International Journal of Bifurcation and Chaos, Vol. 14, No. 10 (2004) 3613–3624

A NOVEL FAST IMAGE ENCRYPTION SCHEME BASED ON 3D CHAOTIC BAKER MAPS*

YAOBIN MAO
Department of Automation, Nanjing University of Science and Technology, Nanjing, P. R. China
maoyaobin@163.com

GUANRONG CHEN
Department of Electronic Engineering, City University of Hong Kong, Hong Kong SAR, P. R. China

SHIGUO LIAN
Department of Automation, Nanjing University of Science and Technology, Nanjing, P. R. China

Received June 27, 2003; Revised November 5, 2003

Symmetric block encryption schemes, designed on invertible two-dimensional chaotic maps on a torus or a square, prove feasible and secure for real-time image encryption according to the commonly used criteria given in the literature. In this paper, a typical map of this kind, namely, the baker map, is further extended to be three-dimensional and then used to speed up image encryption while retaining its high degree of security. The proposed algorithm is described in detail, along with its security analysis and implementation. Experimental results show that this three-dimensional baker map is 2–3 times faster than the two-dimensional one, showing its great potential in real-time image encryption applications.

Keywords: Chaos; image encryption; 3D baker map.

1. Introduction

Encryption on image or video objects has its own requirements due to the intrinsic characters of images such as bulk data capacity and high redundancy. Traditional symmetric encryption algorithms such as DES, IDEA, Blowfish and RSA are generally not suitable for image encryption due to their slow speed in real-time processing and some other issues such as in handling various data formatting.

Recently, the idea of using chaos in data encryption has been introduced and discussed in, for instance, [Fridrich, 1997, 1998; Li *et al.*, 2002; Mao & Chen, 2004, Chen *et al.*, 2004; Scharinger, 1998]. It has been shown that chaos-based algorithms have advantages in applications of bulk data encryption, which make use of two special features of chaotic maps — the sensitivity to initial conditions and parameters and the mixing property (topological transitivity or ergodicity), [Fridrich, 1998; Kocarev, 2001; Kocarev & Jakimovski, 2001; Masuda & Aihara, 2002]. Sensitivity to initial conditions means that when a chaotic map is iteratively applied to two extremely close initial points, the iterates quickly diverge, and become uncorrelated in the long term. This character is especially useful

*This research was supported by the Hong Kong Research Grants Council under the CERG grant CityU 1115/03E and by the Hong Kong City University Shenzhen Applied R&D Centres.

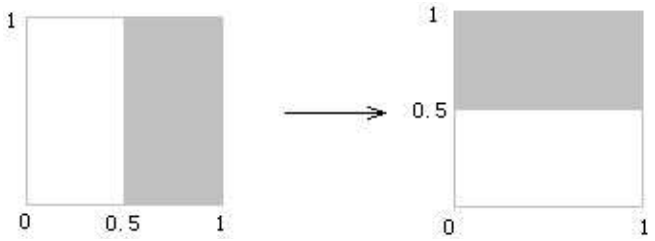

Fig. 1. The standard 2D baker map.

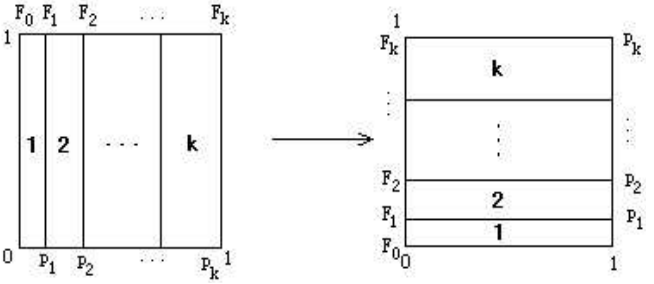

Fig. 2. The generalized 2D baker map.

in image encryption, since two adjacent pixels in an image are highly correlated but while using a chaotic map, they will be uncorrelated after several rounds of iteration. Sensitivity to parameters causes the properties of the map to change quickly when slightly perturbing the parameters on which the map depends. This property of the parameters is just like that of a cipher key, therefore, in a chaotic based encryption scheme, those parameters are often used as keys. Mixing is the tendency of the system to quickly blend small portions of the state space into an intricate network of filaments. This character can also make correlated information become scattered all over the phase space. These characteristics form a basis of chaotic data encryption.

In practical applications, there are two kinds of methods used for constructing secure encryption algorithms. For quite a long time, many now-classic schemes like DES and IDEA [Schneier, 1995] emphasize more on substitution than on permutation, aiming at the key issue of security alone. Actually, permutation plus diffusion can also compose very good encryption schemes with not only high security but also fast speed. In fact, this observation led to some excellent encryption schemes based on two-dimensional chaotic maps, which were essentially motivated by this observation [Fridrich, 1997, 1998; Scharinger, 1998]. The present paper continues the same pursuit with further improvement, in which the two-dimensional baker map is extended to be three-dimensional and is then used to compose a fast and secure image encryption scheme. As will be shown later in this paper, experimental results show that this three-dimensional baker map is 2–3 times faster than the two-dimensional one, showing its great potential in real-time image encryption applications.

The remainder of the paper is arranged as follows: Section 2 describes the two-dimensional baker map and its extension to be three-dimensional. Section 3 constructs a new image encryption scheme based on the extended three-dimensional chaotic baker map. Some analysis of the proposed image encryption scheme is then given in Sec. 4. Section 5 shows some test results and, finally, Sec. 6 concludes the paper with some discussions.

2. Extending the 2D Baker Map to 3D

2.1. *The 2D baker map*

The standard 2D baker map, denoted by B hereafter, is described by [Fridrich, 1997, 1998]

$$B(x) = \begin{cases} \left(2x, \dfrac{y}{2}\right) & 0 \le x < \dfrac{1}{2} \\ \left(2x-1, \dfrac{y}{2}+\dfrac{1}{2}\right) & \dfrac{1}{2} \le x \le 1 \end{cases} \tag{1}$$

This 2D baker map is a chaotic bijection of the unit square $I \times I$ onto itself, as depicted in Fig. 1.

The generalized baker map [Pichler & Scharinger, 1995, 1996] is defined as follows (see Fig. 2): divide the unit square into k vertical rectangles, $[F_{i-1}, F_i) \times [0,1)$, $i = 1, \ldots, k$, $F_i = p_1 + p_2 + \cdots + p_i$, $F_0 = 0$, such that $p_1 + \cdots + p_k = 1$. The lower right corner of the ith rectangle is located at $F_i = p_1 + \cdots + p_i$. The generalized baker map stretches each rectangle horizontally by the factor $1/p_i$; at the same time, the rectangle is contracted vertically by the factor p_i. Formally, the map is defined by

$$B(x,y) = \left(\frac{1}{p_i}(x - F_i), p_i y + F_i\right)$$

for $(x,y) \in [F_i, F_i + p_i) \times [0,1)$.

The discretized baker map is required to assign a pixel to another pixel in a bijective manner. As pointed out in [Fridrich, 1998], if the discretized

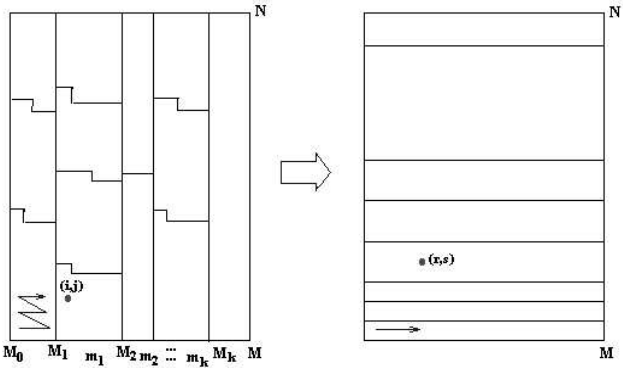

Fig. 3. The generalized discrete 2D baker map.

map satisfied the following formula:

$$\lim_{N\to\infty} \max_{0\le i,j<N} |B_c(i/N, j/N) - B_d(i,j)| = 0$$

then it could inherit the basic properties of its original continuous version, namely, the discretized map will be close to the continuous one, as the number of pixels tends to infinity. In the above formula, B_c stands for the continuous baker map and B_d, its discretized version.

If one divides an $N \times N$ square into vertical rectangles with N pixels high and N_i pixels wide, then the discretized baker map can be expressed as follows:

$$B_d(r,s) = \left(\frac{N}{n_i}(r - N_i) + s \bmod \frac{N}{n_i}, \frac{n_i}{N}\left(s - s \bmod \frac{N}{n_i}\right) + N_i\right)$$

where the pixel (r,s) is with $N_i \le r < N_i + n_i$, $0 \le s < N$. The sequence of k integers, $n_1, n_2, \ldots, n_k$, is chosen such that each integer n_i divides N, and $N_i = n_1 + n_2 + \cdots + n_i$, $N = n_1 + \cdots + n_k$. If not all of the integers $n_1, n_2, \ldots, n_k$ divide N, and furthermore if the unit is an $M \times N$ rectangle, then the map from pixel (i,j) to (r,s) can be obtained according to the next formula, which is illustrated in Fig. 3.

As can be seen from Fig. 3, an $M \times N$ square is divided into vertical rectangles of N-pixel height and m_i-pixel wide. The sequence of k integers, $m_1, \ldots, m_k$, is chosen such that $M_i = m_1 + \cdots + m_i$, $M = m_1 + m_2 + \cdots + m_k$, and $M_0 = 0$. Thus, the generalized discrete 2D baker map is expressed as

$$\begin{aligned}(r,s) &= B_d(i,j)\\ &= (\lfloor (M_{i-1} \times N + j \times m_i + i - M_{i-1})/M \rfloor,\\ &\quad (M_{i-1} \times N + j \times m_i + i - M_{i-1}) \bmod M)\end{aligned}$$

2.2. *The 3D baker map*

A direct extension of the standard two-dimensional baker map to a three-dimensional setting can be accomplished by the following procedure.

Firstly, divide an unit cube into four even narrow stripes of small cubes, and then press each of them and pile them up one by one to form a new unit cube that has the same volume with the original. Mathematically, it is described by

$$B(x,y,z) = \begin{cases} \left(2x,\, 2y,\, \frac{z}{4}\right) & 0 \le x < \frac{1}{2},\, 0 \le y < \frac{1}{2} \\ \left(2x,\, 2y-1,\, \frac{z}{4} + \frac{1}{2}\right) & 0 \le x < \frac{1}{2},\, \frac{1}{2} \le y \le 1 \\ \left(2x-1,\, 2y,\, \frac{z}{4} + \frac{1}{4}\right) & \frac{1}{2} \le x < 1,\, 0 \le y < \frac{1}{2} \\ \left(2x-1,\, 2y-1,\, \frac{z}{4} + \frac{3}{4}\right) & \frac{1}{2} \le x \le 1,\, \frac{1}{2} \le y \le 1 \end{cases} \tag{2}$$

which is illustrated in Fig. 4.

Compared with 2D baker map, the extended 3D baker map has more intensive chaotic characters, which turns out to be propitious to image encryption. This can be verified by comparing the Lyapunov Exponents (LE) of the two baker maps.

Recall the 2D baker map defined in formula (1). A unit square is first divided evenly into two parts, and each part is stretched horizontally whilst contracted vertically. Then, they are piled up one over another. Since the stretching in the horizontal direction is magnified by a factor of 2, and the measure

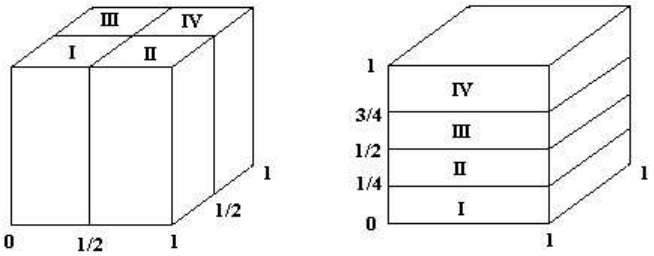

Fig. 4. The standard 3D baker map.

is uniform in the same direction, the probability of the iterations falling in these two regions is evenly equal to 1/2. Thus, the LE in the horizontal direction is

$$\lambda_h = \frac{1}{2}\ln 2 + \frac{1}{2}\ln 2 = \ln 2 > 0\,.$$

Similarly, the contraction in the vertical direction for the two parts are both 1/2, so another LE in the vertical direction is

$$\lambda_v = \frac{1}{2}\ln\frac{1}{2} + \frac{1}{2}\ln\frac{1}{2} = -\ln 2 < 0\,.$$

Now, observe the LEs of the 3D baker map. Since the 3D baker map is a three-dimensional map, there should be three LEs, which indicate the exponential divergences of the map in three principal directions, respectively. In Fig. 4, the direction of $\mathrm{I}\to\mathrm{II}$ is denoted as the x-direction, $\mathrm{I}\to\mathrm{III}$ as the y-dimension, and the last is the z-dimension. Each of the four parts in the unit cube is stretched in both x- and y-directions with equivalent probability. Therefore, the map has two LEs,

$$\lambda_x = \frac{1}{4}\ln 2 + \frac{1}{4}\ln 2 + \frac{1}{4}\ln 2 + \frac{1}{4}\ln 2 = \ln 2 > 0$$

and

$$\lambda_y = \frac{1}{4}\ln 2 + \frac{1}{4}\ln 2 + \frac{1}{4}\ln 2 + \frac{1}{4}\ln 2 = \ln 2 > 0\,,$$

respectively. In the z-direction, the unit cube is contracted, so the LE in this direction is

$$\lambda_z = 4\times\frac{1}{4}\ln\frac{1}{4} = -\ln 4 < 0\,.$$

Since there are two LEs larger than 0 in the 3D baker map, the map is hyperchaotic, implying that it is "more chaotic" than its 2D counterpart to some extent.

Similarly to the 2D baker map, the 3D baker map also has its general form. As can be seen from Fig. 5, an unit cube is firstly divided into several small stripes, and each stripe is pressed and then piled up to form a new unit cube of the same volume. More precisely, assume that the unit cube is divided into $k\times t$ blocks, $[W_{i-1}, W_i)\times[H_{j-1}, H_j)\times[0,1)$, $i=1,\ldots,k$, $j=1,\ldots,t$, $W_i = w_1, w_2, \ldots, w_i$, $W_0 = 0$, such that $w_1 + w_2 + \cdots + w_k = 1$, and $H_j = h_1 + h_2 + \cdots + h_j$, $H_0 = 0$, with $h_1 + h_2 + \cdots + h_t = 1$. Then, the generalized 3D baker map is given by

$$B_3(x,\,y,\,z) = \left(\frac{1}{w_i}(x - W_i),\ \frac{1}{h_j}(y - H_j), w_i h_j z + L_{ij}\right)$$

for $(x,y,z)\in[W_{i-1},W_i)\times[H_{j-1},H_j)\times[0,1)$, where $L_{ij} = W_i\times h_j + H_j$, $i=1,\ldots,k$, $j=1,\ldots,t$.

The continuous 3D baker map is then discretized, with an arbitrary cube size. Without loss of generality, assume that the cube is $W\times H\times L$, and is split into $k\times t$ blocks. The sequence of k integers, $w_1, w_2, \ldots, w_k$, is chosen such that $W_i = w_1 + w_2 + \cdots + w_i$, $W = w_1 + w_2 + \cdots + w_k$, and $W_0 = 0$. The same is carried out for the sequence of t integers, $h_1, h_2, \ldots, h_t$, namely, $H_j = h_1 + h_2 + \cdots + h_j$, $H = h_1 + h_2 + \cdots + h_t$, and $H_0 = 0$. By using

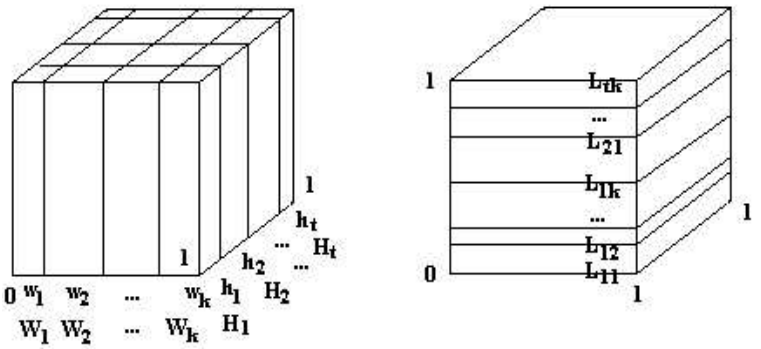

Fig. 5. The generalized 3D baker map.

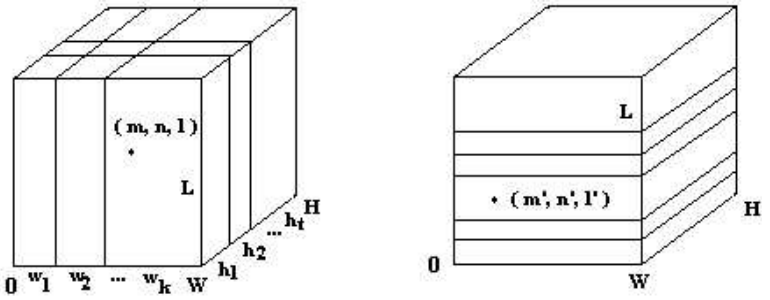

Fig. 6. The generalized discrete 3D baker map.

the formula

$$\begin{aligned} S &= (H_{j-1} \times W + W_{i-1}) \times L + w_i \times h_j \times l + (n - H_{j-1}) \times w_i + (m - W_{i-1})(m', n', l') \\ &= B_{3D}(m, n, l) \\ &= \left((S \bmod (W \times H)) \bmod W, \left\lfloor \frac{S \bmod (W \times H)}{W} \right\rfloor, \left\lfloor \frac{S}{W \times H} \right\rfloor \right) \end{aligned}$$

an arbitrary point (m, n, l) in the original cube is mapped to (m', n', l') in the new cube, as shown by Fig. 6.

3. Chaotic Image Encryption Scheme Based on 3D Baker Map

The discrete 3D baker map designed in Sec. 2 is applied here to construct a fast and secure image encryption scheme.

As discussed in [Fridrich, 1997, 1998], a secure encryption scheme should have a mechanism of diffusion that makes known-plaintext attack infeasible. In this new image encryption scheme, an XOR plus modulo (mod) operation is inserted to each pixel in between every two adjacent rounds of the map used. In the following, the diffusion process is first discussed, and then the complete encryption scheme will be described in detail.

3.1. *Diffusion procedure*

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 logistic map

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

If the next value obtained is in the interval $(0.2, 0.8)$, then go to the next step; otherwise, the iteration goes on until a desired number located in $(0.2, 0.8)$ is obtained. Here, notice that the value of 0.5 is a "bad" point, which leads the iteration to be trapped in the fixed point 0. If this happens, a small disturbance should be applied so that the iteration can continue. Once a proper value is obtained from the logistic map, digitize it by amplification using a proper scaling and sampling. The digitized value is designated as $\phi(k)$ and it is XOR-ed with the values of the currently operated pixel and the 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 operated pixel, in a vector that was strung out from an image, and $C(k)$ is the XOR-ed value. One may set the initial value to be $I(0) = S$. The inverse transform of the above is simple, which is given by

$$\begin{aligned} I(k) = \{&\phi(k) \oplus C(k) \oplus C(k - 1) \\ &+ N - \phi(k)\} \bmod N \,. \end{aligned}$$

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

3.2. *Image encryption scheme*

The integrated image encryption scheme is illustrated in Fig. 7, which consists of five steps of operations:

Step 1. *Key generation.* Select a sequence with 128 bits as the key, and split them into six groups

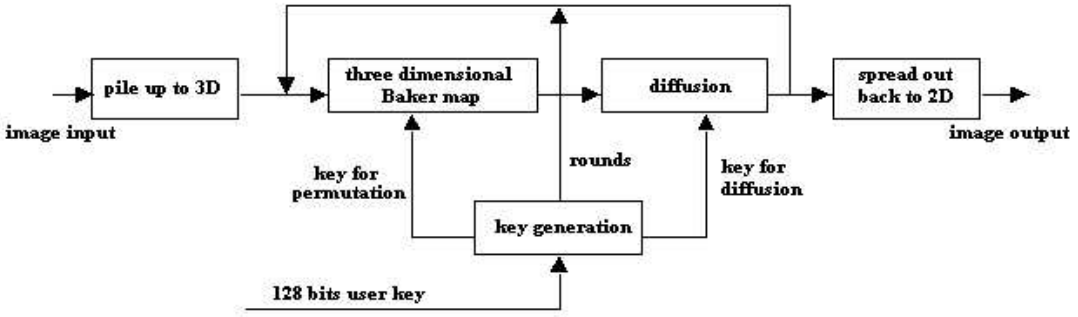

Fig. 7. Block diagram of image encryption using the 3D baker map.

among which the first four groups contain 24 bits each and the last two groups contain 16 bits each. Map these six groups of bits into six numbers, k_1, k_2, k_3, k_4, k_5, and k_6, where k_1, k_2 and k_3 are floating numbers in $(0, 1)$, while the rest are integers.

Step 2. *Pile up the two-dimensional image to three-dimensional.* Suppose that the image to be encrypted is with W pixels wide and H pixels high. First, one needs to pile up all pixels of the image to form a cube of size $M \times N \times L$. Since the number of total pixels is unchanged, the integers M, N and L must be chosen such that $M \times N \times L = W \times H$. The decomposition algorithm for M, N, L is described as follows:

(1) Set $T = W \times H$, and then factor out all prime numbers of T and list them out as a sequence, $\{p_1, p_2, \ldots, p_n\}$, such that $T = p_1 \times p_2 \cdots \times p_n \times 1$.
(2) Permute the sequence $\{p_1, p_2, \ldots, p_n, 1\}$, and then regroup them into three groups. During the permutation process, two integers are needed: one is used as the seed and the other determines the shuffle rounds. Here, k_5 and k_6 are used for these purposes, respectively.

Step 3. *Perform the three-dimensional baker map.* Select k_1 and k_2 as two initial values to perform the logistic map, respectively. After several rounds of mappings, followed by a floating to integer transformation, one can select two sequences, $\{m_1, m_2, \ldots, m_k\}$ and $\{n_1, n_2, \ldots, n_t\}$, such that $M = m_1 + m_2 + \cdots + m_k$ and $N = n_1 + n_2 + \cdots + n_t$. Then, perform the discrete 3D baker map as described in Sec. 2 on the image cube to get a shuffled image.

Step 4. *Process diffusion.* Set $k_3 = L_i$ and $k_4 = S$, and then perform the diffusion process once according to the algorithm described in the first part of this subsection.

Step 5. Transform the 3D cube back to a 2D image. Finally, in this step, the 3D cube is mapped back to a 2D image for display or storage.

Note that operations in Steps 3 and 4 are often interleaved for several rounds, which depend on the requirement of security.

To this end, the deciphering procedure is similar to that of the enciphering process illustrated above, but with the reverse operational sequences to that described in Steps 3 and 4. Since deciphering and enciphering procedures possess similar structures, they have the same algorithmic complexity and time consumption.

4. Security Analysis and Test Results

Compared with other similar encryption schemes, the new one described above has very high security and can resist many kinds of attacks known to us, such as the known-plaintext attack, ciphertext-only attack, statistical attack, differential attack, and brute-force attack, etc. Here, some security analysis results on the scheme are described, including some important ones like key space analysis, statistical analysis and differential analysis.

4.1. *Key space and sensitivity analysis*

A good encryption algorithm should be sensitive to the cipher key, and the key space should be large enough to make brute-force attack infeasible. For the proposed image encryption algorithm, designed on the generalized 3D baker map, the analysis and test results are summarized as follows:

- *Number of secret keys.* This algorithm is a 128-bit encryption scheme whose key space size is $2^{128} \approx$

3.4028×10^{38}. Since this scheme takes advantage of the 3D baker map, an opponent may try to bypass guessing the key and directly guess the possible combinations of the sequences $\{m_1, m_2, \ldots, m_k\}$ and $\{n_1, n_2, \ldots, n_t\}$, as well as the possible decomposition of M, N and L, which are used in the 3D baker map. Therefore, the combinations of the baker map control parameters should be large enough to prevent such exhaustive searching. In [Fridrich, 1998], the possible combinations of control parameters for a 2D baker map was estimated. According to a conservative estimate for an $N \times N$ image, the total number of ciphering keys is about $K(N,t) = \binom{N}{t}$, where t is the length of the key sequence $\{n_1, n_2, \ldots, n_t\}$. For a 2D image, since the key sequences of width and height are different, the size of the key space will be twice of this estimate. If each ciphering round of the baker map uses different ciphering keys, then the increase of round numbers will also enlarge the key space. Compared with the 2D baker map, the key space of the 3D one is further enlarged, for the key space of the 2D map is just a subspace of the 3D one. For example, suppose that an image size is $W \times H$. In order to perform the 3D baker map, the image must be piled up to a cube of size $M \times N \times L$ such that $W \times H = M \times N \times L$. Among all possible decompositions, $W \times H \times 1$ is a special case that reduces the 3D map to a 2D one. Therefore, one can conclude that the 3D baker map has a much larger key space than that of the 2D one.

- *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 is 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 "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 99.59% differences from the image encrypted by the key "1234567890123457" in terms of pixel gray-scale values, although there is only one bit difference in the two keys. Figure 8 shows the test result. Moreover, if a 16-character key is used to encrypt an image while another trivially modified key is used to decrypt the ciphered image, then the decryption should not succeed. Figure 9 has

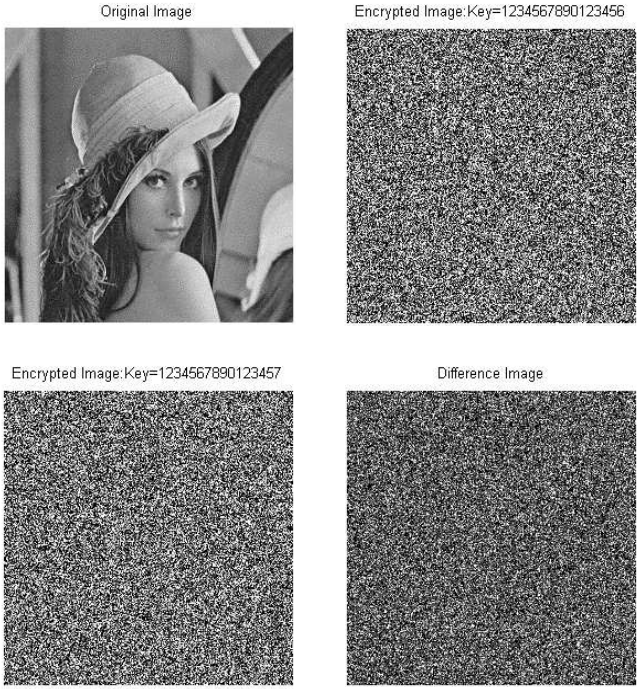

Fig. 8. Key sensitive test: result 1.

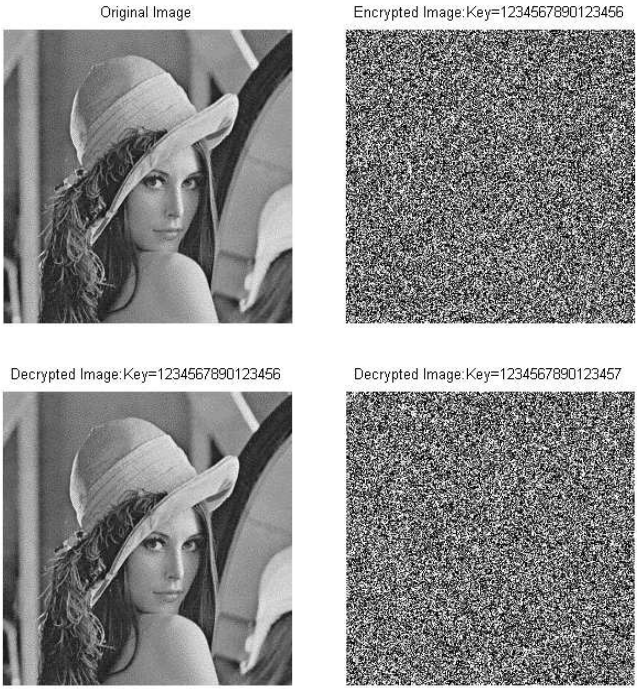

Fig. 9. Key sensitive test: result 2.

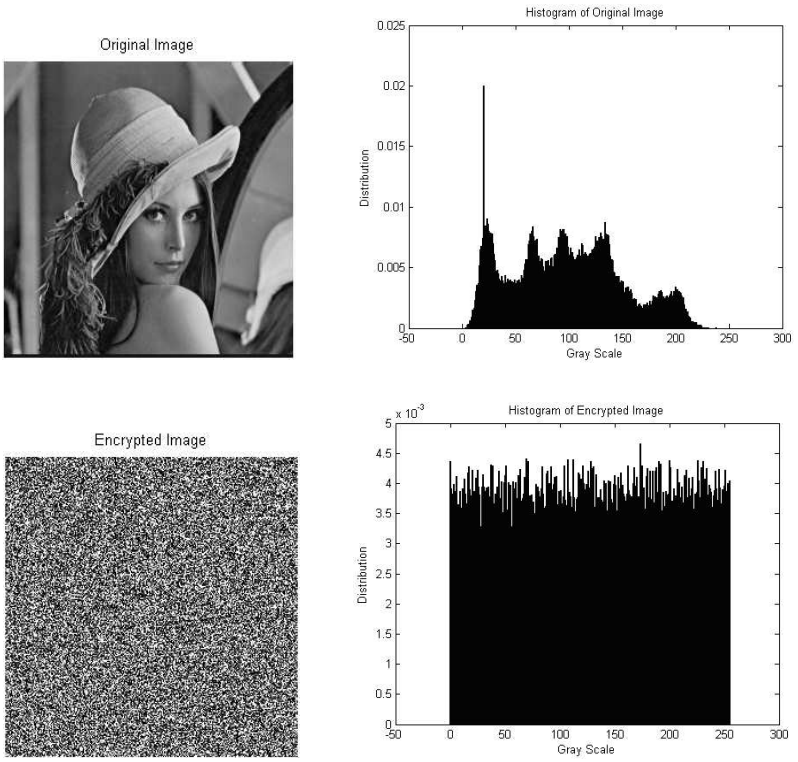

Fig. 10. Histograms of the plain image and the ciphered image.

verified this, where the image encrypted by the key "1234567890123456" was not be correctly decrypted by using the key "1234567890123457." Here, there is also only one bit difference between the two keys.

4.2. *Statistical analysis*

In his masterpiece, Shannon [1949] said, "It is possible to solve many kinds of ciphers by statistical analysis," and, therefore, he suggested two methods of diffusion and confusion for the purpose of frustrating the powerful statistical analysis.

Here, it is demonstrated that the new image encryption scheme, designed on the generalized 3D baker map, has very good confusion and diffusion properties. This is shown by a test on the histograms of the ciphered images and on the correlations of adjacent pixels in the ciphered image.

1. *Histograms of ciphered images.* Select several 256 gray-scale images with size of 512×512 that have different contents, and calculate their histograms. One typical example among them is shown in Fig. 10. 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 in a ciphered image, respectively, the procedure is performed as follows: First, randomly select 1000 pairs of two adjacent pixels from an image. Then, calculate their correlation coefficient 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 gray-scale values of two adjacent pixels in the image. In numerical computation, the following discrete formulas were used:

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

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

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

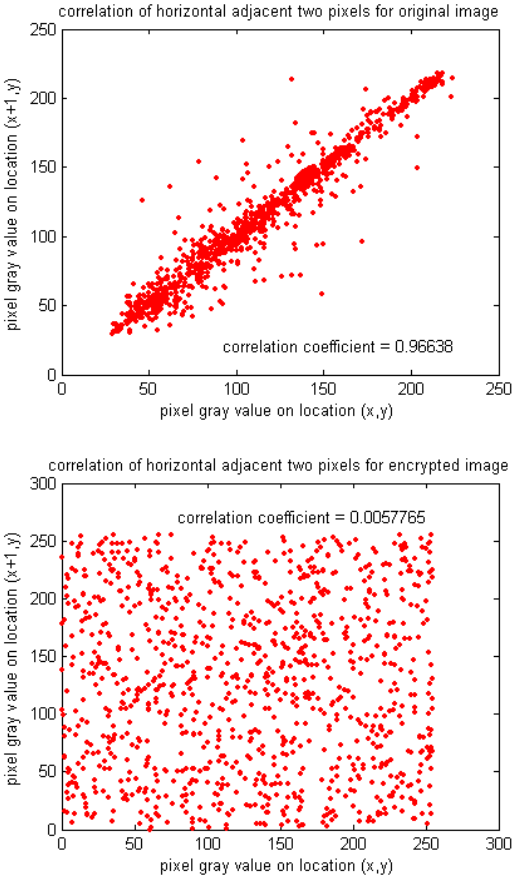

Fig. 11. Correlations of two horizontally adjacent pixels in the plain image and in the ciphered image.

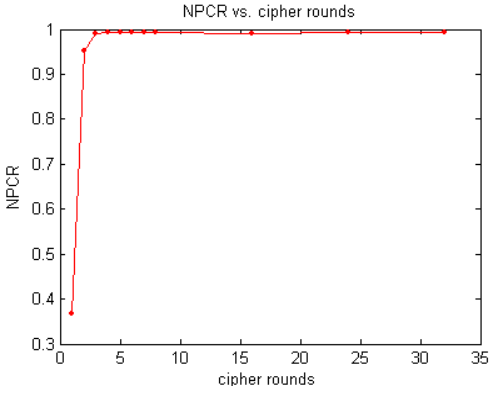

Fig. 12. NPCR versus ciphering rounds.

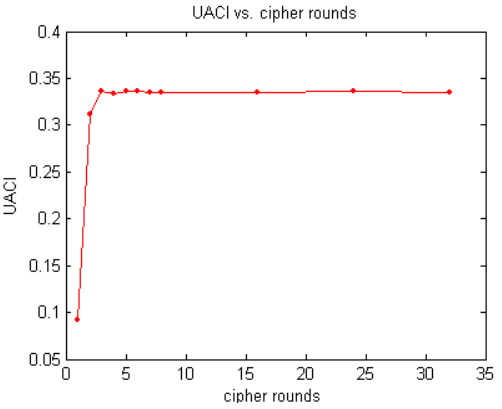

Fig. 13. UACI versus ciphering rounds.

Table 1. Correlation coefficients of two adjacent pixels in two images.

	Plain-Image	Ciphered-Image
horizontal	0.97653	0.04454
vertical	0.97961	0.02843
diagonal	0.95025	0.02066

Figure 11 shows the correlations of two horizontally adjacent pixels in the plain-image and that in the ciphered-image: the correlation coefficients are 0.97653 and 0.044535, respectively. Similar results for diagonal and vertical directions were obtained and are shown in Table 1.

4.3. *Some other analysis*

Usually, an opponent would make a slight change (e.g. modify only one pixel) of the encrypted image so as to observe the change of the corresponding result. In doing so, he might be able to find out a meaningful relationship between the plain-image and the iphered-image. If one minor change in the plain-image can cause a significant change in the ciphered-image, with respect to both diffusion and confusion, then this "differential attack" may become inefficient.

To test the influence of one-pixel change on the whole image, encrypted by the proposed chaos-based algorithm, two common measures may be used: *Number of Pixels Change Rate* (*NPCR*) and *Unified Average Changing Intensity* (*UACI*). Let two ciphered-images, whose corresponding plain-images have only one pixel difference, be denoted by C_1 and C_2. Label the gray-scale values of the pixels at grid (i, j) in 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 images C_1 and 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

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

Table 2. Comparison of ciphering speed between the 2D baker map and the 3D baker map schemes.

Image Size (in pixels)	Colors	2D Baker Map (in seconds)	3D Baker Map (in seconds)
256×256	2	< 0.3	< 0.3
256×256	16	< 0.3	< 0.3
256×256	256	< 0.3	< 0.3
256×256	16777216	< 0.3	< 0.3
512×512	2	1	< 0.3
512×512	16	1	< 0.3
512×512	256	1.1	< 0.3
512×512	16777216	1	< 0.3
1024×1024	2	3.3	1.0
1024×1024	16	3.3	1.1
1024×1024	256	3.3	1.2
1024×1024	16777216	3.3	1.3
2048×2048	2	13.6	3.4
2048×2048	16	13.5	3.2
2048×2048	256	14.0	3.4
2048×2048	16777216	13.6	4.3

Test Conditions:
(1) The configuration of the computer used in this test is Pentium IV 1G CPU with 256M memory and 40G hard disk capacity.
(2) Theoretically, both algorithms are symmetric, i.e. both encipher and decipher procedures have the same complexity. But, due to the programming realization issue, the decipher procedure may consume a little more time than enciphering. The time recorded in Table 2 is the average time of the encipher and decipher procedures.

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

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

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

A performed test is on the one-pixel change influence on a 256 gray-scale image of size 512×512. The test results are shown in Figs. 12 and 13. 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 so as to achieve higher security; yet, this is at the expense of processing time.

5. Other Test Results

Apart from the security consideration, some other issues on an image encryption scheme are also important. These include running speed, ability of surviving from image compression, and so on.

5.1. *Enciphering/deciphering speeds*

The proposed image encryption algorithm is quite fast. Simulation shows that the average enciphering/deciphering speed is 1.2 MB/sec., and the peak speed can reach up to 2.8 MB/sec., on a 1 GHz Pentium IV computer. The designed cipher based on the 2D baker map is different from that suggested in [Fridrich, 1998] on the diffusion operation. Taking into account the improvement in the computer technology, the speeds of these two ciphers in implementation are about the same. The encryption rate of the algorithm of [Fridrich, 1998] is about 1 Mb with an unoptimized C code on a 60 MHz Pentium computer.

5.2. *Tolerance of image processing*

Variation tolerance of image processing operations, including noise addition, JPEG compression, and

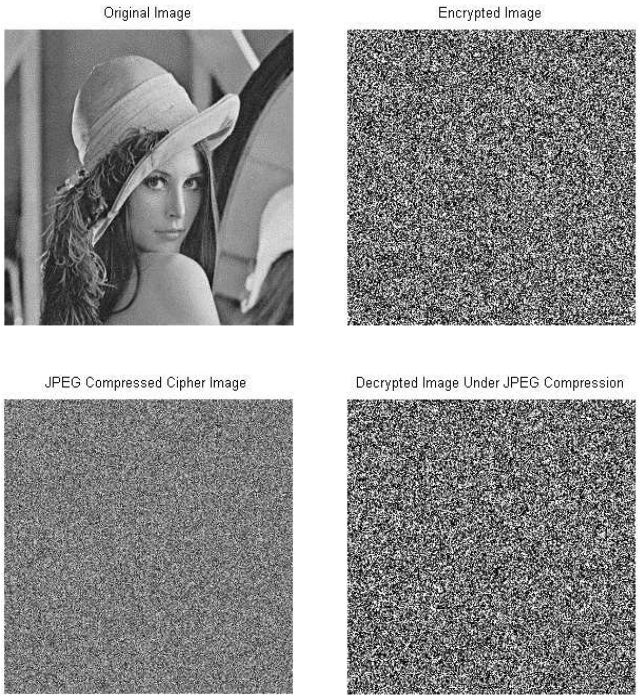

Fig. 14. Test on image encryption with JPEG compression.

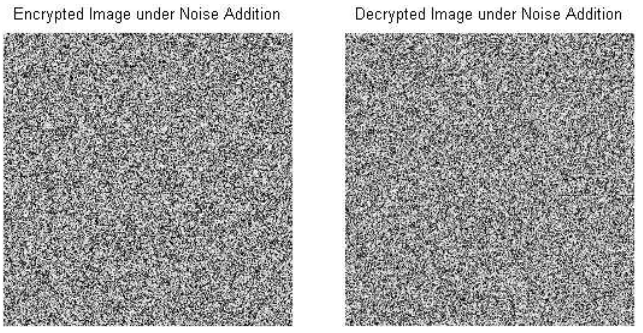

Fig. 15. Test of noise addition.

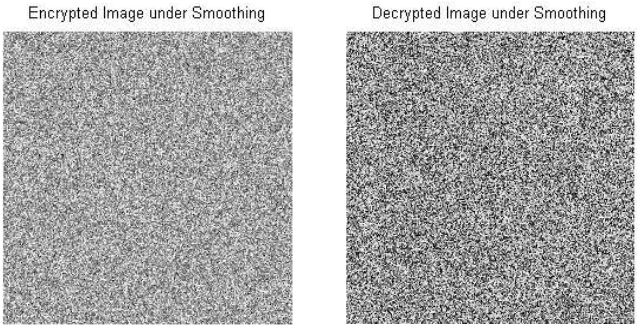

Fig. 16. Test of image under smoothing.

image smoothing, has been performed on the proposed encryption scheme. Test results show that, due to the sensitivity to ciphered-image of the proposed algorithm, all the image operations significantly affect the decryption process. In other words, the proposed image encryption scheme can only be used in an error-free scenario.

5.2.1. *Tolerance of JPEG compression*

In testing the tolerance of JPEG compression, the results show that due to the sensitivity of the proposed chaos-based scheme to ciphered-images, JPEG compression significantly affects the decryption process. In other words, the designed image encryption scheme can only be used in an error-free scenario. An encrypted image, if being JPEG compressed and then transferred, cannot be decrypted correctly at the receiver side. A test is shown in Fig. 14, where the quality factor used by the JPEG compression is 65. Here, quality factor is a kind of measure for JPEG compression, commonly within a range between 1 to 100: the bigger the factor, the better the quality of the image after compression and, correspondingly, the smaller the compression rate. The PSNR of the JPEG compressed image is 12.39 dB.

5.2.2. *Noise addition*

Figure 15 shows the result of a decrypted image under additions of Gaussian noise. Here, the PSNR of the noise contaminated image is 15.08 dB.

5.2.3. *Smoothing*

Image smoothing also greatly affects decrypted result, as shown in Fig. 16, where the PSNR = 15.82 dB.

6. Conclusions

In this paper, the two-dimensional baker map has been extended to three-dimensional, and an image encryption scheme based on this three-dimensional map is proposed. Comparing with existing similar schemes that were designed on the two-dimensional baker map, the new scheme has higher security and faster enciphering/deciphering speeds. This makes it a very good candidate for real-time image encryption applications. Experiments and analysis have both demonstrated the feasibility and efficiency of the new algorithm.

References

Chen, G., Mao, Y. B. & Chui, C. K. [2004] "A symmetric image encryption scheme based on 3D chaotic cat maps," *Chaos Solit. Fract.* **21**, 749–761.

Fridrich, J. [1997] *Secure Image Ciphering Based on Chaos: Final Report for AFRL*, Rome, New York, USA.

Fridrich, J. [1998] "Symmetric ciphers based on two-dimensional chaotic maps," *Int. J. Bifurcation and Chaos* **8**, 1259–1284.

Kocarev, L. [2001] "Chaos-based cryptography: A brief overview," *IEEE Circuits Syst. Mag.* **1**, 6–21.

Kocarev, L. & Jakimovski, G. [2001] "Chaos and cryptography: From chaotic maps to encryption algorithms," *IEEE Trans. Circuits Syst.-I* **48**, 163–169.

Li, S. J., Zheng, X., Mou, X. & Cai, Y. [2002] "Chaotic encryption scheme for real-time digital video," *Real-Time Imaging VI, Proc. SPIE*, Vol. 4666, San Jose CA USA, pp. 149–160.

Mao, Y. B. & Chen, G. [2004] "Chaos-based image encryption," *Handbook of Computational Geometry for Pattern Recognition, Computer Vision, Neurocomputing and Robotics* (Springer-Verlag, NY), in press.

Masuda, N. & Aihara, K. [2002] "Cryptosystems with discretized chaotic maps," *IEEE Trans. Circuits Syst.-I* **49**, 28–40.

Pichler, F. & Scharinger, J. [1995] "Ciphering by Bernoulli-shifts in finite Abelian groups," in *Contributions to General Algebra*, Vol. 9, eds. Kaiser, H. K., Muller, W. B. & Pilz. G. F., pp. 249–256.

Pichler, F. & Scharinger, J. [1996] "Ciphering by Bernoulli shifts in finite Abelian groups," in *Contributions to General Algebra: Proc. Linz-Conf.*, pp. 465–476.

Scharinger, J. [1998] "Fast encryption of image data using chaotic Kolmogorov flows," *J. Elect. Imag.* **7**, 318–325.

Schneier, B. [1995] *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, 2nd edition (Wiley, NY).

Shannon, C. E. [1949] "Communication theory of secrecy system," *Bell Syst. Techn. J.* **28**, 656–715.