transformation to the original images and after that regain the original images through forward transformation.

Nowadays researches of image scrambling based on position switching are in the majority. However the existing algorithm has many disadvantages. And researches of image scrambling based on pixel gray value switching are relatively less. But this algorithm is better than the previous one. This paper studies the pixel gray value switching in the image scrambling and proposes an image scrambling algorithm based on spiral filling of bits. The experimental results have shown that this algorithm generates better scrambling effect.

2. Introduction of Relevant Image Scrambling Algorithm

Image scrambling means the process of disorganizing the texture, color and shape etc. of digital information, turning it into an unrecognizable picture. While using a mathematical matrix to describe an image, the matrix's line number and column number to represent pixel counts of the image's height and width and the matrix's element value to represent the image's pixel color value, then the definition of image scrambling is as follow [15]:

The image is given as $G = [g(i, j)]_{m \times n}$. The transformational matrix $T = [t(i, j)]_{m \times n}$ is a sequence from 1 to $m \times n$ in which m is the line number of the matrix and n is the column number of the matrix. Make G and T correspond to each other in the light of ranks. And then remove the color values of pixels at positions $k(1 \leq k < m \times n)$ in G to corresponding positions $k+1$. At the same time remove the color values of pixels at positions $m \times n$ to corresponding position 1. By doing so a new image N is generated. N is the image changed from G by means of T scrambling.

An empirical example is given below to illustrate this definition:

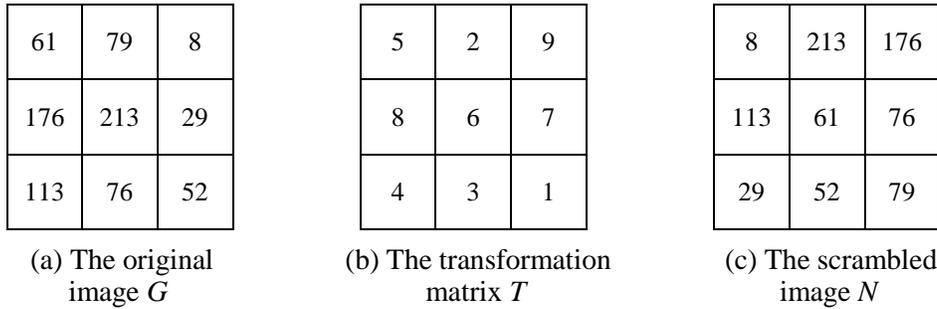


Figure 1. A Particular Example of Image Scrambling

2.1. Image Scrambling based on Position Switching

Image scrambling based on position switching is a method that regards the image as a matrix and conducts realignment to some or all elements' positions in the matrix. This method only changes the pixel positions of image and the pixel values remain unchanged. So the histograms of the original image and the scrambling image are the same, which leads to a low security. Common methods include Arnold transformation, Fibonacci transformation, Hilbert curvilinear transformation, magic square transformation, affine transformation and knight-tour transformation and so on. An introduction of image scrambling method adopting magic square transformation is given below [16]:

Magic square is a very ancient mathematical problem whose definition is: an n -order matrix where the integers in each row, each column and each diagonal line, all add up to the same number M (M is called magic square sum). The mathematical expressions of n -order magic square matrix are equation (1) and (2) in which $i, j \in \{1, 2, 3 \dots n\}$ and n is a random positive integer.

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1(n-2)} & a_{1(n-1)} & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2(n-2)} & a_{2(n-1)} & a_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{(n-2)1} & a_{(n-2)2} & \dots & a_{(n-2)(n-2)} & a_{(n-2)(n-1)} & a_{(n-2)n} \\ a_{(n-1)1} & a_{(n-1)2} & \dots & a_{(n-1)(n-2)} & a_{(n-1)(n-1)} & a_{(n-1)n} \\ a_{n1} & a_{n2} & \dots & a_{n(n-2)} & a_{n(n-1)} & a_{nn} \end{bmatrix} \tag{1}$$

$$M = \sum_{j=1}^n a_{ij} = \sum_{i=1}^n a_{ij} = \sum_{i=1}^n a_{ii} = \sum_{i+j=n+1}^n a_{ij} = \frac{n(n^2 + 1)}{2} \tag{2}$$

2.2. Image Scrambling based on Pixel Gray Value Transformation

Image scrambling based on pixel gray value transformation conducts coding or relevant arithmetical operations to the pixel values of the original image changing the gray values of the image pixels. Since the gray values are changed in this method, the histograms of the scrambling image and the original image are then different. Thus the security is upgraded.

However the arithmetical operations are complicated that result in low efficiency in the transformation. Relatively speaking this method is difficult in today's researches. Common methods include Gray code transformation, the chaotic mapping transformation and exclusive OR transformation and so on. An introduction of image scrambling method based on pixel gray value switching adopting exclusive OR transformation is given below:

Assuming that the size of the image is $m \times n$, a random number sequence S is generated from an original random number. In the sequence the i th element is $s_i (0 \leq i < m \times n)$

If the scrambling process transforms the three color components (r_i, g_i, b_i) of the pixel in the original image into the three color components (r'_i, g'_i, b'_i) of the pixel in the scrambling image through equation (3), then this method is called exclusive OR transformation.

$$\begin{cases} r'_i = r_i \oplus s_i, 0 \leq i < m \times n \\ g'_i = g_i \oplus s_i, 0 \leq i < m \times n \\ b'_i = b_i \oplus s_i, 0 \leq i < m \times n \end{cases} \quad (3)$$

3. The Proposed Algorithm

This paper studies image scrambling based on position switching focusing on the positions of pixels and further proposes an image scrambling algorithm based on spiral filling of bits.

3.1. The Specific Process of Scrambling

The scrambling process of this algorithm is divided into 5 steps, and this paper explains the steps taking gray image as example. The disposal of colorful image is the same as the one of gray image.

3.1.1. Reading of Image Data: Assuming that the size of the original image is $m \times n$ (unit: pixel, the same below), and A is a one-dimensional integer number array whose length is $m \times n$. The way to transform the pixel values of the original image to the array A successively is: 1), first storage the pixel values of all the odd-numbered rows in the image from top to bottom successively in the array A ; for pixels in the same row, read them from left to right; 2), and then storage the pixel values of all the even-numbered rows in the image from top to bottom successively in the array A ; for pixels in the same row, read them from left to right as well.

3.1.2. Conversion to Binary System: Since the range of every element value in A is $[0, 255]$, which is the numerical range of one byte, every element value in A can be transformed into 8 binary numbers. B is a one-dimensional integer number array with length being $m \times n \times 8$. The way to transform into binary system is: turn every element into 8 binary numbers, thus there are $m \times n \times 8$ unit digits. Then save these digits successively in the array B .

3.1.3. Reverse Array: This step is to arrange the bit sequences in B that correspond to all the odd-numbered rows in the original image in a reverse order. For example, if the bit sequences in B that correspond to an odd-numbered row in the original image are: 01110111 10010101 00010110 ... 11010110 01110011 11110001; then the results after reverse arrangement are 10001111 11001110 01101011 ... 01101000 10101001 11101110.

3.1.4. Spiral Filling: C is a two-dimensional integer number array that has m rows and $n \times 8$ columns. Fill in sequence the two-dimensional number array C with elements in B in the arrow direction in picture 2. S is the starting point of the filling while E is the terminal point.

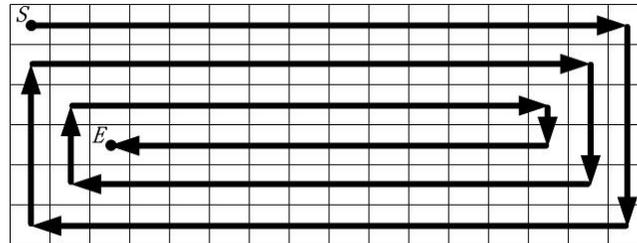


Figure 2. The Spiral Filling

Make $n_8 = n \times 8$ then the implementation code of spiral filling is as follow:

```
void Spiral(int m,int n_8,int **C)
{
    int k,t=0;
    int max=m>n_8?m:n_8;
    int min=m>n_8?n_8:m;
    for(int i=0;i <min/2;i++)
    {
        for(k=i;k<n_8-i-1;k++)
            C[i][k]=B[t++];
        for(k=i;k<m-i-1;k++)
            C[k][n_8-1-i]=B[t++];
        for(k=n_8-i-1;k>i;k--)
            C[m-i-1][k]=B[t++];
        for(k=m-i-1;k>i;k--)
            C[k][i]=B[t++];
    }
    if(min%2==1)
    {
        if (m==n_8)
            C[m/2][m/2]=B[t];
        else if(m>n_8)
        {
            for(int k=n_8/2;k<n_8/2+m-(n_8/2)*2;k++)
                C[k][n_8/2]=B[t++];
        }
        else
        {
            for(int k=m/2;k<m/2+n_8-(m/2)*2;k++)
                C[m/2][k]=B[t++];
        }
    }
}
```

3.1.5. Transformation to Decimal System: Take every 8 adjacent elements in C as 8 binary digits and transform them into a decimal integer number. The range of the integer number is $[0, 255]$ and there are totally $m \times n$ decimal integer numbers. Then save these $m \times n$ decimal numbers in sequence to the corresponding positions in the original image. A scrambling image is generated.

3.2. Scrambling Algorithm

For an image of random size, the scrambling algorithm proposed in this paper is as follow:

- (1) Using method introduced in 3.1.1, load the data of the original image into number array A .
- (2) Using method introduced in 3.1.2, transform all the element values in A into binary digits and save them in B .
- (3) Using method introduced in 3.1.3, arrange the bit sequences in B that correspond to all the odd-numbered rows of the original image in a reverse order.
- (4) Using method introduced in 3.1.4, fill in sequence the two-dimensional number array C by using elements in B in the heliciform direction.
- (5) Using method introduced in 3.1.5, take every 8 adjacent elements in C as 8 binary digits and transform them into a decimal integer number and then save them in order into all the corresponding positions of the original image. The scrambling image is therefore generated.

3.3. Inverse Scrambling Algorithm

Inverse scrambling algorithm is similar with the scrambling algorithm which can be described as follow:

- (1) Conduct the inverse operation of the method in 3.1.5, a two-dimensional number array C is generated.
- (2) Then conduct the inverse operation of the method in 3.1.4 to C , the one-dimensional number array B is generated.
- (3) Conduct the inverse operation of the method in 3.1.3 to B .
- (4) Conduct the inverse operation of the method in 3.1.2 to B , get A .
- (5) Conduct the inverse operation of the method in 3.1.1 to A to recover the image (original image).

4. Experimental Results

In order to verify the scrambling effect of the proposed algorithm, tests are conducted to a large number of images. From the histograms, it can be seen that this algorithm has terrific scrambling effect. And it can also recover the images without distortion. Due to the limited length of writing, here only 4 tests of gray image are offered, including three standard test images and one tree image shoot.

Test 1: the original image is a Lena gray image with size being 512×512 which is demonstrated in picture 3(a). By using the inverse scrambling algorithm in this paper, recover picture 3(b) to picture 3(c). Pictures 3(d), 3(e) and 3(f) are the histograms of pictures 3(a), 3(b) and 3(c). From two obtained pictures in tests and three histograms it can be seen that the pixel values in scrambling image picture 3(b) distribute homogeneously and disorderly. Information of picture 3(a) is completely unrecognizable. The recovered picture 3(c) is exactly the same as picture 3(a).

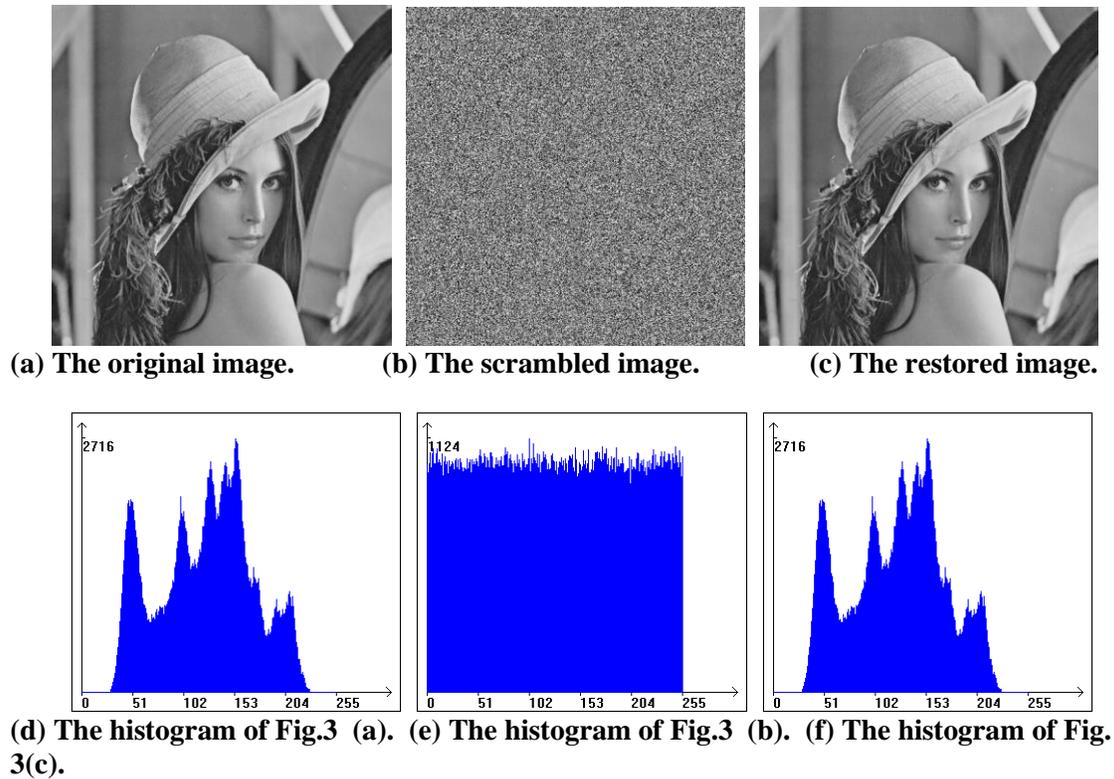
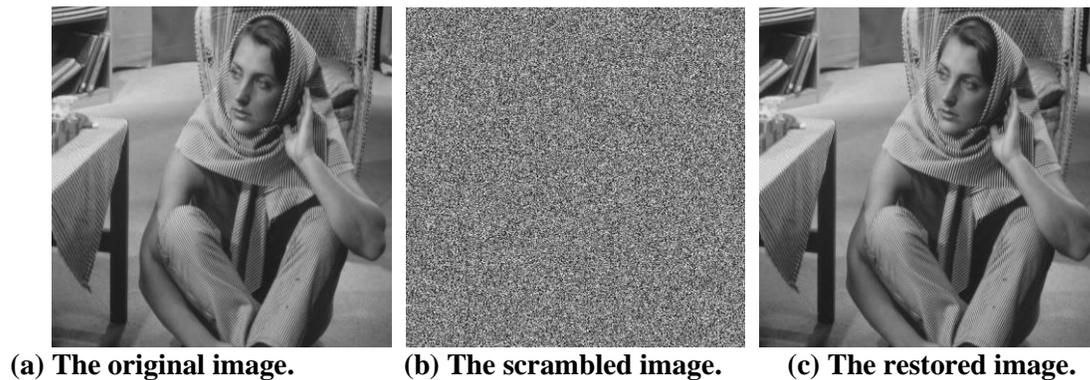
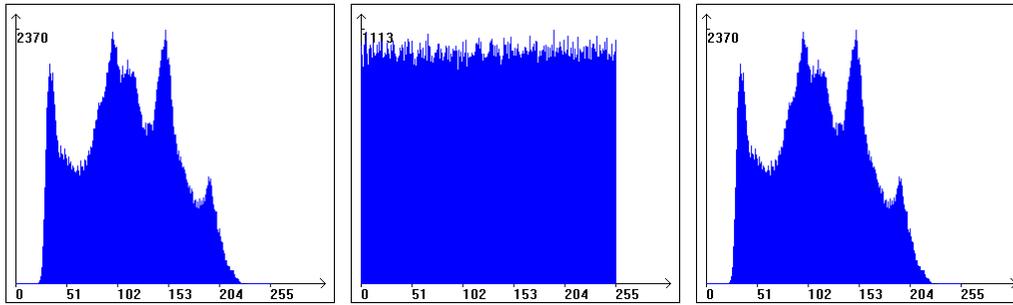


Figure 3. The Experiment on the Lena Image

Test 2: the original picture is a Barbara gray image with size being 512×512 as is shown in picture 4(a). By using the scrambling algorithm introduced in this paper, conduct scrambling process to picture 4(a), get scrambling image as is shown in picture 4(b). By using the inverse scrambling algorithm in this paper, recover picture 4(b) to picture 4(c). Pictures 4(d), 4(e) and 4(f) are the histograms of pictures 4(a), 4(b) and 4(c). Similarly, it is easy to see that information of picture 4(a) is totally unrecognizable in the scrambling image picture 4(b). The recovered picture 4(c) is exactly the same as picture 4(a).

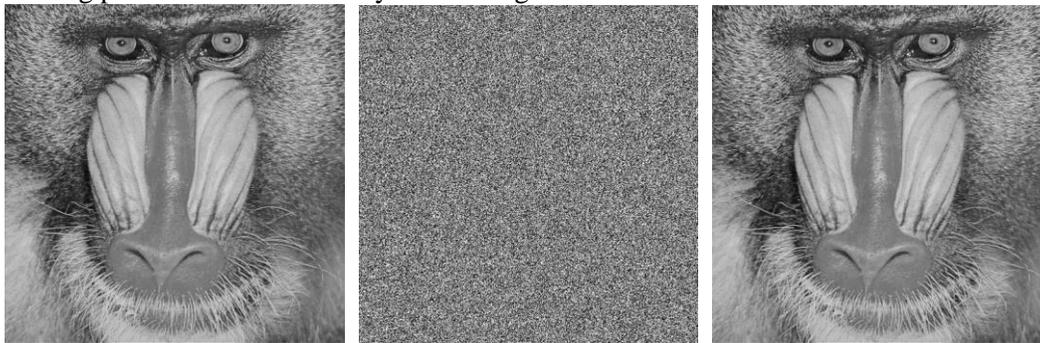




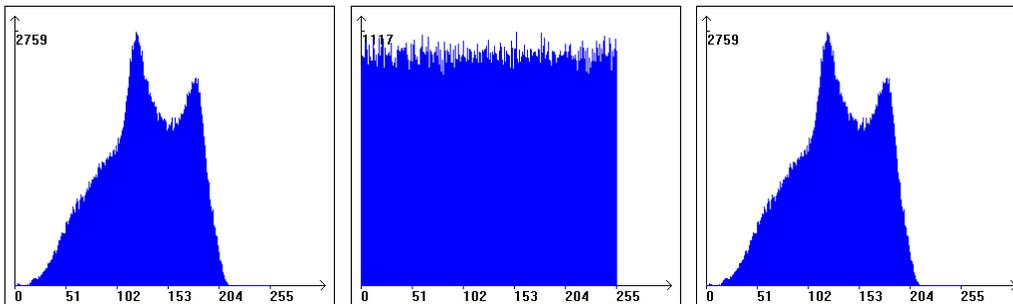
(d) The histogram of Fig.4 (a). (e) The histogram of Fig.4 (b). (f) The histogram of Fig. 4(c).

Figure 4. The Experiment on the Barbara Image

Test 3: the original picture is a Baboon gray image with size being 512×512 as is shown in picture 5(a). By using the scrambling algorithm introduced in this paper, conduct scrambling process to picture 5(a), get scrambling image as is shown in picture 5(b). By using the inverse scrambling algorithm in this paper, recover picture 5(b) to picture 5(c). Pictures 5(d), 5(e) and 5(f) are the histograms of pictures 5(a), 5(b) and 5(c). The results have shown that the scrambling process and the recovery of the image had ideal effects.



(a) The original image. (b) The scrambled image. (c) The restored image.

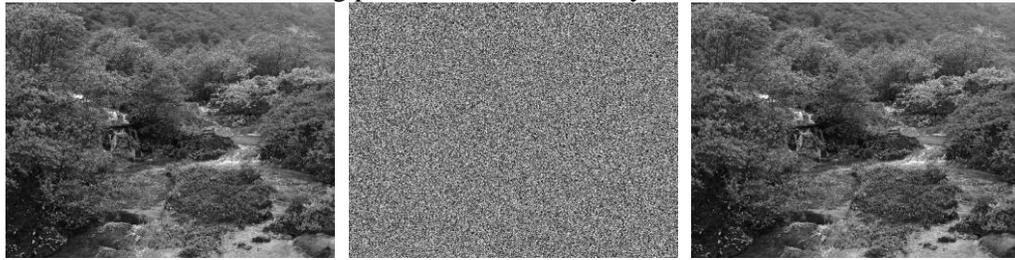


(d) The histogram of Fig.5 (a). (e) The histogram of Fig.5 (b). (f) The histogram of Fig. 5(c).

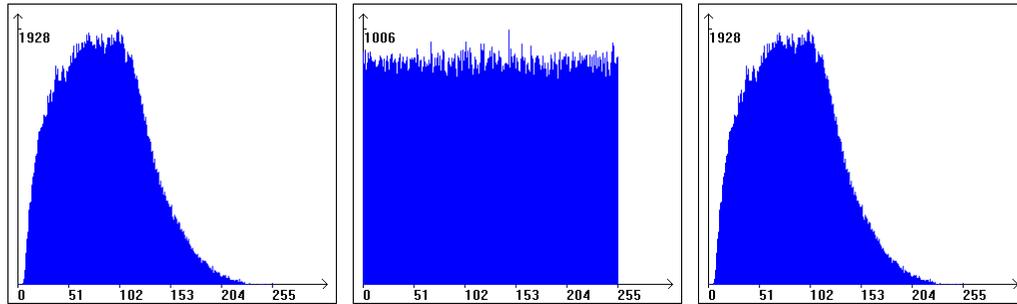
Figure 5. The Experiment on the Baboon Image

Test 4: the original picture is a tree gray image shoot with size being 540×420 as is shown in picture 6(a). By using the scrambling algorithm introduced in this paper, conduct scrambling process to picture 6(a), get scrambling image as is shown in picture 6(b). By using the inverse scrambling algorithm in this paper, recover picture 6(b) to picture 6(c). Pictures

6(d), 6(e) and 6(f) are the histograms of pictures 6(a), 6(b) and 6(c). The results have shown that the effects of both scrambling process and the recovery are fine.



(a) The original image. (b) The scrambled image. (c) The restored image.



(d) The histogram of Fig.6 (a). (e) The histogram of Fig.6 (b). (f) The histogram of Fig.6 (c).

Figure 6. The Experiment on the Tree Image

5. Discussion and Conclusion

This paper proposed an image scrambling algorithm based on spiral filling of bits. This algorithm conducts scrambling process to images in a good way and is able to recover the scrambling images without distortion. From the histograms in the tests it can be seen that the pixel values of scrambling image have a homogeneous distribution and the pixel value distribution of the recovered image is the same as the one in the original image. The algorithm proposed is an image scrambling based on pixel position switching. The algorithm is simple efficient and easy to be realized. Meanwhile it generates good scrambling effects. However, relatively speaking, the security is low. To enhance the security of this algorithm, the algorithm proposed in this paper could possibly be combined with the existing image scrambling method based on pixel gray value transformation. That might be our research content in the next step.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (61462008), the Natural Science Foundation of Guangxi (2013GXNSFAA019336, 2013GXNSFBA019268), the Program of Liuzhou Science and Technology Bureau (2014010401), and the Science Research Program of Guangxi Education Department (LX2014188, LX2014189, and YB2014208). The authors would like to thank the anonymous referees for their valuable comments and suggestions.

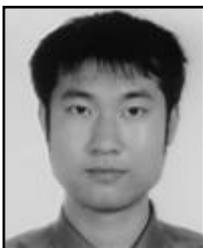
References

- [1] Z. Wang, S. Li, S. Su, M. Liu and Q. Cai, "A quickly-speed running watermarking algorithm based on quantum evolutionary algorithm", *Journal of Optoelectronics, Laser*, vol. 21, no. 5, (2010), pp. 737-742.
- [2] L. Zhao, A. Adhikari, D. Xiao and K. Sakurai, "On the security analysis of an image scrambling encryption of pixel bit and its improved scheme based on self-correlation encryption", *Communications in Nonlinear Science and Numerical Simulation*, vol. 17, no. 8, (2012), pp. 3303-3327.
- [3] L. Shao, Z. Qin, B. Liu, H. Gao and J. Qin, "2D bi-scale rectangular mapping and its application in image scrambling", *Journal of Computer-Aided Design & Computer Graphics*, vol. 21, no. 7, (2009), pp. 1025-1034.
- [4] G. Ye, "Image scrambling encryption algorithm of pixel bit based on chaos map", *Pattern Recognition Letters*, vol. 31, no. 5, (2010), pp. 347-354.
- [5] Z. Tang, X. Lu, W. Wei and S. Wang, "Image scrambling based on bit shuffling of pixels", *Journal of Optoelectronics, Laser*, vol. 18, no. 12, (2007), pp. 1486-1488, 1495.
- [6] Z. Zhong, J. Chang, M. Shan and B. Hao, "Double image encryption using double pixel scrambling and random phase encoding", *Optics Communications*, vol. 285, no. 5, (2012), pp. 584-588.
- [7] S. Liu and J. T. Sheridan, "Optical encryption by combining image scrambling techniques in fractional Fourier domains", *Optics Communications*, vol. 287, (2013), pp. 73-80.
- [8] L. Jiang, H. Yuan, J. Jiang, Y. Zhang and J. He, "Image scrambling algorithm based on grouping calculation", *International Journal of Security and Its Applications*, vol. 8, no. 3, (2014), pp. 209-220.
- [9] Z. Lei, Q. Sun and X. Ning, "Image scrambling algorithms based on knight-tour transform and its applications", *Journal of Chinese Computer Systems*, vol. 31, no. 5, (2010), pp. 984-989.
- [10] T. Fan, Z. Zhang and P. Zhang, "Image scrambling algorithm based on SOMA CUBE square matching", *Computer Science*, vol. 40, no. 6, (2013), pp. 308-310, 314.
- [11] B. He, "A New Digital Image Scrambling Algorithm Basic on Chaos Theory", *Journal of Weinan Normal University*, vol. 29, no. 7, (2014), pp. 16-19.
- [12] D. Qi, J. Zou and X. Han, "A new class of scrambling transformation and its application in the image information covering", *Science in China (Series E)*, vol. 43, no. 3, (2000), pp. 304-412.
- [13] L. Shao, Z. Qin, X. Heng and H. Gao, "Solution for the Inverse Problem of Matrix Transform Based Image Scrambling", *Acta Electronica Sinica*, vol. 36, no. 7, (2008), pp. 1355-1363.
- [14] L. Shao, Z. Qin, X. Heng, H. Gao and X. Wang, "Avalanche image scrambling transformation based on high-dimension matrix transformation", *Journal of Image and Graphics*, vol. 13, no. 8, (2008), pp. 1429-1436.
- [15] Y. Li, "Research on digital image scrambling algorithm", *XiDian University*, (2011).
- [16] J. Li, "Optical image encryption based on scrambling technique and fractional fourier transform", *Jilin University*, (2014).

Authors



Haohao Yuan, she received the M.S. degree in precision instruments and machinery from North University of China, China in 2007. Currently, she is a lecturer at Guangxi University of Science and Technology, China. She is working toward the Ph.D. degree in the School of Information Engineering, Wuhan University of Technology, China. Her research interests include signal processing and wireless sensor networks. E-mail address: yuanhao_1027@163.com.



Lianyuan Jiang, he received the M.S. degree in computer application technology from Guangxi Normal University, China in 2007. Currently, he is an associate professor at Guangxi University of Science and Technology, China. He is working toward the Ph.D. degree in the College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, China. His research interests include pattern recognition and image processing. He is the corresponding author. E-mail address: jly_jly_jly@163.com.