



# 3D variable-structure chaotic system and its application in color image encryption with new Rubik's Cube-like permutation

Jiuyang Xin · Hanping Hu · Jun Zheng

Received: 27 September 2022 / Accepted: 30 December 2022 / Published online: 10 January 2023  
© The Author(s), under exclusive licence to Springer Nature B.V. 2023

**Abstract** Nowadays, images have an increasingly deep impact on human life, so it is essential to encrypt images. Among various image encryption methods, chaotic encryption is particularly prominent due to the unpredictability and initial state sensitivity of chaos. The security of chaos-based cryptosystems depends largely on the performance of the adopted chaotic systems. This paper proposes a three-dimensional variable-structure chaotic system (3DVSCS) whose structure is time-varying. The distinguished dynamical characteristics of 3DVSCS are verified by various metrics, such as lyapunov exponent, approximate entropy, etc. Taking full advantage of the chaotic sequences generated by the 3DVSCS, a novel cryptosystem (3DVSCS-IES) with the confusion–diffusion architecture is presented. To further enhance security, a novel Rubik's Cube-like permutation method is designed to severely scramble adjacent pixels, and an iterative diffusion algorithm is employed to completely infect the entire cipher image with a slight change in the plaintext image. The security analyses show that the proposed image encryption algorithm owns bet-

ter security performance than some typical state-of-art methods.

**Keywords** Chaos · Variable structure · Rubik's Cube-like permutation · Image encryption

## 1 Introduction

Images are playing an increasingly important role in human life. The rapid development of sharing platforms on social networking sites has resulted in an ever-increasing amount of image data [1]. Most of the massive image data are stored in plaintext to save the cost of storage and computing resources, which poses a huge security risk. For example, some people use other people's photos to take away the express delivery of the parties through facial recognition. Therefore, it is crucial to protect the images [2, 3]. Methods to protect digital images can be divided into two categories: (1) Information hiding, including watermarking [4], anonymity [5], and steganography [6]. (2) Encryption [7], including conventional encryption [8] and other methods, such as chaotic encryption [9]. Of the two categories, encryption is the more straightforward approach.

In order to encrypt an image, one approach is to treat the pixels of the plain image as a stream of binary data, which is then encrypted using traditional text encryption methods, such as RSA [10], data encryption standard (DES) [11], advanced encryption standard (AES) [12], and International Data Encryption Algo-

---

J. Xin · J. Zheng (✉)  
School of Cyberspace Security, Huazhong University of Science and Technology, Wuhan 430074, China  
e-mail: M202171791@hust.edu.cn; zhengj@hust.edu.cn

H. Hu  
School of Artificial Intelligence and Automation,  
Huazhong University of Science and Technology, Wuhan  
430074, China  
e-mail: hhphustemail@foxmail.com

rithm (IDEA) [13]. The purpose of this approach is to utilize the existing text encryption methods for encrypting images, as an alternative to developing new methods specifically for image encryption. However, unlike text encryption, images have some inherent characteristics, including large data volume, high correlation between adjacent pixels, and strong redundancy [14]. To solve this problem, many encryption techniques have been proposed, such as chaos encryption [15], permutation encryption [16], optical-based encryption [17], and DNA-based encryption [18]. Among these techniques, chaos encryption has received extensive attention due to its outstanding performance.

The chaos-based image encryption method draws on the sound characteristics of chaos theory to build an encryption system [19]. The advantages of a chaos-based image encryption system include large key space, simple implementation, and fast encryption speed. However, the security of most encryption schemes depends heavily on the performance of the chaotic system used by the encryption system [20].

Chaos possesses many unique properties that can satisfy the requirements of image encryption, such as good randomness, unpredictability, and initial state sensitivity [21, 22]. In a chaos-based encryption scheme, the security level strongly relies on the complexity and performance of the core chaotic system [23]. However, existing chaotic maps may exhibit shortcomings in different aspects when implemented in digital computers and digital circuits [24–27]. Firstly, most of the existing chaotic systems have a single structure and lack of change, so it is difficult to resist phase space reconstruction and parameter identification attacks [28]. Secondly, many existing chaotic systems have weak chaotic performance, which is characterized by short periods, non-ergodicity, low linear complexity, etc. Moreover, their chaotic ranges are either narrow or discontinuous. If a chaotic map has narrow or discontinuous chaotic ranges, its chaos properties may be destroyed when its parameters are disturbed by certain external factors such as noise [29].

In order to improve the above shortcomings, most of the schemes obviously increase the implementation cost. Based on this, we propose a 3D variable-structure chaotic system (3DVSCS) which is low in cost and easy to control. The 3DVSCS constructs a safe and controllable time-varying chaotic source to ensure that its output has non-stationary statistical characteristics. Compared with slow parameter changes, the structural

transformations in chaotic systems reflect the abrupt change of dynamical behaviors, which can produce more complex non-stationary characteristics. Variable structure and nonstationarity make the 3DVSCS resistant to mathematical analysis and statistical analysis.

Taking 3DVSCS as the nonlinear chaotic source, a 3DVSCS-based image encryption system (3DVSCS-IES) composed of a new Rubik's Cube-like permutation method and an iterative diffusion algorithm is presented, where permutation is used to severely scramble adjacent pixels, and diffusion is employed to completely infect the entire cipher image with a small change in the plaintext image. The contributions and novelty of this work are summarized as follows.

- (1) The 3DVSCS is first presented to obtain higher complexity and security than fixed-structure chaotic systems and variable parameter systems;
- (2) The variable-structure chaotic system is heuristic, which can be easily extended to any complex system;
- (3) A new Rubik's Cube-like permutation is introduced, which has lower implementation cost and better pixel scrambling effect compared with other permutation schemes;
- (4) Based on the 3DVSCS, an image encryption system (3DVSCS-IES) composed of a new Rubik's Cube-like permutation method and an iterative diffusion algorithm is proposed. Security analysis demonstrates that 3DVSCS-IES can achieve a high level of security.

The remainder of this paper is organized as follows. Section 2 gives the 3DVSCS model and mathematically analyzes its chaotic behavior. In Sect. 3, the performance of the chaotic map generated by 3DVSCS on a series of chaotic metrics is presented. Based on the 3DVSCS, Sect. 4 raises the model of 3DVSCS-IES. In Sect. 5, the simulation results of 3DVSCS-IES are shown. Section 6 verifies the security of the proposed encryption system and its resistance to various types of attacks. In Sect. 7, the conclusion is given.

## 2 3DVSCS

This section introduces the model and theoretical analysis of 3DVSCS. A concrete method to generate the 3DVSCS map is also given.

### 2.1 Mathematical model of 3DVSCS

3DVSCS is presented to obtain a chaotic map with a more extensive chaotic range and more complex chaotic behavior. The simple mathematical model of 3DVSCS is as follows:

$$\mathbf{x}_{n+1} = W_i \mathbf{x}_n \text{ mod } 1, \quad i \in \{1, 2\}, \tag{1}$$

where  $\mathbf{x} = (x, y, z)^T$ , is the state vector of the chaotic system, mod is the modulo function used to constrain the phase space of state variable.  $W_1$  and  $W_2$  are the parameter matrices of the system. Different parameter matrices correspond to different structures.  $W_i$  can be expressed as follows:

$$\begin{bmatrix} w_{11}^{(i)} & w_{12}^{(i)} & w_{13}^{(i)} \\ w_{21}^{(i)} & w_{22}^{(i)} & w_{23}^{(i)} \\ w_{31}^{(i)} & w_{32}^{(i)} & w_{33}^{(i)} \end{bmatrix}, \quad i \in \{1, 2\}$$

Then, the above Eq. (1) can also be transformed into another form:

$$\begin{cases} x(n+1) = (w_{11}^{(i)}x(n) + w_{12}^{(i)}y(n) \\ \quad + w_{13}^{(i)}z(n)) \text{ mod } 1 \\ y(n+1) = (w_{21}^{(i)}x(n) + w_{22}^{(i)}y(n) \\ \quad + w_{23}^{(i)}z(n)) \text{ mod } 1 \\ z(n+1) = (w_{31}^{(i)}x(n) + w_{32}^{(i)}y(n) \\ \quad + w_{33}^{(i)}z(n)) \text{ mod } 1 \end{cases}, \quad i \in \{1, 2\} \tag{2}$$

As can be seen from the formula, 3DVSCS is first stretched by multiplying the parameter matrix of the current iteration by the state variables of the system and then folded by the boundary function. In this way, complex chaotic behavior can be obtained.

### 2.2 Theoretical analysis

The basic feature of chaotic system is the extreme sensitivity of the system to the initial value. The trajectories generated by two initial values that are almost the same are separated exponentially over time. The Lyapunov exponent (LE) is an indicator that quantitatively describes this phenomenon. Chaos in the sense of LE can be defined as Definition 1

**Definition 1** A discrete system is said to be chaotic in the sense of LE if it satisfies the two conditions:

- (1) it has at least one positive LE;
- (2) its phase space region is globally bounded.

For a chaotic map  $x \rightarrow f(x)$ , LE can be defined as follows:

$$LE = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=0}^{n-1} \ln |f'(x_i)|, \tag{3}$$

as for 3DVSCS, the LEs can be calculated as:

$$LE_k = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{j=0}^{n-1} \ln |\lambda_k^{x_j}|, \tag{4}$$

where  $\lambda_k^{x_j}$  is the  $k$ th eigenvalue of the Jacobian matrix at the  $j$ th iteration of 3DVSCS. The Jacobian matrix of 3DVSCS described in Eq. (2) can be expressed as:

$$J(\mathbf{x}_j) = \begin{bmatrix} w_{11}^{(i)} & w_{12}^{(i)} & w_{13}^{(i)} \\ w_{21}^{(i)} & w_{22}^{(i)} & w_{23}^{(i)} \\ w_{31}^{(i)} & w_{32}^{(i)} & w_{33}^{(i)} \end{bmatrix},$$

which is exactly the same as the parameter matrix  $W_i$ . It can be concluded that the value of the Jacobian matrix of 3DVSCS is equal to the value of the parameter matrix of the current iteration of the system. Then the LEs of 3DVSCS can also be expressed as:

$$\begin{aligned} LE_k &= \lim_{n \rightarrow \infty} \frac{1}{n} \left( \sum_{j=0}^{n_1-1} \ln |\lambda_k^{(1)}| + \sum_{j=0}^{n_2-1} \ln |\lambda_k^{(2)}| \right) \\ &= \frac{n_1}{n} \ln |\lambda_k^{(1)}| + \frac{n_2}{n} \ln |\lambda_k^{(2)}|, \end{aligned} \tag{5}$$

where  $n_1$  and  $n_2$  are respectively the number of times  $W_1$  and  $W_2$  are used as parameter matrices in the iteration.

For a square matrix, the trace is equal to the sum of all eigenvalues. Therefore, the sum of all eigenvalues of the parameter matrix  $W_i$  is equal to the trace  $\text{tr}(W_i)$  of  $W_i$ :

$$\lambda_1^{(i)} + \lambda_2^{(i)} + \lambda_3^{(i)} = \text{tr}(W_i) = w_{11}^{(i)} + w_{22}^{(i)} + w_{33}^{(i)},$$

where  $\lambda_1^{(i)} \geq \lambda_2^{(i)} \geq \lambda_3^{(i)}$ . If  $w_{11}^{(i)}$ ,  $w_{22}^{(i)}$ , and  $w_{33}^{(i)}$  are all greater than 1, this leads to

$$\lambda_1^{(i)} + \lambda_2^{(i)} + \lambda_3^{(i)} > 3,$$

which suggests that at least one of  $\lambda_1^{(i)}, \lambda_2^{(i)}, \lambda_3^{(i)}$  has a value greater than 1. It is easy to get  $\lambda_1^{(1)} > 1$  and  $\lambda_1^{(2)} > 1$ , which means that at least one LE is positive according to Eq. (5). It can be concluded that the proposed 3DVSCS satisfies condition (1) in Definition 1 if the diagonal elements of  $W_i$  are all greater than 1.

The modulo function makes the 3D chaotic systems in Eq. (2) fold over bounded phase space. That is, the phase space region of 3DVSCS is globally bounded and thus satisfies condition (2) of Definition 1. In summary, Proposal 1 is given for the 3DVSCS parameter matrices to make sure the system exhibits chaotic characteristics.

**Proposition 1** *The 3D parameter matrices  $W_1, W_2$  of the 3D variable-structure chaotic system in Eq. (1) should satisfy the following two conditions, respectively:*

$$W_1 : w_{ii} \geq w_{ij} + 3, \quad i = [1, 2, 3], j \neq i, w > 1$$

$$W_2 : w_{ii} \leq w_{ij} - 3, \quad i = [1, 2, 3], j \neq i, w > 1$$

Among the two matrices  $W_1, W_2$  in Proposition 1,  $W_1$  is characterized by larger diagonal elements, and  $W_2$  is characterized by smaller diagonal elements. Taking  $W_1$  and  $W_2$  as parameter matrices, two different structures of the chaotic system are obtained, and then by alternately using  $W_1$  and  $W_2$  in the iteration, the chaotic system with variable structure is realized.

### 2.3 Method to generated 3DVSCS map

In order to construct two parameter matrices  $W_1$  and  $W_2$  that satisfy Proposal 1, five parameters  $o, r_1, r_2, d_1,$  and  $d_2$  are used, where  $o \in (0, \infty)$  is used as the center element  $w_{22}$  of  $W_1$  and  $W_2$ .  $r_1, r_2 \in (0, \infty)$  are used to generate the other two diagonal elements. The other elements of  $W_1$  and  $W_2$  are generated by  $d_1, d_2 \in (0, 1)$ . This generation method can be described as:

$$W_1 = \begin{bmatrix} r_1o + 4 & r_1od_1 + 1 & r_1od_2 + 1 \\ od_1 + 1 & o + 4 & od_2 + 1 \\ r_2od_1 + 1 & r_2od_2 + 1 & r_2o + 4 \end{bmatrix}$$

$$W_2 = \begin{bmatrix} r_1o + 1 & r_1o(d_1 + 1) + 4 & r_1o(d_2 + 1) + 4 \\ o(d_1 + 1) + 4 & o + 1 & o(d_2 + 1) + 4 \\ r_2o(d_1 + 1) + 4 & r_2o(d_2 + 1) + 4 & r_2o + 1 \end{bmatrix} \tag{6}$$

The transformation of the chaotic system structure is achieved by using different parameter matrices in different iterations.  $W_1$  is set as the parameter matrix when

$x_i < 0.5$ , and  $W_2$  is set as the parameter matrix when  $x_i \geq 0.5$ . The two matrices  $W_1$  and  $W_2$  constitute a dual form, so the generated variable-structure chaotic system can possess relatively balanced dynamics. The detailed chaotic map generated by 3DVSCS is as follows:

$$x_{n+1} = \begin{cases} W_1x_n \text{ mod } 1, & x_n < 0.5 \\ W_2x_n \text{ mod } 1, & x_n \geq 0.5 \end{cases} \tag{7}$$

### 3 Performance evaluation of 3DVSCS map

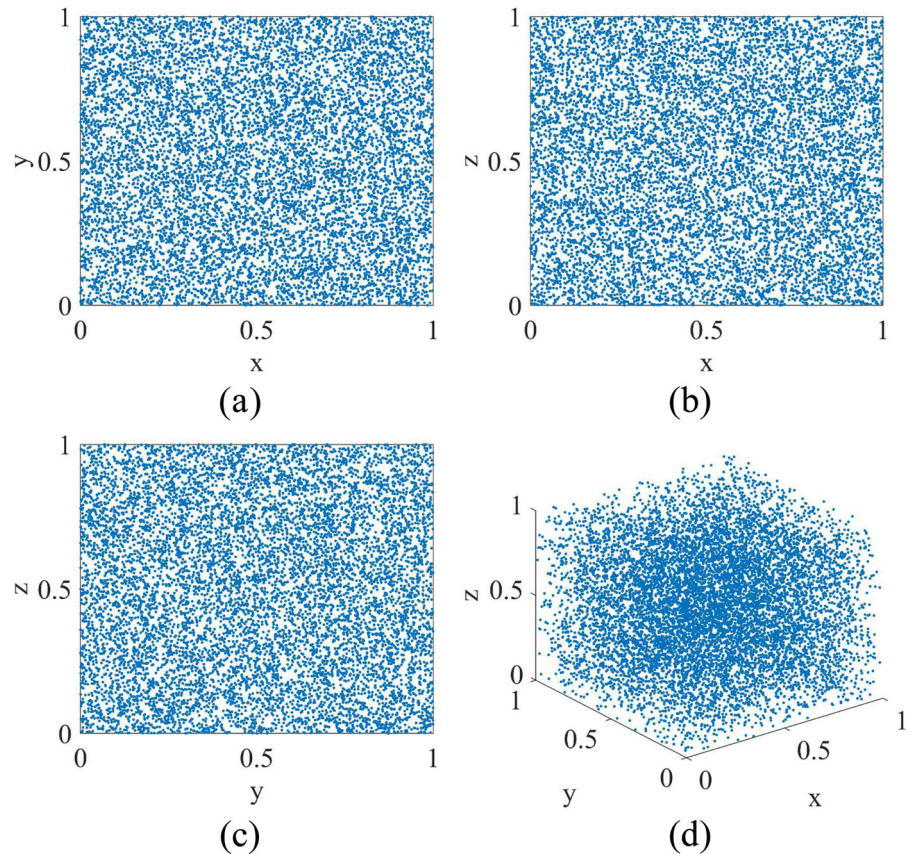
This section shows the performance of the chaotic map generated by 3DVSCS on different aspects, including phase diagram, bifurcation diagram, lyapunov exponent, approximate entropy and randomness test.

#### 3.1 Phase diagram

Phase diagram is a method to describe the dynamic behavior of a chaotic system directly. The phase diagram of a chaotic system is its unclosed trajectory in a two-dimensional phase plane or a three-dimensional phase space. Systems with complex phase diagrams usually have complex chaotic behaviors. For the 3DVSCS map in Eq. (7), the initial values  $(x_0, y_0, z_0)$  are set as (0.01, 0.02, 0.03), and the parameters  $(o, r_1, r_2, d_1, d_2)$  are set as (101, 5.1, 3.8, 0.8, 0.6). The 2D and 3D phase diagrams of the 3DVSCS map presented in Fig. 1 are all noise-like patterns, which means that the chaotic behavior of this variable structure system is sufficiently complex.

#### 3.2 Bifurcation diagram

The bifurcation diagram shows the dynamical evolution of the system under different control parameters by plotting the values of system variables under different parameters. By observing the bifurcation diagram, we can clearly get the chaotic range of the system on certain parameter. For 3DVSCS in this section, the initial values  $(x_0, y_0, z_0)$  are set as (0.01, 0.02, 0.03), and the values of parameters  $(o, r_1, r_2, d_1, d_2)$  are set to (101, 5.1, 3.8, 0.8, 0.6) by default. Figure 2 gives the bifurcation diagrams of 3DVSCS for different parameters. It can be seen from the figure that 3DVSCS

**Fig. 1** Phase diagrams of 3DVSCS map

exhibits chaotic behavior under all parameters within the value range, which means that when the parameters of 3DVSCS can be used as equally strong keys in the encryption system without weak keys.

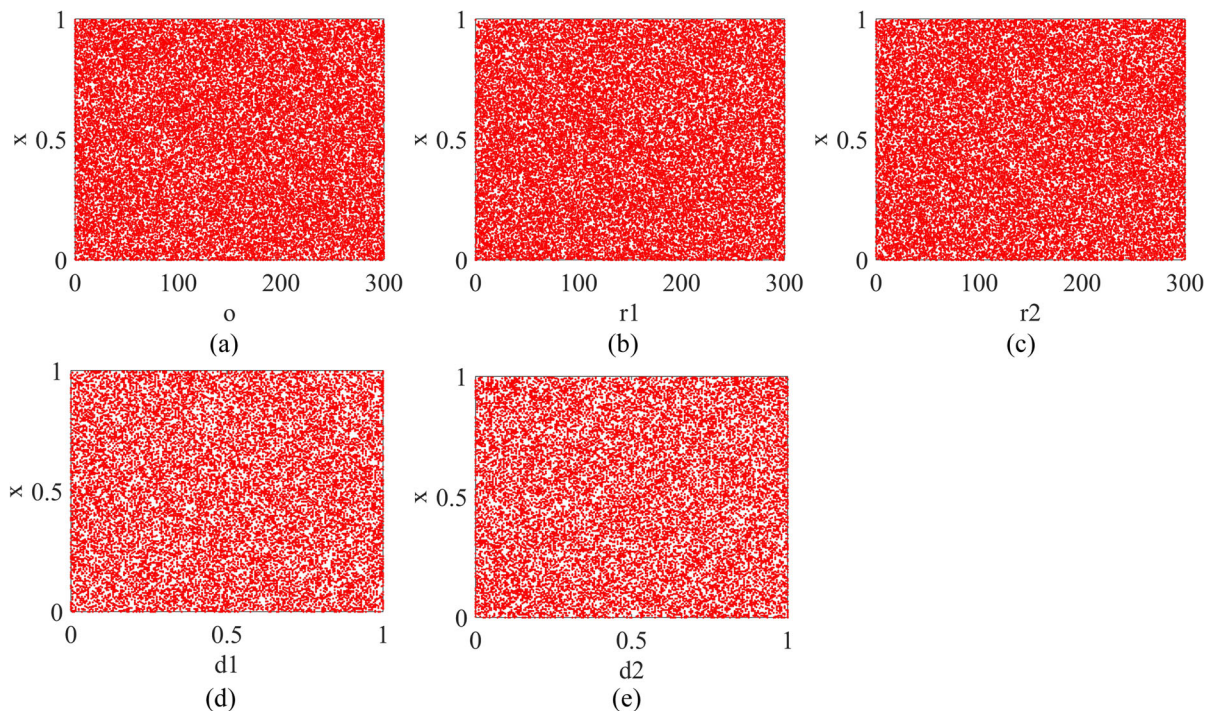
### 3.3 Lyapunov exponent

The Lyapunov exponent (LE) is an important quantitative index to measure the chaotic characteristics of the system, which characterizes the average exponential rate of convergence or divergence of the system between adjacent orbitals in the phase space. Whether there is dynamic chaos in the system can be judged intuitively by whether the maximum LE is greater than zero. A positive LE means that in the phase space of the system, no matter how small the initial distance between the two trajectories is, the difference will become unpredictable with an exponential rate increase over time, which is the phenomenon of chaos. Figure 3 depicts the Lyapunov exponents of 3DVSCS map for different parameters and compares them with the Lyapunov exponents of different mul-

tidimensional maps. For 3DVSCS in this section, the initial values  $(x_0, y_0, z_0)$  are set as  $(0.01, 0.02, 0.03)$ , and the values of parameters  $(o, r_1, r_2, d_1, d_2)$  are set to  $(101, 5.1, 3.8, 0.8, 0.6)$  by default. Figure 3a–e are the Lyapunov exponent plots for the five parameters, respectively. Figure 3f–h are the Lyapunov exponent diagrams of LSM [43], CSCM [33] and LSCM [20], respectively. As can be seen from the figure, 3DVSCS possesses three positive and high Lyapunov exponents for all parameters. This illustrates that the newly generated chaotic map exhibits hyperchaos. And compared with different chaotic systems, 3DVSCS always exhibits hyperchaos in its parameter range, and the Lyapunov exponents of 3DVSCS are larger, which means higher chaos. It can be concluded that 3DVSCS can generate chaotic maps with hyperchaotic behavior.

### 3.4 Complexity analysis

Approximate entropy (ApEn) is a nonlinear dynamic indicator used to quantify the regularity and unpredictability of fluctuations in time series. It uses a non-



**Fig. 2** Bifurcation diagrams of 3DVSCS map with different parameters

negative number to represent the complexity of a time series, reflecting the possibility of new information occurring in the time series. The more complex the time series, the larger the approximate entropy. ApEn is defined as follows:

**Definition 2** Given a time series  $\{x_1, x_2, \dots, x_N\}$  and pre-specified parameters  $m$  and  $r$ , where  $m$  is the embedding dimension, and  $r$  is the similarity tolerance, also known as the filtering level, the ApEn of this series is defined as:

$$\begin{cases} \text{ApEn} = \lim_{N \rightarrow \infty} [\phi^m(r) - \phi^{m+1}(r)], \\ \phi^m(r) = \frac{1}{N-m+1} \sum_{i=1}^{N-m+1} \ln C_i^m(r), \end{cases} \quad (8)$$

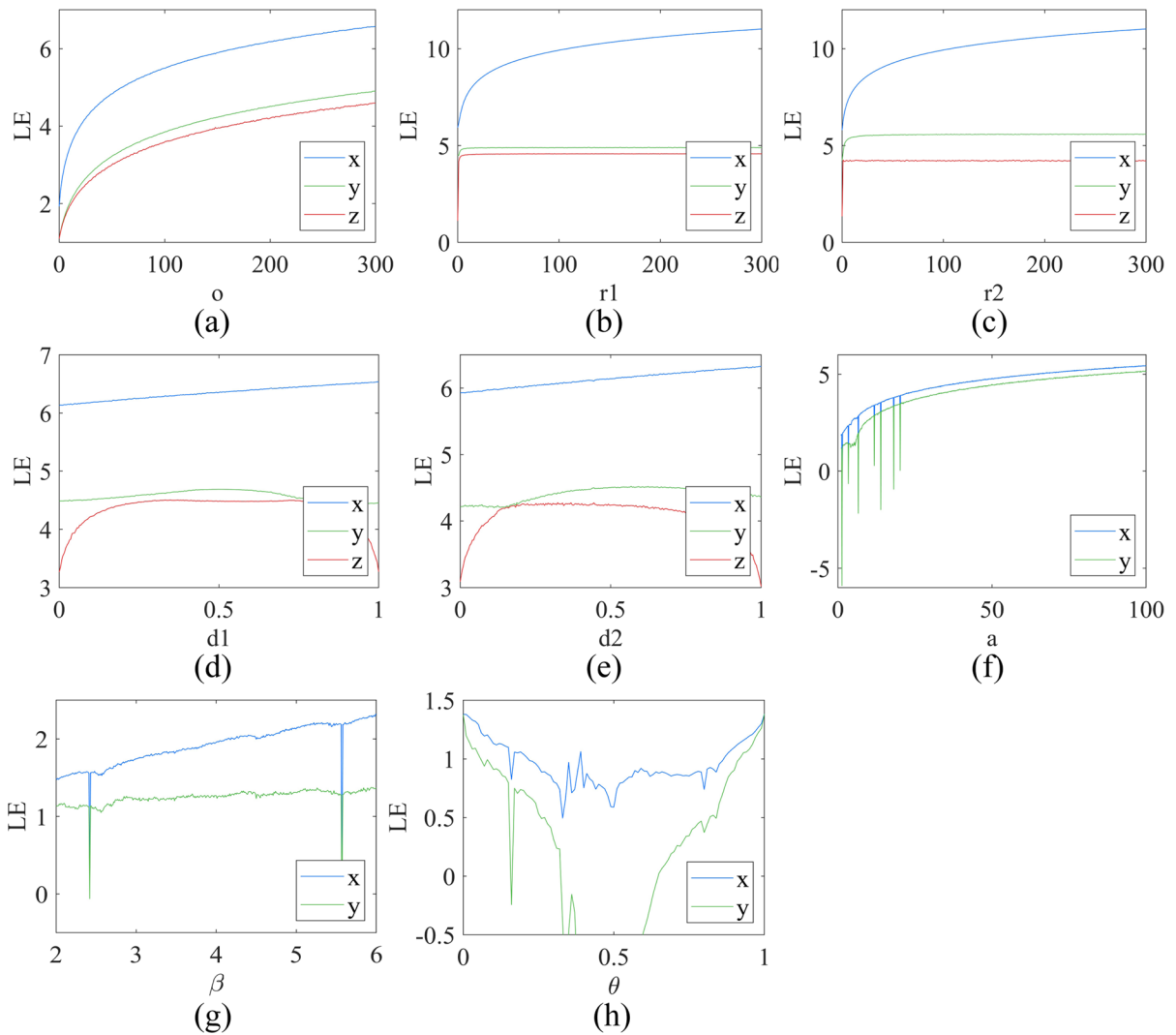
where  $C_i^m$  is obtained by counting the number of vectors having  $d[X(i), X(j)]$  less than  $r$  for each value of  $i$  and then calculating the ratio of this number to the total number of distances.  $d[X(i), X(j)]$  is the Chebyshev distance between  $X(i)$  and  $X(j)$ .

Figure 4 shows the comparison of ApEn values of 3DVSCS and different chaotic maps. For 3DVSCS in this section, the initial values and the parameter settings are the same as in Sect. 3.3. Figure 4a–c describes

the ApEn values when the parameters  $o$ ,  $r_1$ , and  $d_1$  are changed while other parameters remain unchanged, respectively. Figure 4d–f are the ApEn values of LSM, CSCM and LSCM, respectively. As can be seen from the figure, 3DVSCS has high ApEn values in all parameter ranges, which are more stable than those of different chaotic maps. The conclusion is that the novel 3DVSCS map can generate sequences with high complexity over a wide range of parameters.

### 3.5 Randomness test for 3DVSCS

In this section, two test suites NIST SP800-22 and TestU01 are used to verify the randomness of the chaotic sequences generated by 3DVSCS. NIST SP800-22 is a standard for the randomness of test sequences published by the National Institute of Standards and Technology. At present, this standard has been internationally recognized, including 15 items such as runs test and linear complexity test. Each test result corresponds to a calculated normalized P value, and the P value is compared with 0.01. If the P value is greater than 0.01, it means that the test requirements are met, otherwise it is a failure. Selecting random parameters and initial



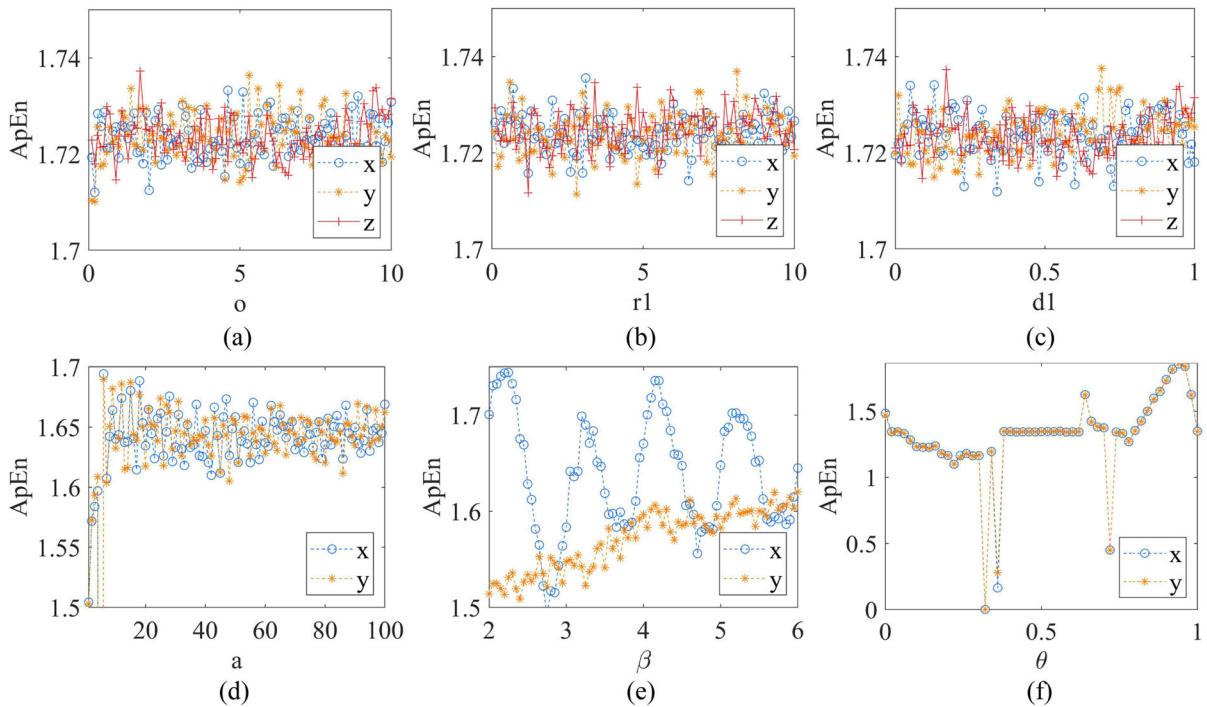
**Fig. 3** Lyapunov exponents of different maps. **a–e** 3DVSCS; **f** LSM; **g** CSCM; **h** LSCM

values, the 3DVSCS map generates 1000 sequences of length  $10^6$ , and the values in the sequences are simply quantized, where values greater than 0.5 are quantized to 1, and other values are quantized to 0. These sequences are subjected to the NIST test suite, and the test results are shown in Table 1. TestU01 is a more stringent randomness test suite that contains multiple sets of tests, each of which assesses the randomness of the sequence from a different aspect. The sets of tests used for testing the randomness of binary sequences are Alphabit, Rabbit, and BlockAlphabit. In this experiment, limited to the storage of the computer, the binary sequences of length  $2^{20}$ ,  $2^{25}$ , and  $2^{28}$  generated by

3DVSCS are used to carry out the TestU01 test, and the results are shown in Table 2. From the results in the two tables, it can be concluded that the sequences generated by 3DVSCS pass all the tests. In summary, the proposed 3DVSCS can generate sequences with high randomness, which are obviously adequate for image encryption.

#### 4 3DVSCS-IES

By using the 3D chaotic system, this section raises a 3DVSCS-based image encryption system (3DVSCS-IES). It uses the classic confusion–diffusion structure,



**Fig. 4** Approximate entropy of different maps. **a–c** 3DVSCS; **d** LSM; **e** CSCM; **f** LSCM

**Table 1** Results of the NIST SP800-22 test suite

Statistical test	<i>p</i> -value	Pass rate	Result
Frequency	0.206103	0.996	Success
BlockFrequency	0.412939	0.992	Success
CumulativeSums forward	0.517509	0.990	Success
CumulativeSums reverse	0.527450	0.994	Success
Runs	0.416251	0.986	Success
LongestRun	0.653244	0.996	Success
Rank	0.582809	0.998	Success
FFT	0.578869	0.990	Success
NonOverlappingTemplate	0.413036	0.988	Success
OverlappingTemplate	0.498931	0.991	Success
Universal	0.48764	0.996	Success
ApproximateEntropy	0.541599	0.992	Success
RandomExcursions	0.479646	0.989	Success
RandomExcursionsVariant	0.459153	0.991	Success
Serial	0.573903	0.991	Success
LinearComplexity	0.381075	0.996	Success



**Table 2** Results of the TestU01 test suite

Bits length	2 <sup>20</sup> bits	2 <sup>25</sup> bits	2 <sup>28</sup> bits
	Alphabit		
Pass ratio	17/17	17/17	17/17
	Rabbit		
Pass ratio	38/38	39/39	40/40
	BlockAlphabit		
Pass ratio	102/102	102/102	102/102

which is well-known for its high-security performance. The structure of 3DVSCS-IES is illustrated in Fig. 5, in which the security key generates the initial states and control parameters of 3DVSCS. 3DVSCS is iterated to obtain chaotic sequences  $(x, y, z)$ , which together with the security key generate the initial values and parameters of the logistic map. Then the sequences  $(x, y, z)$  generated by 3DVSCS are converted into  $(x_a, y_a, z_a)$  and  $(x_b, y_b, z_b)$ , these two sequences are used to control the cyclic shift distance in the proposed Rubik’s Cube-like confusion and the pixel change in the iterative diffusion method, respectively. The sequence generated by the logistic map is sorted to obtain the sequence  $c$  that controls the rotation order of the Rubik’s Cube-like confusion method. 3DVSCS-IES performs confusion–diffusion operation two times in total, which is enough to result in high cryptographic security.

4.1 Generate initial states and parameters

A security key that is too short makes the image encryption system vulnerable to brute-force attacks. In this paper, the length of the secure key is set to 256 bits, and its structure is illustrated in Fig. 6. The security key is divided into eight parts:  $K = \{x_0, y_0, z_0, o, r, d, a, m\}$ , each of which is 32 bits long, and  $r, d, a, m$  all conclude two 16-bit strings,  $r = (r_1, r_2)$ ,  $d = (d_1, d_2)$ ,  $a = (a_1, a_2)$ ,  $m = (m_1, m_2)$ .

The variables  $x_0, y_0, z_0$  are all fixed-point numbers within  $[0, 1)$ , the value of them can be obtained from a 32-bit sequence by  $V = \sum_{n=1}^{32} \text{Bit}_n \times 2^{-n}$ .  $o$  is fixed-point number with eight binary digits before the decimal point, the value of it is obtained by  $V = \sum_{n=1}^8 \text{Bit}_n \times 2^{8-n} + \sum_{n=9}^{32} \text{Bit}_n \times 2^{-(n-8)}$ . Moreover,  $r_1, r_2$  are fixed-point numbers with one binary digit before the decimal point, which can be calculated by  $V = \text{Bit}_1 + \sum_{n=2}^{16} \text{Bit}_n \times 2^{-(n-1)}$ .  $d_1, d_2$

are fixed-point numbers in the range of  $[0, 1)$ , each of them can be obtained by  $V = \sum_{n=1}^{16} \text{Bit}_n \times 2^{-n}$ .  $a_1, a_2, m_1, m_2$  are all integer numbers which can be gained by  $V = \sum_{n=1}^{16} \text{Bit}_n \times 2^{16-n}$ .

$(x_0, y_0, z_0)$  and  $(o, r, d)$  are the initial values and the parameters of the 3DVSCS chaotic map, respectively. The initial value  $a_0$  of the logistic map is generated by  $a$  as follows.

$$a_0 = \frac{x(a_1) + y(a_1) + z(a_1) + x(a_2) + y(a_2) + z(a_2)}{6}$$

And the parameter of the Logistic map will be obtained by the following formula.

$$\mu = 3.57 + 0.43 \times \frac{x(m_1) + y(m_1) + z(m_1) + x(m_2) + y(m_2) + z(m_2)}{6}$$

since the logistic map has chaotic behavior when the parameter  $\mu \in (3.5699456, 4]$ . Using all the initial states and parameters obtained by these methods above, sufficiently complex chaotic sequences can be generated for image confusion and diffusion.

4.2 Rubik’s Cube-like confusion method

The Rubik’s Cube-like confusion is raised with reference to the rotation of the Rubik’s Cube in three planes. Figure 7 shows the rotation of a third-order Rubik’s Cube in three planes. As is depicted in this figure, the operation of the Rubik’s Cube can be divided into three types, (a) rotate the  $x0z$  plane; (b) rotate the  $y0z$  plane; (c) rotate the  $x0y$  plane.

For an  $M \times N \times 3$  color RGB image, the rotation method of the Rubik’s Cube cannot be used directly because  $M$  and  $N$  are generally not equal to 3. In order to implement a Rubik’s Cube rotation operation on the image, a Rubik’s Cube-like rotation (RCLR) is raised.

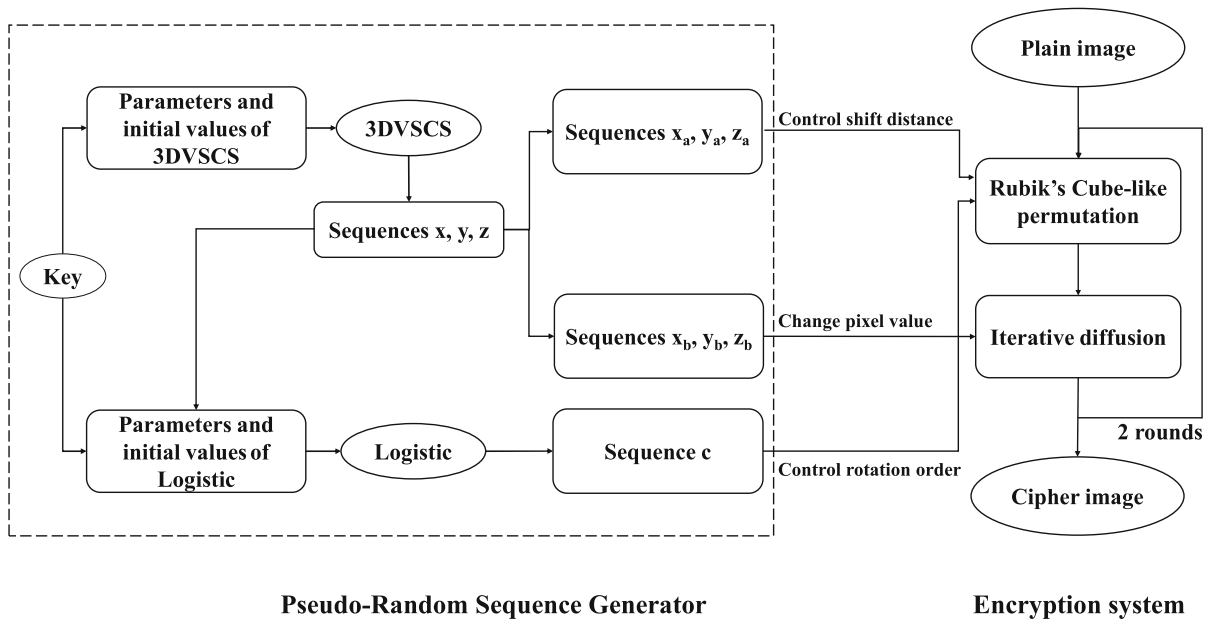


Fig. 5 The structure of 3DVSCS-IES

Fig. 6 Key structure of 3DVSCS-IES

$x_0$	$y_0$	$z_0$	$o$	$r$	$d$	$a$	$m$
32 bits	32 bits	32 bits	32 bits	32 bits	32 bits	32 bits	32 bits

Fig. 7 Three types of operations on the Rubik's Cube

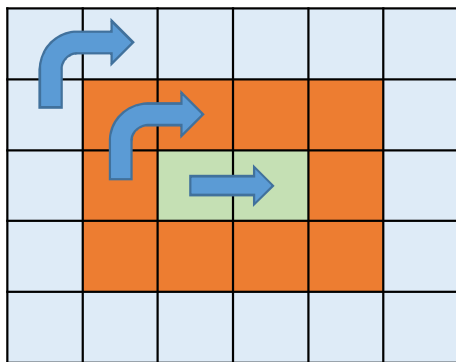
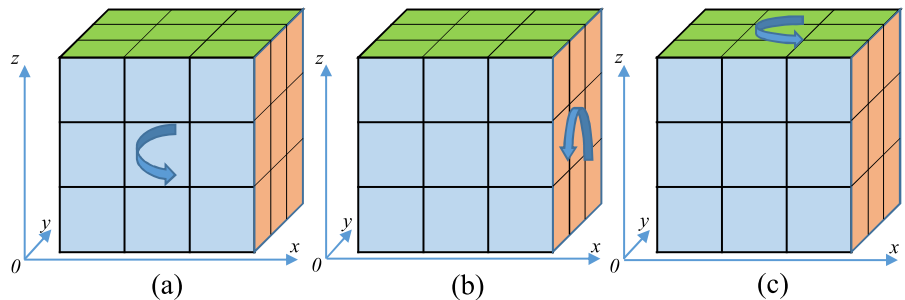


Fig. 8 Rubik's Cube-like rotation

As shown in Fig. 8, each matrix can be divided into  $\lfloor \min(M, N)/2 \rfloor$  rings, and if  $\min(M, N)$  is odd, there is also a one-dimensional array in the matrix. Let  $d$  denote the total number of rings and one-dimensional arrays of a matrix  $T$ , and then use a sequence  $s$  of length  $d$  to set the cyclic shift distance of each ring and one-dimensional array to achieve the RCLR operation of the matrix. The RCLR operation is presented as:  $RCLR(T, s)$ .

The entire flow of Rubik's Cube-like confusion is described below.

*Step 1* For the  $M \times N \times 3$  input pixel matrix  $D$  of Rubik's Cube confusion, there are 3 matrix slices of

$M \times N$  size on the  $x0z$  plane,  $M$  matrix slices of  $N \times 3$  size on the  $x0y$  plane, and  $N$  matrix slices of  $M \times N$  size on the  $y0z$  plane, these matrix slices are labeled and arranged as  $D^1 \dots D^3, D^4 \dots D^{3+M}, D^{4+M} \dots D^{3+M+N}$ . The  $d$  of the slice matrix on the three planes is represented by  $d_x, d_y,$  and  $d_z$  respectively, where  $d$  is the total number of rings and one-dimensional arrays in the slice matrix.

*Step 2* Iterate the 3DVSCS map  $M \times N \times 2 + 1000$  times and discard the first 1000 times to avoid the influence of the transient state. Then we get three sequences  $x, y$  and  $z$ , which are further scaled to obtain three integer sequences  $x_a, y_a, z_a$  as follows.

$$\begin{cases} x_a = [2 \times (M + N) \times x(1 : 3 \times \max(M, N) \times 2)] \\ y_a = [2 \times (M + N) \times y(1 : 3 \times \max(M, N) \times 2)] \\ z_a = [2 \times (M + N) \times z(1 : 3 \times \max(M, N) \times 2)] \end{cases} \tag{9}$$

*Step 3* Since the rotation of each layer of the Rubik’s Cube is out of order, a random sequence needs to be used to control the rotation order of the pixel matrix slices. Iterate the Logistic map  $(M + N + 3) \times 2 + 1000$  times and discard the first 1000 times to avoid the influence of the transient state. Divide the generated sequence into two parts and sort each part separately to obtain the index sequence  $c$ , and the two parts in  $c$  are used for 2 rounds of confusion–diffusion operation.

*Step 4* Perform RCLR operations on each slice of  $D$  in the order specified by sequence  $c$ . The detailed process is as follows,

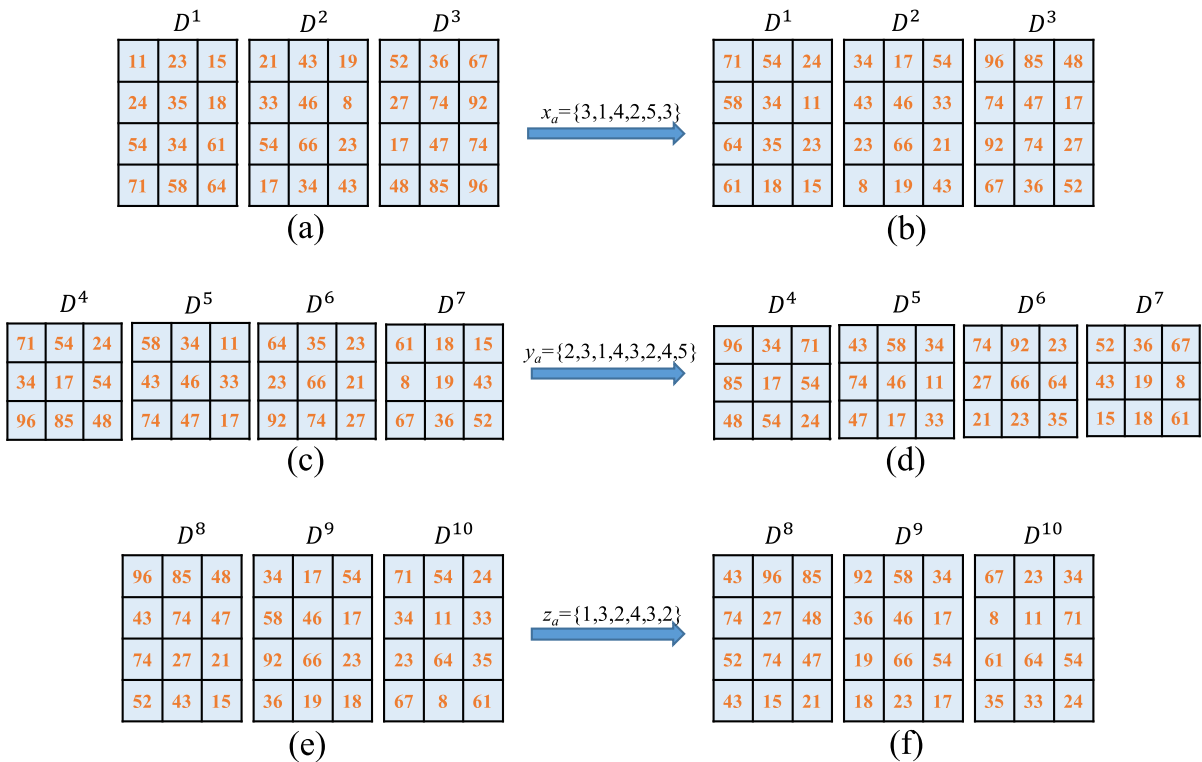
$$D_{n+1} = \begin{cases} \text{RCLR}(D_n^{c_n}, x_a(1 + (n - 1) \times d_x : n \times d_x)), & 1 < c_n \leq 3 \\ \text{RCLR}(D_n^{c_n}, y_a(1 + (n - 1) \times d_y : n \times d_y)), & 3 < c_n \leq 3 + M \\ \text{RCLR}(D_n^{c_n}, z_a(1 + (n - 1) \times d_z : n \times d_z)), & 3 + M < c_n \leq 3 + M + N \end{cases} \tag{10}$$

where  $n$  is from 1 to  $3 + M + N$ , and  $d_x, d_y, d_z$  are the  $d$  values of the slice matrices on the three planes of  $D$  respectively.  $D_n^{c_n}$  represents the  $c_n$ -th slice of the matrix  $D$  at the  $n$ -th rotation. The confusion result  $P$  is equal to  $D_{M+N+4}$ .

To further illustrate the Rubik’s Cube-like confusion, an image of size  $4 \times 3 \times 3$  is used as a numerical

example. Suppose that the three 3DVSCS sequences have been obtained as:  $x_a = \{3, 1, 4, 2, 5, 3\}$ ,  $y_a = \{2, 3, 1, 4, 3, 2, 4, 5\}$ ,  $z_a = \{1, 3, 2, 4, 3, 2\}$ , and for convenience, the sequence  $c$  is given as  $c = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ , that is, the rotation of the three planes is performed sequentially. The whole process is shown in Fig. 9.

- Figure 9a gives the slice matrices ( $D^1, D^2, D^3$ ) on the  $x0z$  surface of the initial image. The outermost circle of  $D^1$  is (11, 23, 15, 18, 61, 64, 58, 71, 54, 24), and the corresponding value in the sequence  $x_a$  representing the shift distance is 3. Then, the outermost circle performs a cyclic shift of distance 3 to obtain the sequence (71, 54, 24, 11, 23, 15, 18, 61, 64, 58). By performing this Rubik’s Cube-like rotation determined by the sequence  $x_a$  on the three matrices  $D^1, D^2,$  and  $D^3$ , respectively, the result of the rotation is obtained and displayed in Fig. 9b.
- The four slice matrices ( $D^4, D^5, D^6, D^7$ ) on the  $x0y$  surface of the pixel matrix obtained after the rotation of the  $x0z$  surface are shown in Fig. 9c. The outermost circle of the  $D^4$  matrix is (71, 54, 24, 54, 48, 85, 96, 34), and the corresponding value in the sequence  $y_a$  representing the shift distance is 2. Then, the outermost circle performs a cyclic shift of distance 2 to obtain the sequence (96, 34, 71, 54, 24, 54, 48, 85). By performing this Rubik’s Cube-like rotation determined by the sequence  $y_a$  on the four matrices  $D^4, D^5, D^6,$  and  $D^7$ , respectively, the result of the rotation is obtained and displayed in Fig. 9d.
- The three slice matrices ( $D^8, D^9, D^{10}$ ) on the  $y0z$  surface of the pixel matrix obtained after the rotation of the  $x0y$  surface are shown in Fig. 9e. The outermost circle of the  $D_8$  matrix is (96, 85, 48, 47, 21, 15, 43, 52, 74, 43), and the corresponding value in the sequence  $z_a$  representing the shift distance is 1. Then, the outermost circle performs a cyclic shift of distance 1 to obtain the sequence (43, 96, 85, 48, 47, 21, 15, 43, 52, 74). By performing this Rubik’s Cube-like rotation on the three matrices  $D^8, D^9, D^{10}$ , respectively, whose cyclic shift distance is determined by the sequence  $z_a$ , the result of the rotation is obtained and displayed in Fig. 9f.



**Fig. 9** The Rubik’s Cube-like confusion example

4.3 Iterative diffusion

Merely shuffling the positions of adjacent pixels does not provide sufficient security. In order to hide the statistical properties of plaintext, a diffusion operation is also required. In 3DVSCS-IES, an iterative diffusion method is applied to ensure that small changes in the plaintext image can result in a completely different ciphertext image. By further scaling the three sequences  $x, y, z$  generated by the 3DVSCS map as follows, we get three chaotic sequences  $x_b, y_b, z_b$  which are respectively used to change the pixel values of the three color channels.

$$\begin{cases} x_b = \lceil M \times N \times x(1 : 2 \times M \times N) \rceil \\ y_b = \lceil M \times N \times y(1 : 2 \times M \times N) \rceil \\ z_b = \lceil M \times N \times z(1 : 2 \times M \times N) \rceil \end{cases} \quad (11)$$

The current pixel is then changed by the previous pixel and the corresponding chaotic sequence value. Suppose both permutation result  $P$  and diffusion result  $D$  are of size  $M \times N \times 3$ , and transform all channels of  $P$

and  $D$  into sequences of length  $M \times N$ , denoted by  $P_R, P_G, P_B, D_R, D_G, D_B$ . The iterative diffusion can be described as follows:

$$\begin{aligned} R : & \begin{cases} D_R(n) = P_R(n) + x_b(n) \pmod{256}, & n = 1 \\ D_R(n) = D_R(n - 1) + P_R(n) + x_b(n) \pmod{256}, & n > 1 \end{cases} \\ G : & \begin{cases} D_G(n) = D_R(M * N) + P_G(n) + y_b(n) \pmod{256}, & n = 1 \\ D_G(n) = D_G(n - 1) + P_G(n) + y_b(n) \pmod{256}, & n > 1 \end{cases} \\ B : & \begin{cases} D_B(n) = D_G(M * N) + P_B(n) + z_b(n) \pmod{256}, & n = 1 \\ D_B(n) = D_B(n - 1) + P_B(n) + z_b(n) \pmod{256}, & n > 1 \end{cases} \end{aligned} \quad (12)$$

where the first pixel in the  $G$  channel of  $P$  is diffused by the last pixel in the  $R$  channel of  $P$ . Similarly, the first pixel in the  $B$  channel of  $P$  is diffused by the last pixel in the  $G$  channel of  $P$ . By this operation, subtle changes that occur in the  $R$  channel can be passed to the  $G$  and  $B$  channels.

### 4.4 Decryption process

3DVSCS-IES iterates the confusion–diffusion operation two times in the encryption process, so in the decryption process, it is necessary to perform the inverse operation of diffusion and then the inverse operation of confusion on the cipher image, and cycle the above operations two times to obtain the plain image.

The diffusion process is from the confusion result  $P$  to the diffusion result  $D$ , then the inverse process of diffusion is the process from  $D$  to  $P$ , where in the first iteration of decryption, the diffusion result  $D$  represents the cipher image. Still assuming that the size of  $P$  and  $D$  are both  $M \times N \times 3$ , and transform all channels of  $P$  and  $D$  into sequences of length  $M \times N$  denoted by  $P_R, P_G, P_B, D_R, D_G, D_B$ . The inverse operation of the iterative diffusion is described as follows.

$$\begin{aligned}
 B : & \begin{cases} P_B(n) = D_B(n) - D_B(n - 1) - z_b(n) \bmod 256, & n > 1 \\ P_B(n) = D_B(n) - D_G(M * N) - z_b(n) \bmod 256, & n = 1 \end{cases} \\
 G : & \begin{cases} P_G(n) = D_G(n) - D_G(n - 1) - y_b(n) \bmod 256, & n > 1 \\ P_G(n) = D_G(n) - D_R(M * N) - y_b(n) \bmod 256, & n = 1 \end{cases} \\
 R : & \begin{cases} P_R(n) = D_R(n) - D_R(n - 1) - x_b(n) \bmod 256, & n > 1 \\ P_R(n) = D_R(n) - x_b(n) \bmod 256, & n = 1 \end{cases} \tag{13}
 \end{aligned}$$

The confusion process is from the plain image or the result  $D$  of the diffusion operation of the previous encryption iteration to the confusion result  $P$ , then the inverse operation of confusion in the decryption process is to obtain  $D$  from  $P$ , where  $D$  here represents the plain image in the second decryption iteration.

Same as the operation in the encryption process,  $P$  is divided and arranged into  $P^1 \dots P^3, P^4 \dots P^{3+M}, P^{4+M} \dots P^{3+M+N}$ , chaotic sequences  $x_a, y_a, z_a$  are generated by 3DVSCS, and the  $d$  of the slice matrix on the three planes is represented by  $dx, dy,$  and  $dz$  respectively, where  $d$  is the total number of rings and one-dimensional arrays in the slice matrix. We denote the inverse operation of RCLR as  $RCLR^{-1}(T, s)$ , which cyclically shifts the rings and the possibly one-dimensional array of the matrix  $T$  in the opposite direction of RCLR by the distance specified by the control sequence  $s$ . The inverse operation of Rubik’s Cube-like confusion is described below.

$$P_{n-1} = \begin{cases} RCLR^{-1}(P_n^{c_n}, z_a(1 + (n - 1) \times d_z : n \times d_z)), & 3 + M < c_n \\ & \leq 3 + M + N \\ RCLR^{-1}(P_n^{c_n}, y_a(1 + (n - 1) \times d_y : n \times d_y)), & 3 < c_n \leq 3 + M \\ RCLR^{-1}(P_n^{c_n}, x_a(1 + (n - 1) \times d_x : n \times d_x)), & 1 < c_n \leq 3 \end{cases} \tag{14}$$

where  $n$  is from  $3 + M + N$  to 1,  $P_n^{c_n}$  represents the  $c_n$ -th slice of the matrix  $P$  after rotation  $3 + M + N - n$  times. The result  $D$  of this reverse process of confusion is equal to  $P_0$ .

### 5 Simulation result

MatlabR2019a is used to implement the proposed image encryption system on a private computer configured as intel(R) Core(TM) i7-10700, CPU 2.90 GHZ, memory 16 GB, operation system Microsoft Windows 10. Test images in this section include Lena(256 × 256), Peppers(512 × 512), Mandrill(512 × 512), and San Diego(1024 × 1024). The 256-bit long key used in this simulation is:

```

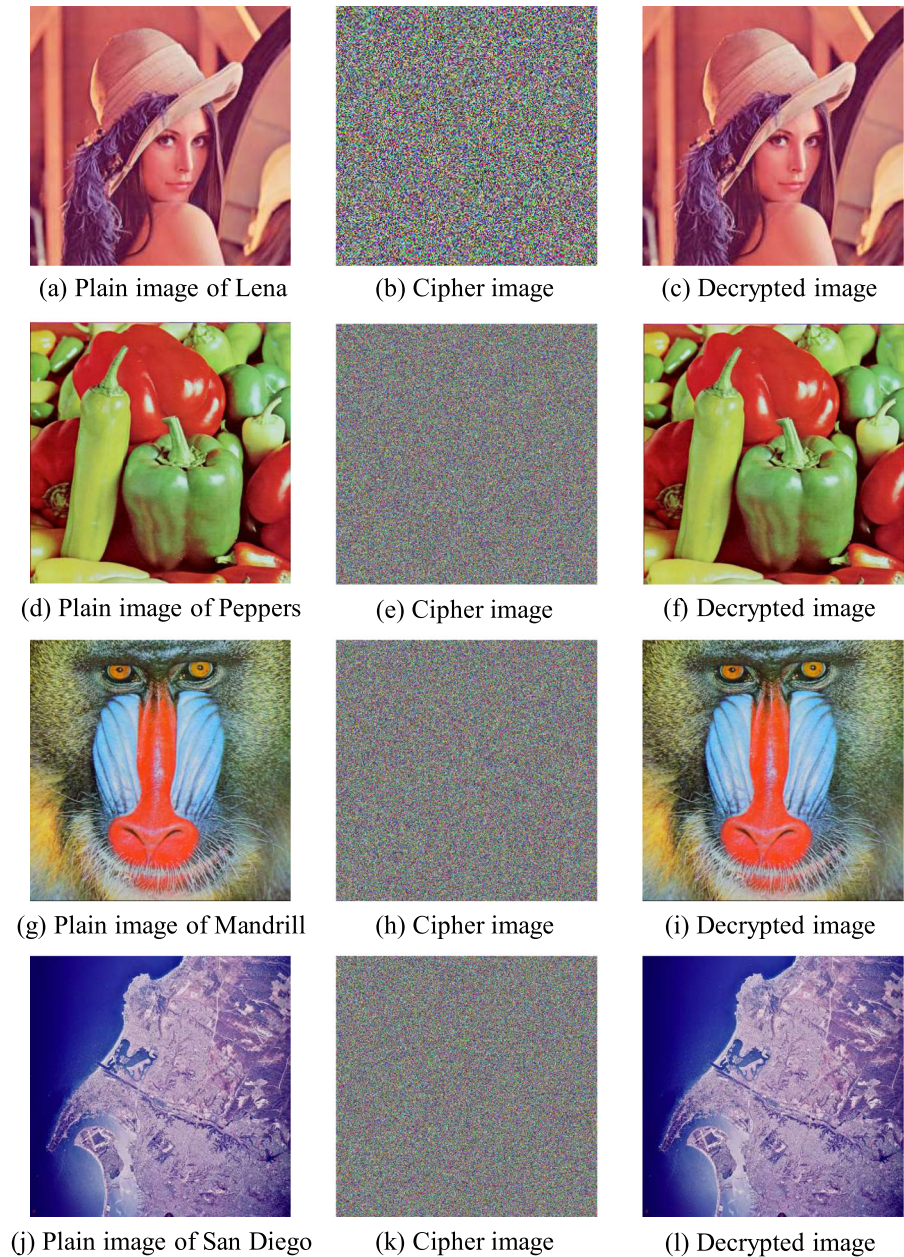
1E3A2453543FD22EB3FC48
A71EA26C34FD4A2E9136D90AF99B6BD6
4A24F5C6E7.
    
```

The simulation results are shown in Fig. 10, where the plain images are in the first column, the cipher images are in the second column, and the decrypt images are in the third column. It can be seen that the cipher images are all noise-like images and the decrypt images are the same as the plain images. Thus it can be concluded that our image encryption system can successfully encrypt and decrypt color images.

### 6 Security analysis

This section verifies the security of the proposed encryption system and its resistance to various types of attacks.

**Fig. 10** Simulation results of 3DVSCS-IES

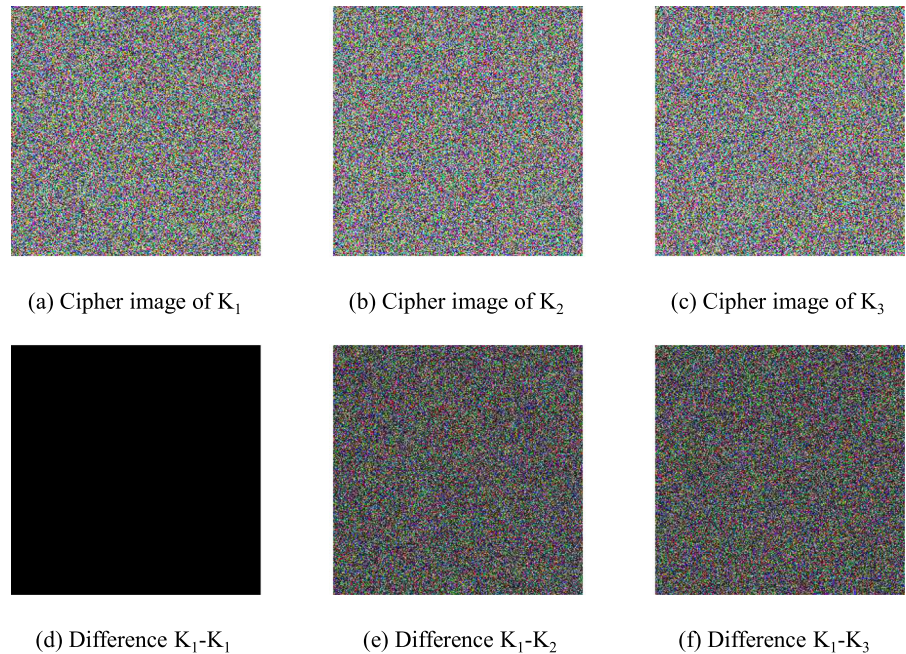


### 6.1 Key space

The key space refers to the set of all possible keys, and the size of the key space depends on the length of the security key, which is one of the most important characteristics that determine the strength of a cryptosystem. For a binary security key of length  $L$ , its key space size is  $2^L$ . If an attacker wants to attack the encryption system utilizing brute force, theoretically, it needs to calculate

$2^L$  times to ensure the success of the attack. Generally, the key space should be larger than  $2^{100}$  to provide high-security performance. 3DVSCS-IES employs keys of length 256, which can provide a key space with the size of  $2^{256}$  that is much larger than  $2^{100}$  and can effectively resist brute force attacks.

**Fig. 11** Key sensitivity analysis in encryption process



## 6.2 Key sensitivity

A qualified image encryption algorithm needs to be sensitive to keys in both the encryption process and the decryption process. Otherwise, there may be many equivalent keys, which seriously reduce the key space. Key sensitivity can be quantified by calculating the ratio of pixel grayscale differences between the ciphertext image generated under the slightly changed key and the ciphertext image generated with the original key. In the experiment, Lena ( $256 \times 256$ ) is chosen as the initial image, and the key given in Sect. 5 is used as the initial key,  $K_1$ . Two slightly changed keys  $K_2$ ,  $K_3$  are generated by changing one bit of  $K_1$ :

$$K_1 = 1E3A2453543FD22EB3FC48A71 \\ EA26C34FD4A2E9136D90AF99B6 \\ BD64A24F5C6E7$$

$$K_2 = 3E3A2453543FD22EB3FC48A71 \\ EA26C34FD4A2E9136D90AF99B6 \\ BD64A24F5C6E7$$

$$K_3 = 1E3A2453543FD22EB3FC48A71 \\ EA26C34FD4A2E9136D90AF99B6 \\ BD64A24F5C6E6$$

The result of the key sensitivity analysis in the encryption process is shown in Fig. 11, where (a), (b), and

(c) are the cipher images of Lena generated by  $K_1$ ,  $K_2$ ,  $K_3$ . (d), (e) and (f) are the difference between the three ciphertext images and the cipher image generated by the initial key, respectively. Table 3 lists the pixel grayness change ratio between the cipher image generated under the slightly changed key and the cipher image generated by the original key  $K_1$ . From Fig. 11 and Table 3, it can be observed that small changes to the key during encryption can result in a completely different cipher image, and the pixel grayness change ratios of the three color channels are all above 99%, so it can be concluded that 3DVSCS-IES has high key sensitivity in the encryption process.

The key sensitivity in the decryption process is analyzed by using  $K_2$  and  $K_3$  to decrypt the cipher image generated by the initial key  $K_1$  in Fig. 10b. The result is shown in Fig. 12, where (a), (b), and (c) are the images decrypted from Fig. 10b using  $K_1$ ,  $K_2$ , and  $K_3$ , respectively. Table 4 lists the pixel grayness change ratio between the image decrypted by the slightly changed key and the image decrypted correctly by the original key  $K_1$ . In Fig. 12 and Table 4, the small changes to the key during decryption can result in a completely different decrypted image, and the pixel grayness change ratios of the three color channels are all above 99%. Thus 3DVSCS-IES has high key sensitivity in the decryption process.

**Table 3** Pixel grayness change ratio of the cipher images

Image	Key	Pixel grayness change ratio		
		Red component	Green component	Blue component
Figure 10a	$K_1$	0	0	0
	$K_2$	99.6613	99.6124	99.6246
	$K_3$	99.6002	99.6262	99.6078

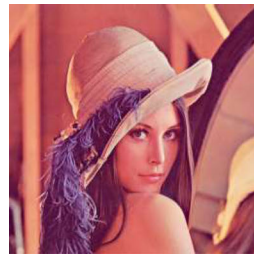
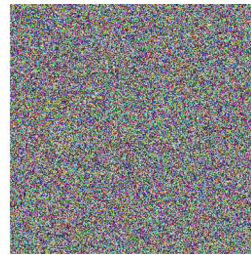
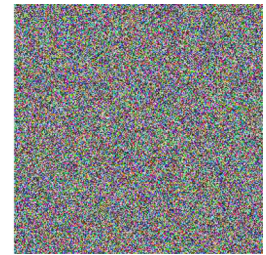
**Fig. 12** Key sensitivity analysis in decryption process(a) Decrypted image by  $K_1$ (b) Decrypted image by  $K_2$ (c) Decrypted image of  $K_3$ **Table 4** Pixel grayness change ratio of the decrypted images

Image	Key	Pixel grayness change ratio		
		Red component	Green component	Blue component
Figure 10b	$K_1$	0	0	0
	$K_2$	99.6170	99.6201	99.6170
	$K_3$	99.5972	99.6582	99.6094

We can summarize that 3DVSCS-IES owns high key security in both the encryption process and the decryption process.

### 6.3 The equality in key strength

Equal key strength means that all keys are required to be equally strong, preventing attackers from obtaining partial information of the plain image by decrypting the cipher image with keys close to the real key. Moreover, equal key strength can effectively avoid the avalanche effect (whenever the key used for decryption is close to the real key, the difference between the plaintext and the recovered plaintext tends to 0) [30, 31].

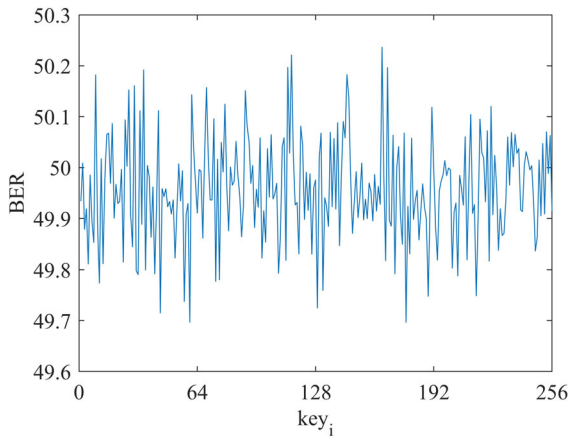
To evaluate the key equivalence of 3DVSCS-IES, we use  $K_1$  as the real key to encrypt Lena, use different keys around  $K_1$  to try to recover the plain image, and then calculate the Bit Error Ratio (BER) between the recovery result of each different key and Lena. The results are shown in Fig. 13, where  $key_i$  refers to the key obtained by inverting the  $i$ -th bit of the real key.

It can be clearly seen that even if there is only 1 bit difference from the real key, the BER value remains around 0.5. No matter how close to the correct key, it is impossible to obtain any partial information of the plain image, which means that 3DVSCS-IES has equal key strength.

### 6.4 Histogram analysis

The image histogram describes the distribution of pixel values in an image. For an RGB image, the three channels can be regarded as three  $M \times N$  matrices, and the value range of each element in the matrices is  $[0, 255]$ , then the image histogram  $h(g)$  is defined as the number of elements with pixel value  $g$ . The pixel distribution of a typical color image is generally uneven, and when it is encrypted, the original pixel distribution should be broken to present a uniform distribution, which means that the distribution information is hidden. Figure 14 shows the histograms of Lena and its cipher image, where (a), (b), (c) are the histograms of the three color





**Fig. 13** BER value of keys around  $K_1$

channels of Lena, and (d), (e), (f) are the histograms of the three color channels of the cipher image of Lena. We can obtain that the plain image has an uneven distribution while the ciphertext image has a uniform pixel distribution. Conclusively, 3DVSCS-IES can generate cipher images that hide the pixel distributions of the plain images.

### 6.5 Differential attack

Differential attack is a common way of cracking cipher images, and its essence is a selective plaintext attack. The attacker first makes minor changes to the plaintext image and then uses the proposed encryption method to encrypt the original plain image and the altered plain image separately. By comparing the two cipher images, the attacker can analyze the relationship between the original image and the cipher image, thereby cracking the cipher image. When the difference between the two cipher images is huge, it is difficult for the attacker to find the connection between the plain image and the cipher image, which makes it difficult to perform differential attacks. Therefore, whether an encryption method can resist differential attacks can be judged by its sensitivity to the plain image. The ability of encryption methods to resist differential attacks can be quantified by the number of pixels change rate (NPCR) and the unified average changing intensity (UACI) which are obtained by measuring the cipher images of the original plain image and slightly modified plain image. Equations (15)–(17) give the method to calculate the

values of NPCR and UACI.

$$NPCR = \frac{\sum_{i,j} D(i, j)}{M \times N} \times 100\%, \tag{15}$$

$$D(i, j) = \begin{cases} 1, & C(i, j) \neq C'(i, j) \\ 0, & C(i, j) = C'(i, j) \end{cases}, \tag{16}$$

$$UACI = \frac{1}{M \times N} \times \left[ \sum_{i,j} \frac{|C(i, j) - C'(i, j)|}{255} \right] \times 100\%, \tag{17}$$

where  $C$  and  $C'$  are the cipher images of the original plain image and the altered plain image, respectively, and  $M \times N$  is the size of each channel of the cipher images. For an image with an 8-bit grayscale value, the ideal NPCR value is 99.6094% and the ideal UACI value is 33.4635%. The closer the calculated values are to these two ideal values, the stronger the ability of the encryption method to resist differential attacks. The NPCR and UACI values of different test images are given in Table 5, and the NPCR and UACI values of our method and some other methods are compared in Table 6. It can be seen from the two tables that 3DVSCS-IES has values of NPCR and UACI that are closer to the theoretical values, which means our method is better at resisting differential attacks.

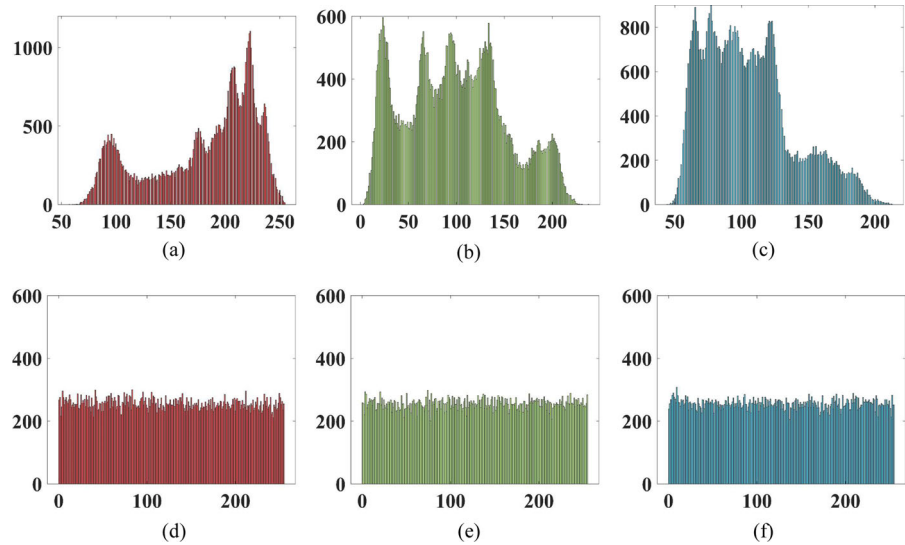
### 6.6 Correlation analysis

The correlation of adjacent pixels reflects the degree of correlation of pixel values in adjacent positions of the image. A qualified image encryption algorithm should reduce the correlation of adjacent pixels and try to achieve zero correlation to prevent attackers from analyzing the correlation of neighboring pixels to obtain adequate information about the image. The correlation coefficients  $r_{x,y}$  can be calculated as:

$$\begin{cases} r_{x,y} = \frac{E((x-E(x))(y-E(y)))}{\sqrt{D(x)D(y)}}, \\ E(x) = \frac{1}{N} \sum_{i=1}^N x_i, \\ D(x) = \frac{1}{N} \sum_{i=1}^N (x_i - E(x))^2, \end{cases} \tag{18}$$

where  $x_i$  and  $y_i$  are a couple of adjacent pixels,  $E(x)$  and  $D(x)$  are the expectation and variance of variable

**Fig. 14** Histograms of Lena and its cipher image: **a** Red component of Lena; **b** Green component of Lena; **c** Blue component of Lena; **d** Red component of cipher image; **e** Green component of cipher image; **f** Blue component of cipher image (color figure online)



**Table 5** NPCR and UACI results for different images

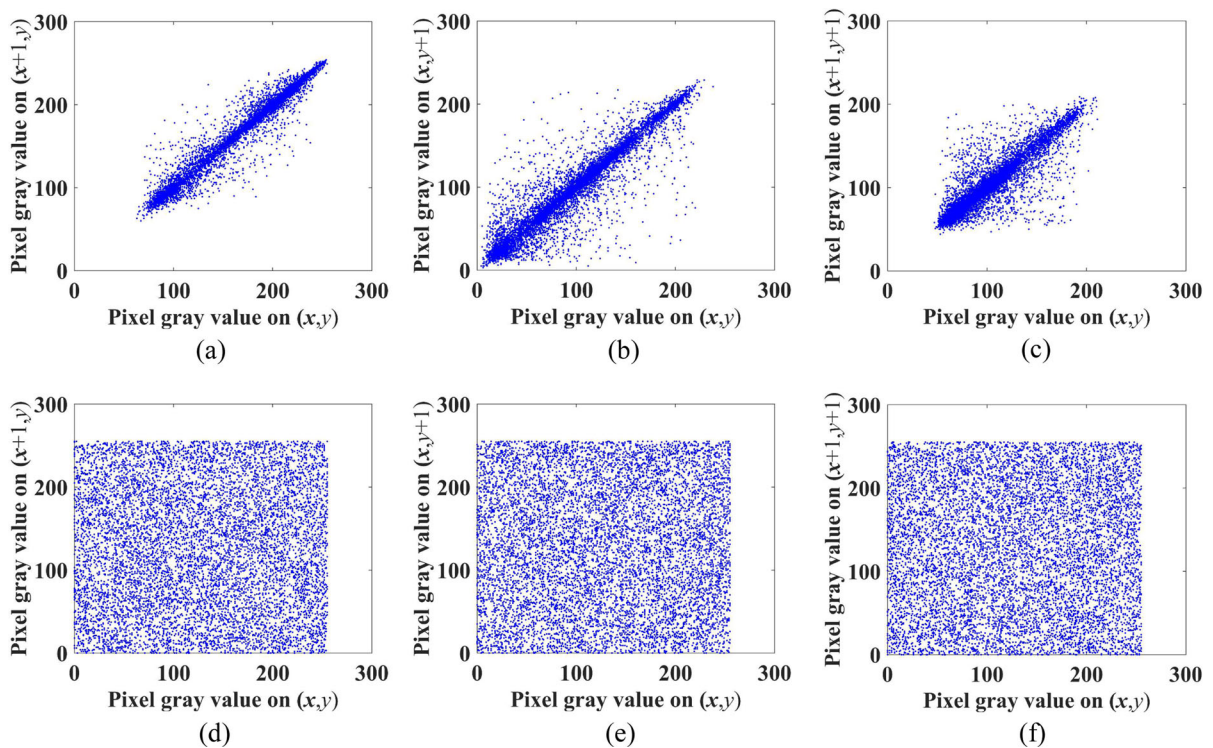
Image	NPCR (%)			UACI (%)		
	Red	Green	Blue	Red	Green	Blue
Lena	99.6140	99.6017	99.6140	33.5627	33.5218	33.4339
Peppers	99.5899	99.6037	99.6090	33.4482	33.4100	33.5403
Mandrill	99.6299	99.6037	99.6090	33.4236	33.4326	33.4886
San Diego	99.5971	99.6157	99.6060	33.4308	33.4589	33.4183
Mean	99.6077	99.6062	99.6095	33.4663	33.4558	33.4703

**Table 6** Comparison of NPCR and UACI results with other methods

Method	NPCR (%)			UACI (%)		
	Red	Green	Blue	Red	Green	Blue
Ours	99.6077	99.6062	99.6095	33.4663	33.4558	33.4703
[32]	99.6182	99.611	99.6114	33.4775	33.46	33.4564
[33]	99.6025	99.6077	99.6123	33.4710	33.4632	33.4597
[34]	99.60	99.61	99.60	33.55	33.44	33.45
[35]	99.6821	99.6827	99.6811	33.5002	33.4993	33.5167
[36]	99.73	99.66	99.68	33.26	34.13	33.28
[37]	99.6106	99.6085	99.6107	33.4784	33.4582	33.4673

$x$ , respectively, and  $r_{x,y}$  is the correlation coefficient. Generally, three aspects of the horizontal, vertical, and diagonal pixels of the image should be analyzed. Figure 15 gives the pixel distribution of the plain image Lena and its cipher image in H(horizontal), V(vertical), and D(diagonal) directions, where (a) shows the horizontal direction of red component of the plain image, (b) is the vertical direction of green component of the

plain image, (c) depicts the diagonal direction of blue component of the plain image, (d) draws the horizontal direction of red component of the cipher image, (e) is the vertical direction of green component of the cipher image, and (f) shows the diagonal direction of blue component of the cipher image. It can be seen that the pixel distribution of the plaintext image is concentrated near the diagonal, which means that the adja-



**Fig. 15** Pixels distribution of Lena and its cipher image: **a** horizontal direction of red component of Lena; **b** vertical direction of green component of Lena; **c** diagonal direction of blue component of Lena; **d** horizontal direction of red component of the

cipher image; **e** vertical direction of green component of the cipher image; **f** diagonal direction of blue component of the cipher image (color figure online)

cent pixels have a high correlation. In contrast, the pixel distribution of the ciphertext image is close to a uniform distribution, and the correlation is effectively removed. Table 7 lists the correlation coefficients of different images before and after encryption, and Table 8 compares the correlation coefficients of 3DVSCS-IES and other encryption algorithms after encrypting Lena. Obviously, the proposed encryption method can significantly reduce the correlation between adjacent pixels in cipher images, and our method has lower correlation coefficients relative to the other methods listed. It can be concluded that 3DVSCS-IES is better in removing the correlation of adjacent pixels.

## 6.7 Information entropy

Mathematically, information entropy is the expectation of the amount of information. It is an indicator used to evaluate and measure the amount of information in a piece of information and is often used to measure the randomness of cipher images. It is calculated as

follows:

$$H(m) = \sum_{i=0}^{2^n-1} p(m_i) \log \frac{1}{p(m_i)}, \quad (19)$$

where  $p(m_i)$  means the probability of  $m_i$ . For a digital image with a grayscale pixel value of 8 bits, the ideal information entropy should be 8. Table 9 gives the information entropy values of the cipher images generated by the 3DVSCS-IES and other methods. The table shows that the values of these information entropies are all close to the ideal value, and our method is significantly more relative to the ideal value than other methods, which means that 3DVSCS-IES has high security.

## 6.8 Noise and occlusion attack analysis

Images will inevitably suffer from data loss or noise during physical transmission, and if the encryption method is not robust, accurate plaintext informa-

**Table 7** Correlation coefficients of different images before and after encryption

Image	Direction	Correlation coefficients					
		Plain image			Cipher image		
		Red	Green	Blue	Red	Green	Blue
Lena	H	0.9507	0.9318	0.9128	0.0006	-0.0004	0.0001
	V	0.9755	0.9691	0.9482	-0.0012	-0.0007	0.0005
	D	0.9242	0.9098	0.8970	0.0008	0.0007	0.0006
Peppers	H	0.9621	0.9789	0.9616	-0.0005	-0.0004	-0.0006
	V	0.9646	0.9774	0.9628	0.0004	0.0002	0.0003
	D	0.9513	0.9599	0.9401	0.0007	0.0004	0.0006
Mandrill	H	0.9224	0.8631	0.9079	-0.0003	0.0008	0.0008
	V	0.8624	0.7584	0.8788	-0.0002	-0.0002	0.0008
	D	0.8495	0.7317	0.8368	-0.0011	0.0005	0.0025
San Diego	H	0.9254	0.9178	0.9067	-0.0003	0.0006	0.0019
	V	0.9220	0.9151	0.9048	0.0003	0.0007	0.0025
	D	0.9032	0.8946	0.8822	-0.004	0.0002	-0.0003

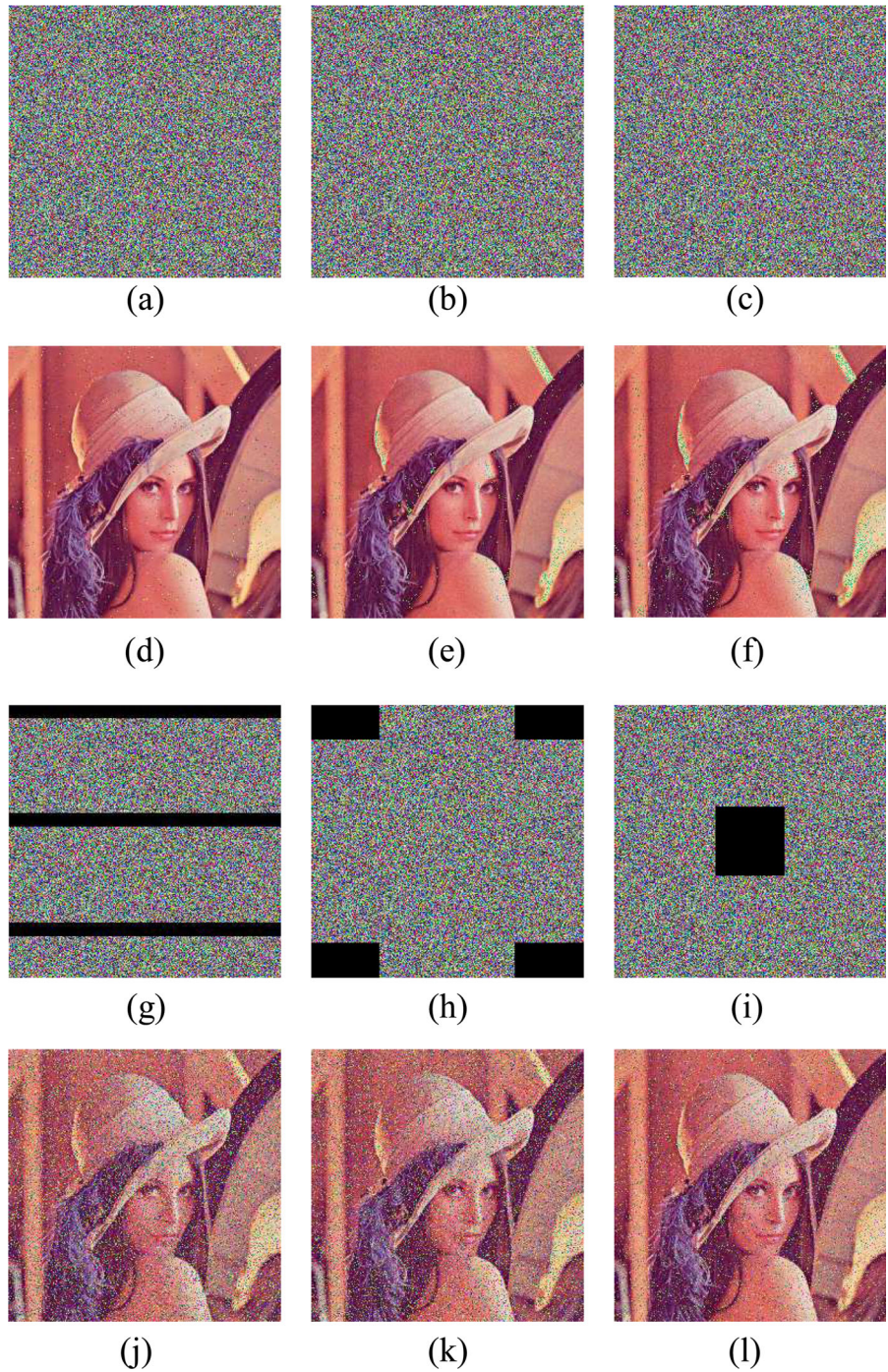
**Table 8** Comparing results of correlation coefficients for Lena

Method	Correlation coefficients			
	Horizontal	Vertical	Diagonal	Average
Plain image	0.9318	0.9643	0.9103	0.9355
Ours	0.0003	0.0008	0.0007	0.0006
[33]	0.003	0.0002	0.0001	0.0011
[34]	-0.0027	0.0033	-0.0035	0.0031
[38]	0.0016	0.0011	0.0011	0.0013
[39]	0.0076	0.0055	0.0032	0.0054
[40]	0.0173	0.0143	0.0085	0.0134
[41]	0.0002	0.0022	-0.0015	0.0013
[42]	0.0048	0.0029	0.0031	0.0036

**Table 9** Information entropy of cipher images of different encryption methods

Image	Information entropy		
	R	G	B
Lena	7.9976	7.9973	7.9971
Peppers	7.9991	7.9992	7.9992
Mandrill	7.9992	7.9993	7.9991
San Diego	7.9998	7.9998	7.9998
Lena [32]	7.9912	7.9913	7.9914
Lena [34]	7.9973	7.9969	7.9971
Lena [43]	7.9974	7.9974	7.9974

**Fig. 16** The resistance to noise and occlusion attack: **a** noisy images by SPN with density = 0.005; **b** noisy images by SN with density = 0.001; **c** noisy images by GN with density = 0.0005; **d** decrypted image of (a); **e** decrypted image of (b); **f** decrypted image of (c); **g** stripe clipping image; **h** corner clipping image; **i** center clipping image; **j** decrypted image of (g); **k** decrypted image of (h); **l** decrypted image of (i) (color figure online)



**Table 10** Running time of different images

Image	Size	Encryption time (s)	Decryption time (s)
Lena	256 × 256	0.426247	0.457733
Peppers	512 × 512	0.801175	0.893177
Mandrill	512 × 512	0.922003	0.841457
San Diego	1024 × 1024	2.425249	2.416450

**Table 11** Running time of different algorithms

Method	Encryption time (s)	Decryption time (s)
Ours	0.922003	0.841457
[32]	1.796105	0.847575
[33]	1.305322	1.026494
[34]	1.431	1.633
[44]	1.4933	7.8065
[45]	1.1168	3.2687
[46]	8.6078	9.6359

tion cannot be obtained after decryption. A qualified encryption algorithm is resistant to data loss and noise, and Fig. 16 shows the decryption results of the cipher images of Lena after suffering data loss or noise, where (a), (b), (c) are the noisy cipher images contaminated by Salt & Pepper noise (SPN) with density = 0.005, Speckle noise (SN) with density = 0.001, and Gaussian noise (GN) with density = 0.0005, respectively. (d), (e), (f) are the decrypted images of (a), (b), and (c), respectively. (g) is the stripe clipping image, (h) is the corner clipping image, and (i) is the center clipping image. (j), (k), and (l) are the decrypted image of (g), (h), and (i), respectively. It can be seen that the cropped or noised cipher image can still retain the primary information of the plain image after decryption. Therefore, the raised encryption algorithm is highly resistant to noise and occlusion attacks.

### 6.9 Runtime analysis

In addition to security, encryption and decryption speed is also an important aspect in measuring the performance of an encryption algorithm. Table 10 lists the time required to encrypt and decrypt different images using 3DVSCS-IES, and Table 11 compares the running time of 3DVSCS-IES with other algorithms on

Peppers(512 × 512). As can be seen from the table, our encryption algorithm has excellent running speed compared with other encryption methods.

## 7 Conclusion

This paper presents a 3DVSCS that has innovative and mutational dynamical behaviors. The structure is transformed by switching two coupling matrices in each iteration. The chaotic performance of 3DVSCS is analyzed by the phase diagram, bifurcation diagram, Lyapunov exponent, approximate entropy, and randomness test. The analysis results demonstrate that 3DVSCS has complex chaotic behavior and can be applied to encryption algorithms. By using 3DVSCS as the core chaotic map, we further propose an image encryption algorithm, 3DVSCS-IES, which follows the well-known confusion–diffusion mechanism. Because the generated chaotic sequences have good properties, a better efficiency Rubik’s Cube-like permutation method and an iterative diffusion algorithm are raised to realize confusion and diffusion, respectively. The simulation results show that 3DVSCS-IES can effectively encrypt and decrypt color images of different sizes. Then various security tests on the presented cryptosystem are performed, including key space, key sensitivity, key strength equality, histogram analysis, the ability to defend the differential attack, correlation analysis, and information entropy. Comparisons with some advanced methods are also given. The analysis and comparison results show that 3DVSCS-IES has high-security performance and surpasses some typical state-of-art methods. Our future work will investigate more complex variable structure mechanisms on chaotic systems and corresponding hardware implementation.

**Funding** This work is supported by Key R&D Program of Hubei Province (Grant No. 2020BAB104), the National Natural Science Foundation of China (Grant No. 62202183).

**Data availability** The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

#### Declarations

**Conflict of interest** The authors declare that they have no conflict of interest concerning the publication of this manuscript.

#### References

- Walker, S.J.: *Big Data: A Revolution That Will Transform How We Live, Work, and Think*, vol. 33, pp. 181–183. Taylor & Francis, London (2014)
- Li, X.W., Lee, I.K.: Robust copyright protection using multiple ownership watermarks. *Opt. Express* **23**(3), 3035–3046 (2015)
- Zhang, L.Y., Liu, Y., Pareschi, F., Zhang, Y., Wong, K.W., Rovatti, R., Setti, G.: On the security of a class of diffusion mechanisms for image encryption. *IEEE Trans. Cybern.* **48**(4), 1163–1175 (2017)
- Dragoi, I.C., Coltuc, D.: On local prediction based reversible watermarking. *IEEE Trans. Image Process.* **24**(4), 1244–1246 (2015)
- Nissenbaum, H.: The meaning of anonymity in an information age. *Inf. Soc.* **15**(2), 141–144 (1999)
- Cheddad, A., Condell, J., Curran, K., et al.: Digital image steganography: survey and analysis of current methods. *Signal Process.* **90**(3), 727–752 (2010)
- Li, X., Xiao, D., Wang, Q.H.: Error-free holographic frames encryption with CA pixel-permutation encoding algorithm. *Opt. Lasers Eng.* **100**, 200–207 (2018)
- Yegireddi, R., Kumar, R. K.: A survey on conventional encryption algorithms of Cryptography. In: 2016 International Conference on ICT in Business Industry & Government (ICTBIG), pp. 1–4. IEEE (2016)
- Ahmad, I., Shin, S.: A novel hybrid image encryption-compression scheme by combining chaos theory and number theory. *Signal Process. Image Commun.* **98**, 116418 (2021)
- Shand, M., Vuillemin, J.: Fast implementations of RSA cryptography. In: *Proceedings of IEEE 11th Symposium on Computer Arithmetic*, pp. 252–259. IEEE (1993)
- Pub F.: Data Encryption Standard (DES). FIPS PUB. 46-3 (1999)
- Heron, S.: Advanced encryption standard (AES). *Netw. Secur.* **12**, 8–12 (2009)
- Basu, S.: International data encryption algorithm (IDEA)—a typical illustration. *J. Glob. Res. Comput. Sci.* **2**(7), 116–118 (2011)
- Zhang, Y., Xiao, D.: An image encryption scheme based on rotation matrix bit-level permutation and block diffusion. *Commun. Nonlinear Sci. Numer. Simul.* **19**(1), 74–82 (2014)
- Hua, Z., Zhou, Y., Huang, H.: Cosine-transform-based chaotic system for image encryption. *Inf. Sci.* **480**, 403–419 (2019)
- Jolfaei, A., Wu, X.W., Muthukkumarasamy, V.: On the security of permutation-only image encryption schemes. *IEEE Trans. Inf. Forensics Secur.* **11**(2), 235–246 (2015)
- Faragallah, O.S., El-sayed, H.S., Afifi, A., El-Shafai, W.: Efficient and secure opto-cryptosystem for color images using 2D logistic-based fractional Fourier transform. *Opt. Lasers Eng.* **137**, 106333 (2021)
- Chen, J., Zhu, Z.L., Zhang, L.B., Zhang, Y., Yang, B.Q.: Exploiting self-adaptive permutation-diffusion and DNA random encoding for secure and efficient image encryption. *Signal Process.* **142**, 340–353 (2018)
- Kocarev, L.: Chaos-based cryptography: a brief overview. *IEEE Circuits Syst. Mag.* **1**(3), 6–21 (2001)
- Hua, Z., Jin, F., Xu, B., Huang, H.: 2D Logistic-Sine-coupling map for image encryption. *Signal Process.* **149**, 148–161 (2018)
- Zheng, J., Hu, H.: Bit cyclic shift method to reinforce digital chaotic maps and its application in pseudorandom number generator. *Appl. Math. Comput.* **420**, 126788 (2022)
- Zheng, J., Hu, H., Ming, H., Liu, X.: Theoretical design and circuit implementation of novel digital chaotic systems via hybrid control. *Chaos Solitons Fractals* **138**, 109863 (2020)
- Wu, Y., Noonan, J.P., Yang, G., Jin, H.: Image encryption using the two-dimensional logistic chaotic map. *J. Electron. Imaging* **21**(1), 013014 (2012)
- Liu, M., Zhang, S., Fan, Z., Qiu, M.:  $H_\infty$  State estimation for discrete-time chaotic systems based on a unified model. *IEEE Trans. Syst. Man Cybern. Part B (Cyber.)* **42**(4), 1053–1063 (2012)
- Lin, L., Shen, M., So, H.C., Chang, C.: Convergence analysis for initial condition estimation in coupled map lattice systems. *IEEE Trans. Signal Process.* **60**(8), 4426–4432 (2012)
- Srivastava, A.N., Das, S.: Detection and prognostics on low-dimensional systems. *IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.)* **39**(1), 44–54 (2008)
- Xia, X., Zheng, J.: A novel method to improve the dynamical degradation of digital chaotic systems. In: 2018 3rd International Conference on Mechanical, Control and Computer Engineering (ICMCCE), pp. 379–384. IEEE (2018)
- Chen, Z., Yuan, X., Yuan, Y., Iu, H.H.C., Fernando, T.: Parameter identification of chaotic and hyper-chaotic systems using synchronization-based parameter observer. *IEEE Trans. Circuits Syst. I Regul. Pap.* **63**(9), 1464–1475 (2016)
- Zeraouia, E.: *Robust Chaos and its Applications*, vol. 79. World Scientific (2012)
- Merah, L., Ali-Pacha, A., Hadj-Said, N.: Real-time cryptosystem based on synchronized chaotic systems. *Nonlinear Dyn.* **82**(1), 877–890 (2015)
- Merah, L., Adnane, A., Ali-Pacha, A., Ramdani, S., Hadj-said, N.: Real-time implementation of a chaos based cryptosystem on low-cost hardware. *Iran. J. Sci. Technol. Trans. Electr. Eng.* **45**(4), 1127–1150 (2021)
- Teng, L., Wang, X., Yang, F., Xian, Y.: Color image encryption based on cross 2D hyperchaotic map using combined cycle shift scrambling and selecting diffusion. *Nonlinear Dyn.* **105**(2), 1859–1876 (2021)
- Qiu, H., Xu, X., Jiang, Z., Sun, K., Xiao, C.: A color image encryption algorithm based on hyperchaotic map and Rubik's Cube scrambling. *Nonlinear Dyn.* 1–19 (2022)

34. Chai, X., Fu, X., Gan, Z., Lu, Y., Chen, Y.: A color image cryptosystem based on dynamic DNA encryption and chaos. *Signal Process.* **155**, 44–62 (2019)
35. Seyedzadeh, S.M., Mirzakuchaki, S.: A fast color image encryption algorithm based on coupled two-dimensional piecewise chaotic map. *Signal Process.* **92**(5), 1202–1215 (2012)
36. Liao, X., Lai, S., Zhou, Q.: A novel image encryption algorithm based on self-adaptive wave transmission. *Signal Process.* **90**(9), 2714–2722 (2010)
37. Wu, X., Kan, H., Kurths, J.: A new color image encryption scheme based on DNA sequences and multiple improved 1D chaotic maps. *Appl. Soft Comput.* **37**, 24–39 (2015)
38. Sun, J.: A chaotic image encryption algorithm combining 2D chaotic system and random XOR diffusion. *Phys. Scr.* **96**(10), 105208 (2021)
39. Jasra, B., Moon, A.H.: Color image encryption and authentication using dynamic DNA encoding and hyper chaotic system. *Expert Syst. Appl.* **206**, 117861 (2022)
40. Wu, X., Wang, K., Wang, X., Kan, H., Kurths, J.: Color image DNA encryption using NCA map-based CML and one-time keys. *Signal Process.* **148**, 272–287 (2018)
41. Wang, X., Gao, S.: Image encryption algorithm based on the matrix semi-tensor product with a compound secret key produced by a Boolean network. *Inf. Sci.* **539**, 195–214 (2020)
42. ur Rehman, A., Liao, X., Ashraf, R., Ullah, S., Wang, H.: A color image encryption technique using exclusive-OR with DNA complementary rules based on chaos theory and SHA-2. *Optik* **159**, 348–367 (2018)
43. Hua, Z., Zhu, Z., Chen, Y., Li, Y.: Color image encryption using orthogonal Latin squares and a new 2D chaotic system. *Nonlinear Dyn.* **104**(4), 4505–4522 (2021)
44. Ahmad, I., Shin, S.: A novel hybrid image encryption-compression scheme by combining chaos theory and number theory. *Signal Process. Image Commun.* **98**, 116418 (2021)
45. Chai, X., Bi, J., Gan, Z., Liu, X., Zhang, Y., Chen, Y.: Color image compression and encryption scheme based on compressive sensing and double random encryption strategy. *Signal Process.* **176**, 107684 (2020)
46. Xuejing, K., Zihui, G.: A new color image encryption scheme based on DNA encoding and spatiotemporal chaotic system. *Signal Process. Image Commun.* **80**, 115670 (2020)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.