Journal of Information Security, 2011, 2, 139-150
doi:10.4236/jis.2011.24014 Published Online October 2011 (http://www.SciRP.org/journal/jis)

New Approach for Fast Color Image Encryption Using Chaotic Map

Kamlesh Gupta[1*], Sanjay Silakari[2]
[1]*Department of Computer Science*, *Jaypee University of Engineering and Technology*, *Guna*, *India*
[2]*Department of Computer Science*, *Rajiv Gandhi Technical University*, *Bhopal*, *India*
E-mail: {*Kamlesh_rjitbsf, ssilakari}@yahoo.com
Received June 6, 2011; *revised July* 5, 2011; *accepted July* 16, 2011

Abstract

Image encryption using chaotic maps has been established a great way. The study shows that a number of functional architecture has already been proposed that utilize the process of diffusion and confusion. However, permutation and diffusion are considered as two separate stages, both requiring image-scanning to obtain pixel values. If these two stages are mutual, the duplicate scanning effort can be minimized and the encryption can be accelerated. This paper presents a technique which replaces the traditional preprocessing complex system and utilizes the basic operations like confusion, diffusion which provide same or better encryption using cascading of 3D standard and 3D cat map. We generate diffusion template using 3D standard map and rotate image by using vertically and horizontally red and green plane of the input image. We then shuffle the red, green, and blue plane by using 3D Cat map and standard map. Finally the Image is encrypted by performing XOR operation on the shuffled image and diffusion template. Theoretical analyses and computer simulations on the basis of Key space Analysis, statistical analysis, histogram analysis, Information entropy analysis, Correlation Analysis and Differential Analysis confirm that the new algorithm that minimizes the possibility of brute force attack for decryption and very fast for practical image encryption.

Keywords: Chaotic Map, 3D Cat Map, Standard Map, Confusion and Diffusion

1. Introduction

With the fast development of image transmission through computer networks especially the Internet, medical imaging and military message communication, the security of digital images has become a most important concern. Image encryption, is urgently needed but it is a challenging task because it is quite different from text encryption due to some intrinsic properties of images such as huge data capacity and high redundancy, which are generally difficult to handle by using conventional techniques. Nevertheless, many new image encryption schemes have been suggested in current years, among which the chaos-based approach appears to be a hopeful direction.

General permutation-diffusion architecture for chaos-based image encryption was employed in [1,2] as illustrated in **Figure 1**. In the permutation stage, the image pixels are relocated but their values stay unchanged. In the diffusion stage, the pixel values are modified so that a minute change in one-pixel spreads out to as many pixels as possible. Permutation and diffusion are two different and iterative stages, and they both require scanning the image in order to gain the pixel values. Thus, in the encryption process, each round of the permutation-diffusion operation requires at least twice scanning the same image.

In this paper, we generate diffusion template using 3D standard map and rotated image by using vertically and horizontally red and green plane of the input image. We then shuffle the red, green, and blue plane by using 3D

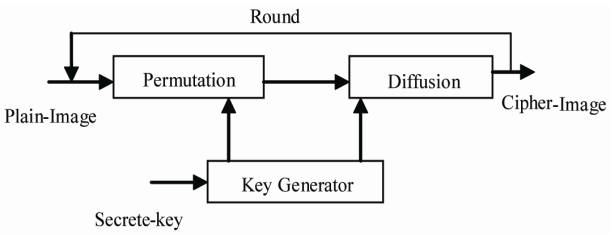


Figure 1. Permutation and diffusion based image cryptosystem.

Cat map and standard map. Finally the Image is encrypted by performing XOR operation on the shuffled image and diffusion template. The objectives of this new design includes: 1) to efficiently extract good pseudorandom sequences from a cascading of 3D cat and standard map for color image and 2) to simultaneously perform permutation and diffusion operations for fast encryption.

The rest of this paper is organized as follows: Section 2 focuses on the efficient generation of pseudorandom sequences. In Section 3, proposed algorithm is described in detail. Section 4 presents simulation results and performance analyses. In Section 5, conclusions and future work.

2. Efficient Generation of Pseudorandom Sequences

The generation of pseudorandom is based on two cascaded chaotic maps behave as a single chaotic map in present case. The 3D cat map & 3D standard map are taken for encryption. The pseudorandom matrix generated by this method is given below. (The explanation for pseudorandom sequences generation is given in Section 3).

3. Proposed Algorithm

The proposed algorithm are divided into several stages and explained below.

Table 1. Generation of pseudorandom values by proposed method.

'FC'	'EE'	'C4'	'D1'	'E6'	'D3'	'E8'	'E9'	'EA'	'CB'	'EA'	'ED'
'D7'	'EC'	'D9'	'EE'	'5'	'FF'	'16'	'17'	'9'	'EF'	'FF'	'1B'
'26'	'9'	'A'	'E2'	'FA'	'2'	'FC'	'F5'	'1C'	'2F'	'1E'	'1F'
'2D'	'33'	'10'	'26'	'0'	'28'	'11'	'2A'	'4'	'17'	'15'	'F8'
'19'	'38'	'15'	'2B'	'3F'	'1E'	'19'	'41'	'2A'	'1C'	'1D'	'45'
'3C'	'40'	'10'	'33'	'47'	'44'	'58'	'37'	'5A'	'35'	'5C'	'3F'
'5F'	'41'	'51'	'5E'	'31'	'54'	'24'	'61'	'35'	'27'	'28'	'47'
'4D'	'6C'	'4F'	'6E'	'45'	'42'	'56'	'71'	'3A'	'59'	'3C'	'5B'
'63'	'89'	'74'	'6D'	'49'	'6F'	'69'	'61'	'6B'	'5D'	'6D'	'6E'
'8F'	'62'	'7B'	'64'	'89'	'66'	'7F'	'8C'	'53'	'54'	'64'	'9F'
'80'	'72'	'A6'	'74'	'99'	'67'	'7D'	'7E'	'9D'	'AD'	'80'	'81'
'87'	'97'	'89'	'99'	'9A'	'9B'	'9C'	'BB'	'70'	'AE'	'72'	'73'
'A2'	'A3'	'BF'	'A5'	'B5'	'A7'	'C2'	'9A'	'A6'	'C5'	'A8'	'9E'
'9D'	'BC'	'B2'	'90'	'AC'	'A1'	'AE'	'AF'	'A0'	'C8'	'C9'	'A3'
'CE'	'CF'	'DF'	'C1'	'BC'	'C3'	'E2'	'CE'	'CF'	'E5'	'E6'	'D1'
'E7'	'F7'	'CE'	'DB'	'D8'	'DD'	'DA'	'BD'	'DC'	'C5'	'E4'	'B1'

3.1. Diffusion Template

According to the proposal the diffusion template must have the same size as main image. Let the main image have m number of rows n number of columns then the diffusion template is created as follows

$$(i, j, k) = \text{round}\left(\frac{255}{n} \times j\right) \tag{1}$$

where $1 \le i \le m$, $1 \le j \le n$ and $1 \le k \le 3$.

Equation (1) form the matrix with all rows filled with linearly spaced number in between 0 to 255. The sequence is randomized by 3D standard map in discrete form as given below.

The 3D standard map randomizes the pixels by reallocating it in new position by utilizing its property of one to one mapping. **Figure 2** shows the final diffusion template by using 3D standard map.

$$i' = (i + j) \bmod m \tag{2}$$

$$j' = \left[j + k + K1 \times \sin\left(i \times \frac{c}{2 \times pi} \right) \right] \bmod n \tag{3}$$

$$k' \left[k + K1 \times \sin\left(i \times \frac{p}{2 \times pi} \right) + K2 \times \sin\left(j \times \frac{p}{2 \times pi} \right) \right] \bmod p \tag{4}$$

where the $K1$, $K2$ are the integers, p = 3 for the case of color image and i', j', k' shoes the transformed location of i, j, k.

$$I'_{\text{diff}}(i', j', k') = I'_{\text{diff}}(i, j, k).$$

3.2. Image Encryption

Step 1. The main image is divided into three separate images I_R, I_G and I_B as follows

$$I_R(x, y) = I(x, y, 1) \quad I_G(x, y) = I(x, y, 2)$$
$$I_B(x, y) = I(x, y, 3)$$

where $1 \le x \le m$ and $1 \le y \le n$.

Step 2. The Red and Green image are transform verti-

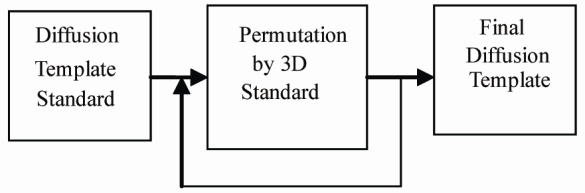


Figure 2. Diffusion template.

cally and horizontally respectively. The blue image remains same and reconstructs the new image.

$$I'_R(x,y) = I_R\left[\left(x+\frac{m}{2}\right)\right] \bmod m, y$$

$$I'_G(x,y) = I_R\left[x\left(y+\frac{n}{2}\right)\right] \bmod n$$

$$I'_B(x,y) = I_B(x,y) \quad I_{\text{new}}(x,y,1) = I'_R(x,y)$$

$$I_{\text{new}}(x,y,2) = I'_G(x,y) \quad I_{\text{new}}(x,y,3) = I'_B(x,y).$$

Step 3. Perform the first level confusion by using 2D cat map. Slice the plane normal to *R*, *G*, *B* Planes by

$$I'_{\text{new}}(i,j,k) = I'_{\text{SRGB}}(j',k') = I_{\text{new}}(i,j,k)$$

where $1 \le i \le m$, $1 \le j \le n$ and $1 \le k \le 3$.

$$j' = \left(j + r_x + r_y + p \times k\right) \bmod m$$

$$k' = \left(q \times j + r_y + (p \times q + 1) \times k\right) \bmod n$$

where j' and k' are obtained by 2D cat map and p and q are integer > 0 and r_x, r_y are offset integer such that $0 \le r_x \le m$, $0 \le r_y \le n$.

Step 4. Generate Final confusion stage by two cascade 3D maps first by cat map then by standard map. So the transformation of location (i, j, k) into (i'', j'', k'') is performed by following equations.

$$i' = \left[\left(1 + a_x a_z b_y\right) \times i + a_z \times j + \left(a_y + a_x \times a_z + a_x \times a_y \times a_z \times b_y\right) \times k\right] \bmod m$$

$$i' = \left[\left(b_z + a_x \times b_y + a_x \times a_z \times b_y \times b_z\right) \times i + \left(a_z \times b_z + 1\right) \times j + \left(a_y \times a_z + a_x \times a_y \times a_z \times b_y \times b_z + a_x \times a_z \times b_y + a_y \times b_y + 1\right) \times k\right] \bmod n$$

$$k' = \left[\left(a_x \times b_x \times b_y + b_y\right) \times i + b_x \times j + \left(a_x \times a_y \times b_x \times b_y + a_x \times b_x + a_y \times b_y + 1\right) \times k\right] \bmod p$$

$$i'' = \left[\left(i' + k'\right)\right] \bmod m$$

$$j'' = \left[\left(i' + j'\right) + K1 \times \sin\left(i'' \times \frac{n}{2 \times pi}\right)\right] \bmod n$$

$$k'' = \left[k' + K1 \times \sin\left(i'' \times \frac{p}{2 \times pi}\right) + K2 \times \sin\left(j + \frac{p}{2 \times pi}\right)\right] \bmod p$$

$$I_{\text{con}} f\left(i'', j'', k''\right) = I'(i,j,k)$$

where a_x, a_y, a_z, b_x, b_y, b_z and $K1$, $K2$ are integers >0.

Each confusion step is followed by diffusion obtained by EXOR operations performed between each pixels of I_{conf} and diffusion I_{diff}. The proposed image encryption architecture is given in **Figure 3**.

$$I_{\text{encp}} = I_{\text{conf}} \oplus I_{\text{diff}}.$$

3.3. Key Generation Process

The proposed method has a large number of variables which can be used as key parameters but to avoid the exceptionally large key and decreased key sensitivity, the parameter which does not having great affects on encryption are avoid or scaled. The selected key parameters and their length are given below

Step 1. Diffusion template shuffling $\mathbf{D_s}$ = 8 bits.

Step 2. Diffusion template offset value $\mathbf{D_xD_yD_z}$ = 8 + 8 + 2 = 18 bits.

Step 3. Diffusion template variables $\mathbf{D_{k1}D_{k2}}$ = 8 + 8 = 16 bits.

Step 4. Sliced RGB plane Shuffling = $\mathbf{S_s}$ = 8 bits.

Step 5. Sliced RGB plane offset values $\mathbf{S_xS_y}$ = 8 bits.

Step 6. Sliced RGB Plane Variables $\mathbf{S_pS_q}$ = 8 + 8 = 16 bits.

Step 7. Final Confusion shuffling $\mathbf{C_s}$ = 8 bits.

Step 8. Confusion offset of cat map $\mathbf{C_xC_yC_z}$ = 8 + 8 + 2 = 18 bits.

Step 9. Confusion cat map variables $\mathbf{C_aC_b}$ = 8 + 8 = 16 bits.

Step 10. Confusion offset of standard map $\mathbf{C_x'C_y'C_z'}$ = 8 + 8 + 2 = 18 bits.

Step 11. Confusion standard map variables $\mathbf{C_{k1}C_{k2}}$ = 8 + 8 = 16 bits.

Final key structure

$$\mathbf{D_sD_xD_yD_zD_{k1}D_{k2}S_sS_xS_yS_pS_q}$$
$$\mathbf{C_sC_xC_yC_zC_aC_bC_x'C_y'C_z'C_{k1}C_{k2}}$$

Total bits = 8 + 18 + 16 + 8 + 8 +16 + 8 + 16 + 16 + 18 + 16 = 148 bits.

3.4. Image Decryption

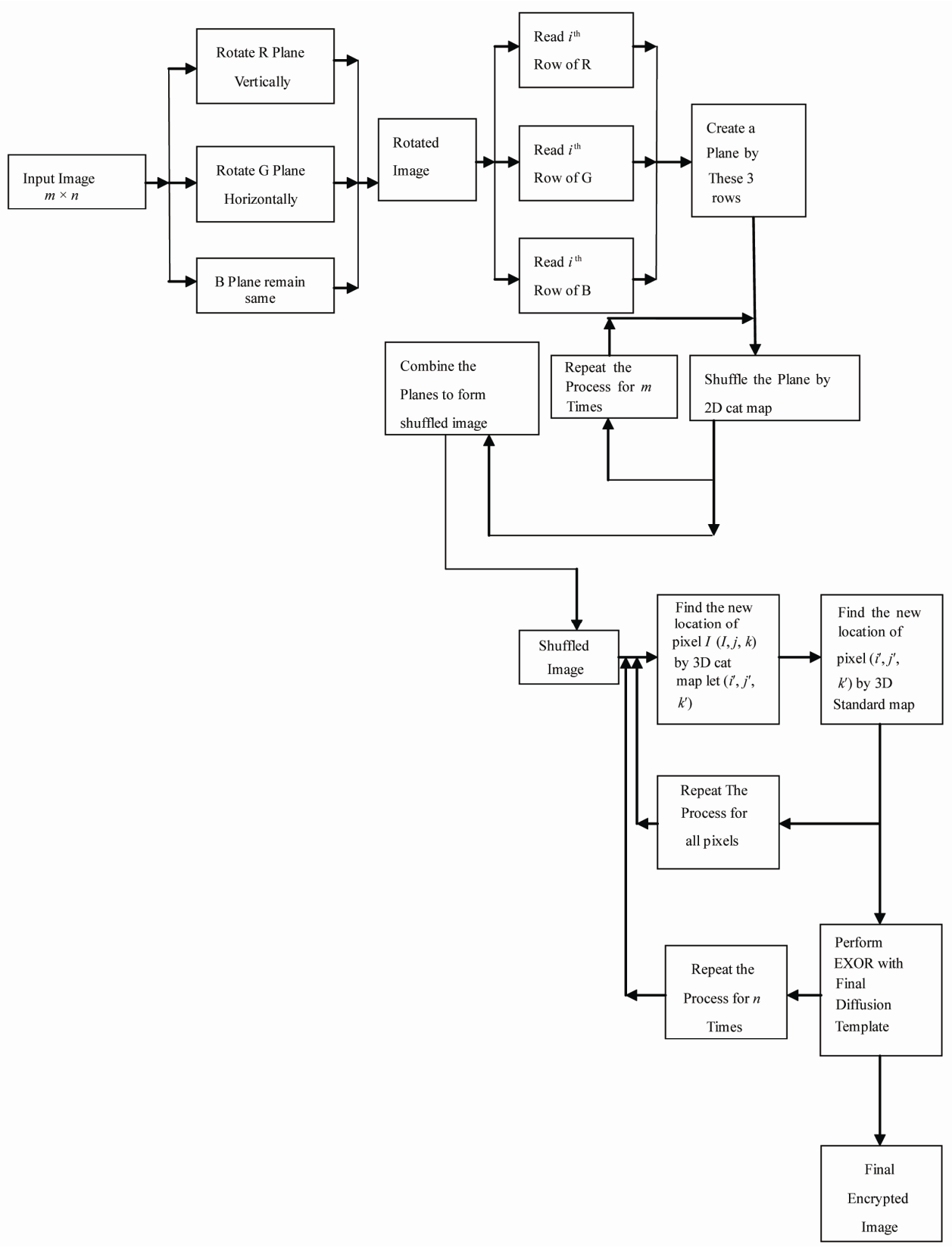
Step 1. Generate the diffusion template in same way as in encryption section.

Step 2. Re-transformation of location is done by two cascaded 3D maps firstly by standard map then by cat map.

So the re-transformation of location (i'', j'', k'') into (i, j, k) is performed by following equations

$$i' = \left[\left(i'' + k''\right)\right] \bmod m$$

$$j' = \left[i'' + j'' + k1 \times \sin\left(i' \times \frac{n}{2 \times pi}\right)\right] \bmod n$$



Figuer 3. Image encryption by using confusion and diffusion.

$$k' = \left[k'' + k1 \times \sin\left(j' \times \frac{n}{2 \times pi} \right) + k2 \times \left(j' \times \frac{n}{2 \times pi} \right) \right] \bmod p$$

$$i = \left[\left(1 + a_x a_z b_y\right) \times i' + a_z \times j' + \left(a_y + a_x \times a_z + a_x \times a_y \times a_z \times b_y\right) \times k' \right] \bmod m$$

$$i = \left[\left(b_z + a_x \times b_y + a_x \times a_z \times b_y \times b_z\right) \times i' + \left(a_z \times b_z + 1\right) \times j' + \left(a_y \times a_z + a_x \times a_y \times a_z \times b_y \times b_z + a_x \times a_y \times b_y + a_x\right) \times k' \right] \bmod n$$

$$k = \left[\left(a_x \times b_x \times b_y + b_y\right) \times i' + b_x \times j' + \left(a_x \times a_y \times b_x \times b_y + a_x \times b_x + a_y \times b_y + 1\right) \times k' \right] \bmod p$$

$$I_{\text{retransf}}(i, j, k) = I_{\text{encp}}(i, j, k)$$

where a_x, a_y, a_z, b_x, b_y, b_z and $K1$, $K2$ are integers > 0.

Each confusion step is followed by diffusion obtained by EXOR operations performed between each pixels of I_{retrnsf} and diffusion I_{diff}

$$I'_{\text{dencp}} = I_{\text{retrnsf}} \oplus I_{\text{diff}} .$$

Step 3. Performing inverse of First level confusion.

Slicing the plane normal to *R*, *G*, *B* Planes

$$I'_{\text{SRGB}}(j,k) = I'_{\text{dencp}}(i, j, k)$$

for each value of *i*, *j* changed from 0 to *m*, *k* changed from 0 to 3.

De-shuffling the sliced plane

$$I_{\text{DRGB}}(j,k) = I'_{\text{SRGB}}(j,k)$$

where *j*′ and *k*′ are obtained by 2D cat map given below

$$j' = \left(j + r_x + r_y + p \times k\right) \bmod m$$

$$k' = \left(q \times j + r_y + \left(p \times q + 1\right) \times k\right) \bmod n$$

where *p* and *q* are integers > 0, and r_x, r_y are offset integers such that $0 \le r_x \le$ m and $0 \le r_y \le n$.

Recombining the planes for forming 3D matrix for next operation

$$I'(i, j, k) = I'_{\text{DRGB}}(j,k) .$$

Step 4. Re-rotating the image planes

Dividing main image into three separate images I_R, I_G and I_B as follows

$$I_R(x,y) = I'(x,y,1) \quad I_g(x,y) = I'(x,y,2)$$

$$I_B(x,y) = I'(x,y,3)$$

where $1 \le x \le m$ and $1 \le y \le n$.

Scrolling the red plane vertically

$$I_R(x,y) = I_R\left[\left(x + \frac{m}{2}\right) \bmod m, y\right].$$

Scrolling the green plane horizontally

$$I_G(x,y) = I_R\left[x\left(y + \frac{n}{2}\right) \bmod n\right].$$

Blue plane remain intact.

$$I_B(x,y) = I_B(x,y) .$$

Step 5. Next recombination of planes are performed to form final decrypted image

$$I_{final}(x,y,1) = I_R(x,y) \quad I_{final}(x,y,2) = I_G(x,y)$$

$$I_{final}(x,y,3) = I_B(x,y)$$

4. Performance Analysis

4.1. Key Space Analysis

The strong point of the proposed algorithm is the generation of the permutation sequence with the chaos sequence. The key space should also be suitably large to make brute-force attack not feasible. In the proposed algorithm, we use 148 bit key (37 Hex number) is used. It has been observed in **Figures 4(a)** and **(b)** that with slightly varying the initial condition of the chaotic sequence. It has been almost impossible to decrypt the image.

4.2. Statistical Analysis

It is well known that passing the statistical analysis on cipher-text is of crucial importance for a cryptosystem actually, an ideal cipher should be strong against any statistical attack. In order to prove the security of the proposed image encryption scheme, the following Statistical tests are performed.

4.2.1. Histogram Analysis

To prevent the access of information to attackers, it is important to ensure that encrypted and original images do not have any statistical similarities. The histogram analysis clarifies that, how the pixel values of image are distributed. A number of images are encrypted by the encryption schemes under study and visual test is performed.

An example is shown in **Figure 5**. In **Figure 5** shows histogram analysis on test image using proposed algorithm. The histogram of original image contains great sharp rises followed by sharp declines as shown in **Figure 5** and the histograms of the encrypted images for different round as shown in **Figures 5(a)-(f)** have uniform distribution which is significantly different from original image and has no statistical similarity in ap-

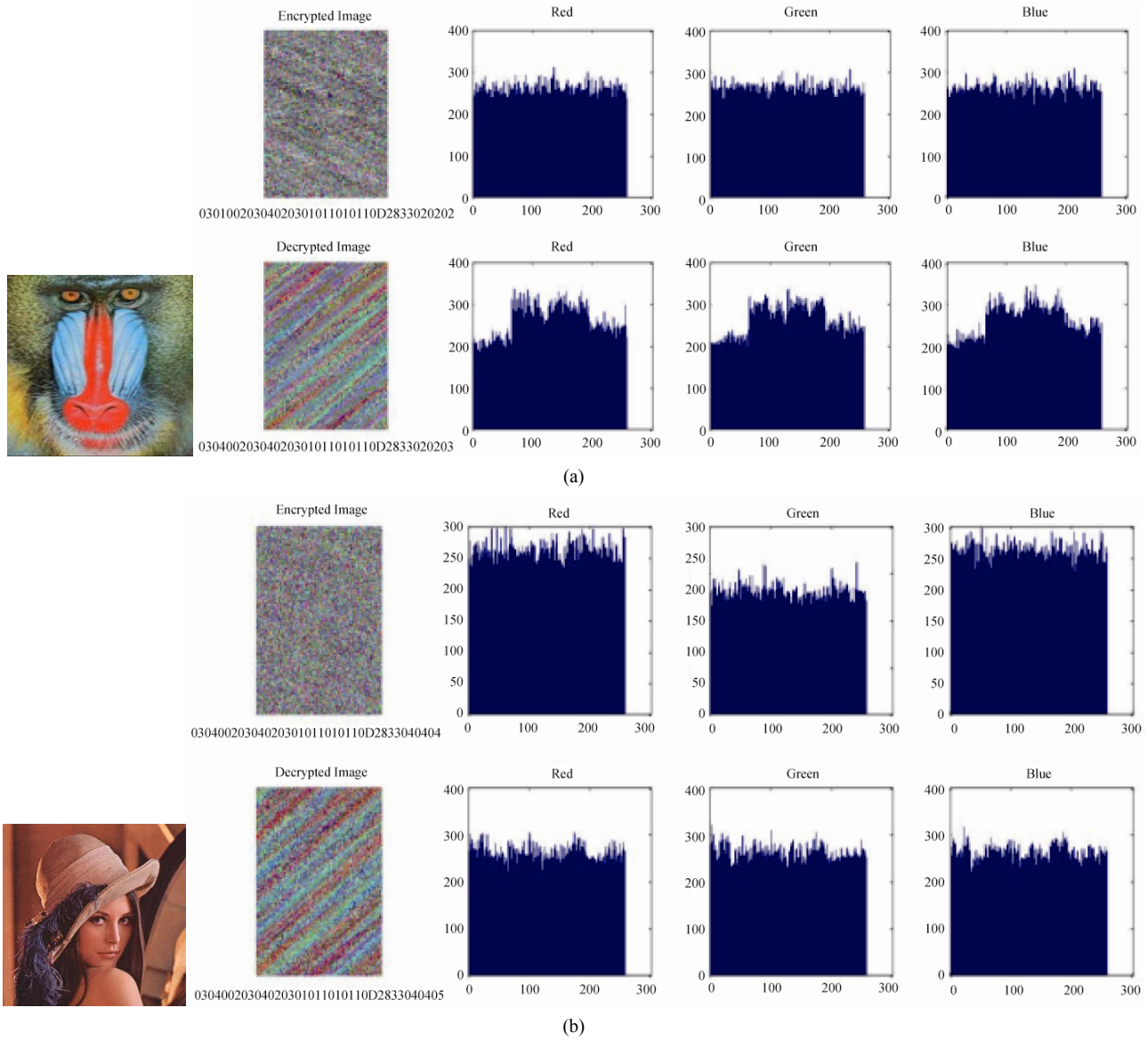


Figure 4. (a) Input image encrypted with 0304002030402 0301011010110D2833020202 and Decrypted by 0304002030402030-1011010110D2833020203; (b) Input lenna image encrypted with 0304002030402030101101011 0D2833040404 and Decrypted by 0304002030402030101101011 0D2833040405.

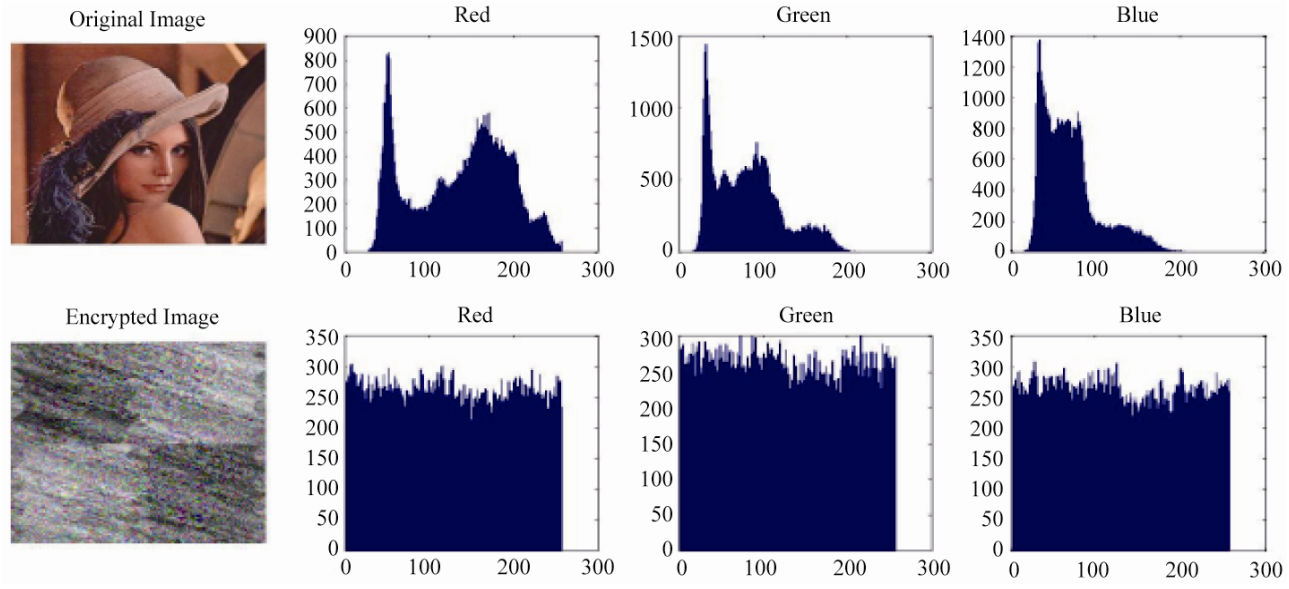
pearance. So, the surveyed algorithms do not provide any clue for statistical attack. The encrypted image histogram, approximated by a uniform distribution, is quite different from plain-image histogram.

4.2.2. Correlation Analysis

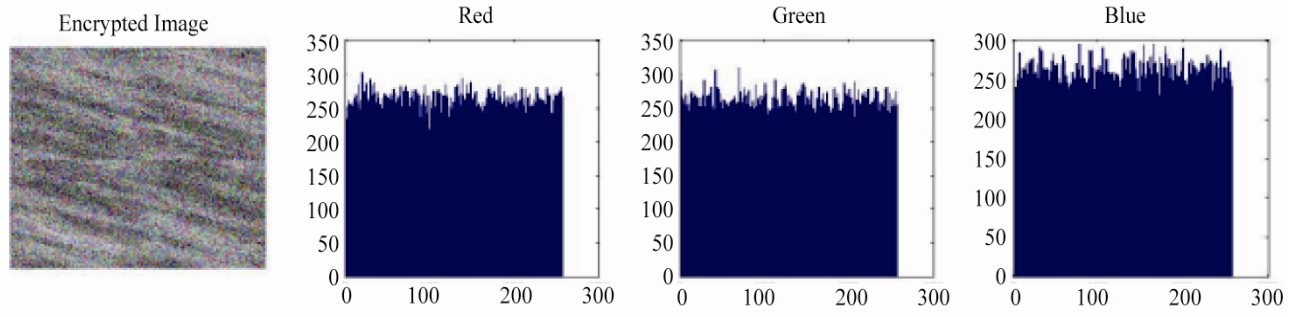
There is a very good correlation among adjacent pixels in the digital image [3]. Equation (5) is used to study the Correlation between two adjacent pixels in horizontal, vertical and diagonal orientations. This is shown in **Figure 6**.

x and *y* are intensity values of two neighboring pixels in the image and *N* is the number of adjacent pixels selected from the image to calculate the correlation. 1000 pairs of two adjacent pixels are selected randomly from image to test correlation. The correlation coefficient be-

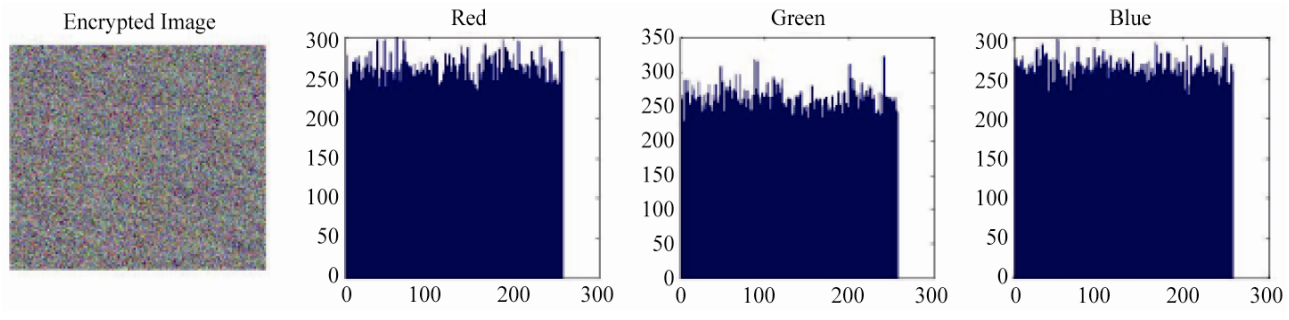
$$c_r = \frac{N\sum_{j=1}^{N}\left(x_j \times y_j\right) - \sum_{j=1}^{N} x_j \times \sum_{j=1}^{N} y_j}{\sqrt{\left(N\sum_{j=1}^{N} x_j^2 - \left(\sum_{j=1}^{N} x_j\right)^2\right) \times \left(N\sum_{j=1}^{N} y_j^2 - \left(\sum_{j=1}^{N} y_j\right)^2\right)}} \quad (5)$$



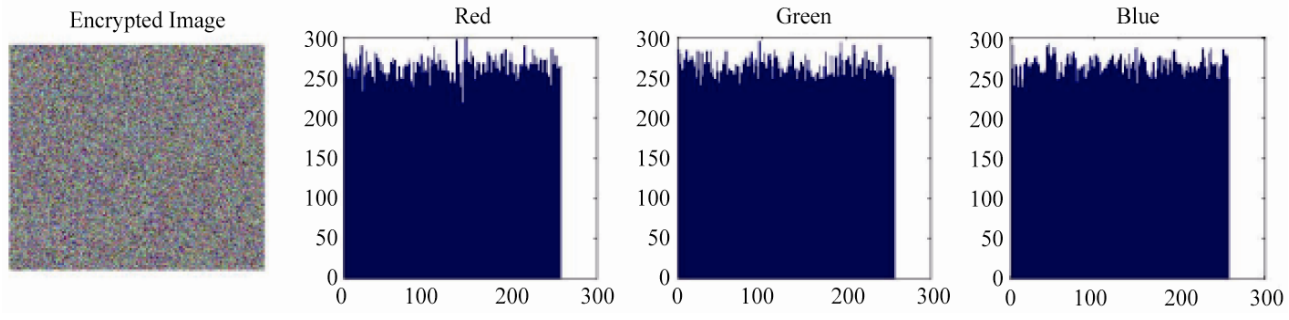
(a)



(b)



(c)



(d)

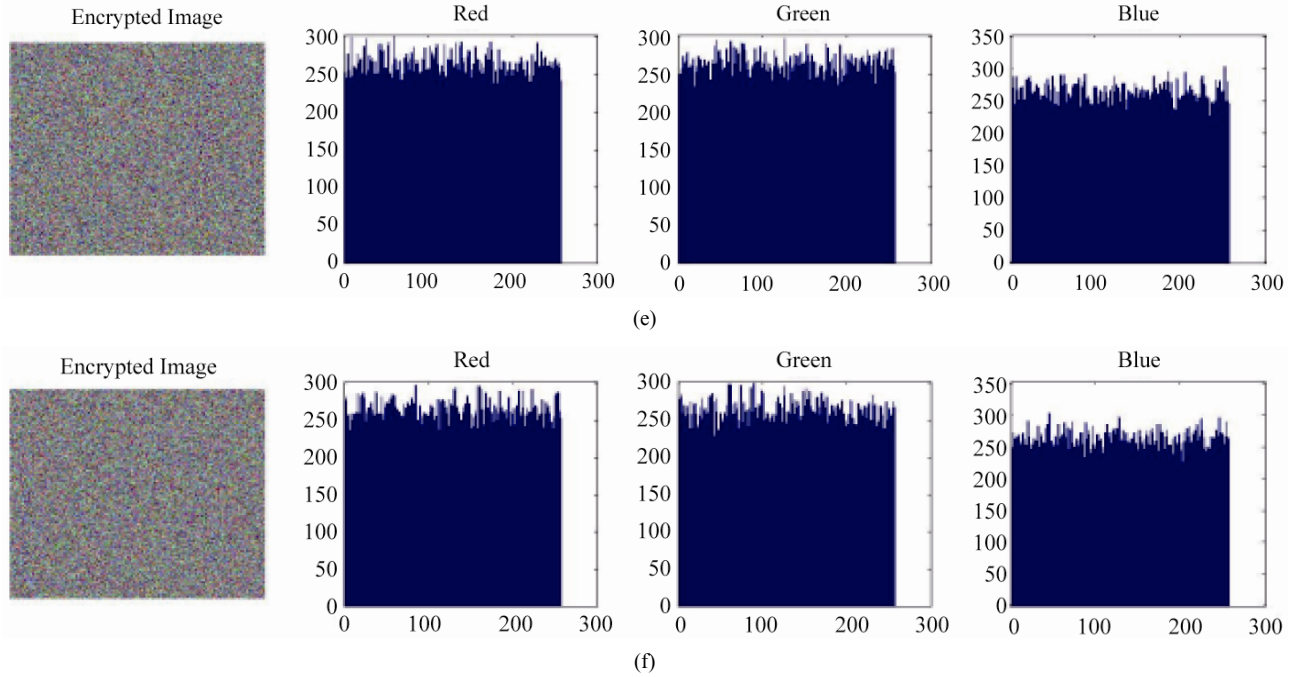


Figure 5. (a) Histogram for red, green and blue plane of original and encrypted image for *R* = 1; (b) Histogram for red, green and blue plane of encrypted image for *R* = 2; (c) Histogram for red, green and blue plane of encrypted image for *R* = 4; (d) Histogram for red, green and blue plane of encrypted image for *R* = 8; (e) Histogram for red, green and blue plane of encrypted image for *R* = 16; (f) Histogram for red, green and blue plane of encrypted image for *R* = 32.

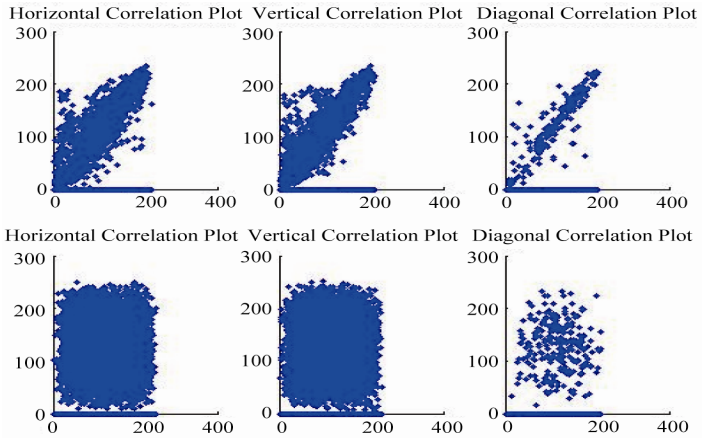


Figure 6. Correlation for horizontal, vertical and diagonal.

tween original and cipher image is calculate in **Table 6**.

4.3. Key Space Analysis

Key space size is the total number of different keys that can be used in the cryptography. Cryptosystem is totally sensitive to all secret keys. A good encryption algorithm should not only be sensitive to the cipher key, but also the key space should be large enough to make brute-force attack infeasible. The key space size for initial conditions and control parameters is over than 2^{148}. Apparently, the key space is sufficient for reliable practical use.

4.4. Differential Analysis

In general, a desirable characteristic for an encrypted image is being sensitive to the little changes in plain-image (e.g. modifying only one pixel). Adversary can create a small change in the input image to observe changes in the result. By this method, the meaningful relationship between original image and cipher image can be found. If one little change in the plain-image can

cause a significant change in the cipher-image, with respect to diffusion and confusion, then the differential attack actually loses its efficiency and becomes almost useless. There are three common measures were used for differential analysis: MAE, NPCR and UACI. Mean Absolute Error (MAE). The bigger the MAE value, the better the encryption security. NPCR means the Number of Pixels Change Rate of encrypted image while one pixel of plain-image is changed. UACI which is the Unified Average Changing Intensity, measures the average intensity of the differences between the plain-image and Encrypted image.

Let $C(i, j)$ and $P(i, j)$ be the color level of the pixels at the ith row and jth column of a $W \times H$ cipher and plain-image, respectively. The MAE between these two images is defined in

$$\mathrm{MAE} = \frac{1}{W \times H} \sum_{j=1}^{W} \sum_{i=1}^{H} \left| c(i, j) - p(i, j) \right|. \quad (6)$$

Consider two cipher-images, $C1$ and $C2$, whose corresponding plain-images have only one pixel difference.

The NPCR of these two images is defined in

$$\mathrm{NPCR} = \frac{\sum_{i,j} D(i, j)}{W \times H} \times 100\% \quad (7)$$

where W and H are the width and height of the image and $D(i, j)$ is defined as

$$D(i, j) = \begin{cases} 0, & \text{if } C1(i, j) = C2(i, j) \\ 1, & \text{if } C1(i, j) \neq C2(i, j) \end{cases}.$$

Another measure, UACI, is defined by the following formula:

$$\mathrm{UACI} = \frac{1}{W \times H} \sum_{i,j} \left[\frac{c1(i, j) - c2(i, j)}{255} \right] \times 100\%. \quad (8)$$

Tests have been performed on the encryption schemes on a 256-level color image of size 256 × 256 shown in **Figures 5(a)-(f)**. The MAE, NPCR and UACI experiment result is shown in **Tables 4** and **2**. The **Tables 3** and **5** compare the result of Yong previous related work based on chaotic map and our. Results obtained from NPCR show that the encryption scheme's sensitivity to little changes in the input image is under 0.01%. According to the UACI estimation result, the rate influence due to one pixel change is very low. The results demonstrate that a swiftly change in the original image will result in a negligible change in the ciphered image.

4.5. Information Entropy Analysis

It is well known that the entropy $H(m)$ of a message source m can be measured by

$$H(m) = \sum_{i=0}^{m-1} p(m_i) \log \frac{1}{p(m_i)} \quad (9)$$

where M is the total number of symbols $m_i \in m$; $p(m_i)$ represents the probability of occurrence of symbol mi and log denotes the base 2 logarithm so that the entropy is expressed in bits. For a random source emitting 256 symbols, its entropy is $H(m) = 8$ bits. This means that the cipher-images are close to a random source and the proposed algorithm is secure against the entropy attack. The test result on different image for different round is defined in **Table 7**.

4.6. Speed Analysis

Apart from the security consideration, some other issues on image encryption are also important. This includes the encryption speed for real-time processes. In general, encryption speed is highly dependent on the CPU/MPU structure, RAM size, Operating System platform, the programming language and also on the compiler options. So, it is senseless to compare the encryption speeds of two ciphers image.

Without using the same developing atmosphere and optimization techniques. Inspire of the mentioned difficulty, in order to show the effectiveness of the proposed image encryption scheme over existing algorithms. We

Table 2. NPCR, UACI and Entropy for key sensitivity test.

Lenna Error Image	$R = 2$	$R = 3$	$R = 4$
NPCR	99.5966593424	99.6098836263	99.651082356
UACI	52.5394813687	51.6816741344	50.603535970
Entropy	7.99913068980	7.99912231127	7.9991865161
Baboon Error Image			
NPCR	99.6103922526	99.5905558268	99.599711100
UACI	46.9998348460	47.8581327550	48.719709807
Entropy	7.99901078968	7.99905756543	7.9991734549

Table 3. Comparison of NPCR and UACI with Yong Wong *et al.*

Name of image		$R = 1$		$R = 2$		$R = 3$	
		Our	Yong Wang *et. al*	Our	Yong Wang *et. al*	Our	Yong Wang *et al.*
Airplane	NPCR	**99.62**	97.621	99.60	99.637	99.62	99.634
	UACI	**33.19**	32.909	33.10	33.575	33.35	33.580

Table 4. NPCR and UACI for different round on different color image.

Image		$R = 1$	$R = 2$	$R = 3$	$R = 4$	$R = 8$	$R = 10$	$R = 16$	$R = 32$
Baboon	NPCR	99.55	99.57	99.59	99.59	99.61	99.60	99.60	99.61
	UACI	37.17	38.68	39.00	38.78	38.69	38.88	39.03	38.89
	MAE	71.65	74.52	75.29	75.18	75.23	75.37	75.34	75.50
Lenna	NPCR	99.63	99.64	99.59	99.62	99.62	99.61	99.62	99.59
	UACI	28.87	27.51	27.33	27.42	27.43	27.64	27.51	27.37
	MAE	80.84	77.24	77.46	77.58	77.76	77.84	77.67	77.54
Pepper	NPCR	99.62	99.62	99.58	99.58	99.63	99.62	99.62	99.62
	UACI	38.05	38.26	37.99	38.03	38.34	38.20	38.33	38.11
	MAE	75.20	74.68	74.27	74.48	74.89	74.62	74.62	74.61
Airplane	NPCR	99.62	99.60	99.59	99.62	99.63	99.61	99.59	99.60
	UACI	33.19	33.10	33.35	33.27	33.329	33.28	33.32	33.31

Table 5. The round number of scanning-image, permutation and diffusion to achieve NPCR > 0.996 and UACI > 0.287.

	NPCR	UACI	No. of Round for Confusion and Diffusion
Our	>0.996	>0.287	1
Ref. [3]	>0.996	>0.333	2
Ref. [4]	>0.996	>0.333	18
Ref. [5]	>0.996		5
Ref. [6]	>0.996	>0.333	6
Ref. [7]	>0.996	>0.333	6

evaluated the performance of encryption schemes with an un-optimized MATLAB 7.0 code. Performance was measured on a machine with Intel core 2 Duo 2.00 GHz CPU with 2 GB of RAM running on Windows XP. The time for encryption and decryption is measured for different round is shown in **Tables 8** and **9**.

4.7. FIPS 140 Testing

We also show that our proposed algorithm pass the FIPS 140-2 randomness tests. There are four tests: Mono-bit, Poker, Runs tests and Long run tests. Each of the tests was designed to test the randomness of a sample sequence length of 20,000 bits as follows:

4.7.1. The Monobit Test

1) Calculate x which is the number of ones in the 20,000 bit stream.

2) The test is passed if $9725 < x < 10{,}275$.

4.7.2. The Poker Test

1) Divide the 20,000 bit stream into 5000 contiguous 4 bit segments. Count and store the number of occurrences of each of the 16 possible 4 bit values. Denote $g(i)$ as the number of each 4 bit value i where 0 - 15.

2) Calculate x by

$$X = \frac{16}{5000}\sum_{i=0}^{15} g(i)^2 - 5000 \qquad (10)$$

3) The test is passed if $2.16 < x < 46.17$.

4.7.3. The Runs Test

1) A run represents a maximal sequence of consecutive bits of all ones or all zeros. The incidences of runs of all lengths in the sample stream should be counted and stored.

2) The test is passed if the number of runs is each within the corresponding interval specified below **Table 10**.

4.7.4. The Long Run Test

1) Find the longest run in the 20,000 bits.

2) If the length of the longest run in the bit stream of 20,000 bit (both of one and zero) is smaller than 26, the test is passed.

Table 6. Correlation coefficient for plain and cipher image.

	Correlation coefficient of plain image			Correlation coefficient of Cipher image		
Images	Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
Baboon	0.8646	0.8293	0.8114	0.004	0.007	0.037
Lena	0.9156	0.8808	0.8603	0.001	0.006	0.091
Pepper	0.9376	0.9364	0.8935	0.005	0.006	0.023

Table 7. Entropy test for different color image.

Image	$R = 1$	$R = 2$	$R = 3$	$R = 4$	$R = 8$	$R = 10$	$R = 16$	$R = 32$	Yong Wang *et al.* [7]
				Our Scheme					
Baboon	7.9988	7.9990	7.9991	7.9992	7.9990	7.9990	7.9990	7.9991	-
Lenna	7.9981	7.9991	7.9991	7.9992	7.9991	7.9990	7.9990	7.9990	7.9990
Pepper	7.9987	7.9991	7.9991	7.9989	7.9989	7.9992	7.9992	7.9992	7.9990
Airplan	7.9989	7.9989	7.9991	7.9991	7.9991	7.9991	7.9991	7.9992	-
Boat	7.9986	7.9992	7.9991	7.9990	7.9990	7.9990	7.9991	7.9991	-

Table 8. Encryption time in second for different round.

Image	$R = 1$	$R = 2$	$R = 3$	$R = 4$	$R = 8$	$R = 10$	$R = 16$	$R = 32$
				Our Scheme				
Baboon	0.510	0.87	1.20	1.56	2.94	3.65	5.73	11.24
Lenna	0.521	0.87	1.197	1.561	2.933	3.662	5.697	11.225
Pepper	0.521	0.87	1.197	1.561	2.933	3.662	5.697	11.225
Airplan	0.521	0.87	1.197	1.561	2.933	3.662	5.697	11.225
Boat	0.521	0.87	1.197	1.561	2.933	3.662	5.697	11.225

Table 9. Decryption time in second for different round.

Image 256 × 256	$R = 1$	$R = 2$	$R = 3$	$R = 4$	$R = 8$	$R = 10$	$R = 16$	$R = 32$
				Our Scheme				
Baboon	0.43	0.77	1.12	1.470	2.85	3.55	5.62	11.17
Lenna	0.429	0.77	1.137	1.471	2.869	3.548	5.618	11.20
Pepper	0.430	0.78	1.139	1.472	2.869	3.549	5.619	11.22
Airplan	0.429	0.77	1.137	1.471	2.869	3.548	5.618	11.20
Boat	0.434	0.722	1.065	1.414	2.807	3.523	5.594	11.15

We need to change the testing algorithm to suit to image data so we randomly chose 100 streams of 20,000 consecutive bits from the ciphered images of image A. Then we find statistics of the randomly chosen 100 streams for each test and compared them to the acceptance ranges. **Table 11** show the numbers of the samples among 100 randomly chosen samples, which passed the Mono-bit, Poker, Long run tests and run tests.

5. Conclusions

This paper presents a technique which replaces the traditional preprocessing complex system and utilizes the basic operations like confusion, diffusion which provide same or better encryption using cascading of 3D standard and 3D cat map. We generate diffusion template using 3D standard map and rotate image by using vertically and horizontally red and green plane of the input image. We then shuffle the red, green, and blue plane by using 3D Cat map and standard map. Finally the Image is encrypted by performing XOR operation on the shuffled image and diffusion template. Completion of the design, both theoretical analyses and experimental tests have been carried out, both confirming that the new cipher

Table 10. FIPS-140 test range.

Length of the run	1	2	3	4	5	≥6
Required interval	2315 - 2685	1114 - 1386	527 - 723	240 - 384	103 - 209	103 - 209

Table 11. FIPS-140 test P = pass, F = fail.

Name of image		$R = 1$	$R = 2$	$R = 3$	$R = 4$	$R = 8$	$R = 10$	$R = 16$	$R = 32$
Baboon	runs	10P, 10P	11P, 12P	13P, 15P	12P, 14P	15P, 11P	12P, 12P	16P, 12P	17P, 14P
	pocker	374.7F	9.8944P	15.443P	12.985P	13.644P	8.678P	12.556P	12.556P
	mono	10082P	9975P	9969P	9952P	9990P	10031P	9913P	10032P
Lenna	runs	10P, 11P	13P, 15P	13P, 15P	17P, 13P	12P, 12P	13P, 13P	15P, 20P	12P, 12P
	pocker	317.4F	24.30P	14.022P	8.9920P	18.227P	21.856P	11.558P	18.752P
	mono	9956P	10025P	10103P	9967P	9938P	10054P	9900P	10112P
Pepper	runs	12P, 16P	13P, 15P	12P, 13P	12P, 14P	12P, 13P	13P, 13P	13P, 11P	14P, 14P
	pocker	138.41F	13.196P	12.057P	13.337P	18.227P	20.454P	10.227P	27.929P
	mono	10085P	9982P	10001P	9967P	10048P	9994P	9913P	10036P
Airplan	runs	12P, 10P	14P, 13P	17P, 23P	12P, 14P	14P, 13P	15P, 12P	12P, 13P	12P, 11P
	pocker	880.25F	30.016P	13.504P	12.134P	16.761P	20.108P	12.800P	9.568P
	mono	10030P	10075P	9975P	9996P	9940P	10071P	9946P	9973P

possesses high security and fast encryption speed. In conclusion, therefore, the new cipher indeed has excellent potential for practical image encryption applications.

6. References

[1] G. Chen, Y. Mao and C. K. Chui, "A Symmetric Image Encryption Scheme Based on 3D Chaotic Cat Maps," *Chaos Solitons & Fractals*, Vol. 21, No. 3, 2004, pp. 749-761. doi:10.1016/j.chaos.2003.12.022

[2] D. Xiao, X. Liao and P. Wei, "Analysis and Improvement of a Chaos-Based Image Encryption Algorithm," *Chaos Solitons & Fractals*, Vol. 40, No. 5, December 2007.

[3] Y. Wanga, K.-W. Wong, X. F. Liao and G. R. Chen, "A New Chaos-Based Fast Image Encryption Algorithm," *Applied Soft Computing*, Vol. 11, No. 1, 2011, pp. 514-522. doi:10.1016/j.asoc.2009.12.011

[4] Z. Guan, F. Huang and W. Guan, "Chaos-Based Image Encryption Algorithm," *Physics Letters A*, Vol. 346, No. 1-3, 2005, pp. 153-157. doi:10.1016/j.physleta.2005.08.006

[5] S. Lian, J. Sun and Z. Wang, "A Block Cipher Based on a Suitable Use of the Chaotic Standard Map," *Chaos Solitons & Fractals*, Vol. 26, No. 1, 2005, pp. 117-129. doi:10.1016/j.chaos.2004.11.096

[6] K. W. Wong, B. S. Kwok and W. S. Law, "A Fast Image Encryption Scheme Based on Chaotic Standard Map," *Physics Letters A*, Vol. 372, No. 15, 2008, pp. 2645-2652. doi:10.1016/j.physleta.2007.12.026

[7] Y. Mao, G. Chen and S. Lian, "A Novel Fast Image Encryption Scheme Based on 3D Chaotic Baker Maps," *International Journal of Bifurcation and Chaos*, Vol. 14, No. 10, 2004, pp. 3613-3624. doi:10.1142/S021812740401151X

