

Fractal sorting vector-based least significant bit chaotic permutation for image encryption

Yong-Jin Xian(咸永锦), Xing-Yuan Wang(王兴元), Ying-Qian Zhang(张盈谦), Xiao-Yu Wang(王晓雨), and Xiao-Hui Du(杜晓慧)

Citation: Chin. Phys. B, 2021, 30 (**6**): 060508. DOI: 10.1088/1674-1056/abda35
Journal homepage: http://cpb.iphy.ac.cn; http://iopscience.iop.org/cpb

What follows is a list of articles you may be interested in

An image encryption algorithm based on improved baker transformation and chaotic S-box

Xing-Yuan Wang(王兴元) , Huai-Huai Sun(孙怀怀) , and Hao Gao(高浩)
Chin. Phys. B, 2021, 30 (**6**): 060507. DOI: 10.1088/1674-1056/abdea3

Control of chaos in Frenkel-Kontorova model using reinforcement learning

You-Ming Lei(雷佑铭) and Yan-Yan Han(韩彦彦)
Chin. Phys. B, 2021, 30 (**5**): 050503. DOI: 10.1088/1674-1056/abd74f

A multi-directional controllable multi-scroll conservative chaos generator: Modelling, analysis, and FPGA implementation

En-Zeng Dong(董恩增), Rong-Hao Li(李荣昊), and Sheng-Zhi Du(杜升之)
Chin. Phys. B, 2021, 30 (**2**): 020505. DOI: 10.1088/1674-1056/abc239

A novel method of constructing high-dimensional digital chaotic systems on finite-state automata

Jun Zheng(郑俊), Han-Ping Hu(胡汉平)
Chin. Phys. B, 2020, 29 (**9**): 090502. DOI: 10.1088/1674-1056/aba60f

Hidden attractors in a new fractional-order discrete system: Chaos, complexity, entropy, and control

Adel Ouannas, Amina Aicha Khennaoui, Shaher Momani, Viet-Thanh Pham, Reyad El-Khazali
Chin. Phys. B, 2020, 29 (**5**): 050504. DOI: 10.1088/1674-1056/ab820d

---

Fractal sorting vector-based least significant bit chaotic permutation for image encryption*

Yong-Jin Xian(咸永锦)[1], Xing-Yuan Wang(王兴元)[1,†], Ying-Qian Zhang(张盈谦)[2], Xiao-Yu Wang(王晓雨)[1], and Xiao-Hui Du(杜晓慧)[1]

[1] *School of Information Science and Technology, Dalian Maritime University, Dalian* 116026, *China*
[2] *School of Information Science and Technology, Xiamen University Tan Kah Kee College, Zhangzhou* 363105, *China*

(Received 2 December 2020; revised manuscript received 10 January 2021; accepted manuscript online 11 January 2021)

The image's least significant bit (LSB) covers lots of the details that have been commonly used in image encryption analysis. The newly proposed fractal sorting vector (FSV) and FSV-based LSB chaotic permutation (FSV-LSBCP) is a novel chaotic image encryption cryptosystem introduced in this article. The FSV-LSBCP effectively strengthens the security of the cryptographic scheme concerning the properties of the FSV. Key analysis, statistical analysis, resistance differential attack analysis, and resistance to cropping attacks and noise attacks are the focus of the suggested image encryption cryptosystem. The security experiment shows that the cryptosystem is adequate to achieve the desired degree of security.

Keywords: chaotic image encryption, fractal sorting vector, bit-level permutation, FSV-based LSB chaotic permutation

PACS: 05.45.Gg, 05.45.Df, 07.05.Pj, 05.45.Df **DOI:** 10.1088/1674-1056/abda35

1. Introduction

Research on image security has gained widespread publicity in recent years.[1–4] With the development of cryptography as an important image authentication tool, image encryption has attracted a great deal of attention.[5,6] The researchers incorporated the chaos technique into image encryption and provided several useful chaotic image encryption studies.[7–17]

Image encryption algorithm based on chaos consists of pixel scrambling and pixel diffusion within the context of the cryptanalysis image encryption scheme of Fridrich.[18] In the design of chaotic image encryption algorithms, bit-level permutation simultaneously realized the pixel uncertainty and pixel diffusion, which enhances the security of image encryption algorithm.[19–22] In Ref. [23], the original image was divided into 8 binary images, then the 8 binary images were encrypted respectively, and then synthesized into a gray-scale encrypted image. In Refs. [24,25], a bit-level permutation approach was implemented to color the image encryption. The pixel-based bit exchange could exert a certain encryption effect, but it would do better with bit-level scrambling based on bit planes. In Ref. [26], the bit in the pixel was scrambled into its neighboring pixel, so that the direction of the bit varied between each plane. In Ref. [27], by applying rank and column encryption to each binary bit plane, the image algorithm realized the bit-level permutation. In Ref. [28], the 8 bits of each pixel were split into the four most important bits on the left and the four least significant bits on the right. The LSBs were independently encrypted. In Ref. [29], the right four bits of each pixel were scrambled separately, while the left four bits were encrypted as a whole part. In Ref. [30], the bit positions of the image's LSB planes were scrambled, and thus leaving the other planes on the left unchanged.

By analyzing the above-mentioned bit-level permutation methods, they can be divided into two categories: bit permutation on all bit planes and bit permutation on least significant bit planes. The bit permutation on LSB planes does not scramble all bits, while the result is almost the same as that on all planes. The normal LSB-based bit-level image encryption method only performs independent bit scrambling in each bit plane. There is no bit position conversion between bit layers in the process of scrambling. Therefore, there is also more space for progress in the encryption effect obtained by the present chaotic image encryption methods based on bit permutation on LSB planes. To solve these problems, a novel LSB chaotic permutation based on FSV is proposed to exchange all necessary bit positions of the image. The proposed method changes the bit positions of the four different planes of the LSB of the image and reduces the information about the relation among the four LSB planes that contain the main information about the image. The proposed bit-level permutation can not only increase the range of pixel values changed by bit permutation, but also improve the ability to resist cryptanalysis with the ap-

*Project supported by the National Natural Science Foundation of China (Grant No. 61672124), the Password Theory Project of the 13th Five-Year Plan National Cryptography Development Fund, China (Grant No. MMJJ20170203), the Liaoning Provincial Science and Technology Innovation Leading Talents Program Project, China (Grant No. XLYC1802013), the Key Research and Development Projects of Liaoning Province, China (Grant No. 2019020105-JH2/103), and the Jinan City 20-University Funding Projects for Introducing Innovation Team Program, China (Grant No. 2019GXRC031).

†Corresponding author. E-mail: xywang@dlmu.edu.cn

© 2021 Chinese Physical Society and IOP Publishing Ltd

plication of chaotic systems. It is worth noting that the iteratively generated FSV will enhance the algorithm's encryption performance to some degree.

The rest of this article is structured as follows. The basic principles and iterative algorithm of FSV, and chaotic LSB permutation based on FSV, are given in detail in Section 2. In Section 3, the FSV-LSBCP based image encryption and decryption algorithms are presented. In Section 4, the experimental results are discussed. Finally, some conclusions are drawn from the present study in Section 5.

2. Proposed methods

Inspired by Refs. [31–37], we introduce a new vector with fractal properties in order and analyze its characteristics in the application of image encryption in this section.

2.1. Basic concepts of FSV

Definition 1 A vector $V=\{v_1,v_2,\ldots,v_n\}\in\mathbb{R}^{1\times n}$ is called a sorting vector if the elements in the vector are composed of positive integers $1,2,\ldots,N$ and the element values at any two different positions are different. In other words, V is a sorting vector such that any of the following statements holds true:

$$\begin{cases} v_i\in\mathbb{N}^+, \quad i=1,2,\ldots,n,\\ \min(V)=1,\\ \max(V)=n,\\ v_a=v_b \quad \text{if and only if} \quad a=b. \end{cases} \tag{1}$$

Based on Definition 1, we define a class of vector with properties similar to the fractals as follows.

Definition 2 Sorting vector $V=\{v_1,v_2,\ldots,v_n\}\in\mathbb{R}^{1\times n}$ is called fractal sorting vector (FSV), if there are the following properties:

1) the distribution of elements in the vector is overall irregular;

2) the ordering of elements in each sub-vector with the same length is self-similar;

3) vector V can be generated by iteration.

2.2. Iterative algorithm of FSV

According to the definitions above, we give a class of fractal sorting vector construction method based on addition and multiplication between vector and number as follows.

Based on any initial vector $V^{[1]}\in\mathbb{R}^{1\times l(V^{[1]})}$, where $l(V)$ represents the length of the vector V, we iterate it according to the following steps to obtain FSV V^*.

Step 1 Calculate the iteration vector $V^{[2]}$ according to the following iteration rules:

$$\begin{cases} V^{[2]}=\{V^{[2]}\{1\},V^{[2]}\{2\},\ldots,V^{[2]}\{l(V^{[1]})\}\}\in\mathbb{R}^{1\times l(V^{[2]})},\\ V^{[2]}\{i\}=V_M^{[1]}\times(V^{[1]}(i)-1)+V^{[1]},\\ i=1,2,\ldots,l(V^{[1]}),\\ l(V^{[2]})=l(V^{[1]})\times l(V^{[1]})=l^2(V^{[1]}), \end{cases} \tag{2}$$

where $V\{i\}$ represents the i-th sub-vector of V, and $V(i)$ denotes the i-th element of V.

Step 2 Repeat the following general term formulas to obtain the higher-order FSV, $V^{[n]}$.

$$\begin{cases} V^{[n]}=\{V^{[n]}\{1\},V^{[n]}\{2\},\ldots,V^{[n]}\{l(V^{[n-1]})\}\}\in\mathbb{R}^{1\times l(V^{[n]})},\\ V^{[n]}\{i\}=V_M^{[n-1]}\times(V^{[n-1]}(i)-1)+V^{[1]},\\ i=1,2,\ldots,l(V^{[n-1]}),\\ l(V^{[n]})=l(V^{[n-1]})\times l(V^{[1]})=l^n(V^{[1]}). \end{cases} \tag{3}$$

Step 3 Obtain an FSV V^* of any length from the following equation:

$$V^*=\{v\in V^{[m]}|v\le l(V^*)\}, \tag{4}$$

where $V^{[m]}$ is a generated FSV calculated from steps 1 and 2, and $l(V^{[m]})\ge l(V^*)$.

Through the above steps, we can obtain the FSV V^* of any length, by combining the actual needs and the appropriate number of iterations. The relationship between FSV and the initial iteration vector can be expressed as a function:

$$V^*=f(V^{[1]},l(V^*)). \tag{5}$$

Based on these steps, we propose the pseudo-code to implement the iterative calculation of FSV as shown in Table 1.

Table 1. Pseudo-code of algorithm for FSV.

Algorithm 1. The pseudo-code of iteratively generating algorithm for FSV

```
Set initial vector V with the length n
While length(V)<length(V*), Do
  V_max ← max(V)% Calculate the maximum value of V.
  For j = 1 : n
    B{j} ← (V(j) − 1) × V_max + V
  End
  A ← [B{1}, B{2}, ..., B{n}]
  n ←length(V)
  i ← i + 1
End
For k = 1 : n
  If V(k) >=length(V*), Delate(V(k))
End
Return V* ← V
```

2.3. FSV based LSB chaotic permutation

The least significant bit of the image covers most of the information,[28] and the eight binary planes of peppers are shown in Fig. 1. The human eyes can clearly distinguish the information among the first four planes, but nothing from the last four. The information weight of the eight binary planes and their cumulative rates are shown in Table 2. The right four bits cumulatively contain 94.12% of the information. Processing the right four bits can be applied to the image encryption to improve the efficiency of the algorithm while ensuring security.

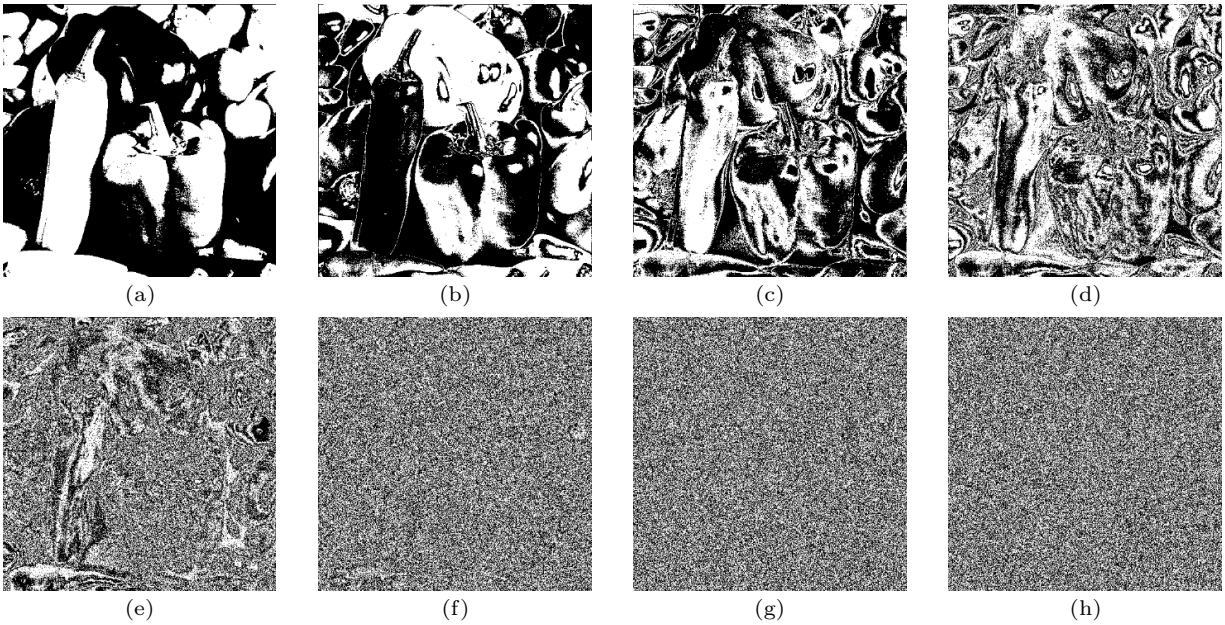


Fig. 1. Eight binary planes of peppers: (a) 8th, (b) 7th, (c) 6th, (d) 5th, (e) 4th, (f) 3rd, (g) 2nd, (h) 1st.

Table 2. Information weights of bit planes.

Bit plane	Weight	Rate (%)	Cumulative rate (%)
8th	128	50.20	50.20
7th	64	25.10	75.29
6th	32	12.55	87.84
5th	16	6.27	94.12
4th	8	3.14	97.25
3rd	4	1.57	98.82
2nd	2	0.78	99.61
1st	1	0.39	100.00

Based on FSV and LSB permutation, a novel FSV-based LSB chaotic permutation method is proposed as follows.

Step 1 Represent each pixel in grayscale image into 8 bits. And use a tensor T_S to represent the image represented by the bit.

Step 2 Denote the 8 layers of the tensor as P_8, P_7, P_6, P_5, P_4, P_3, P_2, P_1 in sequence, where P_8 is a binary image composed of the highest bit and P_1 is a binary image consisting of the lowest bit.

Step 3 Press the upper left, upper right, lower left, and lower right sub-block positions to compose P_8, P_7, P_6, P_5 into a binary image P_{LSB}.

Step 4 Exchange the position of each bit in P_{LSB} by the permutation vectors of the FSV and the ascending chaotic sequence index, successively.

Step 5 Split the new permuted binary image P^*_{LSB} obtained in Step 4 into P^*_8, P^*_7, P^*_6, P^*_5 according to the sub-block positions in Step 3.

Step 6 Restore P^*_8, P^*_7, P^*_6, P^*_5, P_4, P_3, P_2, P_1 to a new grayscale image.

With the above six steps, the FSV-based LSB shaotic permutation of the image is realized. For better understanding, the flow chart is shown in Fig. 2.

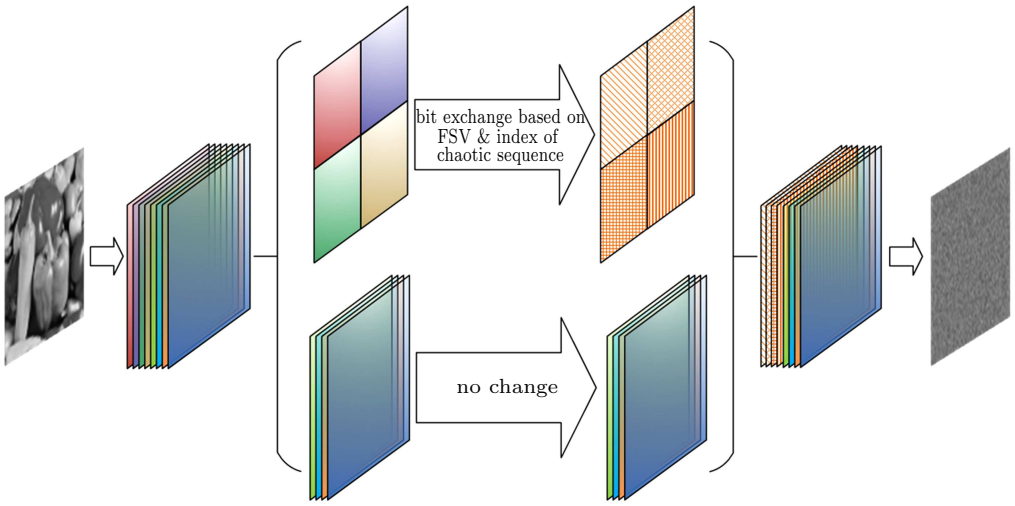


Fig. 2. Process of FSV-based LSB chaotic permutation.

3. Image encryption algorithm based on FSV-LSBCP

3.1. Two-dimensional chaotic system

In many chaotic systems, the Henon map has simple operations and can generate two chaotic sequences. And the Henon map is defined as[38]

$$\begin{cases} x_{n+1} = 1 - a \cdot x_n^2 + y_n, \\ y_{n+1} = b \cdot x_n, \end{cases} \tag{6}$$

where $a = 1.4$ and $b = 0.3$ make the map reach a chaotic state. The application of the Henon map can make the image encryption method more difficult to predict and efficient to operate.

3.2. Encryption method

Image encryption method based on FSV-LSBCP is introduced in detail step by step in this subsection. Due to the flexibility of the proposed FSV construction algorithm, the proposed encryption algorithm, which is suitable to the images of any size and shape, mainly consists of six steps as follows.

Step 1 Read the plaintext image into the computer and mark it as P.

Step 2 Obtain an FSV of the appropriate length according to the proposed construction algorithm.

Step 3 Generate a 128-bit character string Λ by using SHA-512 from P, and convert Λ into 3 decimals $K = [k_1,\ k_2,\ k_3]$ as follows:

$$k_\iota = \frac{a(2\iota - 1) + a(2\iota) + 1}{a(2\iota) + a(2\iota + 1) + 1}, \quad \iota = 1, 2, \tag{7}$$

$$k_3 = (a(6) + 1) \times (a(7) + a(8) + 1) + 1000, \tag{8}$$

$$a(i) = \overset{15}{\underset{j=0}{\oplus}} (\Lambda(i + 8j))_{10}, \quad i = 1, 2, \ldots, 8. \tag{9}$$

Step 4 Obtain the appropriate length chaotic sequences C_1 and C_2 by Henon map with the parameters $[k_1',\ k_2',\ k_3]$, where k_3 is the length of the deleted sequence to ensure chaotic state, while k_1' and k_2' are the initial parameters of chaotic system. Setting private key $[k_4,\ k_5]$, the parameters of the generated chaotic sequence are calculated as follows:

$$\begin{cases} k_1' = \mod(k_1 \times k_4, 1), \\ k_2' = \mod(k_2 \times k_5, 1), \end{cases} \tag{10}$$

where mod(,) is the remainder function, and $[k_4,\ k_5]$ is the private keystream.

Step 5 Permutate P into P' by FSV-LSBCP according to the FSV and the chaotic sequence C_1.

Step 6 Diffuse re-segmented image P' with C_2 by using the method shown below and denote the diffused image as $P'' = [p''(i)]$:

$$p''(i) = \begin{cases} p'(i) \oplus h(i), & i = 1, \\ p'(i) \oplus p''(i-1) \oplus h(i), & i \geq 2, \end{cases} \tag{11}$$

$$h(i) = \mod(\left\lfloor C_2(i) \times 7 \times 10^5 \right\rfloor, 256), \tag{12}$$

where $\oplus$ denotes XOR operation, and $\lfloor\ \rfloor$ is the round-up function.

The ciphertext image is obtained by using the above six steps. The key sequence consists of $[k_1, k_2, k_3, k_4, k_5]$, which are used for both image encryption and decryption. For easy understanding, flow charts and examples of the above basis are shown in Figs. 3 and 4, respectively.

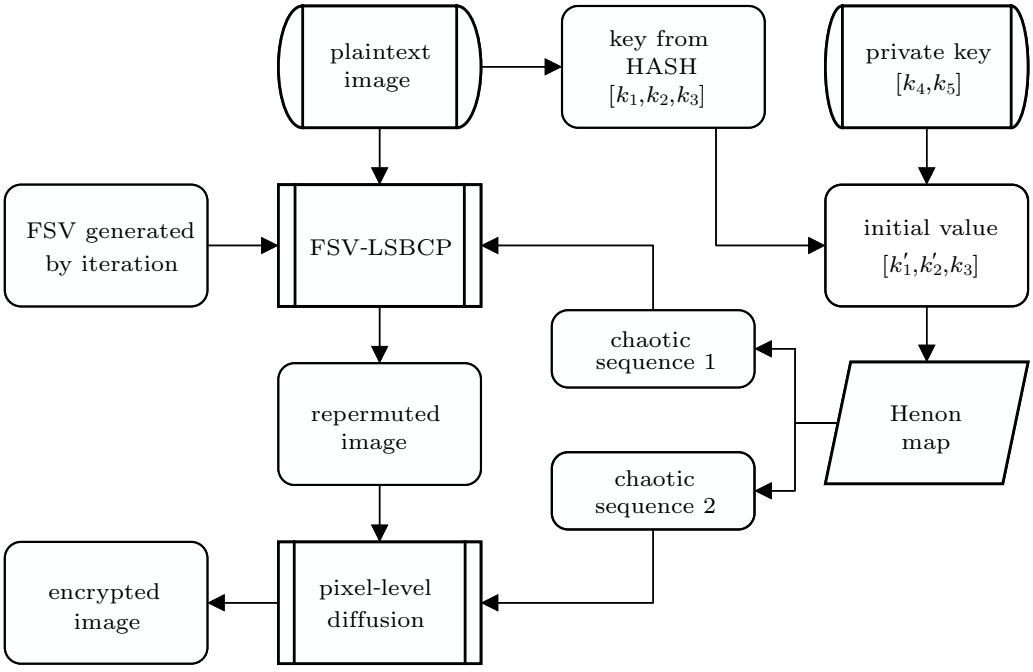


Fig. 3. Flow chart of encryption algorithm based on FSV-LSBCP.

Remark 1 When the parameters of Eq. (6) are determined, different initial values can obtain different pseudo-random sequences after the same iteration. The k_1' and k_2' calculated from Eq. (10) are used as the initial values of Eq. (6), that is, $x_0 = k_1'$ and $y_0 = k_2'$. Then the pseudo-random sequences generated by k_1' and k_2' can be obtained. The lengths of these two sequences are determined by the size of the image. At the same time, to avoid the unsatisfactory pseudo-randomness caused by the system in the front part of the sequence not entering the mixed state, the first k_3 elements of each sequence are deleted and k_3 elements are iterated and converged at the end. The resulting pseudo-random sequences conform to the size of the image and can be well applied to encryption algorithms.

Remark 2 The hexadecimal sequence obtained by SHA-512, after the calculation of Eqs. (7)–(9) can obtain k_1, k_2, k_3. These three keys are used as the keys generated by the plaintext image, which can make the encryption algorithm have good resistance to brute force attacks. The subjectively set keys k_4 and k_5 can effectively improve the ability of the encryption algorithm to resist the selected plaintext attacks. This key design method guarantees the comprehensive security of the algorithm. At the same time, the keys pass through a secure channel to avoid the possibility of the keys being stolen, thereby ensuring the security of ciphertext information in the key management.

3.3. Decryption algorithm

Using inverse transformation, the decryption and corresponding experimental test can be completed by using the key sequence provided in the encryption algorithm.

Step 1 Read the encrypted image into the computer. Call the image matrix $P_{\mathrm{E}} = [p_{\mathrm{e}}(i)]$.

Step 2. Obtain an FSV of the appropriate length according to the proposed construction algorithm.

Step 3. Obtain the chaotic sequences C_1 and C_2 by using the previously described system and processing method with the group key $[k_1, k_2, k_3, k_4, k_5]$.

Step 4. Back-diffuse re-segmented matrix P_{E} with C_2 through using the method shown below and denote the back-diffused image matrix as $P_{\mathrm{E}}' = [p_{\mathrm{e}}'(i)]$:

$$p_{\mathrm{e}}'(i) = \begin{cases} p_{\mathrm{e}}(i) \oplus h(i), & i = 1, \\ p_{\mathrm{e}}(i) \oplus p_{\mathrm{e}}(i-1) \oplus h(i), & i \geq 2. \end{cases} \tag{13}$$

Step 5 Permutate P_{E}' into P_{E}'' by reverse process of FSV-LSBCP.

This concludes the decryption process, which is the reverse of the encryption process. The demonstration is no longer illustrated as space limitations.

3.4. Encryption and decryption simulation results

MATLAB 2017a software is used to conduct the encryption and decryption in this study. The operation system of

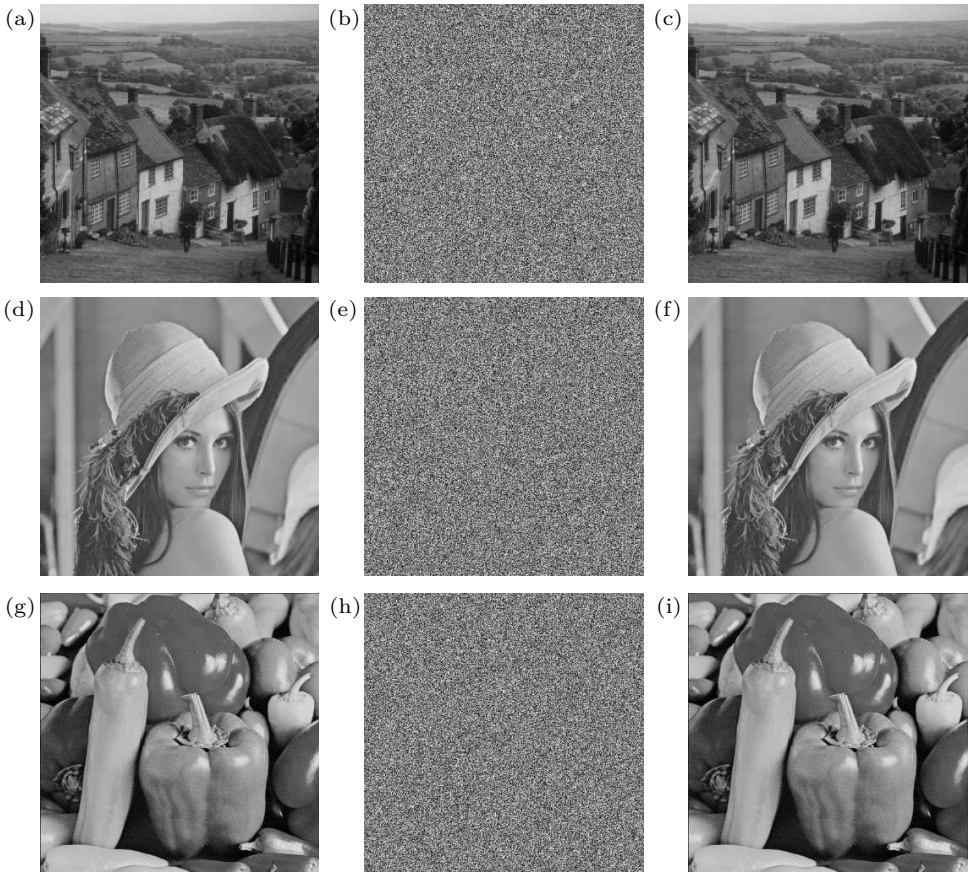


Fig. 4. Image encryption/decryption results, showing (a) image of Fullgold, (b) encrypted Fullgold, (c) decrypted Fullgold, (d) image of Lena, (e) encrypted Lena, (f) decrypted Lena, (g) image of peppers, (h) encrypted peppers, (i) decrypted peppers.

the computer is a Microsoft Windows 10 system with Intel(R) Core(TM) i5-7500 CPU. The encryption and decryption simulation results of three representative images, each with a size of 512×512, are shown in Fig. 4. The simulation results illustrate the accurate process of encryption and decryption in terms of visual effect.

3.5. Encryption runtime comparison

LSB chaotic permutation reduces the operation of bit position transformation. This design can effectively reduce the running time of the proposed encryption algorithm, and its comparisons with the running times of some classic algorithms are shown in Table 3. Through the comparisons, it can be found that the proposed encryption method can effectively improve the efficiency of image encryption.

Table 3. Comparisons among runtimes.

Method	Ref. [21]	Ref. [22]	Ref. [24]	Ref. [27]	Ref. [30]	Proposed
Runtime (s)	7.750	0.459	2.414	0.324	1.283	0.189

4. Security analysis

Numbers of experimental results and data comparisons are provided in this section to check the supremacy and protection of proposed image encryption and decryption algorithms based on FSV-LSBCP. The images, each with a size of 512×512 used for the experimental test, are shown in Fig. 5.



Fig. 5. Images used for experimental test on (a) baboon, (b) Barbara, (c) bridge, (d) couple, (e) crowd, (f) Fullgold, (g) Lena, (h) man, (i) peppers, (j) plane.

4.1. Key analysis

According to the conclusion in Ref. [39], the cryptosystem with keyspace larger than 2^{100} is better at resisting brute-force attacks. Due to the calculation accuracy and the selected space of the initial matrix, the keyspace can be calculated as follows:

$$10^{14\times4}\times10^{3}\geq2^{195}, \tag{14}$$

where the selection space of k_1, k_2, k_4, k_5 is 10^{14} as the calculation accuracy of the computer is assumed to be 10^{-14}, the selection space of k_3 is 10^3.

The key sensitivity test can detect the effect of small changes of the correct key on the decryption of ciphertext. Any accurate image cannot be decrypted unless the right key $K_{\rm r}=[k_1,\ k_2,\ k_3,\ k_4,\ k_5]$ is used in the proposed cryptosystem. The error keys $K_1=[k_1+10^{-14},\ k_2,\ k_3,\ k_4,\ k_5]$ and $K_2=[k_1,\ k_2+10^{-14},\ k_3,\ k_4,\ k_5]$ are used to decrypt the ciphertext, respectively. There is no doubt that the wrong key cannot obtain the correct decrypted image, and even if the two images are subtracted, no relevant information can be obtained. The detailed simulation process of key sensitivity analysis is shown in Fig. 6.

4.2. Statistical analysis

Kinds of statistical experiments and analysis are enumerated to verify the reliability of the proposed algorithm as follows.

4.2.1. Histogram analysis

The ciphertext image histogram is a method used to calculate the ability of an encryption system to object to a statistical analysis attack. This suggests that it is difficult to attack the ciphertext image obtained by the encryption algorithm when the histogram is smooth. As shown in Fig. 7, the histograms of the encrypted and decrypted images have obvious characteristics, where the histograms of the original images have various fluctuations but the histograms of these ciphertext images are very stable.

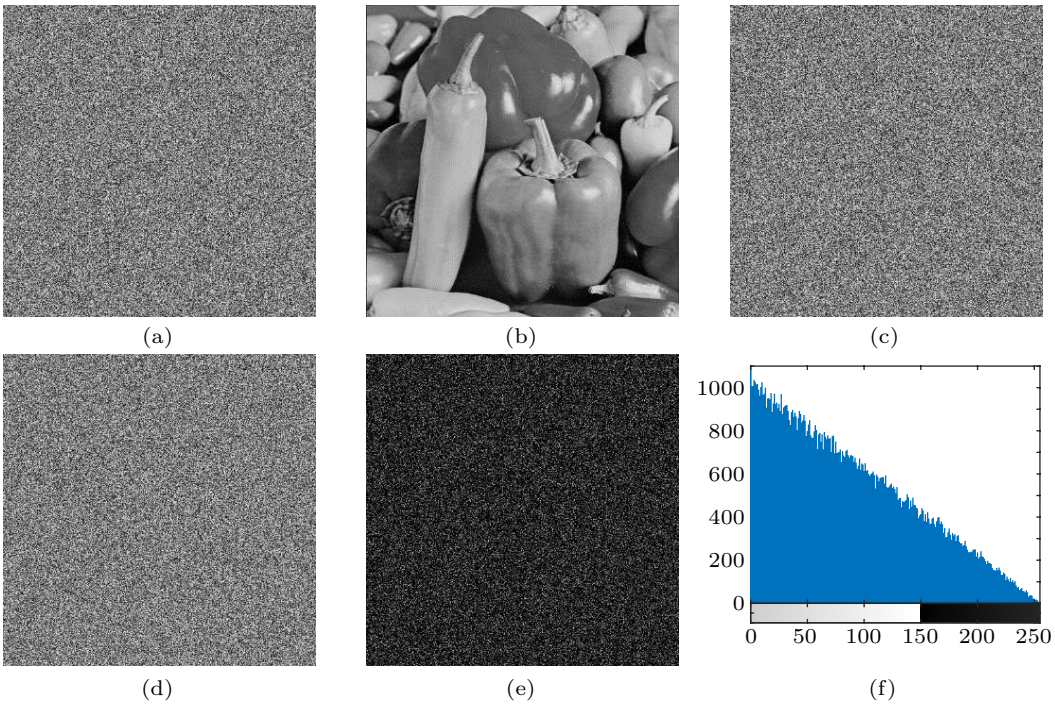


Fig. 6. Key sensitivity test on (a) encrypted peppers with K_r, (b) decrypted peppers with K_r, (c) decrypted peppers with K_1, (d) decrypted peppers with K_2, (e) absolute value of (c) minus (d), and (f) histogram of panel (e).

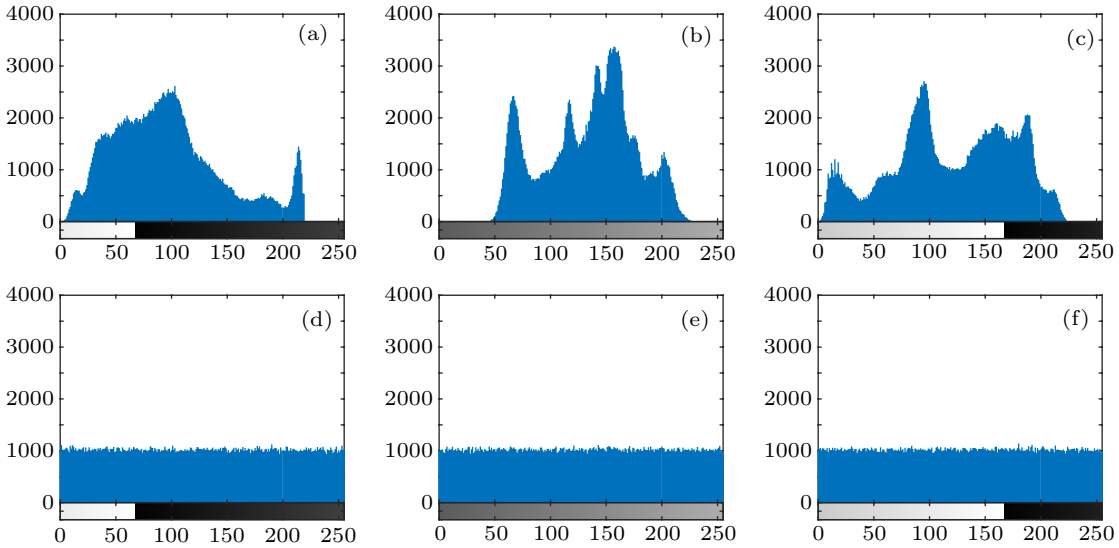


Fig. 7. Histograms of (a) Fullgold, (b) Lena, (c) peppers, (d) encrypted Fullgold, (e) encrypted Lena, and (f) encrypted peppers.

4.2.2. Correlation coefficient

The correlation coefficient is a description of the correlation between two adjacent pixels. For a flat image, its adjacent pixel relationship is divided into three categories: horizontal, vertical, and diagonal. The correlation coefficients of the three directions of the ciphertext image obtained by a secure image encryption algorithm should be small enough to resist attacks. The correlation coefficients of the flat image are calculated as follows:[40]

$$r_{xy} = \frac{\operatorname{cov}(x,y)}{\sqrt{D(x)}\sqrt{D(y)}}, \tag{15}$$

where

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

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

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

Table 4 shows the comparisons of correlation coefficients between the ciphertext images and the original images. The correlation coefficients of the representative encrypted images are generally very small. The mean values in the horizontal,

vertical, and diagonal directions of the correlation coefficients are −0.000271, −0.000358, and 0.000654, respectively. The small standard deviations in the three directions indicate that the proposed method is safe and stable.

The visual correlation results of the image peppers before encryption and after encryption are shown in Fig. 8. The correlations in the three directions of the two original images are all concentrated around the line $y = x$ of the coordinate system, while most of the points are close to the line, which means that the original images are relevant to each other. In contrast to the correlation image of the ciphertext image, each point is evenly distributed in the image range, which reveals that the correlation of the encrypted image is weak.

Table 4. Correlation coefficients of images before and after being encrypted.

Test image	Plaintext image			Ciphertext image		
	Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
Baboon	0.865114	0.759020	0.726597	-0.000592	0.000489	0.001954
Barbara	0.862016	0.957878	0.840457	–0.002477	–0.000677	–0.000377
Bridge	0.940173	0.928434	0.898313	0.002905	0.002018	–0.001207
Couple	0.946397	0.953172	0.912011	0.001141	–0.002344	–0.000530
Crowd	0.973059	0.967565	0.943311	–0.002019	0.003986	0.000013
Fullgold	0.971099	0.974115	0.952639	0.001068	–0.002412	0.001086
Lena	0.970906	0.984655	0.958729	0.000488	–0.000549	0.004539
Man	0.960615	0.969541	0.942613	0.002485	0.000598	0.000837
Peppers	0.980582	0.982251	0.970974	–0.004707	–0.000211	–0.001087
Plane	0.971448	0.964058	0.942496	–0.001002	–0.004474	0.001316
Mean				–0.000271	–0.000358	0.000654
Std				0.002364	0.002390	0.001726

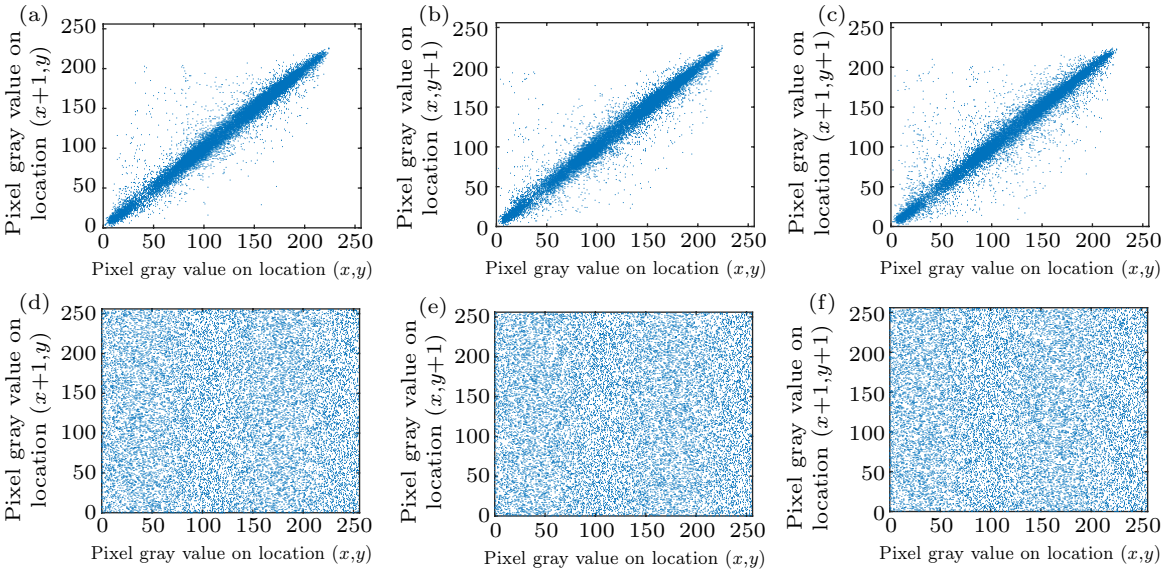


Fig. 8. Adjacent pixel correlation in (a) horizontal direction of peppers, (b) vertical direction of peppers, (c) diagonal direction of peppers, (d) horizontal direction of encrypted peppers, (e) vertical direction of encrypted peppers, and (f) diagonal direction of encrypted peppers.

4.2.3. Information entropy

An important parameter that represents the randomness of information is information entropy, which is expressed as[41]

$$H(s,L) = \sum_{i=0}^{2^L-1} p(s_i) \log_2 \frac{1}{p(s_i)}, \tag{16}$$

where descriptions of parameters such as $s = [s_i]_{2^L}$, $p(s_i)$, and s_i are detailed in Ref. [41].

Table 5 accurately reflects the change in information entropy of the image before and after encryption, that the information entropy after encryption is closer to the theoretical limit value. The standard deviation is 0.000060, which is small, implying that the proposed algorithm is both safe and stable.

Table 5. Information entropy of plaintext and ciphertext images.

Test image	Plaintext image	Ciphertext image
Baboon	7.139099	7.999388
Barbara	7.459910	7.999256
Bridge	5.705560	7.999415
Couple	7.058103	7.999345
Crowd	6.789261	7.999411
Fullgold	7.477784	7.999253
Lena	7.218498	7.999309
Man	7.192588	7.999281
Peppers	7.592451	7.999341
Plane	6.705888	7.999310
Mean	–	7.999331
Std	–	0.000060

4.3. Analysis of resistance to differential attack

The resistance of encryption algorithms to differential attacks can be tested by the number of pixels changing rate (NPCR) and unified averaged changed intensity (UACI), which are acquired as follows:[17]

$$D(i,j) = \begin{cases} 0, & c_1(i,j) = c_2(i,j), \\ 1, & c_1(i,j) \neq c_2(i,j), \end{cases} \tag{17}$$

$$\text{NPCR} = \frac{\sum_{i,j} D(i,j)}{W \times H} \times 100, \tag{18}$$

$$\text{UACI} = \frac{1}{M \times N} \left[\sum_{i,j} \frac{|c_1(i,j) - c_2(i,j)|}{255} \right] \times 100, \tag{19}$$

where $c_1(i,j)$ and $c_2(i,j)$ are two pixel values of the same position (i,j) in the two different encrypted images; M and N represent the image dimensions.

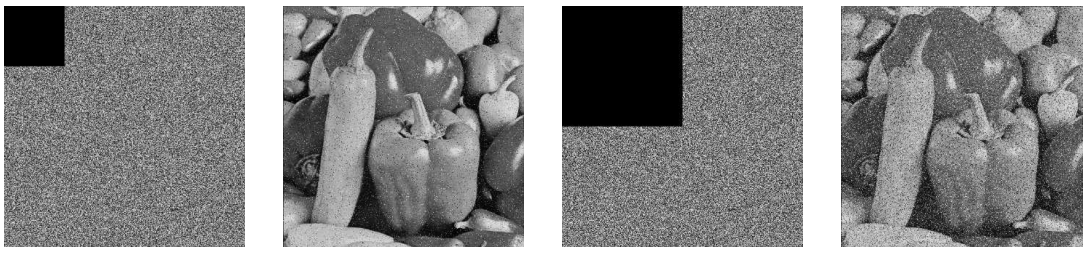
Table 6 shows the average values of the NPCR and UACI among 100 random pixels at different positions. Selecting the mean value of NPCR and UACI at random positions can avoid the particularity of special positions in the calculation process and the waste of time in obtaining the overall value. The NPCR and UACI values of each image and their average values are close to the theoretical level 99.6094% and 33.4635%, respectively. Although the standard deviation of UACI is slightly larger than that of NPCR, they are not very large. This makes our algorithm stable while being safe, and the proposed method is suitable for various types of images. In general, the proposed method has a high resistance to the differential attack. Therefore, the proposed encryption scheme is verified to resist differential attacks.

Table 6. NPCR and UACI values of ciphertext images (%).

Test image	NPCR	UACI
Baboon	99.6086	33.4709
Barbara	99.6080	33.4727
Bridge	99.6147	33.4799
Couple	99.6147	33.4626
Crowd	99.6101	33.4436
Fullgold	99.6078	33.4874
Lena	99.6094	33.4215
Man	99.6125	33.4595
Peppers	99.6087	33.4707
Plane	99.6072	33.4633
Mean	99.6102	33.4632
Std	0.0028	0.0189

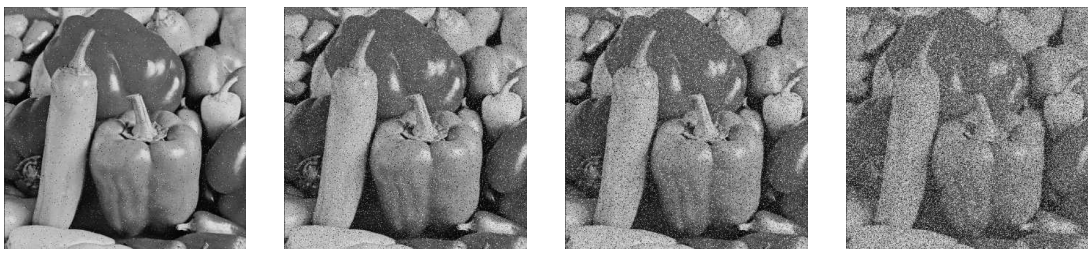
4.4. Resistance to cropping attacks and noise attacks

The cropping attacks and noise attacks are used to disrupt the integrity of the ciphertext image, which will prevent the attacker from obtaining the correct decryption information. In the resisting cropping attacks experiment, the encrypted images at the degrees of 1/16 and 1/4 cropping are decrypted as shown in Fig. 9.



(a) (b) (c) (d)

Fig. 9. Decryption results after cropping attacks, showing (a) encrypted peppers with 1/16 degree cropping, (b) decrypted image of panel (a), (c) encrypted peppers with 1/4 degree cropping, and (d) decrypted image of panel (c).



(a) (b) (c) (d)

Fig. 10. Decryption results after salt pepper noise attacks, showing (a) decrypted Lena after 1% noise, (b) decrypted Lena after 5% noise, (c) decrypted Lena after 10% noise, and (d) decrypted Lena after 25% noise.

Table 7. Comparisons of proposed method with other bit-level permutation encryption methods.

Criteria		Ref. [21]	Ref. [22]	Ref. [26]	Ref. [27]	Ref. [42]	Proposed
Information entropy		7.999350	7.998754	7.9976	7.998683	7.9967	7.999331
Correlation	H	0.001465	–0.001600	0.0047	0.001847	–0.0020	–0.000271
	V	0.002711	0.000300	0.0118	0.001098	0.0010	–0.000358
	D	0.001816	0.002200	0.0021	0.002304	–0.0259	0.000654
NPCR		0.995937	0.996093	0.9959	0.996083	0.9965	0.996102
UACI		0.334607	0.334640	0.3349	0.334521	0.3332	0.334632

In the experiment on resisting salt and pepper noise attacks, the encrypted images with noise strengths of 0.01, 0.05, 0.10, and 0.25 are decrypted as shown in Fig. 10. The proposed cryptosystem can effectively resist cropping attacks and noise attacks and can decrypt identifiable plaintext images from ciphertext images that have been damaged to varying degrees.

4.5. Comparison with other algorithms

In the previous part, the experimental results of several indicators were introduced, and finally, the average values are used to compare with the experimental indicators, and thus comparing the proposed algorithm with other methods. The comparisons of performance between the proposed encryption scheme and other pixel-level permutation image cryptosystems are list in Table 7.

The proposed algorithm presents good comprehensive experimental performance, showing that the proposed cryptosystem is both safe and reliable.

5. Conclusions

A chaotic image encryption cryptosystem based on FSV-LSBCP is proposed for achieving the image with any size in this work. The basic concepts and iterative algorithm of FSV are introduced in detail. The fast iterative FSV can effectively improve the efficiency of encryption algorithms. The bit permutation of LSB can not only ensure the encryption effect but also improve the encryption efficiency. Combined with the dynamic characteristics of FSV, a new FSV-LSBCP is presented. The chaotic image encryption and decryption algorithm based on the FSV-LSBCP is constructed to encrypt and generate ciphertext images with stronger resistance to attacks. Through the comparison of encryption time, statistical analysis, and analysis of resistance to differential attacks, the proposed method proves to have a large enough keyspace, sufficiently sensitive key, and unrivaled resistance to statistical attacks and differential attacks.

References

[1] Wang C P, Wang X Y, Xia Z Q, Ma B and Shi Y Q 2020 *IEEE T. Circ. Syst. Vid.* **30** 4440
[2] Xian Y J and Wang X Y 2021 *Inform. Sci.* **547** 1154
[3] Li Q, Wang X Y, Wang X Y, Ma B, Wang C P and Shi Y Q 2021 *Inform. Sci.* **553** 19
[4] Popoff S, Lerosey G, Fink M, Boccara A C and Gigan S 2010 *Nat. Commun.* **1** 81
[5] Li C Q, Lin D D, Lu J H and Hao F 2018 *IEEE Multimedia* **25** 46
[6] Chen W, Chen X D and Sheppard C J R 2010 *Opt. Lett.* **35** 3817
[7] Chen J J, Yan D W, Duan S K and Wang L D 2020 *Chin. Phys. B* **29** 110504
[8] Wang X Y, Wang Y, Wang S W, Zhang Y Q and Wu X J 2018 *Chin. Phys. B* **27** 110502
[9] Hayat U and Azam N A 2019 *Signal Process.* **155** 391
[10] Wang X Y and Gao S 2020 *Inform. Sci.* **507** 16
[11] Nosrati K, Volos C and Azemi A 2017 *Signal Process.-Image* **58** 35
[12] Alawida M, Samsudin A, Teh J S and Alkhawaldeh R S 2019 *Signal Process.* **160** 45
[13] Zhang H, Wang X Q, Sun Y J and Wang X Y 2020 *Signal Process.-Image* **84** 115829
[14] Zhang Y 2018 *Inform. Sci.* **450** 361
[15] Liu Z Y, Xia T C and Wang J B 2018 *Chin. Phys. B* **27** 030502
[16] Wang Y P, Liu S T and Li H 2020 *Nonlinear Dyn.* **102** 579
[17] Sun F, Liu S T, Li Z Q and Lu Z W 2008 *Chaos Soliton. Fract.* **38** 631
[18] Fridrich J 1998 *Int. J. Bifur. Chaos* **8** 1259
[19] Zhang Y Q and Wang X Y 2014 *Inform. Sci.* **273** 329
[20] Li C Q, Lin D D and Lu J H 2017 *IEEE Multimedia* **24** 64
[21] Wu J H, Liao X F and Yang B 2018 *Signal Process.* **142** 292
[22] Shahna K and Mohamed A 2020 *Appl. Soft Comput.* **90** 106162
[23] Zhu Z L, Zhang W, Wong K W and Yu H 2011 *Inform. Sci.* **181** 1171
[24] Zhang W, Wong K W, Yu H and Zhu Z L 2013 *Commun. Nonlinear Sci.* **18** 584
[25] Tang Z J, Song J, Zhang X Q and Sun R H 2016 *Opt. Laser Eng.* **80** 1
[26] Diaconu A V 2016 *Inform. Sci.* **355** 314
[27] Cao C, Sun K H and Liu W H 2018 *Signal Process.* **143** 122
[28] Liu X, Song Y R and Jiang G P 2019 *Int. J. Bifur. Chaos* **29** 2
[29] Zhang Y Q and Wang X Y 2014 *Nonlinear Dyn.* **77** 687
[30] Zhang L and Zhang X Q 2020 *Multimed. Tools Appl.* **79** 20753
[31] Sun W H and Liu S T 2020 *Discrete Dyn. Nat. Soc.* **2020** 8547685
[32] Liu S T and Wang P 2018 *Fractal Control Theory* (Singapore: Springer) p. 3
[33] Liu S T, Zhang Y P and Liu C A 2020 *Fractal Control and Its Applications* (Singapore: Springer) p. 235
[34] Bi F and Li C F 2013 *Chin. Phys. Lett.* **30** 010306
[35] Hao W Q, Liang Z C, Liu X Y, Zhao R, Kong M M, Guan J F and Zhang Y 2019 *Acta Phys. Sin.* **68** 199501 (in Chinese)
[36] Xing H Y, Gong P and Xu W 2012 *Acta Phys. Sin.* **61** 160504 (in Chinese)
[37] Yu Y M, Yang L C, Zhou Q, Zhao L L and Liu Z P 2016 *Chin. Phys. B* **25** 060503
[38] Hua Z Y and Zhou Y C 2016 *Inform. Sci.* **339** 237
[39] Alvarez G and Li S J 2006 *Int. J. Bifurc. Chaos* **16** 2129
[40] Alawida M, Teh J S, Samsudin A and Alshoura W H 2019 *Signal Process.* **164** 24
[41] Wang X Y, Zhang Y Q and Bao W M 2015 *Opt. Laser Eng.* **73** 53
[42] Li J F, Xiang S Y, Wang H N, Gong J K and Wen A J 2018 *Opt. Laser Eng.* **102** 170