Available online at www.sciencedirect.com
SCIENCE DIRECT®
Chaos, Solitons and Fractals 21 (2004) 749–761
CHAOS SOLITONS & FRACTALS
www.elsevier.com/locate/chaos

A symmetric image encryption scheme based on 3D chaotic cat maps

Guanrong Chen [a,*], Yaobin Mao [b], Charles K. Chui [c,d]

[a] Department of Electronic Engineering, City University of Hong Kong, Hong Kong SAR, Hong Kong
[b] Department of Automation, Nanjing University of Science and Technology, Nanjing, 210094 PR China
[c] Department of Mathematics and Computer Science, University of Missouri at St. Louis, MO 63121, USA
[d] Department of Statistics, Stanford University, Stanford, CA 94305, USA

Accepted 11 December 2003

Communicated by T. Kapitaniak

Abstract

Encryption of images is different from that of texts due to some intrinsic features of images such as bulk data capacity and high redundancy, which are generally difficult to handle by traditional methods. Due to the exceptionally desirable properties of mixing and sensitivity to initial conditions and parameters of chaotic maps, chaos-based encryption has suggested a new and efficient way to deal with the intractable problem of fast and highly secure image encryption. In this paper, the two-dimensional chaotic cat map is generalized to 3D for designing a real-time secure symmetric encryption scheme. This new scheme employs the 3D cat map to shuffle the positions (and, if desired, grey values as well) of image pixels and uses another chaotic map to confuse the relationship between the cipher-image and the plain-image, thereby significantly increasing the resistance to statistical and differential attacks. Thorough experimental tests are carried out with detailed analysis, demonstrating the high security and fast encryption speed of the new scheme.

1. Introduction

The fascinating developments in digital image processing and network communications during the past decade have created a great demand for real-time secure image transmission over the Internet and through wireless networks. To meet this challenge, a variety of encryption schemes have been proposed [1,2,5–7,17,20]. Among them, chaos-based algorithms have shown some exceptionally good properties in many concerned aspects regarding security, complexity, speed, computing power and computational overhead, etc. Due to some intrinsic features of images, such as bulk data capacity and high correlation among pixels, traditional encryption algorithms such as DES, IDEA and RSA are not suitable for practical image encryption, especially under the scenario of on-line communications. The main obstacle in designing image encryption algorithms is that it is rather difficult to swiftly shuffle and diffuse data by traditional means of cryptology. In this respect, chaos-based algorithms have shown their superior performance. It has been proved that in many aspects chaotic maps have analogous but different characteristics as compared with conventional encryption algorithms [8,9,12,13]. For instance, classical encryption algorithms are sensitive to keys, while chaotic maps are sensitive to initial conditions and parameters; cryptographic algorithms shuffle and diffuse data by rounds of encryption, while chaotic maps spread the initial region over the entire phase space via iterations. The main difference between these two

* Corresponding author.
E-mail address: gchen@ee.cityu.edu.hk (G. Chen).

0960-0779/$ - see front matter
doi:10.1016/j.chaos.2003.12.022

techniques is that encryption operations are defined on finite sets, while chaos in a strict mathematical sense is defined on real numbers. Therefore, some elaborated constructions are needed to successfully employ chaos in encryption.

Chaos-based encryption is not a very new idea. As early as in 1989 [15], a chaotic function was already used to design a cryptographic algorithm. Although dedicated chaos-based image encryption schemes do not often appear in the literature, there does exist some, which are briefly discussed here. In [22], an encryption method called CKBA (chaotic key-based algorithm) was proposed. The algorithm first generates a time series based on a chaotic map, and then uses it to create a binary sequence as a key. According to the binary sequence so generated, image pixels are rearranged and then XOR or XNOR operated with the selected key. This method is very simple but has obvious defects in security, as pointed out lately in [10]: this method is very weak to the chosen/known-plain-text attack using only one plain-image, and moreover its security to brute-force attack is also questionable. In [17], a chaotic Kolmogorov-flow-based image encryption algorithm was designed. In this scheme, the whole image is taken as a single block and permuted through a key-controlled chaotic system based on the Kolmogorov flow. In order to confuse the data, a substitution based on a shift-registered pseudo-random number generator is applied, which alters the statistical property of the cipher-image. It was advocated that the scheme is computationally secure and superior to contemporary bulk encryption systems when aiming at efficient image and video data encryption. In [7], a systematical method was suggested for adapting an invertible two-dimensional chaotic map on a torus or on a square, so as to create a symmetric block encryption scheme. This approach to constructing the symmetric block cipher consists of three steps: (1) choose a chaotic map and generalize it by introducing some parameters; (2) discretize it to a finite square lattice of points that represent pixels; (3) extend the discretized map to three-dimensional and compose it with a simple diffusion mechanism. In this design, an example of the two-dimensional standard baker map was given to illustrate the construction procedure and to demonstrate the security. In the existing literature, some other two-dimensional chaotic maps such as the cat map and standard map have also been used for ciphers design. In [11], for example, a video encryption scheme was proposed based on a multiple digital chaotic system, which is called CVES (chaotic video encryption scheme). In this scheme, 2^n chaotic maps, controlled by another single chaotic map, are used to generate pseudo-random signals to mask the video, and to perform pseudo-random permutation of the masked video. It was claimed that the CVES is independent of any video compression algorithms and can provide high security for real-time digital videoing with fast encryption speed. In [11], this method was extended to the so-called RRS-CVES, which supports random retrieval of cipher-video with maximal time-out.

A new approach is suggested in this paper for fast and secure image encryption. Since digital images are usually represented as two-dimensional arrays, in order to fast de-correlate relations among pixels, a higher-dimensional chaotic map is designed and then used to shuffle the positions (and, if desired, grey values as well) of pixels in the image. Meanwhile, to confuse the relationship between cipher-image and plain-image, a diffusion process among pixels is performed. It is found that Arnold's cat map [3] is a good candidate for permutation, thus it is extended to a three-dimensional version, called 3D cat map, and then used for this purpose. Taking advantage of the exceptionally good properties of mixing and sensitivity to initial conditions and parameters of the chaotic 3D cat map, the proposed scheme incorporates Chen's chaotic system [4,21] in key scheming and alternatively uses permutation and diffusion to render the image totally unrecognizable.

The rest of this paper is organized as follows. Section 2 discusses the main features of the chaotic cat map and a way to extend it to three-dimensional. In Section 3, an image encryption scheme based on 3D cat map is proposed and discussed. In Sections 4 and 5, the security of the new scheme is evaluated via both cryptanalysis and experiments. Finally, Section 6 concludes the paper.

2. Extending the cat map to three-dimensional

Throughout this paper, the standard notation "$x \pmod 1$" will be used for the fractional parts of a real number x by subtracting or adding an appropriate integer.

The classical Arnold cat map is a two-dimensional invertible chaotic map [3,14] described by

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} \bmod 1. \tag{1}$$

The map is area-preserving since the determinant of its linear transformation matrix is equal to 1. The Lyapunov characteristic exponents of the map are the eigenvalues σ_1 and σ_2 of the matrix in (1), given by

$$\sigma_1 = \frac{1}{2}(3 + \sqrt{5}) > 1, \qquad \sigma_2 = \frac{1}{2}(3 - \sqrt{5}) < 1. \tag{2}$$

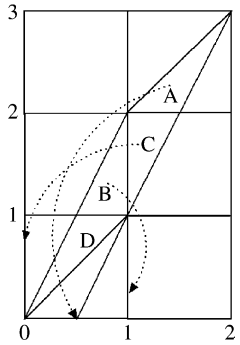

Fig. 1. Geometrical explanation of the 2D chaotic cat map.

The map is known to be chaotic, with geometrical explanation shown in Fig. 1, from which one can see that a unit square is first stretched by the linear transform and then folded by the modulo operation, mod.

The above 2D cat map is now generalized by introducing two control parameters, a and b, as follows:

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & a \\ b & ab+1 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} \bmod 1. \tag{3}$$

Furthermore, the map (3) is extended to three-dimensional by considering the following three maps. The first one is

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \\ z_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & a_z & 0 \\ b_z & a_z b_z + 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \\ z_n \end{bmatrix} \bmod 1, \tag{4}$$

that is, by fixing z_n unchanged it performs the 2D cat map on the x–y plane. The second one is similarly performed, but on the y–z plane while keeping x_n unchanged:

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \\ z_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & a_x \\ 0 & b_x & a_x b_x + 1 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \\ z_n \end{bmatrix} \bmod 1. \tag{5}$$

The last one is performed on the x–z plane:

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \\ z_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & a_y \\ 0 & 1 & 0 \\ b_y & 0 & a_y b_y + 1 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \\ z_n \end{bmatrix} \bmod 1. \tag{6}$$

Then, by combining these three maps together, one obtains a three-dimensional cat map as follows:

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \\ z_{n+1} \end{bmatrix} = \mathbf{A} \begin{bmatrix} x_n \\ y_n \\ z_n \end{bmatrix} \bmod 1, \tag{7}$$

where

$$\mathbf{A} = \begin{bmatrix} 1 + a_x a_z b_y & a_z & a_y + a_x a_z + a_x a_y a_z b_y \\ b_z + a_x b_y + a_x a_z b_y b_z & a_z b_z + 1 & a_y a_z + a_x a_y a_z b_y b_z + a_x a_z b_z + a_x a_y b_y + a_x \\ a_x b_x b_y + b_y & b_x & a_x a_y b_x b_y + a_x b_x + a_y b_y + 1 \end{bmatrix}.$$

As a special case, by simply setting $a_x = b_x = a_y = b_y = a_z = b_z = 1$, one has a direct extension of the original 2D cat map, as follows:

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \\ z_{n+1} \end{bmatrix} = \mathbf{A} \begin{bmatrix} x_n \\ y_n \\ z_n \end{bmatrix} \bmod 1, \quad \mathbf{A} = \begin{bmatrix} 2 & 1 & 3 \\ 3 & 2 & 5 \\ 2 & 1 & 4 \end{bmatrix}. \tag{8}$$

Through numerical calculations, one can easily verify that the three eigenvalues of $\mathbf{A}$ are: $\sigma_1 = 7.1842 > 1$, $\sigma_2 = 0.2430 < 1$ and $\sigma_3 = 0.5728 < 1$, which are actually the three Lyapunov characteristic exponents of the map (8).

Note that the leading Lyapunov characteristic exponent is strictly larger than 1, meaning that the extended 3D cat map is chaotic. Note moreover that since the leading Lyapunov characteristic exponent is larger than that of its 2D version, the 3D map is in a stronger sense chaotic therefore can perform better data mixing.

3. The image encryption scheme based on the 3D cat map

3.1. Discretization of the map

Since encryptions is a kind of transformation operated on a finite set, in order to incorporate a chaotic map into image encryption, one has to discretize it, while reserving some of its useful features such as the mixing property and the sensitivity to initial conditions and parameters.

The map (7) is discretized according to the following formula:

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \\ z_{n+1} \end{bmatrix} = \mathbf{A} \begin{bmatrix} x_n \\ y_n \\ z_n \end{bmatrix} \bmod N, \tag{9}$$

where

$$\mathbf{A} = \begin{bmatrix} 1 + a_x a_z b_y & a_z & a_y + a_x a_z + a_x a_y a_z b_y \\ b_z + a_x b_y + a_x a_z b_y b_z & a_z b_z + 1 & a_y a_z + a_x a_y a_z b_y b_z + a_x a_z b_z + a_x a_y b_y + a_x \\ a_x b_x b_y + b_y & b_x & a_x a_y b_x b_y + a_x b_x + a_y b_y + 1 \end{bmatrix},$$

and a_x, b_x, a_y, b_y, a_z, b_z are all positive integers.

One can easily verify that $\det \mathbf{A} = 1$, which means that the discrete version of the 3D cat map is a 1–1 map, and that its mixing property and the sensitivity to initial conditions and parameters are kept unchanged.

Let $C_d(i,j)$ and $C(x,y)$ denote the discrete and continuous maps in (9) and (7), respectively. Then the above discretization satisfies the following asymptotic property [7]:

$$\lim_{N\to\infty} \max_{0\leqslant i,j\leqslant N} |C(i/N, j/N) - C_d(i,j)| = 0.$$

Unfortunately, not all useful features of chaos can be preserved by discretization. For example, after discretization, an aperiodic chaotic map may become periodic, which will downgrade the security of chaotic encryption. Take the 2D cat map as an example. If one discretizes the continuous map (1) as

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & a \\ b & ab+1 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} \bmod N, \tag{10}$$

where $x_n, y_n \in \{0, 1, \ldots, N-1\}$, and a, b are positive integers, then with the choice of $a = 40$ and $b = 8$, after five rounds of iterations, an image of size 124×124 will turn back to its original. Fig. 2 shows this phenomenon for the above 2D cat

Fig. 2. Periodic phenomenon in discretizing the 2D cat map.

map. Theoretically, this is quite easy to understand. Let the period of the map (10) be P. Then the maximum period of the discrete 2D cat map and the integer N has the following relationship [16]:

$$P(N) = 3N, \quad N = 2 \cdot 5^k, \ k = 1, 2, \ldots,$$

$$P(N) = 2N, \quad N = 5^k \text{ or } N = 6 \cdot 5^k, \ k = 1, 2, \ldots,$$

$$P(N) \leqslant \frac{12N}{7} \quad \text{for all other } N,$$

where N is the width or height of the image. The three-dimensional cat map also has the same problem after discretization.

As a remedy, one resorts to a diffusion process thereby making the map non-invertible. This is further discussed below.

3.2. Diffusion process

There are two reasons for introducing diffusion in an encryption algorithm. On one hand, the diffusion processing can render the discretized chaotic map non-invertible. On the other hand, it can significantly change the statistical properties of the plain-image by spreading the influence of each bit of the plain-image all over the cipher-image. For a secure encryption scheme, a mechanism of diffusion is therefore necessary; otherwise the opponent can break the cryptosystem by comparing a pair of plain-text and cipher-text to discover some useful information. For the purpose of diffusion, the "XOR plus mod" operation will be applied in the new scheme, to each pixel in between every two adjacent rounds of the 3D cat map. This is further detailed below.

First, choose two numbers: one (denoted by L_i) is a floating number in $(0, 1)$, to be used as an initial condition; another (denoted by S) is an integer, to be used as a seed. Then, use L_i as the initial value to compute the chaotic logistic map:

$$x(k+1) = 4x(k)[1 - x(k)].$$

If the next value obtained is within the subinterval $(0.2, 0.8)$, then go to the next step; otherwise, the iteration goes on until a desired number in $(0.2, 0.8)$ is obtained. Here, notice that the value of 0.5 is a 'bad' point, trapping the iterations to the fixed point 0. If this case is encountered, a tiny perturbation should apply. Once a proper value is obtained from the logistic map, digitize it by amplifying it with a proper scaling and sampling. The digitized value is designated as $\phi(k)$ and is XOR-ed with the values of currently operated pixel and previously operated pixel in the image, according to the following formula:

$$C(k) = \phi(k) \oplus \{[I(k) + \phi(k)] \bmod N\} \oplus C(k-1),$$

where $I(k)$ is the currently operated pixel and $C(k-1)$ is the previously output cipher-pixel, in a vector that was strung out from the image, and $C(k)$ is the XOR-ed value, and N is the color level (for a 256 grey-scale image, $N = 256$). One may set the initial value $I(0) = S$. The inverse transform of the above is given by

$$I(k) = \{\phi(k) \oplus C(k) \oplus C(k-1) + N - \phi(k)\} \bmod N.$$

Since in Step k the previous value $C(k-1)$ is known, the value $C(k)$ can be ciphered out.

3.3. Key scheming

According to the basic principle of cryptology [18], a cryptosystem should be sensitive to the key, i.e., the cipher-text should have close correlation with the key. There are two ways to accomplish this requirement: one is to mix the key thoroughly into the plain-text through the encryption process; another is to use a good (ideally, truly random) key generation mechanism.

The key (denoted K_m) directly used in the proposed encryption scheme is the vector of parameters of the chaotic map, which can be floating numbers or integers, whilst the user's input key (denoted K_u) is a string of characters, which can be taken as a sequence of bits. Thus, there is a transform from K_u to K_m, during which a diffusion mechanism is introduced, so as to protect the key from opponent's attacks.

In the proposed scheme, Chen's chaotic system is employed in key scheming, which is modelled by [4,21]

$$\begin{cases} \dot{x} = a(y - x) \\ \dot{y} = (c - a)x - xz + cy \\ \dot{z} = xy - bz, \end{cases}$$

where a, b and c are parameters. When $a = 35$, $b = 3$ and $c \in [20, 28.4]$, the system is chaotic. Simulation shows that the system orbit is extremely sensitive to the parameter c, therefore c is used to control the generation of the cipher key.

The key used in the proposed encryption scheme is a binary sequence of 128 bits. The binary sequence is divided into eight segments, denoted as k_{a_x}, k_{b_x}, k_{a_y}, k_{b_y}, k_{a_z}, k_{b_z}, k_l, k_s, respectively, each with 16-bit long. Parameters k_{a_x}, k_{b_x}, k_{a_y}, k_{b_y}, k_{a_z}, k_{b_z} are used to generate the six control parameters of the 3D cat map (7), while k_l and k_s are used to generate the initial values of the logistic map, L_i, and initial value of the mod operation, S, respectively, as discussed in Section 3.2.

To generate a_x, b_x, the following formulas are first used to compute the control parameter c of Chen's system:

$$c_{a_x} = K_{a_x} \times 8.4 + 20, \tag{11}$$

where $K_{a_x} = \sum_{i=0}^{15} k_{a_x}(i) \times 2^i$, in which $k_{a_x}(i)$ is the ith bit in sequence k_{a_x}. Initial values x_0, y_0, z_0 of Chen's system are also derived from k_{a_x} and k_{b_x}, by using the following formulas:

$$x_{0h} = K_{b_x} \times 80 - 40,$$

$$y_{0h} = K_{a_x} \times 80 - 40,$$

$$z_{0h} = K_{b_x} \times 60,$$

where $K_{b_x} = \sum_{i=0}^{15} k_{b_x}(i) \times 2^i$. Then, in the next step, parameters are set as $a = 35$, $b = 3$, and the other parameters obtained above are used to iterate Chen's system for 100 and 200 times, respectively, yielding two values: z_{100}, z_{200}. Next, then, the following formulas are used to obtain the final parameter values of a_x and b_x for the 3D cat map:

$$a_x = \text{round}(z_{100}/60 \times N),$$

$$b_x = \text{round}(z_{200}/60 \times N),$$

where N is the side length of a cube to be scrambled by the 3D cat map.

A similar process is performed on parameters k_{a_y}, k_{b_y}, k_{a_z}, k_{b_z}, k_l, k_s, to obtain the control parameters of the 3D cat map, a_y, b_z, a_z, b_y, and the initial values of the logistic map, L_i, and the initial value of the mod operation, S. The following two formulas are used instead, to generate L_i and S:

$$L_i = z_{100}/60,$$

$$S = \text{round}(z_{200}/60 \times 255).$$

3.4. Image encryption scheme based on the 3D cat map

The complete image encryption scheme consists of five steps of operations, as shown in Fig. 3.

Step 1. Key generation. Select a sequence of 128 bits as the key, and split them into eight groups, which are further mapped onto several parameters of the 3D cat map and the logistic map, a_x, b_x, a_y, b_y, a_z, b_z, L_i and S, as discussed in Section 3.3.

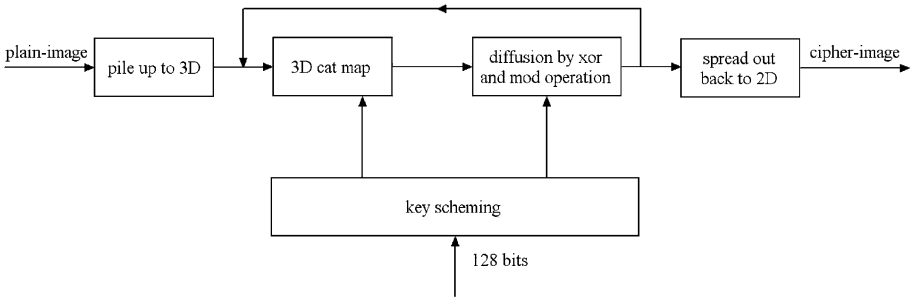

Fig. 3. Block diagram of the image encryption using the 3D cat map.

Step 2. Pile up the two-dimensional image into three-dimensional. Suppose that the image to be encrypted is of W-pixel wide and H-pixel high. First, one needs to pile up all pixels of the image, to form several cubes of size $N_1 \times N_1 \times N_1$, $N_2 \times N_2 \times N_2, \ldots, N_i \times N_i \times N_i$, respectively. To convert an image into several cubes, the following condition must be satisfied:

$$W \times H = N_1^3 + N_2^3 + \cdots + N_i^3 + R, \tag{12}$$

where $N_i \in \{2, 3, \ldots, N\}$ is the side length of each cube, N is the size of the maximum allowable cube, and $R \in \{0, 1, 2, \ldots, 7\}$ is the remainder.

Step 3. Perform the three-dimensional cat map. Use a_x, b_x, a_y, b_y, a_z, b_z as control parameters to perform the three-dimensional discrete cat map (as discussed in Section 3.1) on each image cubes, generating shuffled images.

Step 4. Diffusion process. Set $x(0) = L_i$ and $C(0) = S$, and then perform the diffusion process once according to the algorithm described in Section 3.2.

Step 5. Transform the three-dimensional cubes back to a two-dimensional image. The three-dimensional cubes are appropriately arranged, laying back to a two-dimensional image for display or for storage.

Note that the operations in Steps 3 and 4 are often performed alternatively for several rounds according to the security requirement. The more rounds are processed, the more secure the encryption is, but at the expense of computations and time delays.

To this end, the decipher procedure is similar to that of the encipher process illustrated above, with reverse operational sequences to those described in Steps 3 and 4. Since both decipher and encipher procedures have similar structures, they have essentially the same algorithmic complexity and time consumption.

4. Security analysis

A good encryption scheme should resist all kinds of known attacks, such as known-plain-text attack, cipher-text-only attack, statistical attack, differential attack, and various brute-force attacks. Some security analysis has been performed on the proposed image encryption scheme, including the most important ones like key space analysis, statistical analysis, and differential analysis, which has demonstrated the satisfactory security of the new scheme, as demonstrated in the following.

4.1. Key space analysis

A good image encryption algorithm should be sensitive to the cipher keys, and the key space should be large enough to make brute-force attacks infeasible. For the proposed image encryption algorithm, key space analysis and testing have been carefully performed and completely carried out, with results summarized as follows:

- *Number of control parameters*. This algorithm is a 128-bit encryption scheme, with key space size $2^{128} \approx 3.4028 \times 10^{38}$. Since this scheme takes advantage of the 3D cat map, the opponent may try to bypass guessing the key and instead directly guess all the possible combinations of the control parameters a_x, b_x, a_y, b_y, a_z, b_z. However, the combinations of the 3D cat map control parameters are large enough to prevent such exhaustive searching. A rough estimate of all possible combinations of control parameters is as follows. Suppose that one has a 512×512 image. According to the encryption scheme, it will be piled up to a $64 \times 64 \times 64$ cube. Then, since each parameter among a_x, b_x, a_y, b_y, a_z, b_z is in between 1 and 64, possible combinations of control parameters are $64^6 = 2^{36} \approx 6.8719 \times 10^{10}$. Notice that this is just for one round of the several iterations. If each ciphering round of the 3D cat map uses different ciphering keys, then the increase of round numbers will further enlarge the key space. Compared with the 2D cat map, the key space of the 3D map is much larger than the key space of the 2D map, which is already very large—about $512 \times 512 = 2^{18} \approx 2.621 \times 10^5$.
- *Key sensitivity test*. Assume that a 16-character ciphering key is used. This means that the key consists of 128 bits. A typical key sensitivity test has been performed, according to the following steps:
  1. First, a 512×512 image is encrypted by using the test key "1234567890123456".
  2. Then, the least significant bit of the key is changed, so that the original key becomes, say "1234567890123457" in this example, which is used to encrypt the same image.
  3. Finally, the above two ciphered images, encrypted by the two slightly different keys, are compared.

The result is: the image encrypted by the key "1234567890123456" has 99.61% of difference from the image encrypted by the key "1234567890123457" in terms of pixel grey-scale values, although there is only one bit difference in

the two keys. Fig. 4 shows the test result. Moreover, when a 16-character key is used to encrypt an image while another trivially modified key is used to decrypt the ciphered image, the decryption also completely fails. Fig. 5 clearly shows that the image encrypted by the key "1234567890123456" is not correctly decrypted by using the key "1234567890123457" there, which has also only one bit difference between the two keys.

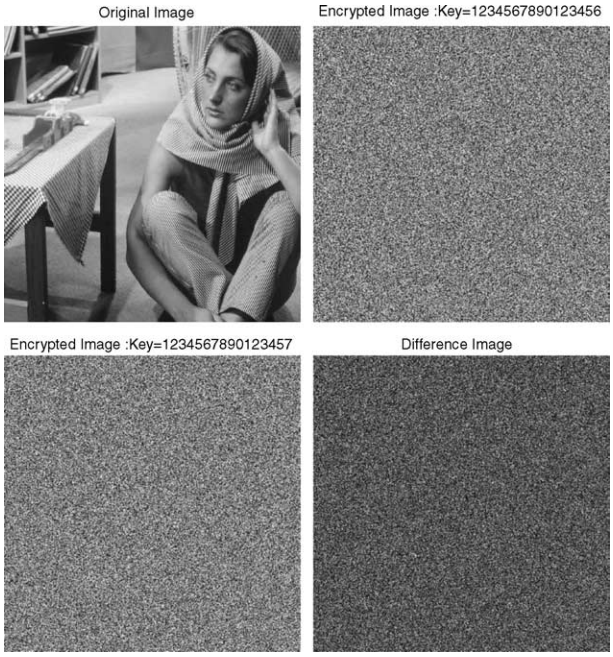

Fig. 4. Key sensitive test: result 1.

Fig. 5. Key sensitive test: result 2.

4.2. Statistical analysis

Shannon once said, in his masterpiece [19], "It is possible to solve many kinds of ciphers by statistical analysis," and, therefore, he suggested two methods of diffusion and confusion in order to frustrate the powerful attacks based on statistical analysis.

Statistical analysis has been performed on the proposed image encryption algorithm, demonstrating its superior confusion and diffusion properties which strongly resist statistical attacks. This is shown by a test on the histograms of the enciphered images and on the correlations of adjacent pixels in the ciphered image.

1. *Histograms of encrypted images*. Select several 256 grey-scale images of size 512×512 that have different contents, and calculate their histograms. One typical example among them is shown in Fig. 6. From the figure, one can see that the histogram of the ciphered image is fairly uniform and is significantly different from that of the original image.
2. *Correlation of two adjacent pixels*. To test the correlation between two vertically adjacent pixels, two horizontally adjacent pixels, and two diagonally adjacent pixels, respectively, in a ciphered image, the following procedure was carried out. First, randomly select 1000 pairs of two adjacent pixels from an image. Then, calculate the correlation coefficient of each pair by using the following two formulas:

$$\operatorname{cov}(x,y) = E(x - E(x))(y - E(y)),$$

$$r_{xy} = \frac{\operatorname{cov}(x,y)}{\sqrt{D(x)}\sqrt{D(y)}},$$

where x and y are grey-scale values of two adjacent pixels in the image. In numerical computation, the following discrete formulas were used:

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i,$$

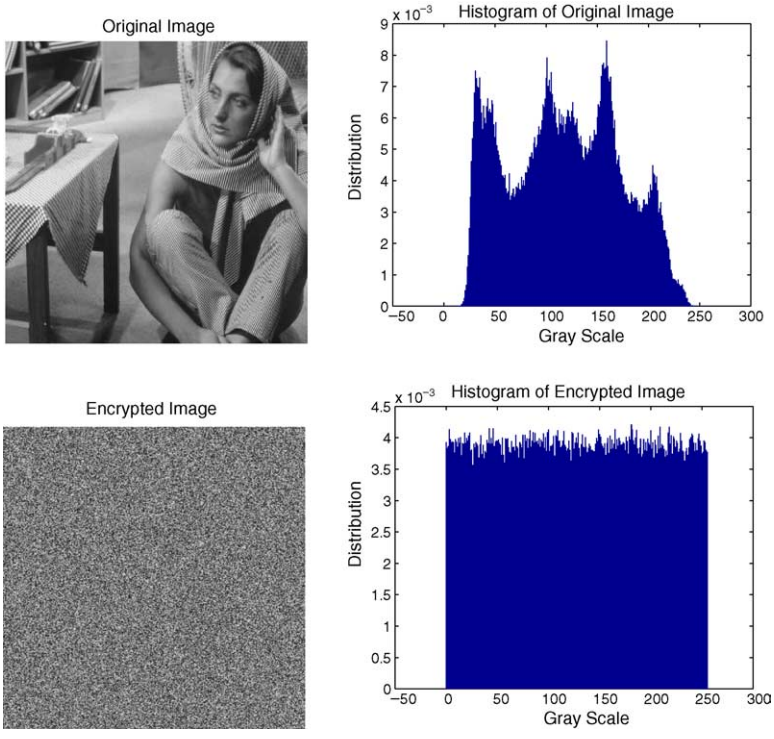

Fig. 6. Histograms of the plain-image and the cipher-image.

$$D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2,$$

$$\mathrm{cov}(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y)).$$

Fig. 7 shows the correlation distribution of two horizontally adjacent pixels in the plain-image and that in the cipher-image: the correlation coefficients are 0.91765 and 0.01183, respectively, which are far apart. Similar results for diagonal and vertical directions were obtained, which are shown in Table 1.

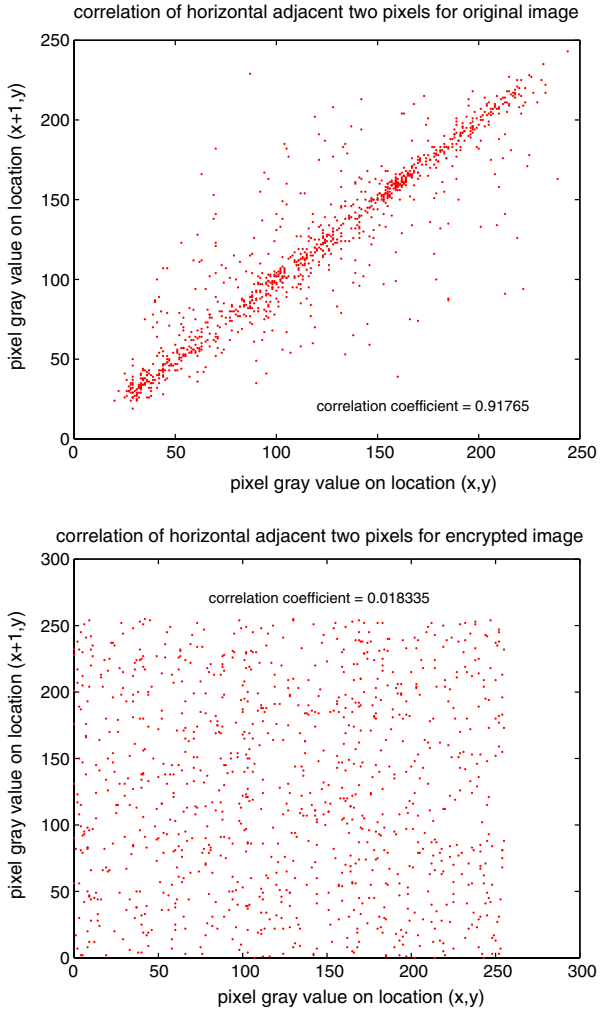

Fig. 7. Correlations of two horizontally adjacent pixels in the plain-image and in the cipher-image.

Table 1
Correlation coefficients of two adjacent pixels in two images

	Plain image	Ciphered image
Horizontal	0.91765	0.01183
Vertical	0.95415	0.00016
Diagonal	0.90205	0.01480

4.3. Differential attack

In general, the opponent may make a slight change (e.g., modify only one pixel) of the encrypted image, and then observes the change of the result. In this way, he may be able to find out a meaningful relationship between the plain-image and the cipher-image. If one minor change in the plain-image can cause a significant change in the cipher-image, with respect to diffusion and confusion, then this differential attack would become very inefficient and practically useless.

To test the influence of one-pixel change on the whole image encrypted by the proposed algorithm, two common measures were used: *number of pixels change rate* (*NPCR*) and *unified average changing intensity* (*UACI*). Denote two cipher-images, whose corresponding plain-images have only one-pixel difference, by C_1 and C_2, respectively. Label the grey-scale values of the pixels at grid (i,j) of C_1 and C_2 by $C_1(i,j)$ and $C_2(i,j)$, respectively. Define a bipolar array, D, with the same size as image C_1 or C_2. Then, $D(i,j)$ is determined by $C_1(i,j)$ and $C_2(i,j)$, namely, if $C_1(i,j) = C_2(i,j)$ then $D(i,j) = 1$; otherwise, $D(i,j) = 0$.

The NPCR is defined as

$$\mathrm{NPCR} = \frac{\sum_{i,j} D(i,j)}{W \times H} \times 100\%,$$

where W and H are the width and height of C_1 or C_2. The NPCR measures the percentage of different pixel numbers between the two images. The UACI, on the other hand, is defined as

$$\mathrm{UACI} = \frac{1}{W \times H}\left[\sum_{i,j} \frac{|C_1(i,j) - C_2(i,j)|}{255}\right] \times 100\%,$$

which measures the average intensity of differences between the two images.

Tests have been performed on the proposed scheme, about the one-pixel change influence on a 256 grey-scale image of size 512×512. The test results are shown in Figs. 8 and 9. Generally, with the increase of ciphering rounds, the influence of one-pixel change is increased. Hence, it is reasonable to increase the ciphering rounds in the test to achieve higher security; yet, this is at the expense of processing speed.

5. Other tests

Apart from the security consideration, some other issues on an image encryption scheme are also important, including the running speed, particularly for real-time Internet applications.

The proposed image encryption algorithm is indeed very fast. Simulation shows that the average enciphering/deciphering speed is 1.0 MB/s, and the peak speed can reach up to 2.1 MB/s, on a 1 GHz Pentium IV personal computer. Table 2 shows the test results of enciphering/deciphering speeds on 256 grey-scale images of different sizes. The computer used in this test is 1 GHz Pentium IV with 256 M memory and 40 G hard-disk capacity.

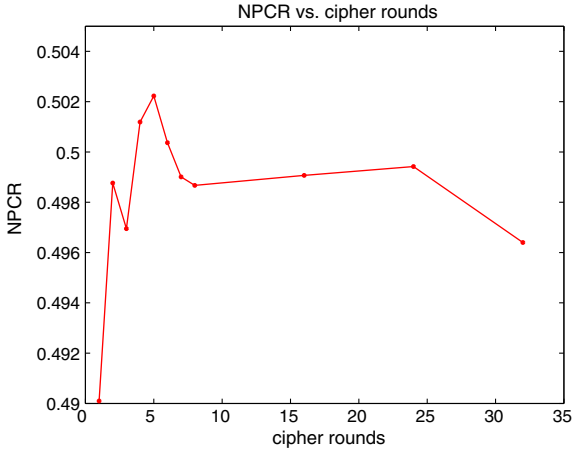

Fig. 8. NPCR vs. ciphering rounds.

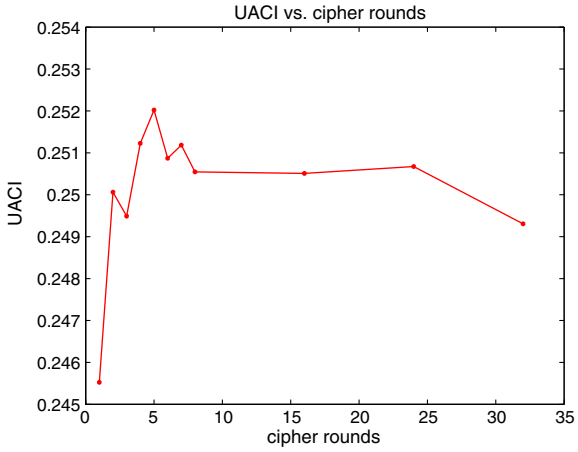

Fig. 9. UACI vs. ciphering rounds.

Table 2
Enciphering/deciphering speed test results

Image size (in pixels)	Encryption (in s)	Decryption (in s)
256×256	<0.4	<0.4
512×512	1	1
1024×1024	3	3
2048×2048	14	14

6. Concluding remarks

In this paper, the well-known two-dimensional chaotic cat map has been generalized to three-dimensional, and then used to design a fast and secure symmetric image encryption scheme. This new scheme employs the 3D cat map to shuffle the positions (and, if desired, grey values as well) of image pixels and uses another chaotic map to confuse the relationship between cipher-image and plain-image, thereby significantly increasing its resistance to various attacks such as the statistical and differential attacks. Thorough experimental tests have been carried out with detailed numerical analysis, demonstrating the high security and fast speed of the new image encryption scheme. This scheme is particularly suitable for real-time Internet image encryption and transmission applications.

Acknowledgement

This work was supported by the Applied R&D Centres, City University of Hong Kong, under the grants 9410011 and 9620004.

References

[1] Chang HKC, Liu JL. A linear quadtree compression scheme for image encryption. Signal Process Image Commun 1997;10(4):279–90.
[2] Chang CC, Hwang MS, Chen TS. A new encryption algorithm for image cryptosystems. J Syst Software 2001;58:83–91.
[3] Chen G, Dong X. From chaos to order: methodologies, perspectives and applications. Singapore: World Scientific; 1998.
[4] Chen G, Ueta T. Yet another chaotic attractor. Int J Bifurcat Chaos 1999;9(7):1465–6.
[5] Cheng H, Li XB. Partial encryption of compressed images and videos. IEEE Trans Signal Process 2000;48(8):2439–51.
[6] Bourbakis N, Alexopoulos C. Picture data encryption using SCAN patterns. Pattern Recognit 1992;25(6):567–81.
[7] Fridrich J. Symmetric ciphers based on two-dimensional chaotic maps. Int J Bifurcat Chaos 1998;8(6):1259–84.
[8] Kocarev L. Chaos-based cryptography: a brief overview. IEEE Circ Syst Mag 2001;1(3):6–21.

[9] Kocarev L, Jakimovski G. Chaos and cryptography: from chaotic maps to encryption algorithms. IEEE Trans Circ Syst—I 2001;48(2):163–9.
[10] Li SJ, Zheng X. Cryptanalysis of a chaotic image encryption method. In: IEEE Int Symposium Circuits and Systems, Scottsdale, AZ, USA, 2002.
[11] Li SJ, Zheng X, Mou X, Cai Y. Chaotic encryption scheme for real-time digital video. In: Proc SPIE on Electronic Imaging, San Jose, CA, USA, vol. 4666, 2002.
[12] Mao YB, Chen G. Chaos-based image encryption. Handbook of Computational Geometry for Pattern Recognition, Computer Vision, Neurocomputing and Robotics. New York: Springer-Verlag; in press, 2004.
[13] Mao YB, Chen G, Lian SG. A novel fast image encryption scheme based on the 3D chaotic baker map, Int J Bifurcat Chaos, accepted, 2003.
[14] http://mathworld.woffram.com/ArnoldsCatMap.html.
[15] Matthews R. On the derivation of a "chaotic" encryption algorithm. Cryptologia 1989;8(1):29–41.
[16] Peterson G. Arnold's cat map, 1997. Available from: http://online.redwoods.cc.ca.us /instruct/ darnold/ maw/catmap.htm.
[17] Scharinger J. Fast encryption of image data using chaotic Kolmogorov flows. J Electron Imaging 1998;7(2):318–25.
[18] Schneier B. Applied cryptography: protocols, algorithms, and source code in C. 2nd ed. New York: Wiley; 1995.
[19] Shannon CE. Communication theory of secrecy system. Bell Syst Tech J 1949;28:656–715.
[20] Uehara T, Safavi-Naini R, Ogunbona P. Securing wavelet compression with random permutations. In: IEEE Pacific Rim Conference on Multimedia, 2000. p. 332–5.
[21] Ueta T, Chen G. Bifurcation analysis of Chens equation. Int J Bifurcat Chaos 2000;10(8):1917–31.
[22] Yen JC, Guo JI. A new chaotic key-based design for image encryption and decryption. In: Proc IEEE Int Conference Circuits and Systems, vol. 4, 2000. p. 49–52.