

<http://ansinet.com/itj>

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSI*net*

Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

2D Triangular Mappings and Their Applications in Scrambling Rectangle Image

Li-Ping Shao, Zheng Qin, Hong-Jiang Gao and Xing-Chen Heng

Research Institute of Electronic Commerce,

School of Electronics and Information Engineering,

Xi'an Jiaotong University, Xi'an 710049, Shaanxi, People's Republic of China

Abstract: The conventional 2D matrix transform represented by 2D Arnold transform and 2D Fibonacci-Q transform is applied widely in the security of image information, because of its easily selected scrambling variables and its abilities in enduring erasing, cropping and compression attacks. However, this scrambling method can only be used to scramble square image. For any rectangle image with its width not equal to its height, it needs to be expanded into square image or divided into several square images before using 2D matrix transform to scramble it, which adds the extra space or increases the computation cost. To address this problem, this study proposes two kinds of 2D matrix transforms called 2D triangular mappings and also gives their corresponding inverse transforms. The proposed mappings can be used to scramble or recover rectangle image directly and their iterative cost is low. The cost to scramble or recover image for one time is only the numbers of pixels and our proposed mappings need not to compute the iterative period in advance. Experiments show the proposed mappings validity in scrambling rectangle image, low cost in scrambling and recovering rectangle image and robustness in enduring erasing, cropping and compressing attacks.

Key words: Image mapping, inverse mapping, rectangle image, image scrambling transform, Arnold transform, Fibonacci-Q transform

INTRODUCTION

With the development of computer science and internet technology, more and more digital image information is transmitted over the network and involved in our daily life and study. At the same time, due to the open nature of internet, the chances of image information being illegal intercepted, modified and ravaged are also being improved, which makes the security of digital image information in open network more important.

The security of image information is rarely been studied by traditional cryptology perhaps because image stores huge information. However, in recent years, with the rapid development of computer technology and especially digital image processing technology, people have obtained some useful results and the security of image is becoming an active research topic involving with mathematics, cryptology and information techniques. The proposed techniques include image watermarking, visual cryptology, information sharing, image scrambling etc. Among them, the scrambling technique is one of the basic means for covering huge image information, which can be used in pre-process or post-process of digital image processing, information hiding, digital watermarking, etc. The main purpose of image scrambling transform is to

generate a chaos image, which prevents human visual or computer vision system from understanding the true meaning of the original image and the scrambled image can be recovered if the operator knows the scrambling method and scrambling variables.

At present, many kinds of different scrambling methods have been proposed, such as methods based on Fibonacci transform (Zou *et al.*, 2004a, 2004b, 2006), Lucas transform (Zou *et al.*, 2004b), 2D matrix transform (Zhu and Chen, 2005; Ma and Qiu, 2003; Zhang *et al.*, 2006; Wang, 2006), 3D matrix transform (Qi *et al.*, 2000; Hong and Zou, 2005), high dimensional matrix transform (Qi *et al.*, 2000), Hilbert curve (Lin and Cai, 2004), John Conway's game (Dai *et al.*, 2004; Ding *et al.*, 2000), knight's tour (Hou *et al.*, 2004), baker mapping (Han *et al.*, 2006), magic square (Wang, 2005; Wang and Jin, 2005), magic cube (Shen *et al.*, 2005; Zhang *et al.*, 2005), etc. The conventional 2D matrix transform, represented by 2D Arnold transform (Qi *et al.*, 2000; Kong and Dan, 2004; Yang *et al.*, 2006) and 2D Fibonacci-Q transform (Qi *et al.*, 2000), is applied widely in the security of image information, because of its easily selected scrambling variables and its abilities in enduring erasing, cropping and compression attacks. However, this method has no abilities to scramble rectangle image. For any

rectangle image where its width is not equal to its height, it needs to be expanded into square image (Kong and Dan, 2004; Wang, 2006) or divided into several square images before using 2D matrix transform to scramble it, which adds the extra space and increases the computation cost (Yang *et al.*, 2006).

To address this problem, in this study, we propose two kinds of 2D matrix transforms called 2D triangular mappings and give their corresponding inverse mappings. The proposed mappings can be used to scramble or recover rectangle image directly and their iterative cost is low. The cost to scramble or recover image for one time is only the numbers of pixels and in our proposed mappings, there is no need to compute the iterative period in advance because of the existed corresponding inverse mappings. Experiments show our methods validity in scrambling rectangle image, low cost in scrambling and recovering rectangle image and robustness in enduring erasing, cropping and compressing attacks.

RELATED WORK ABOUT 2D MATRIX TRANSFORM

2D matrix transform came from continuous 2D Arnold transform, which was originated by Arnold and Avez (1968) in their research of ergodic problems of classical mechanics. Because 2D continuous Arnold transform was usually shown by a cat image, it is also called the cat mapping. In this study, we only care about its discrete form.

Definition 1: Discrete 2D Arnold transform, 2D Arnold transform for short: For a given square matrix with m rows and m columns ($m \times m$ for short), let (x, y) denote the coordinate of pixel in row x and column y . The following is the 2D Arnold transform:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \bmod m \quad (1)$$

Qi *et al.* (2000) gave another 2D transform called 2D Fibonacci-Q transform, which replaced 2D Arnold transform matrix with 2D Fibonacci-Q transform matrix.

Definition 2: 2D Fibonacci-Q transform: For a given $m \times m$ square matrix, let (x, y) denote the pixel coordinate. The following is the 2D Fibonacci-Q transform:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \bmod m \quad (2)$$

Qi *et al.* (2000) also expanded the transform matrix into 2D matrix transform.

Definition 3: 2D matrix transform: For a given $m \times m$ square matrix, let (x, y) denote the pixel coordinate. The following is the 2D matrix transform:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \bmod m \quad (3)$$

Where, a, b, c, d are non negative integers and the determinant of transform matrix $ad-bc$ is prime with m .

A picture can be regard as a matrix of pixels. By stretching and folding operation to rearrange the positions of pixels in the image matrix, 2D matrix transform can be used to scramble image and the shifted pixels determine 2D matrix transform has abilities in enduring erasing, cropping and compressing attacks. However, the matrices in definition 1~3 are limited to $m \times m$ dimensions. So conventional image scrambling methods based on 2D matrix transform can only be used to scramble square image. To expand (3) from $m \times m$ to $m \times n$ simply according to Eq. 4, can not guarantee the expanded 2D matrix transform can be used to scramble rectangle image directly.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \bmod \begin{bmatrix} m \\ n \end{bmatrix} \quad (4)$$

To prove this, we use 2D Arnold transform matrix according to Eq. 4 to scramble 176×260 water-lilies test image for one time and the experimental result is shown in Fig. 1.

In Fig. 1a, the original image is stretched by 2D Arnold transform matrix and cut into six parts by modular arithmetic in Eq. 4 (Fig. 1b). The six parts of the stretched image are folded where part 1 overlaps part 4 and part 6 and part 2 intersects part 5 (Fig. 1c). The experiment in Fig. 1 shows to expand 2D matrix transform simply according to Eq. 4 can not guarantee the scrambled image can be recovered.

To address the problem above, there are two conventional scrambling methods applying 2D matrix transform into rectangle image: one is based on expanding rectangle image into square image (Kong and Dan, 2004; Wang, 2006) and the other is based on dividing rectangle image into square images (Yang *et al.*, 2006).

In the first one, the original $m \times n$ image matrix is expanded into $\max(m, n) \times \max(m, n)$ dimensions first and then the expanded image is scrambled by 2D matrix transform, so the one time iterative cost is $O(\max(m^2, n^2))$, which adds the extra space and increases the computation cost. To prove this, we use 2D Arnold transform to scramble the expanded 260×260 water-lilies image for one time and the experiment result is shown in Fig. 2.

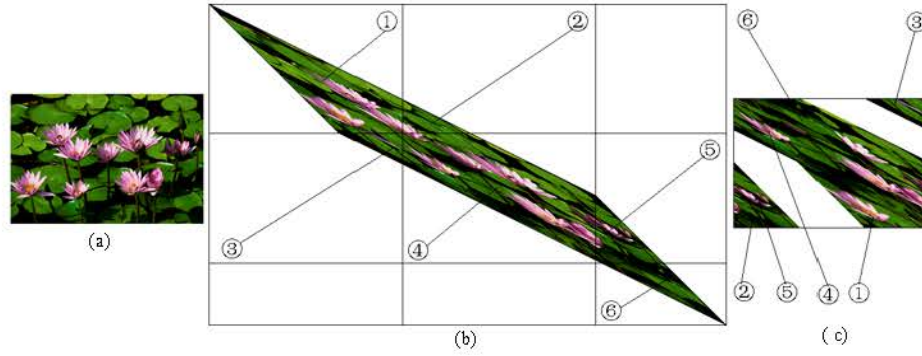


Fig. 1: Scrambling water-lilies image by the expanded 2D Arnold transform, (a) 176×260 water-lilies image, (b) the stretched water-lilies image by 2D Arnold transform matrix and (c) the folded water-lilies image

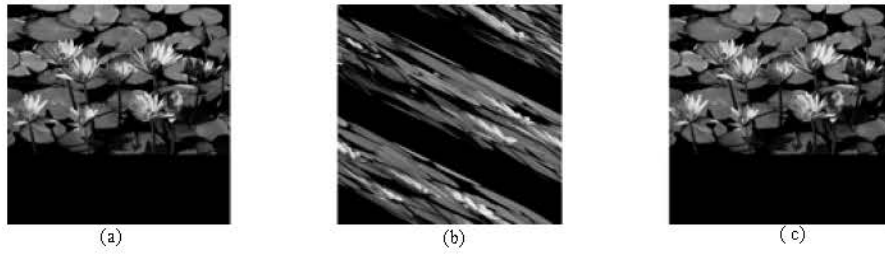


Fig. 2: Scrambling the expanded 260×260 water-lilies image by 2D Arnold transform, (a) the expanded 260×260 water-lilies image, (b) the one time scrambled image by 2D Arnold transform and (c) the recovered image after obverse iterations by 2D Arnold transform for 209 times where the period is 210

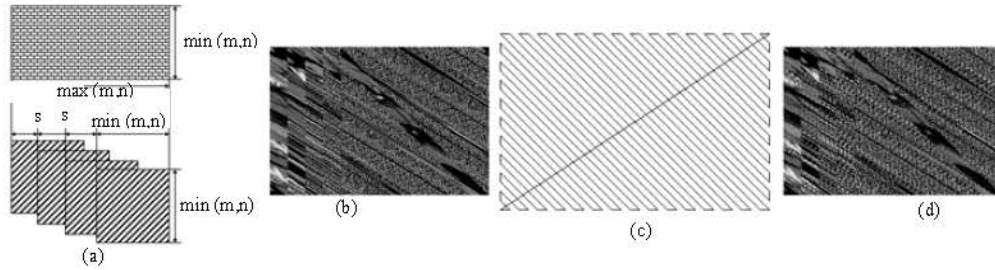


Fig. 3: Scrambling 176×260 water-lilies by dividing rectangle image into squares, (a) changing the original rectangle image into squares, (b) scrambling the squares, respectively by 2D Arnold transform, (c) 20×30 reorder curve and (d) the reordered scrambled image

In Fig. 2, the original 176×260 water-lilies image (Fig. 1a) is expanded into 260×260 water-lilies image (Fig. 2a) and scrambled by 2D Arnold transform for one time (Fig. 2b). Figure 2c is the recovered image.

In the second one, the original $m \times n$ image matrix is divided into several square images and then these images are scrambled, respectively by 2D Arnold transform. To reduce the stripe phenomenon of the scrambled image, this method added the reorder process (Yang *et al.*, 2006). The iterative cost for one time in the second one is

$O(((\max(m,n) - \min(m,n))/s + 1) \times \min(m,n) + m \times n)$ (Fig. 3a), where, s is the offset between every two square images and $s \in \{1, 2, \dots, \min(m,n)\}$. Although the second one need not add the extra space, it really increases the computation cost and operation complexity. To prove this, we divided the 176×260 water-lilies image into five square images where, s is equal to 21 and scrambled them for one time by 2D Arnold transform, respectively. The experiment result is shown in Fig. 3b and 3d and Fig. 3c is the process to reduce the stripe phenomenon.

In Fig. 3c, for clear purpose, we use 20×30 reorder curve to show the reorder process. In our experiment, we use 176×260 reorder curve to reorder the scrambled image. Although the reorder process is added, the strip phenomenon in Fig. 3d still exists.

Form the experiments shown in Fig. 2 and 3, we can see both of the proposed two methods are not good ideas to scramble rectangle image and we need to find another method to solve the problem.

2D TRIANGULAR MAPPINGS

In this section, we give two kinds of 2D triangular mappings that can be used to scramble rectangle image directly. Our scrambling methods are different from the two conventional methods.

Definition 4: 2D lower triangular mapping: For a given $m \times n$ matrix, let (x, y) denote the pixel coordinate. The following is the 2D lower triangular mapping:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & 0 \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \bmod \begin{bmatrix} m \\ n \end{bmatrix} \quad (5)$$

Where, a, c, d are non negative integers; a is prime with m and d is prime with n .

Definition 5: 2D upper triangular mapping: For a given $m \times n$ matrix, let (x, y) denote the pixel coordinate. The following is the 2D upper triangular mapping:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ 0 & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \bmod \begin{bmatrix} m \\ n \end{bmatrix} \quad (6)$$

Where, a, b, d are non negative integers; a is prime with m ; d is prime with n .

The basic idea of 2D triangular mappings is to select a proper transform matrix to permutated every element in the $m \times n$ matrix, in other words, to map all elements with different coordinates into different positions. To ensure it, we give lemma 1 and theorem 1.

Lemma 1: Sequence $\{e_i\}, i' \in [0, m-1]$ is a permutation of sequence $\{e_i\}, i \in [0, m-1]$, if it satisfies (7):

$$e_{i'} \leftarrow e_i, i' = (ai + b) \bmod m \quad (7)$$

Where, a, b are non negative integers and a is prime with m

Proof: Give two different integers $i_1, i_2 \in [0, m-1]$. If we prove $(ai_1+b) \bmod m \neq (ai_2+b) \bmod m$, then we can prove lemma 1. Suppose $(ai_1+b) \bmod m = (ai_2+b) \bmod m$, then $ai_1 -$

$ai_2) \bmod m = 0$. Because a and m are relative primes, we can get $(i_1 - i_2) \bmod m = 0$. From $i_1, i_2 \in [0, m-1]$, we can only get $i_1 = i_2$, which is contrary to $i_1 \neq i_2$. So lemma 1 is proved.

Theorem 1: Elements with different coordinates in the $m \times n$ matrix are mapped to different positions by 2D triangular mapping.

Proof: Give two different coordinates (x_1, y_1) and (x_2, y_2) , if we prove the mapped coordinates $(x'_1, y'_1) \neq (x'_2, y'_2)$ according to definition 4 and 5, then we can prove theorem 1. Suppose $(x'_1, y'_1) = (x'_2, y'_2)$, according to definition 4, we can get $(ax_1) \bmod m = (ax_2) \bmod m$ and $(cx_1 + dy_1) \bmod n = (cx_2 + dy_2) \bmod n$. Because a is prime with m , from lemma 1, if $x'_1 = x'_2$, only $x_1 = x_2$. When $x_1 = x_2$ and d is prime with n , from lemma 1, if $y'_1 = y'_2$, only $y_1 = y_2$. Because $x_1 = x_2$ and $y_1 = y_2$, $(x_1, y_1) = (x_2, y_2)$ which is contrary to two different coordinates (x_1, y_1) and (x_2, y_2) . So 2D lower triangular mapping can be used to map different coordinate to different positions. Similarly, we can prove 2D upper triangular mapping can also be used to map different coordinate to different positions. So theorem 1 is proved.

From theorem 1, we can see the original matrix can be permuted by our proposed 2D triangular mappings. So the proposed mappings can be used to scramble rectangle image directly. From definition 4 and 5, we can also see: using our proposed mappings to map every element in the $m \times n$ matrix to a new position, the one time iterative cost is only $O(m \times n)$. So the proposed mappings one-time iterative cost is low. Because the proposed mappings are still based on permutating pixel coordinates, the proposed mappings can endure erasing, cropping and compressing attacks.

INVERSE MAPPINGS OF THE PROPOSED 2D TRIANGULAR MAPPINGS

In conventional scrambling methods, the iterative period takes an important role in recovering the scrambled image (Wang, 2006). Using the iterative period to recover the original image is usually a time-consuming work because the period is usually from hundreds to thousands or millions times or more and it need to compute the exact iterative period in advance. But, in our proposed 2D triangular mappings, we can also give their inverse mappings to avoid the expensive iterative cost. Therefore, in our mappings, it is no need to compute the iterative period.

Definition 6: Inverse 2D lower triangular mapping: For a given $m \times n$ matrix, let (x, y) denote the pixel coordinate. The following is the inverse 2D lower triangular mapping:

$$\begin{cases} x = (a^{-1}x') \bmod m \\ y = (d^{-1}(y' + \lceil cm/n \rceil \cdot n - cx)) \bmod n \end{cases} \quad (8)$$

Where, $(a^{-1}a) \bmod m = 1$ and $(d^{-1}d) \bmod n = 1$.

Definition 7: Inverse 2D upper triangular mapping: For a given $m \times n$ matrix, let (x, y) denote pixel coordinate. The following is the inverse 2D upper triangular mapping:

$$\begin{cases} y = (d^{-1}y') \bmod n \\ x = (a^{-1}(x' + \lceil bn/m \rceil \cdot m - by)) \bmod m \end{cases} \quad (9)$$

Where, $(a^{-1}a) \bmod m = 1$ and $(d^{-1}d) \bmod n = 1$.

To ensure it, we give theorem 2 and corollary 1.

Theorem 2: If we use 2D triangular mapping to scramble the given matrix for one time, the scrambled matrix can be recovered by the corresponding inverse 2D triangular mapping.

Proof: Given any coordinate of element (x_i, y_i) in the given original matrix, if we prove this coordinate is mapped to the same coordinate (x_i, y_i) by 2D triangular mapping and its corresponding triangular mapping, we can prove theorem 1. From Eq. 5, (x_i, y_i) is mapped to $((ax_i) \bmod m, (cx_i + dy_i) \bmod n)$. In Eq. 8, substitute x' with $(ax_i) \bmod m$, then we can get $(a^{-1}ax_i) \bmod m$. Because $x_i \in [0, m-1]$ and $(a^{-1}a) \bmod m = 1$, $(a^{-1}ax_i) \bmod m = x_i$. In Eq. 8, substitute x with x_i and y' with $(cx_i + dy_i) \bmod n$, then we can get $(d^{-1}((cx_i + dy_i) \bmod n + \lceil cm/n \rceil \cdot n - cx_i)) \bmod n$. By the characteristic of modular arithmetic, this formula can be changed into $(d^{-1}(cx_i + dy_i + \lceil cm/n \rceil \cdot n - cx_i)) \bmod n$. By simplifying, we can get $(d^{-1}dy_i) \bmod n$. Because $y_i \in [0, n-1]$ and $(d^{-1}d) \bmod n = 1$, $(d^{-1}dy_i) \bmod n = y_i$. Because any coordinate of element (x_i, y_i) in the given matrix is mapped to the same coordinate by 2D lower triangular mapping and its corresponding inverse 2D lower triangular mapping, 2D lower triangular mapping can be used to recover the original matrix which is scrambled by the corresponding 2D lower triangular mapping. Similarly, we can prove the inverse 2D upper triangular mapping can be used to recover the original matrix which is scrambled by the corresponding 2D upper triangular mapping. So theorem 2 is proved.

Corollary 1: If we use 2D triangular mapping to scramble the given matrix for several times, the scrambled matrix can be recovered by the corresponding inverse 2D triangular mapping with the same times.

Proof: From theorem 2, if we use 2D triangular mapping to scramble the given matrix for k times where, $k \geq 1$, we can use the corresponding inverse 2D triangular mappings to

return the state of $k-1$ times. Similarly, we can use it to return the state of $k-2$ times until the state of 0 times arriving. So the given matrix is recovered and corollary 1 is proved.

From theorem 2 and corollary 1, we can see, whenever the original image matrix is scrambled for any times by our proposed 2D triangular mapping, it can be recovered by its corresponding inverse matrix. From definition 6 and 7, in our inverse triangular mappings, the one time iterative cost is only $O(m \times n)$.

EXPERIMENTS

To testify the proposed mappings validity in scrambling rectangle image, we scrambled 176×260 waterlilies test images for several times by our proposed 2D triangular mappings and recovered them by the corresponding inverse mappings (Table 1 and Fig. 4).

In Table 1, the first two rows are used to testify 2D lower triangular mappings and the rest rows are used to testify 2D upper triangular mappings. The proposed triangular mappings can scramble rectangle image directly and the scrambled image can be recovered by their corresponding inverse mappings (Table 1 and Fig. 4).

To compare the proposed triangular mappings with the two conventional methods in scrambling and recovering cost, which are shown in Fig. 1 and 2, we gave the periods, one time iterative cost, scrambling times and recovering times of the three kinds of methods in Table 2, where the periods were computed by obverse iterations until the scrambled images are recovered.

In Table 2, row 1-4 are used to testify the proposed triangular mappings; row 5 is used to testify the first conventional method and row 6 is used to testify the second conventional method. From Table 2, we can see the proposed triangular mappings have the least one time iterative cost and the one time iterative cost is only $O(176 \times 260)$. From columns in scrambling times and recovering times, we can see, in our proposed mappings, the scrambling times is equal to the recovering times because of the existing inverse mappings. In our proposed mappings there is no need to compute the period and using the period by obverse iterations to recover the original images. Therefore, the recovering cost of our methods is low.

To testify the proposed mappings robust in enduring erasing and cropping attacks, we attacked the scrambled images shown in Fig. 4 a-d by random cropping square blocks and recovered them by their corresponding inverse mappings. The attack variables and experimental results are shown in Table 3 and Fig. 5, respectively.

The proposed triangular mappings have a favorable scrambling performance in enduring erasing and cropping attacks (Table 3 and Fig. 5).

Table 1: The experimental variables of scrambling and recovering 176×260 images

No.	Scrambling transform matrix				Recovering variables		Iterative times	The scrambled image	The recovered image
	a	b	c	d	a^{-1} modulo 176	d^{-1} modulo 260			
1	3	0	1	3	59	87	13	Fig. 4a	Fig. 4e
2	5	0	2	7	141	223	13	Fig. 4b	Fig. 4e
3	7	1	0	11	151	71	13	Fig. 4c	Fig. 4e
4	13	2	0	17	149	153	13	Fig. 4d	Fig. 4e

Table 2: Comparing the proposed triangular mappings with the two conventional methods in scrambling and recovering cost

No.	Methods	Scrambling matrix variables				Period	One time iterative cost	Scrambling times	Recovering times
		a	b	c	d				
1	Lower triangular	3	0	1	3	300	$O(176 \times 260)$	13	13
2	Lower triangular	5	0	2	7	300	$O(176 \times 260)$	13	13
3	Upper triangular	7	1	0	11	240	$O(176 \times 260)$	13	13
4	Upper triangular	13	2	0	17	120	$O(176 \times 260)$	13	13
5	1st method	1	1	1	2	210	$O(260 \times 260)$	13	197
6	2nd method	1	1	1	2	60	$O(5 \times 176 \times 176 + 176 \times 260)$	13	47

Table 3: The attack variables of erasing and cropping experiments

No.	Randomize seed	Random block size	Quantity	The attacked scrambled image	The recovered image
1	25	10	120	Fig. 5a	Fig. 5e
2	83	11	80	Fig. 5b	Fig. 5f
3	46	12	66	Fig. 5c	Fig. 5g
4	67	13	90	Fig. 5d	Fig. 5h



Fig. 4: The results of experiments to scrambling and recovering 176×260 images where, a-d are the scrambled images and e is the recovered image

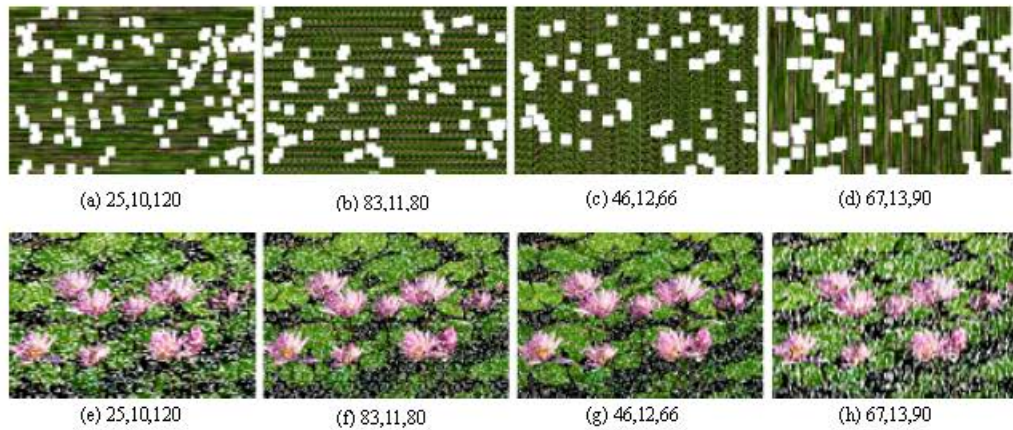


Fig. 5: The results of erasing and cropping attack experiments, a-d are the attacked images by random cropping square blocks and e-h are the recovered images

To testify the proposed mappings robust in enduring compressing attacks, we compressed the scrambled images shown in Fig. 4 a-d, respectively where the

compression qualities were set to 60. The attack variables and experimental results are shown in Table 4 and Fig. 6, respectively.

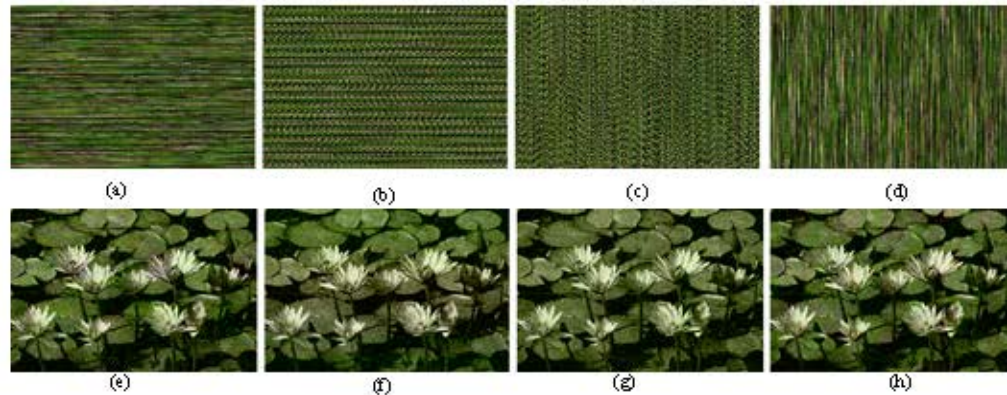


Fig. 6: The results of compressing attack experiments, a-d are the attacked images where the compression qualities are set to 60 and e-h are the recovered images

Table 4: The attack variables of compressing experiments

No.	The compression quality	The attacked scrambled image	The recovered image
1	60	Fig. 6a	Fig. 6e
2	60	Fig. 6b	Fig. 6f
3	60	Fig. 6c	Fig. 6g
4	60	Fig. 6d	Fig. 6h

The proposed triangular mappings have a favorable scrambling performance in enduring compressing attacks (Table 4 and Fig. 6).

CONCLUSIONS AND FUTURE WORK

In this study, we proposed two kinds of 2D triangular mappings and their inverse mappings to address the problem of scrambling rectangle image. To compare with the conventional two scrambling methods, our scrambling methods need not to expand the rectangle image into square or dividing rectangle image into square images. The proposed mappings have their inverse mappings to avoid the expensive iterative cost which is brought by the iterative period and the one time iterative cost is low. From the experiments, we can see our proposed mappings have low cost in scrambling and recovering rectangle image and a wonderful scrambling performance in enduring erasing cropping and compressing attacks.

The future work is to expand the proposed mappings into high dimensional and then use them to scramble image in the color and position space. The aim of our future work is to increase the couples of different elements in the different spaces in order to increase the security and scrambling performance.

ACKNOWLEDGMENT

This research was supported by the National Natural Science Foundation of China under grant No. 60673024.

REFERENCES

- Arnold, V.I. and A. Avez, 1968. Ergodic Problems of Classical Mechanics, New York: W.A. Benjamin Publishing.
- Dai, K.F., W.L. Huang, Z.Y. Chen and L. Tang, 2004. An MPEG-4 motion vector watermarking scheme based on scrambling using game of life. Acta Scientiarum Naturalium Universitatis Sunyatseni, 43: 192-195.
- Ding W., W.Q. Yan and D.X. Qi, 2000. Digital image scrambling and digital watermarking technology based on Conway's game. J. North China Univ. Technol., 12: 1-5.
- Han, F.L., J.K. Hu and X.H. Yu, 2006. A biometric encryption approach incorporating fingerprint indexing in key generation. International Conference on Intelligence Computing, Kunming, China, pp: 675-681.
- Hong C.Y. and W.G. Zou, 2005. Digital image scrambling technology based on three dimension Arnold transform and its period. J. Nanchang Univ. Nat. Sci., 29: 619-621.
- Hou, Q.B., X.F. Yang, Y.S. Wang and X.S. Huang, 2004. An image scrambling algorithm based on wavelet transform and knight's tour. Comput. Res. Dev., 41: 369-375.
- Kong, T. and Z. Dan, 2004. A new anti-Arnold transform algorithm. J. Software, 15: 1558-1564.
- Lin, X.H. and L.D. Cai, 2004. Scrambling research of digital image based on Hilbert curve. Chinese J. Stereol. Image Anal., 9: 224-227.
- Ma, Z.G. and S.S. Qiu, 2003. An image cryptosystem based on general cat map. J. China Inst. Commun., 24: 51-57.
- Qi, D.X., J.C. Zou and X.Y. Han, 2000. A new class of transform and its application in the image transform covering. Sci. China (Series E), 43: 304-312.

- Shen, J.B., X.G. Jin and C. Zhou, 2005. A color image encryption algorithm based on Magic cube transform and modular arithmetic operation. 6th Pacific Rim Conference on Multimedia. Jeju Island, Korea, pp: 270-280.
- Wang, D.M., 2005. The quasi-period of odd order magic square transform on digital image. *J. Zhejiang Univ. Technol.*, 33: 292-294.
- Wang, D.M. and Y.Q. Jin, 2005. Semi-period of doubly even order magic square transformed digital image. *J. Zhejiang Univ. Sci.*, 32: 274-276.
- Wang, Z.H., 2006. On the Period of 2D Random Matrix Scrambling transform and its Application in image hiding. *Chinese J. Comput.*, 29: 2218-2225.
- Yang, D.L., N. Cai and G.Q. Ni, 2006. Digital image scrambling technology based on the symmetry of arnold transform. *J. Beijing Inst. Technol.*, 15: 216-220.
- Zhang, L., S.M. Ji, Y. Xie, Q.L. Yuan, Y.H. Wan and G.J. Bao, 2005. Principle of image encrypting algorithm based on Magic cube transform. *Proceeding of Computational Intelligence and Security*. Xi'an, China, pp: 977-982.
- Zhang, X.H., Y.T. Yang and Y. Zhu, 2006. A secure hierarchical fragile watermarking method with tamper localization. *The International Multi-Symposiums on Computer and Computational Sciences*. Hangzhou, China.
- Zhu, C.X. and Z.G. Chen, 2005. Novel binary image digital watermarking algorithm based on DWT and chaotic scrambling. *Mini-Micro Systems*, 26: 1241-1245.
- Zou, J.C., R.K. Ward and D.X. Qi, 2004a. A new digital image scrambling method based on Fibonacci numbers. *Proceeding of the IEEE ISCAS*, Vancouver, Canada, pp: 965-968.
- Zou, J.C., R.K. Ward and X.D. Qi, 2004b. The generalized fibonacci transform and application to image scrambling. *Proceeding of the IEEE international conference on Acoustic, speech and signal processing*. Montreal, Canada, pp: 385-388.
- Zou, J.C., D.X. Qi and R.K. Ward, 2006. A novel watermarking method based on Fibonacci numbers. *International Conference on Virtual Reality Continuum and Its Applications*, Hongkong, China, pp: 335-338.