fractal and fractional

MDPI

Article

Exploiting Newly Designed Fractional-Order 3D Lorenz Chaotic System and 2D Discrete Polynomial Hyper-Chaotic Map for High-Performance Multi-Image Encryption

Wei Feng [1], Quanwen Wang [1], Hui Liu [1], Yu Ren [1], Junhao Zhang [1], Shubo Zhang [1], Kun Qian [2,3,*] and Heping Wen [4]

[1] School of Mathematics and Computer Science, Panzhihua University, Panzhihua 617000, China; fengwei@pzhu.edu.cn (W.F.); quanwenwang@pzhu.edu.cn (Q.W.); liuhhh2002@126.com (H.L.); renyu2002@126.com (Y.R.); zhangjunhao1618@126.com (J.Z.); 13467266166@163.com (S.Z.)
[2] Key Laboratory of Hunan Province on Information Photonics and Freespace Optical Communications, Hunan Institute of Science and Technology, Yueyang 414006, China
[3] College of Physics and Electronics, Hunan Institute of Science and Technology, Yueyang 414006, China
[4] School of Electronic Information, University of Electronic Science and Technology of China, Zhongshan Institute, Zhongshan 528402, China; wenheping@uestc.edu.cn
* Correspondence: tsienkun@hnist.edu.cn

Abstract: Chaos-based image encryption has become a prominent area of research in recent years. In comparison to ordinary chaotic systems, fractional-order chaotic systems tend to have a greater number of control parameters and more complex dynamical characteristics. Thus, an increasing number of researchers are introducing fractional-order chaotic systems to enhance the security of chaos-based image encryption. However, their suggested algorithms still suffer from some security, practicality, and efficiency problems. To address these problems, we first constructed a new fractional-order 3D Lorenz chaotic system and a 2D sinusoidally constrained polynomial hyper-chaotic map (2D-SCPM). Then, we elaborately developed a multi-image encryption algorithm based on the new fractional-order 3D Lorenz chaotic system and 2D-SCPM (MIEA-FCSM). The introduction of the fractional-order 3D Lorenz chaotic system with the fourth parameter not only enables MIEA-FCSM to have a significantly large key space but also enhances its overall security. Compared with recent alternatives, the structure of 2D-SCPM is simpler and more conducive to application implementation. In our proposed MIEA-FCSM, multi-channel fusion initially reduces the number of pixels to one-sixth of the original. Next, after two rounds of plaintext-related chaotic random substitution, dynamic diffusion, and fast scrambling, the fused 2D pixel matrix is eventually encrypted into the ciphertext one. According to numerous experiments and analyses, MIEA-FCSM obtained excellent scores for key space (2^{541}), correlation coefficients (<0.004), information entropy (7.9994), NPCR (99.6098%), and UACI (33.4659%). Significantly, MIEA-FCSM also attained an average encryption rate as high as 168.5608 Mbps. Due to the superiority of the new fractional-order chaotic system, 2D-SCPM, and targeted designs, MIEA-FCSM outperforms many recently reported leading image encryption algorithms.

Keywords: chaotic system; fractional order; hyper-chaotic map; image encryption; multi-channel fusion; security analysis

Citation: Feng, W.; Wang, Q.; Liu, H.; Ren, Y.; Zhang, J.; Zhang, S.; Qian, K.; Wen, H. Exploiting Newly Designed Fractional-Order 3D Lorenz Chaotic System and 2D Discrete Polynomial Hyper-Chaotic Map for High-Performance Multi-Image Encryption. *Fractal Fract.* **2023**, *7*, 887. https://doi.org/10.3390/fractalfract7120887

Academic Editors: Ravi P. Agarwal and Maria Alessandra Ragusa

Received: 14 November 2023
Revised: 7 December 2023
Accepted: 12 December 2023
Published: 16 December 2023

1. Introduction

In today's highly informationalized and digitalized era, the application of digital images is omnipresent. Compared to text or other carriers, digital images can transmit information promptly and vividly, and they have been integrated into all aspects of society [1,2]. Significantly, the extensive utilization of digital images also incurs a slew of

privacy and security challenges. Hence, there is an urgent need for more secure and efficient methods to safeguard image data. As we know, in contrast to other methods, image encryption is a relatively simple yet effective method to secure image data [3].

In fact, a large number of image encryption algorithms have already been proposed [4]. Unlike traditional encryption algorithms such as data encryption standard (DES), these algorithms are based on novel techniques and methods, such as chaotic maps [5–8], neural networks [9,10], compressive sensing [10,11], deoxyribonucleic acid (DNA) computing [12,13], and quantum computing [14,15].

Notably, chaotic systems possess multiple unique properties that are suitable for constructing cryptographic systems, including unpredictability and randomness. Therefore, a majority of the existing image encryption algorithms are built upon various chaotic systems [16]. Here, we can provide several examples of recent advancements. By exploiting a cellular neural network, Wang et al. [10] presented an encryption algorithm based on compressive sensing to protect image data in an embedded manner. Their algorithm first utilized the wavelet transform to sparsify the input image, and then performed subsequent encryption processing using binary patterns and compressive sensing. In [17], an image encryption algorithm relying on several chaotic maps and optimization algorithms was suggested. This algorithm employed optimization algorithms to modify encryption parameters and ultimately produced the final ciphertext image by performing confusion and diffusion operations. In [18], an encryption algorithm utilizing DNA computing and chaotic systems was introduced by Yu et al. Their algorithm directly utilizes the hash value of the input image to produce chaotic sequences, which are then used for encrypting the image data. In [19], Nan et al. first constructed a hyper-chaotic map called the logistic coupling cubic chaotic map (LCCCM). Then, they suggested an image encryption algorithm using compressive sensing and S-boxes. Benefiting from the randomness of chaotic sequences, this algorithm achieved a satisfactory encryption effect. However, the structure of the chaotic map it adopted is relatively complex, and the encryption efficiency is relatively low. Additionally, their algorithm is also a lossy one, which makes it actually not suitable for encrypting images containing rich details. Employing a 2D logistic map, Liu et al. [20] devised an encryption algorithm based on DNA sequence operations to encrypt images. In their algorithm, plaintext-related information was directly used to generate chaotic sequences. This design requires the algorithm to regenerate the chaotic sequences when encrypting different images. Consequently, its encryption efficiency cannot actually meet the needs of practical applications. By exploiting the Mersenne twister, Masood et al. [21] suggested an image encryption algorithm based on DNA encoding and chaotic sequencing. According to previous studies in cryptanalysis [22–24], although this algorithm passed certain statistical tests, it is unable to effectively withstand plaintext attacks due to the absence of any diffusion operation.

As mentioned above, although many existing encryption algorithms are specially developed for images, these algorithms still have shortcomings in practicality, security, and efficiency, and cannot well meet the needs of practical applications [4,22–24]. Therefore, while ensuring security, in order to further enhance the practicability and efficiency of chaotic image encryption, we first constructed a new fractional-order 3D Lorenz chaotic system and a 2D sinusoidally constrained polynomial hyper-chaotic map (2D-SCPM). Then, we further devised a multi-image encryption algorithm based on the new fractional-order 3D Lorenz chaotic system and 2D-SCPM (MIEA-FCSM). In our proposed MIEA-FCSM, the input images are first reshaped and fused into a 2D pixel matrix. Since the pixel number of the fused pixel matrix is reduced to one-sixth of the original number, there is a considerable decrease in the computational amount required for the subsequent encryption steps. Next, two rounds of plaintext-related chaotic random substitution, dynamic diffusion, and fast scrambling are performed to encrypt the fused matrix into the final ciphertext matrix. Overall, the work presented in this paper has the following contributions and innovations:

- A new fractional-order 3D Lorenz chaotic system with the fourth parameter was constructed to enhance the security of chaos-based image encryption.

- A hyper-chaotic map named 2D-SCPM was proposed. Because of its simple structure and superior chaotic performance, 2D-SCPM is highly appropriate for image encryption.
- To address the shortcomings of existing image encryption algorithms, a multi-image encryption algorithm based on the new fractional-order 3D Lorenz chaotic system and 2D-SCPM was developed.
- Due to the excellent chaotic performance of the fractional-order 3D Lorenz chaotic system and 2D-SCPM, the innovative efficiency advantage of multi-channel fusion, and well-designed full vector-level encryption operations, MIEA-FCSM not only possesses excellent practicability, but also exhibits extremely high security and encryption efficiency.
- Extensive experiments and analyses were performed to demonstrate the superiority of the fractional-order 3D Lorenz chaotic system, 2D-SCPM, and MIEA-FCSM.

The following is the organization of the remaining sections: Section 2 describes the construction of a new fractional-order 3D Lorenz chaotic system. Section 3 presents the proposed 2D-SCPM, evaluates its performance, and compares it to other chaotic maps. Section 4 offers a comprehensive overview of MIEA-FCSM, along with a detailed explanation of each encryption step involved. Section 5 tests and analyzes the security and efficiency of MIEA-FCSM. Finally, the conclusions are given in Section 6.

2. Fractional-Order Chaotic System

The classical 3D Lorenz system is widely applied in the field of chaotic image encryption due to its simple structure and complex dynamical characteristics [1]. The Lorenz system has three control parameters, σ, ρ, and β, which represent the Prandtl number, the Rayleigh number, and the geometric ratio, respectively. This system is chaotic while the control parameters satisfy the conditions $\sigma \in [9, 10]$, $\rho \in [25, 30]$, and $\beta \in [2, 3]$. To expand the key space of our proposed image encryption algorithm and enhance its security, the fourth parameter α is introduced to the 3D Lorenz system through fractional calculus. According to Caputo's definition of fractional derivatives, the proposed 3D fractional-order Lorenz system can be depicted as follows:

$$\begin{cases} D^{\alpha}x = \sigma(y - x), \\ D^{\alpha}y = \rho x - y - xz, \\ D^{\alpha}z = xy - \beta z, \end{cases} \tag{1}$$

where D^{α} is Caputo's differential operator with fractional-order α, $\alpha \in (0, 1]$. According to Caputo's definition of fractional derivatives, the α-order derivative of the function $x(t)$ can be expressed as follows:

$$D^{\alpha}x(t) = \frac{1}{\Gamma(1-\alpha)} \int_0^t (t-\tau)^{-\alpha} x'(\tau) d\tau, \tag{2}$$

where $\Gamma(\bullet)$ is the gamma function. The prediction–correction method of Adams–Bashforth–Moulton (ABM) was utilized to obtain a numerical solution for this fractional-order system [25]. Since the ABM method has an error roughly proportional to h^2, the step-size h was set to 0.001 to obtain an error of 10^{-6}.

Figure 1 shows the phase trajectories of this system when the initial states $\{x_0, y_0, z_0\}$ were set to $\{0.3, 0.3, 0.3\}$, $\{0.4, 0.4, 0.4\}$, $\{0.2, 0.2, 0.2\}$, and $\{0.35, 0.35, 0.35\}$, respectively. At this point, the system parameters $\{\sigma, \rho, \beta\}$ were set to $\{10, 28, 8/3\}$. By observing the phase trajectory diagrams, one can find that the system starts from similar initial states and eventually evolves into different orbits.

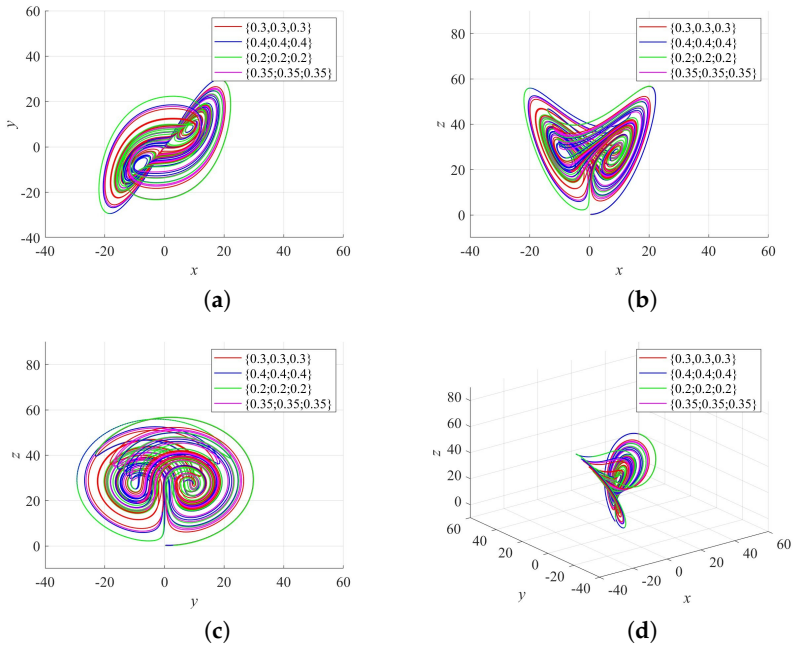


Figure 1. Phase trajectories of the fractional-order 3D Lorenz system with different initial states while the parameters $\{\sigma, \rho, \beta, \alpha\} = \{10, 28, 8/3, 0.995\}$: (**a**) x-y plane; (**b**) x-z plane; (**c**) y-z plane; (**d**) 3D plot.

In addition, the chaotic behavior of the fractional-order system was also analyzed through its Lyapunov exponent spectrums against different parameters, which were calculated through the Benettin–Wolf algorithm [26], as shown in Figure 2. According to Lyapunov exponent spectrums, one can find that the fractional-order 3D Lorenz system has a positive LE, while $\sigma \in [5.81, 17.5]$, $\rho \in [24, 70]$, $\beta \in [1, 3.3]$, and $\alpha \in [0.92, 1]$. Compared to the integer-order 3D Lorenz system, the fractional-order 3D Lorenz system possesses more control parameters and wider chaotic parameter ranges. Therefore, we can employ the fractional-order 3D Lorenz chaotic system to enhance the security of image encryption.

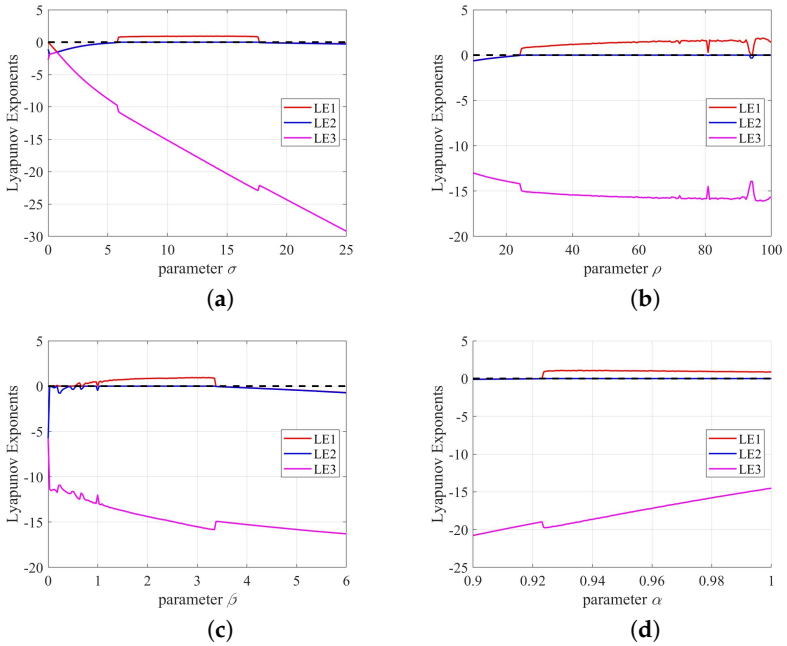


Figure 2. Lyapunov exponent spectrums of the fractional-order 3D Lorenz system: (**a**) Sweep parameter σ while $\{\rho, \beta, \alpha\} = \{28, 8/3, 0.99\}$; (**b**) sweep parameter ρ while $\{\sigma, \beta, \alpha\} = \{10, 8/3, 0.99\}$; (**c**) sweep parameter β while $\{\sigma, \rho, \alpha\} = \{10, 28, 0.99\}$; (**d**) sweep parameter α while $\{\sigma, \rho, \beta\} = \{10, 28, 8/3\}$.

3. Proposed 2D-SCPM

To facilitate the efficiency and security of image encryption, we proposed a 2D hyper-chaotic map characterized by its straightforward structure and outstanding chaotic performance. This section introduces this new map and evaluates its performance with the Lyapunov exponent (LE), bifurcation diagram, Kolmogorov entropy (KE), and the NIST SP800-22 test suite.

3.1. Construction of 2D-SCPM

Currently, chaotic maps find extensive utilization in image encryption. For encrypting images, the structural simplicity and chaotic performance of chaotic maps and their efficiency in generating chaotic sequences are of utmost importance. Put simply, when developing an image encryption algorithm, the chaotic map utilized should possess a straightforward structure and excellent chaotic behaviors. It is widely known that classical maps, such as the tent map, possess straightforward structures; however, their chaotic performances are relatively poor. With the recognition that hyper-chaotic maps generally exhibit superior chaotic behaviors compared to 1D chaotic maps, there has been an increasing number of proposed 2D chaotic maps in recent years [11,19,27–29]. Notably, although these newly proposed hyper-chaotic maps exhibit relatively good chaotic performance, their structures are rather complex, as demonstrated in Table 1. This is obviously not conducive to encryption efficiency or software and hardware implementations.

Table 1. Definitions of five recent chaotic maps.

Ref.	Name	Definition
[27]	SCMCI (2021)	$\begin{cases} x_{i+1} = r\sin(\pi((y_i+h)k\sin(a\pi/x_i))), \\ y_{i+1} = r\sin(kx_{i+1}+h)\sin(a\pi/x_i))). \end{cases}$
[28]	LSM (2021)	$\begin{cases} x_{i+1} = \cos(4ax_i(1-x_i)+b\sin(\pi y_i)+1), \\ y_{i+1} = \cos(4ay_i(1-y_i)+b\sin(\pi x_i)+1). \end{cases}$
[29]	STLFM (2022)	$\begin{cases} x_{i+1} = \sin(\pi(4p_1x_i(1-x_i)+1/(y_i^2+0.1)-p_2y_i)), \\ y_{i+1} = \sin(\pi(4p_1y_i(1-y_i)+1/(x_i^2+0.1)-p_2x_i)). \end{cases}$
[11]	FOCM (2022)	$\begin{cases} x_{i+1} = x_i + (h^v/(\Gamma(1+v)))\cos(2\pi x_i/ \\ (2\mu x_i^4-1)-y_i), \\ y_{i+1} = y_i + (h^v/(\Gamma(1+v)))\cos(\mu\pi \\ x_{i+1}+y_i). \end{cases}$
[19]	LCCCM (2022)	$\begin{cases} x_{i+1} = \cos(\pi^2(4\mu x_i(1-x_i)+py_i(1- \\ y_i^2))+\pi/2), \\ y_{i+1} = \cos(\pi^2(4\mu y_i(1-y_i)+px_i(1- \\ x_{i+1}^2))+\pi/2). \end{cases}$

Therefore, to enhance the security and efficiency of image encryption, we constructed the following hyper-chaotic map, called 2D-SCPM:

$$\begin{cases} x_{i+1} = \sin(10^a x_i y_i + 10^b y_i), \\ y_{i+1} = \sin(10^b x_i y_i + 10^a x_i). \end{cases} \tag{3}$$

In Equation (3), (x_i, y_i) serve as the input states for the i-th iteration of 2D-SCPM, while (x_{i+1}, y_{i+1}) are the resulting output states. Additionally, a and b function as control parameters. In 2D-SCPM, two exponential parameters enable the trajectory to diverge quickly, while the sine function can constrain the trajectory within a specific range. When compared to many newly proposed chaotic maps, 2D-SCPM exhibits better chaotic performance while featuring a simpler construction.

3.2. LE

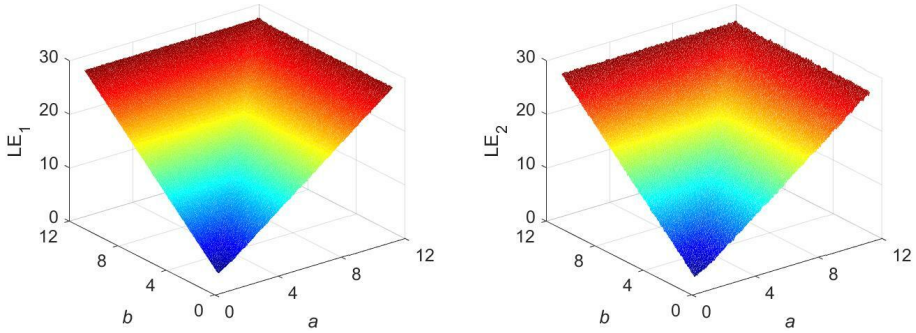
The divergence velocity between the trajectories approaching each other in phase space can be characterized by LE, which is a reliable metric extensively employed to judge

if a dynamical system is chaotic. In particular, if a dynamical system features one LE greater than 0, the system is considered chaotic. Furthermore, if there are multiple positive LEs, then it is classified as hyper-chaotic. For a system $S(x, y)$, such as 2D-SCPM, one can calculate its LEs through

$$LE_k = \lim_{q\to\infty} \frac{1}{q} \sum_{j=1}^{q} \ln\left|\lambda_k(J(x_j, y_j))\right|. \tag{4}$$

In Equation (4), LE_k indicates the two exponents LE_1 and LE_2 to be calculated, q is the quantity of iterations that $S(x, y)$ goes through, and $\lambda_k(J(x_j, y_j))$ represents each eigenvalue of the system's Jacobian matrix $J(x_j, y_j)$.

Figure 3 depicts the LE representations obtained for 2D-SCPM. It is evident that within the continuous interval where $a, b \in [1, 12]$, both LE_1 and LE_2 are always positive. This suggests that 2D-SCPM is in a hyper-chaotic state. As the values of a and b increase, these two exponents will grow rapidly, eventually reaching their maximum values of 27.5045 and 26.9717, respectively.



(**a**) LE_1 (**b**) LE_2

Figure 3. 3D LE presentations for 2D-SCPM.

To further confirm the superiority of 2D-SCPM, additional comparative experiments were carried out. In Table 2, a list of the parameter configurations used in our experiments is provided. Note that the specific values adopted are the ones suggested by the pertinent references [11,19,27–29]. As indicated in Figure 4 and Table 2, SCMCI, FOCM, and LCCM exhibit apparent periodic windows, which are not undesirable for image encryption. LSM also features unstable points where the LE value drops sharply. Although the LE values of STLFM are relatively stable, their values are small, resulting in a relatively low trajectory divergence velocity. In contrast, throughout the entire parameter range, the LE values of 2D-SCPM are not only the most stable but also remarkably high. This suggests that 2D-SCPM possesses the highest state value sensitivity and trajectory divergence velocity, making it more suitable for image encryption.

Table 2. Configurations and results of LE comparative experiments for 2D-SCPM.

Name	Configuration		LE_1		LE_2	
	Invariable	Variable	Average	Std. Dev.	Average	Std. Dev.
SCMCI [27]	$k = 1, h = 2, r = 1$	a	4.7720	0.7903	−0.2738	0.0520
LSM [28]	$b = 50$	a	4.0835	0.0202	3.5923	0.0180
STLFM [29]	$p_2 = 50$	p_1	4.6961	0.0230	4.0224	0.0301
FOCM [11]	$h = 0.5, v = 0.789$	μ	1.6410	0.4487	0.0794	0.0625
LCCCM [19]	$p = 8.78$	μ	6.7625	0.0119	2.6554	1.2474
2D-SCPM	$b = 12$	a	27.0762	0.0010	26.1167	0.0040

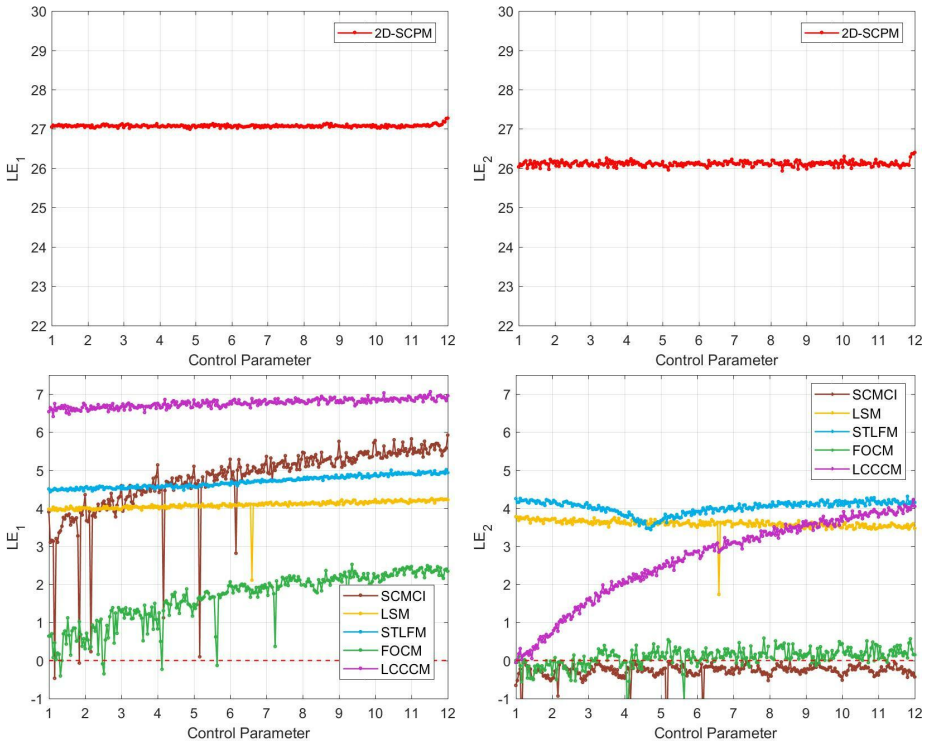


Figure 4. LE experiment results for 2D-SCPM and other five maps: the first column is the LE_1 values of six maps; the second column is the LE_2 values.

3.3. Bifurcation Diagram

Bifurcation diagrams can offer visual representations of how parameters affect the output distributions of chaotic systems. To meet the necessary requirements of image encryption, it is crucial that the distribution of the adopted system's output be uniform. Failure to achieve uniform distribution may result in security vulnerabilities. The top row of Figure 5 exhibits the bifurcation diagrams of LASM [30] and FOCM [11]. It is clearly noticeable that the output distributions of these two recent maps are nonuniform. This lack of uniformity presents limitations and disadvantages for potential applications relying on these maps. Conversely, the output distribution of 2D-SCPM is highly uniform across the entire continuum of $(a, b) \in [1, 12]$, as demonstrated in the bottom row of Figure 5. Consequently, regarding the output distribution, 2D-SCPM is better suited for image encryption.

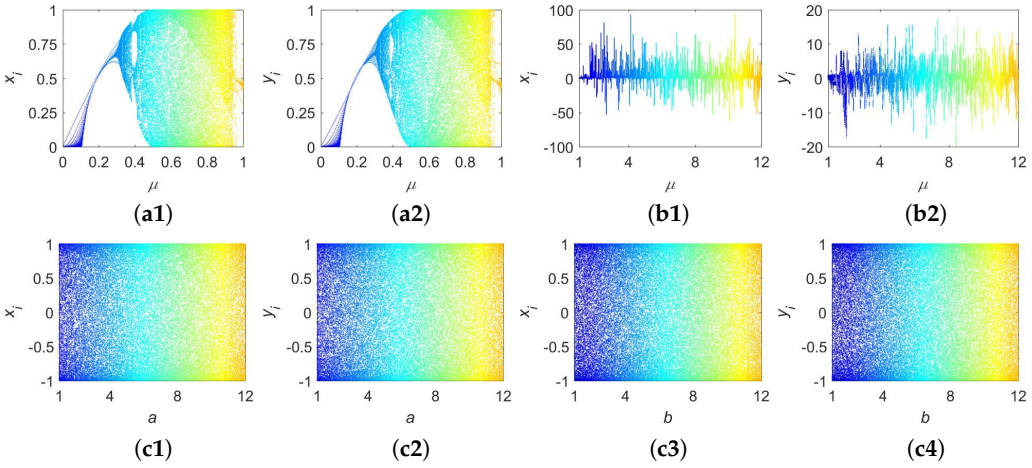


Figure 5. Bifurcation diagrams for three different maps: (**a1**,**a2**) are bifurcation diagrams for LASM; (**b1**,**b2**) are two diagrams for FOCM ($h = 0.5, v = 0.789$); (**c1**–**c4**) are four diagrams for 2D-SCPM.

3.4. KE

As a frequently utilized indicator for evaluating chaos, KE can quantify the information required to predict the future trajectory of a dynamical system based on its current state [31]. If the KE value of a dynamical system exceeds 0, it indicates that the system is in a state of chaos. The chaotic dynamics of the system are further regarded as more complex when the KE value is higher, making it more challenging to predict the trajectory of the system. After dividing the q-dimensional space into infinitely small boxes $(s_1, s_2, \ldots, s_q)$, one can provide a mathematical definition of KE as

$$KE = -\lim_{d\to 0}\lim_{\varepsilon\to 0}\lim_{q\to\infty} d^{-1}\sum_{s_1,s_2,\ldots,s_q}\rho(s_1,s_2,\ldots,s_q)\ln\rho(s_1,s_2,\ldots,s_q), \tag{5}$$

where d stands for the delay, and $\rho(s_1, s_2, \ldots, s_q)$ represents the probability that the trajectory can be properly predicted. A series of experiments were carried out to assess the unpredictability, randomness, and complexity of 2D-SCPM using the approach outlined in [31]. For these experiments, we utilized the same setups as the LE experiments. From Figure 6 and Table 3, it is evident that LSM, STLFM, and LCCCM all exhibit good KE values. However, it is noteworthy that 2D-SCPM outperforms the rest in terms of both average value and stability. This demonstrates that 2D-SCPM offers the best unpredictability, randomness, and complexity, making it better suited for image encryption.

Table 3. Configurations and results of KE comparative experiments for 2D-SCPM.

Name	Configuration		KE_x		KE_y	
	Invariable	Variable	Average	Std. Dev.	Average	Std. Dev.
SCMCI [27]	$k = 1, h = 2, r = 1$	a	2.1476	0.0785	1.8273	0.1467
LSM [28]	$b = 50$	a	2.1210	0.0772	2.1169	0.0733
STLFM [29]	$p_2 = 50$	p_1	2.1125	0.0929	2.1156	0.0591
FOCM [11]	$h = 0.5, v = 0.789$	μ	0.3539	0.1332	0.8251	0.3020
LCCCM [19]	$p = 8.78$	μ	2.1015	0.0810	2.1294	0.0737
2D-SCPM	$b = 12$	a	2.2523	0.0466	2.2246	0.0572

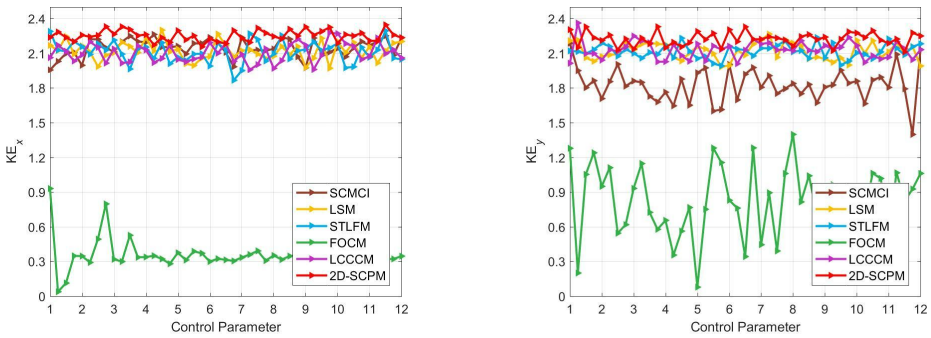


Figure 6. KE experiment results for 2D-SCPM and other five maps: the first column is the KE_x values of six maps; the second column is the KE_y values.

3.5. Randomness Test

As a widely recognized and well-known randomness test suite, NIST SP800-22 contains 15 randomness test items that can comprehensively evaluate the randomness of sequences. For an input data sequence of length 10^6, if the obtained test result (p value) exceeds a given confidence probability (typically 0.01), then the sequence is regarded as highly random. To further demonstrate the performance of 2D-SCPM, we employed the suite to conduct exhaustive experiments on the sequences generated by it. The experiment outcomes we obtained are listed in Table 4. As can be observed, the output p values are considerably greater than the threshold of 0.01, whether they are x sequences or y sequences.

Consequently, 2D-SCPM does possess excellent randomization performance and is ideal for designing cryptosystems.

Table 4. NIST test outcomes for 2D-SCPM.

Item Name	p Value		Result
	x Sequence	y Sequence	
Frequency (Monobit)	0.133171	0.536575	Random
Frequency (Block)	0.124135	0.517819	Random
Runs	0.452929	0.096436	Random
Longest Runs	0.638713	0.353124	Random
Matrix Rank	0.067070	0.872194	Random
Discrete Fourier Transform	0.393422	0.485537	Random
Non-Overlapping Template	0.441853	0.431362	Random
Overlapping Template	0.137611	0.062761	Random
Universal	0.777250	0.120874	Random
Linear Complexity	0.633858	0.953684	Random
Serial 1	0.695361	0.291362	Random
Serial 2	0.768519	0.382965	Random
Approximate Entropy	0.781378	0.229411	Random
Cumulative Sums (Forward)	0.164709	0.548417	Random
Cummulative Sums (Reverse)	0.145096	0.872657	Random
Random Excursions ($x = -1$)	0.311634	0.380218	Random
Random Excursions ($x = +1$)	0.162339	0.423355	Random
Random Excursions Variant ($x = -1$)	0.118109	0.562370	Random
Random Excursions Variant ($x = +1$)	0.177507	0.825329	Random

4. Proposed MIEA-FCSM

Based on the excellent chaotic performance of the fractional-order 3D Lorenz chaotic system and 2D-SCPM, we further designed a highly efficient multi-image encryption algorithm named MIEA-FCSM so as to achieve more secure and efficient image encryption. As shown in Figure 7, MIEA-FCSM is mainly composed of four parts, namely the generation of chaotic sequences, multi-channel fusion, generation of plaintext-related parameters, and two rounds of plaintext-related substitution, diffusion, and scrambling.

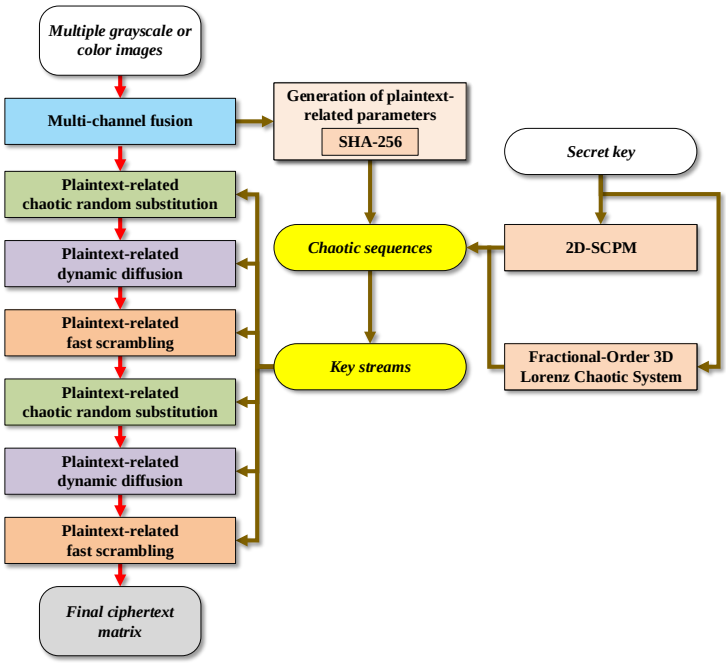


Figure 7. Encryption process of MIEA-FCSM.

4.1. Generation of Chaotic Sequences

In our proposed MIEA-FCSM, the chaotic sequences $\mathbf{G}^{(1)}$ and $\mathbf{G}^{(2)}$ used to encrypt input images are generated by adopting the secret key

$$K = \{x_0^{(1)}, y_0^{(1)}, z_0, \sigma, \rho, \beta, \alpha, x_0^{(2)}, y_0^{(2)}, a, b\}.$$

Specifically, the first seven components $\{x_0^{(1)}, y_0^{(1)}, z_0, \sigma, \rho, \beta, \alpha\}$ of K are input into the fractional-order 3D Lorenz chaotic system, thereby generating the sequence $\mathbf{G}^{(1)}$ comprising $2^{11} + \lceil \tilde{H}/2 \rceil \times \lceil (\tilde{W} \times \tilde{D})/3 \rceil$ elements. Here, $\tilde{H}$, $\tilde{W}$, and $\tilde{D}$ denote the size of the input images with the largest number of pixels to be encrypted, respectively. And $\lceil \bullet \rceil$ returns the smallest integer greater than or equal to the operand. In the first round of substitution, diffusion, and scrambling of MIEA-FCSM, $\mathbf{G}^{(1)}$ will further be converted into the key streams used in these encryption steps.

Similarly, the last four components $\{x_0^{(2)}, y_0^{(2)}, a, b\}$ of K are exploited to iterate 2D-SCPM so as to generate the chaotic sequence $\mathbf{G}^{(2)}$ with the same length as $\mathbf{G}^{(1)}$. And $\mathbf{G}^{(2)}$ will be employed in the second round of substitution, diffusion, and scrambling of MIEA-FCSM.

4.2. Multi-Channel Fusion

The proposed MIEA-FCSM can simultaneously encrypt D 8-bit grayscale images or $D/3$ 24-bit true-color images, where D is an integer divisible by 3. For the input images, we can represent them as a 3D pixel matrix $\mathbf{P}$ of size $H \times W \times D$. Here, H is the number of rows, W is the number of columns, and D represents the number of grayscale images or channels of 24-bit true-color images. To achieve higher encryption efficiency, we perform multi-channel fusion on the input $\mathbf{P}$ according to the following steps:

- **Step 1** : Determine whether $H/2$ is an integer. If $H/2$ is not an integer, fill $\mathbf{P}$ with zero-valued pixels, thereby letting $H = H + 1$.
- **Step 2**: Reshape $\mathbf{P}$ into a 3D matrix $\mathbf{C}^{(1)}$ of size $H' \times W' \times 6$, where $H' = H/2$ and $W' = (W \times D)/3$.
- **Step 3**: Fuse $\mathbf{C}^{(1)}$ into a 2D matrix $\mathbf{C}^{(2)}$ of size $H' \times W'$. Specifically, let

$$\mathbf{C}^{(2)}(i,j) = \sum_{k=1}^{6} \mathbf{C}^{(1)}(i,j,k) \times 2^{(6-k)\times 8}, \tag{6}$$

  where $i = 1, 2, \ldots, H'$, and $j = 1, 2, \ldots, W'$.

From the above steps, it can be seen that through multi-channel fusion, the number of basic operations required for encrypting $\mathbf{P}$ is reduced to one-sixth of the original number, thus contributing to achieving higher encryption efficiency. For example, if the size of six input grayscale images is 512×512, then the size of the fused pixel matrix that needs to be encrypted is 256×1024; if the size of two input color images is $1024 \times 1024 \times 3$, then the size of the fused pixel matrix is 512×2048.

4.3. Generation of Plaintext-Related Parameters

Due to its extreme sensitivity to input, SHA-256 has been widely employed in image encryption to enhance the plaintext sensitivity of an image encryption algorithm [16]. In MIEA-FCSM, we first use SHA-256 to obtain the 32-byte hash value $\mathbf{h}$ of $\mathbf{C}^{(2)}$. Then, $\mathbf{h}$ is

employed to generate two plaintext-related parameters, $r^{(1)}$ and $r^{(2)}$, for the subsequent encryption steps. Specifically,

$$\begin{cases} r^{(1)} = (\sum_{i=1}^{32} \mathbf{h}(i)) \bmod 2^{11}, \\ r^{(2)} = ((\sum_{j=1}^{5} \sum_{k=6\times j-5}^{6\times j} \mathbf{h}(k) \times 2^{(6\times j-k)}) + \sum_{l=27}^{32} \mathbf{h}(l) \times 2^{(32-l)}) \bmod 2^{48}. \end{cases} \tag{7}$$

4.4. Plaintext-Related Chaotic Random Substitution

To enhance the sensitivity of MIEA-FCSM to plaintext pixels and improve the randomness of ciphertext pixels, we arranged two rounds of plaintext-related chaotic random substitutions in MIEA-FCSM. Specifically, in the first round of plaintext-related chaotic random substitution, a chaotic matrix $\mathbf{B}^{(1)}$ is first formed by reshaping the $H' \times W'$ element of $\mathbf{G}^{(1)}$:

$$\mathbf{B}^{(1)} = reshape(\left\lfloor \left| \mathbf{G}^{(1)}(r^{(1)} + 1 : r^{(1)} + H' \times W') \right| \times 10^{15} \right\rfloor \bmod 2^{48}, H', W'), \tag{8}$$

where $\lfloor \bullet \rfloor$ returns the maximum integer that is less than or equal to the operand. Then, a matrix XOR operation is applied to the input matrix $\mathbf{C}^{(2)}$ to obtain the output matrix

$$\mathbf{C}^{(3)} = ((\mathbf{C}^{(2)} + r^{(2)}) \bmod 2^{48}) \oplus \mathbf{B}^{(1)}. \tag{9}$$

Similarly, in the second round of plaintext-related chaotic random substitution, a chaotic matrix $\mathbf{B}^{(2)}$ is first formed by reshaping the $H' \times W'$ element of $\mathbf{G}^{(2)}$:

$$\mathbf{B}^{(2)} = reshape(\left\lfloor \left| \mathbf{G}^{(2)}(r^{(1)} + 1 : r^{(1)} + H' \times W') \right| \times 10^{15} \right\rfloor \bmod 2^{48}, H', W'). \tag{10}$$

Then, a matrix modular addition operation is applied to the input matrix $\mathbf{C}^{(5)}$ to obtain the output matrix

$$\mathbf{C}^{(6)} = ((\mathbf{C}^{(5)} \oplus r^{(2)}) + \mathbf{B}^{(2)}) \bmod 2^{48}. \tag{11}$$

4.5. Plaintext-Related Dynamic Diffusion

In order to enhance encryption efficiency while ensuring security, we introduced two rounds of plaintext-related dynamic diffusions in MIEA-FCSM. Unlike typical pixel-level diffusion methods adopted in many existing image encryption algorithms, MIEA-FCSM's diffusion operations are performed in the form of multiple rows (the first round) and columns (the second round), thus leading to a significant improvement in encryption efficiency. Furthermore, due to their dynamic nature depending on plaintext pixels, the diffusion operations adopted by MIEA-FCSM also possess significant advantages in resisting plaintext attacks. Specifically, the first round of plaintext-related dynamic diffusion is performed as follows:

- **Step 1**: For the input matrix $\mathbf{C}^{(3)}$ of size $H' \times W'$, the diffusion operation is first performed on the first four rows of $\mathbf{C}^{(3)}$, so

$$\mathbf{C}^{(4)}(1:4,:) = (\mathbf{C}^{(3)}(1:4,:) + \mathbf{C}^{(3)}(H'-3:H',:)) \bmod 2^{48}. \tag{12}$$

- **Step 2**: Let $\alpha = \lceil H'/4 \rceil - 1$.
- **Step 3**: When $i = 2$ to α, repeat the following operations:
  Let $\beta = (r^{(1)} + \mathbf{B}^{(1)}(i,1)) \bmod 2$. If $\beta = 1$, then

$$\mathbf{C}^{(4)}(4i-3:4i,:) = \mathbf{C}^{(3)}(4i-3:4i,:) \oplus \mathbf{C}^{(4)}(4i-7:4i-4,:). \tag{13}$$

Otherwise,

$$\mathbf{C}^{(4)}(4i-3:4i,:) = (\mathbf{C}^{(3)}(4i-3:4i,:) + \mathbf{C}^{(4)}(4i-7:4i-4,:)) \bmod 2^{48}. \quad (14)$$

- **Step 4**: Let $\mu = H' \bmod 4$, and perform the diffusion operation on the remaining μ (when $\mu = 1, 2, 3$) or $\mu + 4$ (when $\mu = 0$) rows of $\mathbf{C}^{(3)}$:

$$\begin{cases} \mathbf{C}^{(4)}(H',:) = \mathbf{C}^{(3)}(H',:) \oplus \mathbf{C}^{(4)}(H'-1,:) & \mu = 1, \\ \mathbf{C}^{(4)}(H'-1:H',:) = \mathbf{C}^{(3)}(H'-1:H',:) \oplus \mathbf{C}^{(4)}(H'-3:H'-2,:) & \mu = 2, \\ \mathbf{C}^{(4)}(H'-2:H',:) = \mathbf{C}^{(3)}(H'-2:H',:) \oplus \mathbf{C}^{(4)}(H'-5:H'-3,:) & \mu = 3, \\ \mathbf{C}^{(4)}(H'-3:H',:) = \mathbf{C}^{(3)}(H'-3:H',:) \oplus \mathbf{C}^{(4)}(H'-7:H'-4,:) & \mu = 0. \end{cases} \quad (15)$$

Similarly, the second round of plaintext-related dynamic diffusion is performed as follows:

- **Step 1**: For the input matrix $\mathbf{C}^{(6)}$ of size $H' \times W'$, the diffusion operation is first performed on the first four columns of $\mathbf{C}^{(6)}$, so

$$\mathbf{C}^{(7)}(:,1:4) = (\mathbf{C}^{(6)}(:,1:4) + \mathbf{C}^{(6)}(:,W'-3:W')) \bmod 2^{48}. \quad (16)$$

- **Step 2**: Let $\alpha' = \lceil W'/4 \rceil - 1$.
- **Step 3**: When $i = 2$ to α', repeat the following operations: Let $\beta' = (r^{(1)} + \mathbf{B}^{(2)}(1,i)) \bmod 2$. If $\beta' = 1$, then

$$\mathbf{C}^{(7)}(:,4i-3:4i) = \mathbf{C}^{(6)}(:,4i-3:4i) \oplus \mathbf{C}^{(7)}(:,4i-7:4i-4). \quad (17)$$

Otherwise,

$$\mathbf{C}^{(7)}(:,4i-3:4i) = (\mathbf{C}^{(6)}(:,4i-3:4i) + \mathbf{C}^{(7)}(:,4i-7:4i-4)) \bmod 2^{48}. \quad (18)$$

- **Step 4**: Let $\mu' = W' \bmod 4$, and perform the diffusion operation on the remaining μ' (when $\mu' = 1, 2, 3$) or $\mu' + 4$ (when $\mu' = 0$) columns of $\mathbf{C}^{(6)}$:

$$\begin{cases} \mathbf{C}^{(7)}(:,W') = \mathbf{C}^{(6)}(:,W') \oplus \mathbf{C}^{(7)}(:,W'-1) & \mu' = 1, \\ \mathbf{C}^{(7)}(:,W'-1:W') = \mathbf{C}^{(6)}(:,W'-1:W') \oplus \mathbf{C}^{(7)}(:,W'-3:W'-2) & \mu' = 2, \\ \mathbf{C}^{(7)}(:,W'-2:W') = \mathbf{C}^{(6)}(:,W'-2:W') \oplus \mathbf{C}^{(7)}(:,W'-5:W'-3) & \mu' = 3, \\ \mathbf{C}^{(7)}(:,W'-3:W') = \mathbf{C}^{(6)}(:,W'-3:W') \oplus \mathbf{C}^{(7)}(:,W'-7:W'-4) & \mu' = 0. \end{cases} \quad (19)$$

4.6. Plaintext-Related Fast Scrambling

Finally, to further enhance the security of MIEA-FCSM, we also incorporated one round of fast scrambling after each round of dynamic diffusion. Specifically, the first round of plaintext-related fast scrambling is executed in the following manner:

- **Step 1**: Let $\theta = (r^{(1)} + r^{(2)}) \bmod 2^{10}$.
- **Step 2**: Sort $\mathbf{G}^{(1)}(\theta+1:\theta+H')$ in ascending order to obtain the row index vector $\bar{\mathbf{V}}^{(r)}$ of length H'.
- **Step 3**: Sort $\mathbf{G}^{(1)}(\theta+H'+1:\theta+H'+W')$ in ascending order to obtain the column index vector $\bar{\mathbf{V}}^{(c)}$ of length W'.
- **Step 4**: For each (i,j), where $i = 1, 2, \ldots, H'$ and $j = 1, 2, \ldots, W'$, let

$$\mathbf{C}^{(5)}(i,j) = \mathbf{C}^{(4)}(\bar{\mathbf{V}}^{(r)}(i), \bar{\mathbf{V}}^{(c)}(j)). \quad (20)$$

Similarly, the second round of plaintext-related fast scrambling is executed in the following manner:

- **Step 1**: Let $\theta' = (r^{(1)} \times r^{(2)}) \bmod 2^{10}$.
- **Step 2**: Sort $\mathbf{G}^{(2)}(\theta' + 1 : \theta' + H')$ in ascending order to obtain the row index vector $\bar{\mathbf{V}}'^{(r)}$ of length H'.
- **Step 3**: Sort $\mathbf{G}^{(2)}(\theta' + H' + 1 : \theta' + H' + W')$ in ascending order to obtain the column index vector $\bar{\mathbf{V}}'^{(c)}$ of length W'.
- **Step 4**: For each (i, j), where $i = 1, 2, \ldots, H'$ and $j = 1, 2, \ldots, W'$, let

$$\mathbf{C}(i, j) = \mathbf{C}^{(7)}(\bar{\mathbf{V}}^{(r)}(i), \bar{\mathbf{V}}^{(c)}(j)). \tag{21}$$

4.7. Complete Process of MIEA-FCSM

To provide a clearer demonstration of the proposed MIEA-FCSM, a comprehensive yet concise description of its encryption and decryption processes is presented here. Suppose the encrypting party (sender) is *Alice* and the decrypting party (receiver) is *Bob*. The size of the 3D pixel matrix $\mathbf{P}$ to be encrypted and transmitted between them is $H \times W \times D$. By exploiting the mutually agreed-upon secret key $K = \{x_0^{(1)}, y_0^{(1)}, z_0, \sigma, \rho, \beta, \alpha, x_0^{(2)}, y_0^{(2)}, a, b\}$, *Alice* will complete the encryption process through the following steps:

- **Step 1**: If K is being used for encryption for the first time, generate chaotic sequences $\mathbf{G}^{(1)}$ and $\mathbf{G}^{(2)}$. Otherwise, proceed directly to **Step 2**. For specific details on the generation of $\mathbf{G}^{(1)}$ and $\mathbf{G}^{(2)}$, please refer to Section 4.1.
- **Step 2**: Fuse $\mathbf{P}$ through multi-channel fusion into a 2D matrix $\mathbf{C}^{(2)}$ of size $H' \times W'$. For specific details on multi-channel fusion, please refer to Section 4.2.
- **Step 3**: Employ SHA-256 to obtain the 32-byte hash value $\mathbf{h}$ of $\mathbf{C}^{(2)}$. And then use $\mathbf{h}$ to generate two plaintext-related parameters $r^{(1)}$ and $r^{(2)}$. For specific details on the generation of these two plaintext-related parameters, please refer to Section 4.3.
- **Step 4**: Perform the first round of plaintext-related chaotic random substitution on $\mathbf{C}^{(2)}$ using $r^{(1)}$, $r^{(2)}$, and $\mathbf{G}^{(1)}$ to obtain the substituted $\mathbf{C}^{(3)}$. For specific details on plaintext-related chaotic random substitution, please refer to Section 4.4.
- **Step 5**: Utilize $r^{(1)}$ and $\mathbf{B}^{(1)}$ to carry out the first round of plaintext-related dynamic diffusion on $\mathbf{C}^{(3)}$ so as to obtain the diffused $\mathbf{C}^{(4)}$. For specific details on plaintext-related dynamic diffusion, please refer to Section 4.5.
- **Step 6**: Conduct the first round of plaintext-related fast scrambling on $\mathbf{C}^{(4)}$ using $r^{(1)}$, $r^{(2)}$, and $\mathbf{G}^{(1)}$ to obtain the scrambled $\mathbf{C}^{(5)}$. For specific details on plaintext-related fast scrambling, please refer to Section 4.6.
- **Step 7**: Perform the second round of plaintext-related chaotic random substitution on $\mathbf{C}^{(5)}$ using $r^{(1)}$, $r^{(2)}$, and $\mathbf{G}^{(2)}$ to obtain the substituted $\mathbf{C}^{(6)}$.
- **Step 8**: Utilize $r^{(1)}$ and $\mathbf{B}^{(2)}$ to carry out the second round of plaintext-related dynamic diffusion on $\mathbf{C}^{(6)}$ so as to obtain the diffused $\mathbf{C}^{(7)}$.
- **Step 9**: Conduct the second round of plaintext-related fast scrambling on $\mathbf{C}^{(7)}$ using $r^{(1)}$, $r^{(2)}$, and $\mathbf{G}^{(2)}$ to obtain the final ciphertext $\mathbf{C}$.

After obtaining the final ciphertext $\mathbf{C}$ through encryption, *Alice* sends $\mathbf{C}$ and the 32-byte hash value $\mathbf{h}$ to *Bob* through the public channel.

The decryption process of MIEA-FCSM is the reverse process of its encryption process, as illustrated in Figure 8.

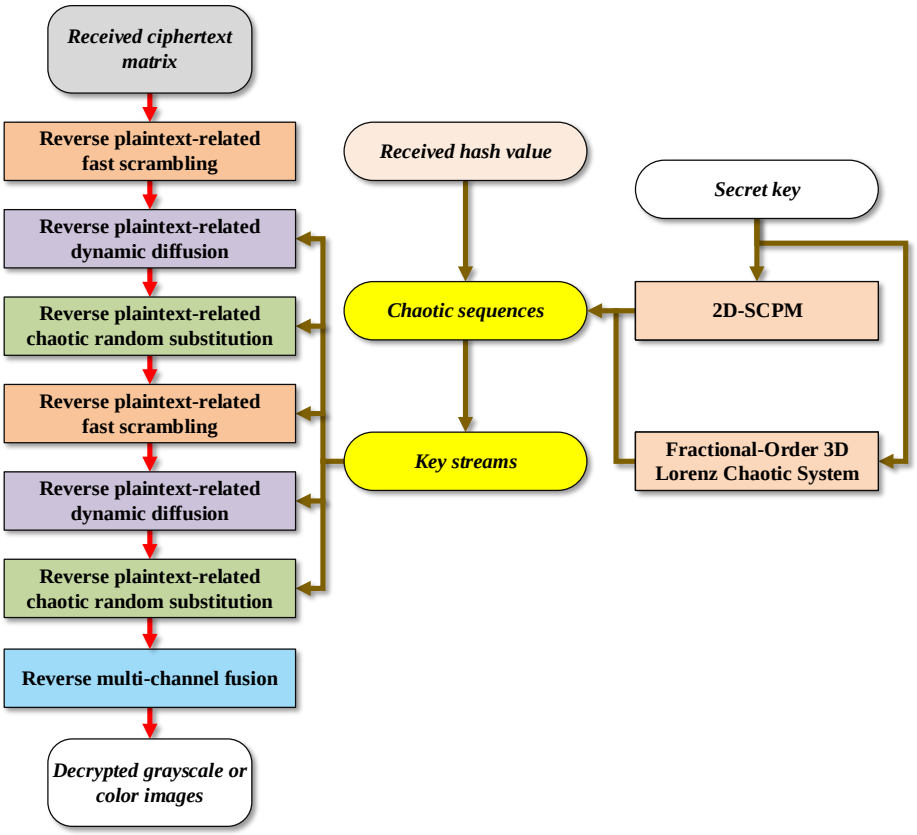


Figure 8. Decryption process of MIEA-FCSM.

By exploiting the received **C** and **h**, *Bob* will complete the decryption process through the following steps:

- **Step 1**: If K is being used for decryption for the first time, generate chaotic sequences $\mathbf{G}^{(1)}$ and $\mathbf{G}^{(2)}$. Otherwise, proceed directly to **Step 2**.
- **Step 2**: Employ **h** to generate two plaintext-related parameters $r^{(1)}$ and $r^{(2)}$.
- **Step 3**: Using $r^{(1)}$, $r^{(2)}$, and $\mathbf{G}^{(2)}$, conduct the reverse operations corresponding to the second round of plaintext-related fast scrambling on **C** so as to obtain $\mathbf{C}^{(7)}$.
- **Step 4**: Employ $r^{(1)}$ and $\mathbf{B}^{(2)}$ to carry out the reverse operations corresponding to the second round of plaintext-related dynamic diffusion on $\mathbf{C}^{(7)}$ so as to obtain $\mathbf{C}^{(6)}$.
- **Step 5**: Using $r^{(1)}$, $r^{(2)}$, and $\mathbf{G}^{(2)}$, perform the reverse operations corresponding to the second round of plaintext-related chaotic random substitution on $\mathbf{C}^{(6)}$ so as to obtain $\mathbf{C}^{(5)}$.
- **Step 6**: Employing $r^{(1)}$, $r^{(2)}$, and $\mathbf{G}^{(1)}$, conduct the reverse operations corresponding to the first round of plaintext-related fast scrambling on $\mathbf{C}^{(5)}$ so as to obtain $\mathbf{C}^{(4)}$.
- **Step 7**: Utilize $r^{(1)}$ and $\mathbf{B}^{(1)}$ to carry out the reverse operations corresponding to the first round of plaintext-related dynamic diffusion on $\mathbf{C}^{(4)}$ so as to obtain $\mathbf{C}^{(3)}$.
- **Step 8**: Employing $r^{(1)}$, $r^{(2)}$, and $\mathbf{G}^{(1)}$, perform the reverse operations corresponding to the first round of plaintext-related chaotic random substitution on $\mathbf{C}^{(3)}$ so as to obtain $\mathbf{C}^{(2)}$.
- **Step 9**: Conduct the reverse operations corresponding to multi-channel fusion on $\mathbf{C}^{(2)}$ so as to obtain the final decrypted 3D pixel matrix **P**.

To maintain brevity, the repetitive explanation of the reverse operations corresponding to each encryption step is omitted here, as there is no substantial difference between them.

5. Simulation Experiments

A series of experiments are presented in this section to validate the efficiency and security superiority of MIEA-FCSM. In these experiments, many test images from the well-known USC-SIPI database were employed. A microcomputer featuring MATLAB R2017a, an Intel CPU E3-1231 v3, and 8 GB of RAM was utilized to carry out these experiments. All experiments employed randomly generated secret keys to guarantee the objectivity of performance evaluation. Furthermore, the final fused ciphertext pixels were divided into 8-bit ciphertext pixels for ease of demonstration and comparison with other image encryption algorithms.

5.1. Visual Effect

To guarantee effective protection for images, it is essential for a suggested image encryption algorithm to have the ability to encrypt them into unrecognizable images that resemble noise. Six grayscale images (5.2.08, 5.2.09, 5.2.10, boat.512, elaine.512, and gray21.512) and two color images (4.2.06 and 4.2.07) were encrypted and then decrypted using MIEA-FCSM. Six grayscale images were encrypted simultaneously in the first round, while two color images were encrypted simultaneously in the second round. As can be observed from Figure 9, after encryption by MIEA-FCSM, these images containing rich details have all become indiscernible noise-like images. However, once decrypted, these unrecognizable images are restored to their original state without any loss. This demonstrates that MIEA-FCSM possesses exceptional encryption and decryption capabilities and thus can provide effective protection for images.

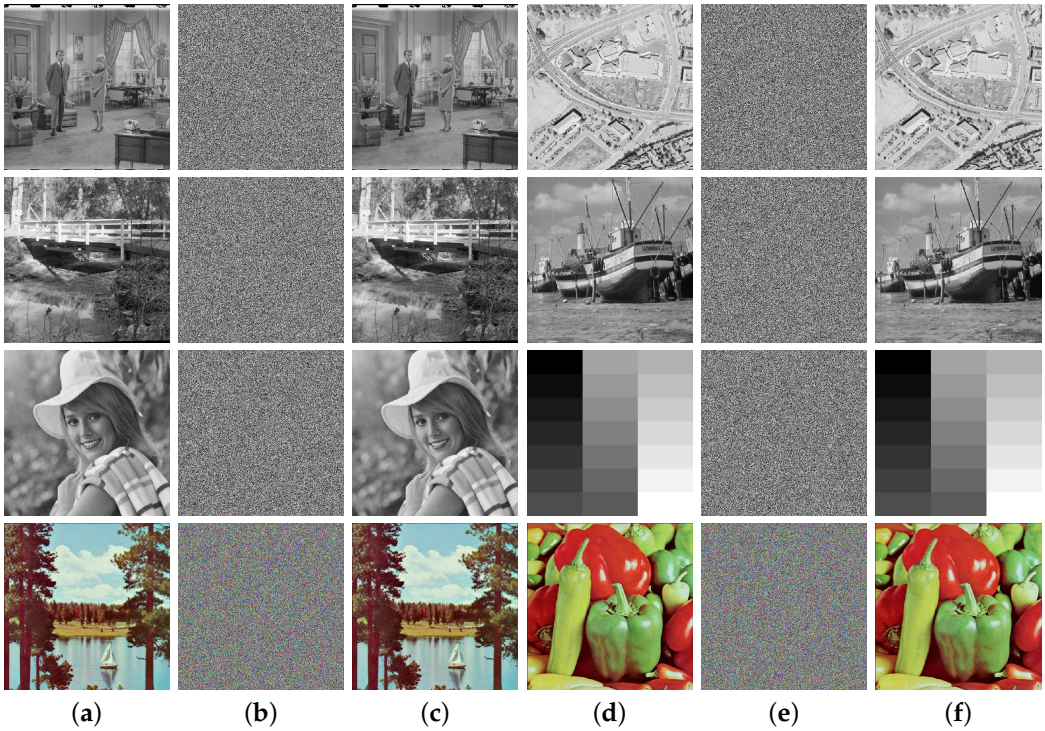


Figure 9. Visual effect experiment results for MIEA-FCSM: (**a**,**d**), six grayscale images and two color images; (**b**,**e**), corresponding encrypted images; (**c**,**f**), corresponding decrypted images.

5.2. Key Space

As a straightforward and easily executable form of attack, brute-force attacks break a cryptosystem by attempting all secret keys within the key space. Currently, it is widely believed that the key space of a cryptosystem should be larger than 2^{128} [4]. Otherwise, it would be challenging to withstand brute-force attacks. As mentioned in Section 4.1, MIEA-FCSM's secret key consists of eleven parts, namely $\{x_0^{(1)}, y_0^{(1)}, z_0, \sigma, \rho, \beta, \alpha, x_0^{(2)}, y_0^{(2)}, a, b\}$.

In MIEA-FCSM, we set the value ranges of these components as $x_0^{(1)} \in [-20, 20]$, $y_0^{(1)} \in [-25, 25]$, $z_0 \in [0, 55]$, $\sigma \in [5.81, 17.5]$, $\rho \in [24, 70]$, $\beta \in [1, 3.3]$, $\alpha \in [0.92, 1]$, $x_0^{(2)} \in (0, 1)$, $y_0^{(2)} \in (0, 1)$, $a \in [1, 12]$, and $b \in [1, 12]$. When the effective calculation precision of floating-point numbers is determined as 10^{-14}, it is possible to determine the size of MIEA-FCSM's key space $\hat{S}^{(K)} = 1.3169 \times 10^{163} \approx 2^{541}$. Given that 2^{541} is significantly larger than 2^{128}, our suggested MIEA-FCSM can effectively withstand brute-force attacks.

5.3. Key Sensitivity

Confusion is an essential principle in the design of cryptosystems, which requires the relationship between the secret key and the ciphertext to be highly complex [4]. Therefore, a qualified image encryption algorithm should be highly sensitive to changes in the secret key. Even with the smallest change to the secret key, the ciphertext should also undergo extremely large changes. In order to demonstrate MIEA-FCSM's sensitivity to the secret key, we encrypted 4.1.07 with a randomly generated key $\tilde{K} = \{\tilde{x}_0^{(1)}, \tilde{y}_0^{(1)}, \tilde{z}_0, \tilde{\sigma}, \tilde{\rho}, \tilde{\beta}, \tilde{\alpha}, \tilde{x}_0^{(2)}, \tilde{y}_0^{(2)}, \tilde{a}, \tilde{b}\}$, where

$$\begin{cases} \tilde{x}_0^{(1)} = 2.97059278176062, \\ \tilde{y}_0^{(1)} = 3.95716694824294, \\ \tilde{z}_0 = 4.48537564872284, \\ \tilde{\sigma} = 10.80028046888880, \\ \tilde{\rho} = 28.14188633862721, \\ \tilde{\beta} = 3.08842794929294, \\ \tilde{\alpha} = 0.96573552518906, \\ \tilde{x}_0^{(2)} = 0.79220732955955, \\ \tilde{y}_0^{(2)} = 0.95949242639290, \\ \tilde{a} = 8.65574069915658, \\ \tilde{b} = 8.03571167857419. \end{cases} \tag{22}$$

Then, we obtained eleven new keys with only minimal differences from $\tilde{K}$ by modifying one component of $\tilde{K}$ each time. After encrypting 4.1.07 with these eleven keys, we calculated the difference image between each ciphertext image and the original ciphertext image. Figure 10 presents the experimental results obtained. Clearly, even though each component of $\tilde{K}$ underwent only the smallest change of 10^{-14}, the resulting ciphertext image was completely changed. And each difference image between the changed ciphertext image and the original one is extremely similar to a noisy image. Thus, MIEA-FCSM has an extremely high key sensitivity.

5.4. Plaintext Sensitivity

Generally, differential attacks are regarded as the most menacing compared to other types of attacks. Differential attacks involve the analysis of the mathematical connection between modifications in plaintext pixels and the consequent changes in ciphertext pixels. Consequently, a qualified image encryption algorithm must be extremely sensitive to minimum changes in plaintext pixels. In other words, even if only one pixel bit is modified, the ciphertext must be completely changed. To demonstrate the plaintext sensitivity of MIEA-FCSM, we inverted the two pixel bits of 2.1.06, as depicted in Figure 11a2,a3. Then, we encrypted the three plaintext images and calculated the related difference images. From Figure 11b1,b2, one can find that each modified plaintext image has almost no difference from 2.1.06. However, the corresponding ciphertext images are completely changed, and the difference images between them and the original ciphertext image resemble random images, as shown in Figure 11d1,d2. This reveals that MIEA-FCSM features an extraordinary level of sensitivity to plaintext pixels.

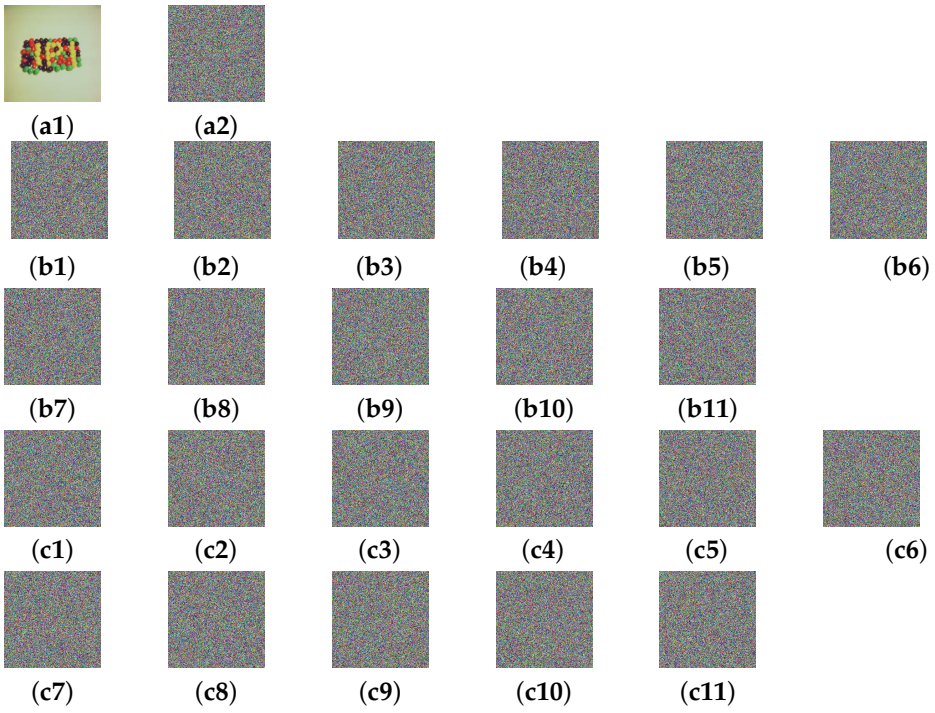


Figure 10. Visual presentation of key sensitivity for MIEA-FCSM: (**a1**) 4.1.07; (**a2**) ciphertext of 4.1.07; (**b1**) cihpertext obtained after $\tilde{x}_0^{(1)} = \tilde{x}_0^{(1)} + 10^{-14}$; (**b2**) $\tilde{y}_0^{(1)} = \tilde{y}_0^{(1)} + 10^{-14}$; (**b3**) $\tilde{z}_0 = \tilde{z}_0 + 10^{-14}$; (**b4**) $\tilde{\sigma} = \tilde{\sigma} + 10^{-14}$; (**b5**) $\tilde{\rho} = \tilde{\rho} + 10^{-14}$; (**b6**) $\tilde{\beta} = \tilde{\beta} + 10^{-14}$; (**b7**) $\tilde{\alpha} = \tilde{\alpha} + 10^{-14}$; (**b8**) $\tilde{x}_0^{(2)} = \tilde{x}_0^{(2)} + 10^{-14}$; (**b9**) $\tilde{y}_0^{(2)} = \tilde{y}_0^{(2)} + 10^{-14}$; (**b10**) $\tilde{a} = \tilde{a} + 10^{-14}$; (**b11**) $\tilde{b} = \tilde{b} + 10^{-14}$; (**c1**) difference between (**b1**) and (**a2**); (**c2**) between (**b2**) and (**a2**); (**c3**) between (**b3**) and (**a2**); (**c4**) between (**b4**) and (**a2**); (**c5**) between (**b5**) and (**a2**); (**c6**) between (**b6**) and (**a2**); (**c7**) between (**b7**) and (**a2**); (**c8**) between (**b8**) and (**a2**); (**c9**) between (**b9**) and (**a2**); (**c10**) between (**b10**) and (**a2**); (**c11**) between (**b11**) and (**a2**).

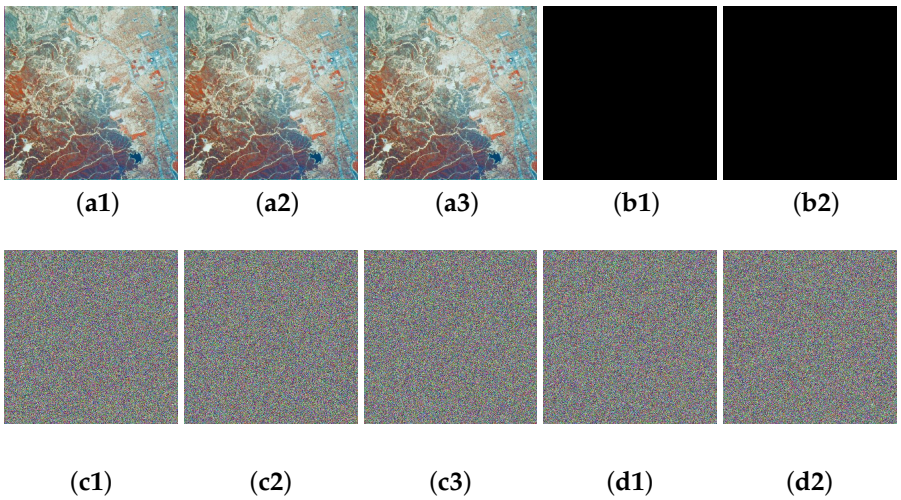


Figure 11. Visual presentation of plaintext sensitivity for MIEA-FCSM: (**a1**) 2.1.06; (**a2**) the lowest bit of the first pixel on the red channel is negated; (**a3**) the lowest bit of the last pixel on the blue channel is negated; (**b1**) difference between (**a1**) and (**a2**); (**b2**) difference between (**a1**) and (**a3**); (**c1**) ciphertext of (**a1**); (**c2**) ciphertext of (**a2**); (**c3**) ciphertext of (**a3**); (**d1**) difference between (**c1**) and (**c2**); (**d2**) difference between (**c1**) and (**c3**).

To further verify the superiority of MIEA-FCSM regarding plaintext sensitivity, we conducted additional quantitative evaluations of MIEA-FCSM using the two commonly used metrics: the number of pixels change rate (NPCR) and the unified average changing

intensity (UACI). For two images **A** and **B** of size $U \times V$, one can calculate their NPCR and UACI values as follows:

$$\mathrm{NPCR}(\mathbf{A}, \mathbf{B}) = \sum_{u=1}^{U} \sum_{v=1}^{V} \mathbf{D}(u, v) / (U \times V) \times 100\%, \tag{23}$$

$$\mathrm{UACI}(\mathbf{A}, \mathbf{B}) = \sum_{u=1}^{U} \sum_{v=1}^{V} \frac{|\mathbf{A}(u, v) - \mathbf{B}(u, v)|}{255 \times U \times V} \times 100\%, \tag{24}$$

where $\mathbf{D}(u, v)$ represents the difference between $\mathbf{A}(u, v)$ and $\mathbf{B}(u, v)$. If $\mathbf{A}(u, v) = \mathbf{B}(u, v)$, then $\mathbf{D}(u, v) = 0$; otherwise, $\mathbf{D}(u, v) = 1$. Through a large number of experiments, we calculated the UPCR and UACI values between the ciphertext images obtained before and after single minimum plaintext changes. By examining Tables 5 and 6, we can observe that MIEA-FCSM attained the average values (99.6098, 33.4659) closest to the ideal values (99.6094, 33.4635) and demonstrated the highest stability (0.0069, 0.0246). This indicates that MIEA-FCSM exhibits an exceedingly high plaintext sensitivity and is capable of effectively defending against diverse differential attacks.

Table 5. NPCR scores of six algorithms.

Size	Name	Channel	MIEA-FCSM	[28]	[32]	[33]	[34]	[35]
512 × 512	2.1.01	Red	99.6089	99.6204	99.6166	99.6215	99.6092	99.6211
	2.1.01	Green	99.6154	99.5853	99.6109	99.6015	99.6184	99.6197
	2.1.01	Blue	99.5986	99.6052	99.6208	99.5973	99.6239	99.6212
	2.1.02	Red	99.5887	99.6201	99.6246	99.6100	99.6052	99.6148
	2.1.02	Green	99.6116	99.6506	99.5899	99.6054	99.5850	99.6115
	2.1.02	Blue	99.6139	99.6414	99.5987	99.6057	99.5918	99.6141
	2.1.05	Red	99.6143	99.6002	99.6067	99.5960	99.6387	99.6231
	2.1.05	Green	99.6070	99.6109	99.5853	99.6037	99.6098	99.6165
	2.1.05	Blue	99.6093	99.6216	99.5998	99.6154	99.5995	99.6232
1024 × 1024	2.2.01	Red	99.6112	99.6277	99.6099	99.6166	99.6095	99.6122
	2.2.01	Green	99.6144	99.5575	99.6076	99.6164	99.6026	99.6131
	2.2.01	Blue	99.6146	99.6490	99.6110	99.6141	99.6006	99.5956
	2.2.08	Red	99.6172	99.6155	99.6086	99.6169	99.6115	99.6021
	2.2.08	Green	99.6109	99.5834	99.6081	99.6190	99.6103	99.5975
	2.2.08	Blue	99.6150	99.6246	99.6113	99.6133	99.6055	99.5956
	2.2.11	Red	99.6044	99.5834	99.6119	99.6209	99.6166	99.6002
	2.2.11	Green	99.6092	99.6292	99.6135	99.6057	99.6044	99.6135
	2.2.11	Blue	99.6125	99.6277	99.6158	99.6015	99.6137	99.5972
	Average	–	99.6098	99.6141	99.6084	99.6101	99.6087	99.6107
	Std. Dev.	–	0.0069	0.0247	0.0098	0.0081	0.0119	0.0100

Table 6. UACI scores of six algorithms.

Size	Name	Channel	MIEA-FCSM	[28]	[32]	[33]	[34]	[35]
512 × 512	2.1.01	Red	33.4941	33.4893	33.4753	33.5109	33.5007	33.4930
	2.1.01	Green	33.4700	33.3720	33.4831	33.4686	33.4449	33.4849
	2.1.01	Blue	33.4736	33.4910	33.5122	33.4622	33.4479	33.4929
	2.1.02	Red	33.4623	33.4873	33.5137	33.4399	33.4839	33.4774
	2.1.02	Green	33.4349	33.5392	33.4412	33.4847	33.4263	33.4671
	2.1.02	Blue	33.4619	33.4500	33.3946	33.4360	33.4408	33.4752
	2.1.05	Red	33.4699	33.5945	33.4818	33.3924	33.4269	33.4778
	2.1.05	Green	33.4513	33.5220	33.4781	33.4470	33.4241	33.4634
	2.1.05	Blue	33.4537	33.3360	33.5460	33.4353	33.5226	33.4777

Table 6. *Cont.*

Size	Name	Channel	MIEA-FCSM	[28]	[32]	[33]	[34]	[35]
1024 × 1024	2.2.01	Red	33.4739	33.4782	33.4600	33.4887	33.4675	33.4286
	2.2.01	Green	33.4236	33.4194	33.4957	33.4808	33.4682	33.4255
	2.2.01	Blue	33.4628	33.6140	33.4181	33.5233	33.4399	33.4399
	2.2.08	Red	33.4825	33.5349	33.4666	33.4685	33.4486	33.4211
	2.2.08	Green	33.4789	33.4775	33.4720	33.4728	33.4585	33.4411
	2.2.08	Blue	33.4588	33.6040	33.4713	33.4726	33.4943	33.4607
	2.2.11	Red	33.5338	33.5236	33.4699	33.4547	33.4316	33.4004
	2.2.11	Green	33.4674	33.5185	33.4829	33.4779	33.4459	33.5004
	2.2.11	Blue	33.4330	33.5079	33.4347	33.5006	33.4450	33.4658
	Average	–	33.4659	33.4977	33.4721	33.4676	33.4565	33.4607
	Std. Dev.	–	0.0246	0.0729	0.0353	0.0309	0.0279	0.0284

5.5. Pixel Distribution

The pixel distribution characteristics in natural images are highly significant, as clearly illustrated in the first and third rows of Figure 12. Undoubtedly, a competent image encryption algorithm must eliminate these characteristics to effectively defend against various attacks based on pixel distribution. To validate MIEA-FCSM's pixel distribution performance, we encrypted two images, 2.1.01 and 2.1.07, and then plotted the pixel distribution diagrams of the output images generated by MIEA-FCSM. It is evident that the pixels, which were initially unevenly distributed for any one of the three channels (red, green, and blue), became remarkably uniform after being encrypted by MIEA-FCSM. This indicates that MIEA-FCSM possesses an exceptional ability to remove the pixel distribution characteristics of the input, thus effectively defending against various attacks that exploit such characteristics.

In addition to plotting histograms, we also conducted the chi-square test on the ciphertext images generated by MIEA-FCSA [36]. For a ciphertext image, we can calculate its chi-square value as follows:

$$\chi^2 = \sum_{k=1}^{m} (q_k - H \times W \times \rho)^2 / (H \times W \times \rho), \tag{25}$$

where q_k denotes the count of pixels with a value of $k-1$. m represents the maximum number of potential pixel values ($m = 256$ for 8-bit pixel depth), and $\rho = 1/m$. H and W correspond to the height and width of the ciphertext image, respectively. Afterwards, it is possible to determine the critical value $\chi^2_{0.05}(255)$ of the chi-square test at a significant level of 0.05, which amounts to 293.2478. If the chi-square value of a cipher image is below 293.2478, it can be considered to have passed the chi-square test successfully. This means that the pixel distribution of the ciphertext image is statistically close to a uniform distribution. Table 7 presents the results of our chi-square test on MIEA-FCSA. As can be observed, all color channels of the six ciphertext images have successfully passed the chi-square test. This demonstrates that the ciphertext images produced by MIEA-FCSM indeed exhibit an extremely uniform pixel distribution.

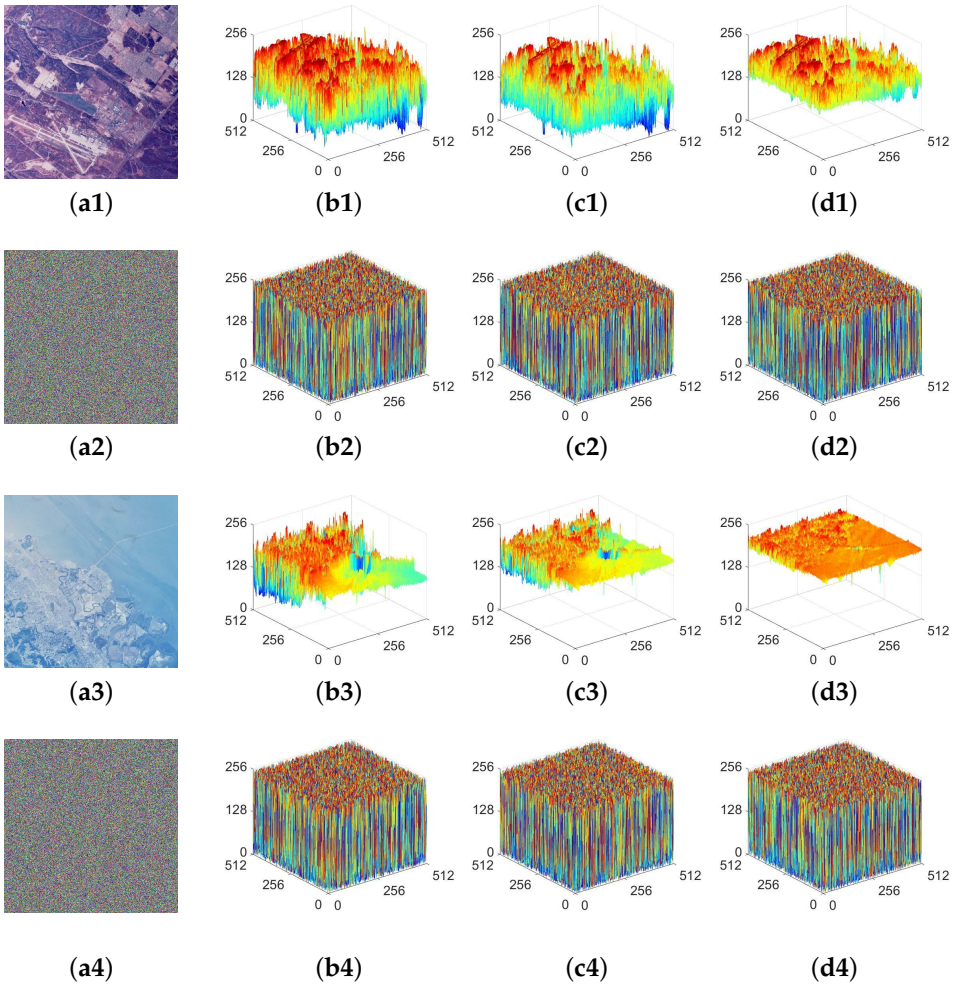


Figure 12. Pixel distribution representations for MIEA-FCSM: (**a1**) 2.1.01; (**b1**,**c1**,**d1**) are pixel distribution diagrams for the red, green, and blue channels of (**a1**); (**a2**) ciphertext of (**a1**); (**b2**,**c2**,**d2**) are three pixel distribution diagrams for (**a2**); (**a3**) 2.1.07; (**b3**,**c3**,**d3**) are three pixel distribution diagrams for (**a3**); (**a4**) ciphertext of (**a3**); (**b4**,**c4**,**d4**) are three pixel distribution diagrams for (**a4**).

Table 7. Chi-square test results of MIEA-FCSM.

Size	Channel	Ciphertext	Chi-Square Value $Ø_{0.05}{}^{2(255)=293.2478}$	Result
512 × 512	2.1.01	Red	249.7539	Pass
	2.1.01	Green	266.6308	Pass
	2.1.01	Blue	247.2011	Pass
	2.1.02	Red	259.5214	Pass
	2.1.02	Green	246.7753	Pass
	2.1.02	Blue	254.2734	Pass
	2.1.05	Red	258.1542	Pass
	2.1.05	Green	258.9863	Pass
	2.1.05	Blue	265.4589	Pass
1024 × 1024	2.2.01	Red	264.2778	Pass
	2.2.01	Green	261.9687	Pass
	2.2.01	Blue	257.9526	Pass
	2.2.08	Red	245.9931	Pass
	2.2.08	Green	256.5932	Pass
	2.2.08	Blue	260.8476	Pass
	2.2.11	Red	243.3730	Pass

Table 7. *Cont.*

Size	Channel	Ciphertext	Chi-Square Value $Ø_{0.05}{}^{2(255)=293.2478}$	Result
	2.2.11	Green	268.8398	Pass
	2.2.11	Blue	257.0625	Pass

5.6. Correlation Analysis

In natural images, significant correlations exist between adjacent pixels. Therefore, to avoid any associated security loopholes, a sound image encryption algorithm should effectively eliminate these correlations. With a randomly generated secret key, we encrypted two images, 2.1.05 and 2.1.06, and then drew the correlation analysis diagrams for the related images. Figure 13 presents the pixel correlations of the images prior to and following encryption in the horizontal, vertical, and diagonal directions. Clearly, the pixel correlations in all three directions are considerably significant for 2.1.05 and 2.1.06, reaching close to 1. Nevertheless, the two encrypted images generated by MIEA-FCSM are in stark contrast to them. MIEA-FCSM has effectively reduced these correlations, making it impossible to observe any discernible features.

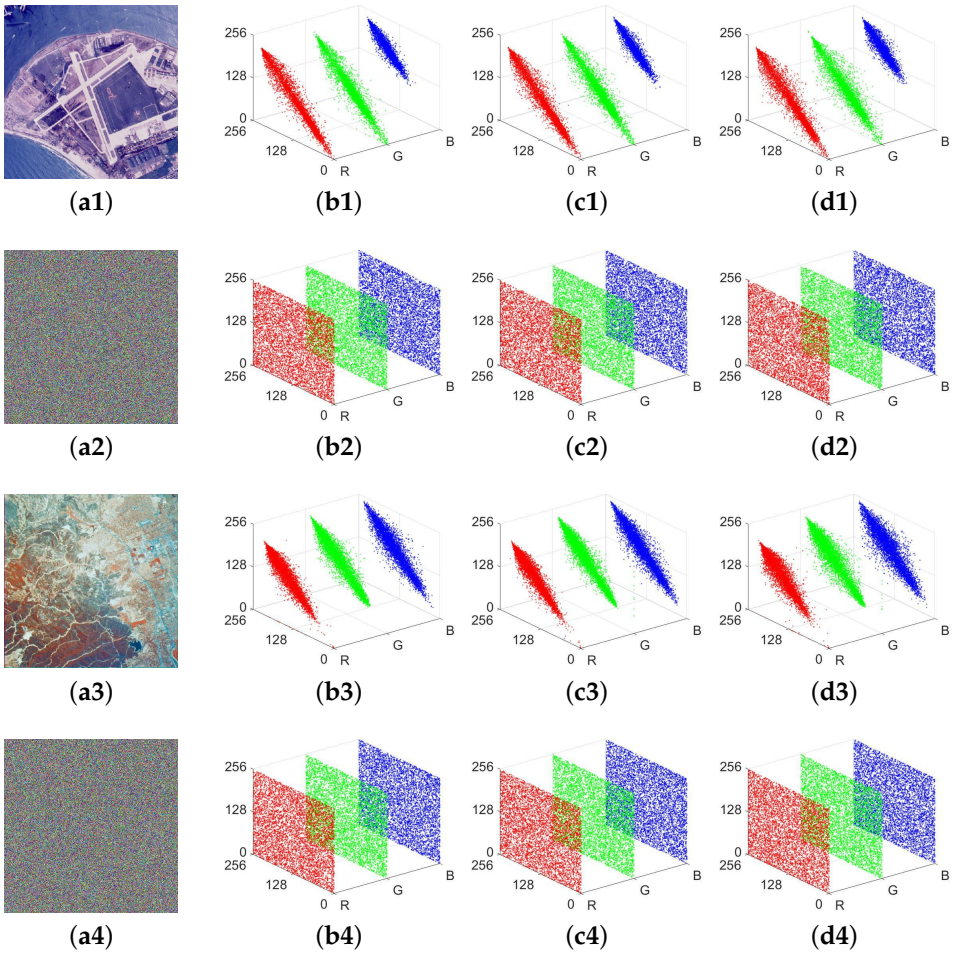


Figure 13. Adjacent pixel correlation representations: (**a1**) 2.1.05; (**b1**) correlation analysis diagram for (**a1**) in the horizontal direction; (**c1**) diagram for (**a1**) in the vertical direction; (**d1**) diagram for (**a1**) in the diagonal direction; (**a2**) ciphertext of 2.1.05; (**b2**,**c2**,**d2**) are correlation analysis diagrams for (**a2**); (**a3**) 2.1.06; (**b3**,**c3**,**d3**) are correlation analysis diagrams for (**a3**); (**a4**) ciphertext of 2.1.06; (**b4**,**c4**,**d4**) are correlation analysis diagrams for (**a2**).

To more accurately evaluate the ability of MIEA-FCSM to reduce pixel correlations, we employed the correlation coefficient (CC) for further quantitative analyses. Specifically, we can determine CC as follows:

$$\mathrm{CC} = \frac{E((A_u - E(A_u))(A_v - E(A_v))}{\sqrt{D(A_u)D(A_v)}}, \tag{26}$$

where A_u and A_v are pixel values, $E(A_u)$ and $E(A_v)$ represent expectations, and $D(A_u)$ and $D(A_v)$ denote variances. Table 8 lists the experimental results that were obtained. It is evident that all images exhibit high CC scores in all directions and channels. However, after undergoing MIEA-FCSM's encryption process, all CC scores drastically decreased to exceptionally low levels. This clearly shows MIEA-FCSM's superior performance in eliminating pixel correlations.

Table 8. CC scores of MIEA-FCSM.

Size	Name	Channel	Plaintext			Ciphertext		
			Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
512 × 512	2.1.01	Red	0.8632	0.8758	0.8516	−0.0025	0.0013	0.0009
	2.1.01	Green	0.8685	0.8837	0.8536	−0.0022	0.0025	0.0029
	2.1.01	Blue	0.8760	0.8812	0.8609	0.0018	−0.0011	−0.0025
	2.1.02	Red	0.8314	0.8498	0.7652	−0.0026	0.0012	0.0018
	2.1.02	Green	0.7552	0.7872	0.6740	−0.0013	−0.0024	0.0019
	2.1.02	Blue	0.7325	0.7413	0.6312	0.0029	0.0003	−0.0010
	2.1.05	Red	0.9570	0.9584	0.9390	0.0005	−0.0023	0.0024
	2.1.05	Green	0.9375	0.9356	0.9100	−0.0015	−0.0020	−0.0007
	2.1.05	Blue	0.9266	0.9193	0.8965	−0.0035	−0.0035	0.0033
1024 × 1024	2.2.01	Red	0.9256	0.9290	0.9049	0.0001	0.0037	0.0006
	2.2.01	Green	0.9142	0.9173	0.8994	−0.0008	0.0023	0.0015
	2.2.01	Blue	0.9031	0.9107	0.8868	0.0010	−0.0036	−0.0016
	2.2.08	Red	0.9181	0.9286	0.9015	0.0039	−0.0012	−0.0026
	2.2.08	Green	0.9141	0.9177	0.8922	0.0009	0.0011	0.0002
	2.2.08	Blue	0.9011	0.8950	0.8710	−0.0012	−0.0026	−0.0024
	2.2.11	Red	0.8111	0.8095	0.7710	0.0004	−0.0034	0.0021
	2.2.11	Green	0.7858	0.7826	0.7186	0.0026	−0.0007	0.0037
	2.2.11	Blue	0.7580	0.7733	0.7061	−0.0022	−0.0036	−0.0027

5.7. Information Entropy

Due to its capability to measure pixel randomness and distribution, information entropy is often employed for testing the security of suggested image encryption algorithms. Generally, if the information entropy value is larger, the ciphertext pixels possess a higher degree of randomness, and their distribution is also more uniform. In a mathematical sense, we can determine the value of information entropy through

$$\mathrm{IE}(\omega) = -\sum_{n=1}^{N} q(\omega_n) \log_2 q(\omega_n). \tag{27}$$

In Equation (27), N is the number of pixel values ω, and $q(\omega_n)$ represents the probability of ω_n. According to Equation (27), one can infer that for an image with an 8-bit pixel depth, the maximum value of its information entropy is 8. We encrypted six images using MIEA-FCSM and calculated the information entropy value for each channel of all images. From Table 9, we can observe that the entropy values of all channels in each original image are relatively small. In contrast, after encrypting, the entropy values of the resulting ciphertext images are all very close to the maximum value of 8. As listed in Table 10, we further compared MIEA-FCSM with several recent algorithms. Among all the algorithms, one can observe that the information entropy score achieved by MIEA-FCSM is closest to 8.

Overall, MIEA-FCSM demonstrates certain advantages when it comes to the randomness and distribution uniformity of ciphertext pixels.

Table 9. Information entropy scores of MIEA-FCSM.

Size	Name	Channel	Plaintext	Ciphertext
512×512	2.1.01	Red	7.5091	7.9994
	2.1.01	Green	7.3542	7.9994
	2.1.01	Blue	6.5966	7.9993
	2.1.02	Red	7.4061	7.9993
	2.1.02	Green	7.4188	7.9994
	2.1.02	Blue	6.5931	7.9994
	2.1.05	Red	7.5580	7.9993
	2.1.05	Green	7.4597	7.9994
	2.1.05	Blue	6.6665	7.9994
1024×1024	2.2.01	Red	7.7575	7.9998
	2.2.01	Green	7.3387	7.9998
	2.2.01	Blue	6.9561	7.9998
	2.2.08	Red	7.7229	7.9998
	2.2.08	Green	7.5289	7.9999
	2.2.08	Blue	6.8318	7.9998
	2.2.11	Red	6.6944	7.9999
	2.2.11	Green	6.3414	7.9998
	2.2.11	Blue	5.1766	7.9998

Table 10. Information entropy scores of nine algorithms.

Algorithm	Entropy Score
[20]	7.9984
[37]	7.9993
[38]	7.9976
[39]	7.9993
[35]	7.9993
[40]	7.9993
[28]	7.9992
[34]	7.9992
MIEA-FCSM	7.9994

To better measure the randomness of ciphertext images, Wu et al. [41] suggested a new performance indicator called local Shannon entropy (LSE). Mathematically, one can define LSE as follows:

$$\tilde{E}_{W,Q}(\mathbf{p}) = \sum_{r=1}^{W} \tilde{E}(\mathbf{p}_r)/W, \tag{28}$$

where $\mathbf{p}_1, \mathbf{p}_1, \ldots, \mathbf{p}_W$ denote W non-overlapping pixel blocks. Each pixel block contains Q pixels. These pixel blocks are randomly chosen from the image that needs to be measured. $\tilde{E}(\mathbf{p}_r)$ represents the information entropy of $\mathbf{p}_r$. With the parameter settings ($W = 30$, $Q = 1936$) recommended in [41], we conducted the LSE test on six ciphertext images generated by MIEA-FCSM. When the significance level is 0.05, the ideal value of LSE is 7.902469317. If the LSE score of a ciphertext image falls within the range of $(7.901901305, 7.903037329)$, it can be considered to have successfully passed the LSE test. Table 11 presents the relevant test results. It is evident that all color channels of the six ciphertext images have successfully passed the LSE test. This indicates that the ciphertext images generated by MIEA-FCSM indeed exhibit excellent randomness.

Table 11. LSE test results of MIEA-FCSM.

Size	Ciphertext	Channel	LSE Value	Result
512 × 512	2.1.01	Red	7.902608	Pass
	2.1.01	Green	7.901998	Pass
	2.1.01	Blue	7.902302	Pass
	2.1.02	Red	7.902425	Pass
	2.1.02	Green	7.902242	Pass
	2.1.02	Blue	7.902601	Pass
	2.1.05	Red	7.902229	Pass
	2.1.05	Green	7.901915	Pass
	2.1.05	Blue	7.902005	Pass
1024 × 1024	2.2.01	Red	7.901944	Pass
	2.2.01	Green	7.902186	Pass
	2.2.01	Blue	7.902156	Pass
	2.2.08	Red	7.902944	Pass
	2.2.08	Green	7.902651	Pass
	2.2.08	Blue	7.902183	Pass
	2.2.11	Red	7.902465	Pass
	2.2.11	Green	7.902302	Pass
	2.2.11	Blue	7.902973	Pass

5.8. Robustness Analysis

Given the network environments of various applications, encrypted images are prone to experiencing data loss or damage while being utilized. Hence, for an image encryption algorithm to be considered reliable, it should be robust enough to withstand certain levels of data loss or damage. By deliberately adding noise and removing pixels, we conducted a series of robustness evaluations on MIEA-FCSM. Specifically, we first added salt and pepper noise with varying intensities (0.04/0.08/0.12/0.16) and then decrypted these noise-contaminated images. Next, we removed 25% of the ciphertext pixels at different locations and decrypted the resulting images. From the first two rows of Figure 14, we can see that when the noise intensity is relatively low, the reconstructed image is slightly affected. Although the noise intensity is very high, MIEA-FCSM can also effectively restore the original image, successfully conveying most of its information. Similarly, from the last two rows of Figure 14, we can observe that when a large amount of data loss occurs on a single channel, there is little impact on the information transfer, and MIEA-FCSM also performs well in reconstructing the original image when a large amount of data loss occurs simultaneously in all channels. To evaluate the quality of decrypted images more objectively, we also introduced two performance indicators: peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM) [42]. For a decrypted image **A**, one can calculate its PSNR value as follows:

$$\mathrm{PSNR} = 10 \times \log_{10} \frac{H \times W \times 255^2}{\sum_{i=1}^{H} \sum_{j=1}^{W} (\mathbf{A}(i,j) - \mathbf{B}(i,j))^2}, \tag{29}$$

where $H \times W$ is the size of **A** and the corresponding plaintext image **B**. In general, a higher PSNR value implies a better quality of **A**. Similar to PSNR, SSIM is also frequently utilized to assess the quality of decrypted images [43]. Mathematically, the SSIM value of **A** can be defined as follows:

$$\mathrm{SSIM}(\mathbf{A}, \mathbf{B}) = \frac{(2\mu_a\mu_b + (0.01R)^2) + (2\sigma_{ab} + (0.03R)^2)}{(\mu_a^2 + \mu_b^2 + (0.01R)^2)(\sigma_a^2 + \sigma_b^2 + (0.03R)^2)}, \tag{30}$$

where μ_a is the mean of **A**, μ_b is the mean of **B**, σ_a is the variance of **A**, σ_b is the variance of **B**, and σ_{ab} is the covariance of **A** and **B**. The range of an SSIM value is $[0, 1]$. If the SSIM value of **A** is closer to 1, it indicates a higher level of similarity between the images. For ciphertext images contaminated by noise of different intensities, we calculated the PSNR

and SSIM values of the corresponding decrypted images. As can be observed from Table 12, when the noise intensity is lower, the quality of the decrypted image is better. The quality of the decrypted image decreases as the noise intensity increases. It is worth noting that even if the noise intensity is as high as 0.16, MIEA-FCSM still maintains a considerable level of image quality. Similarly, for some ciphertext images that suffered data losses, we also calculated the PSNR and SSIM values of the corresponding decrypted images. By observing Table 13, one can find that even if the data loss reaches $128 \times 128 \times 3$ pixels, the decrypted image still has good quality. To summarize, MIEA-FCSM is robust enough to effectively withstand significant data loss or corruption.

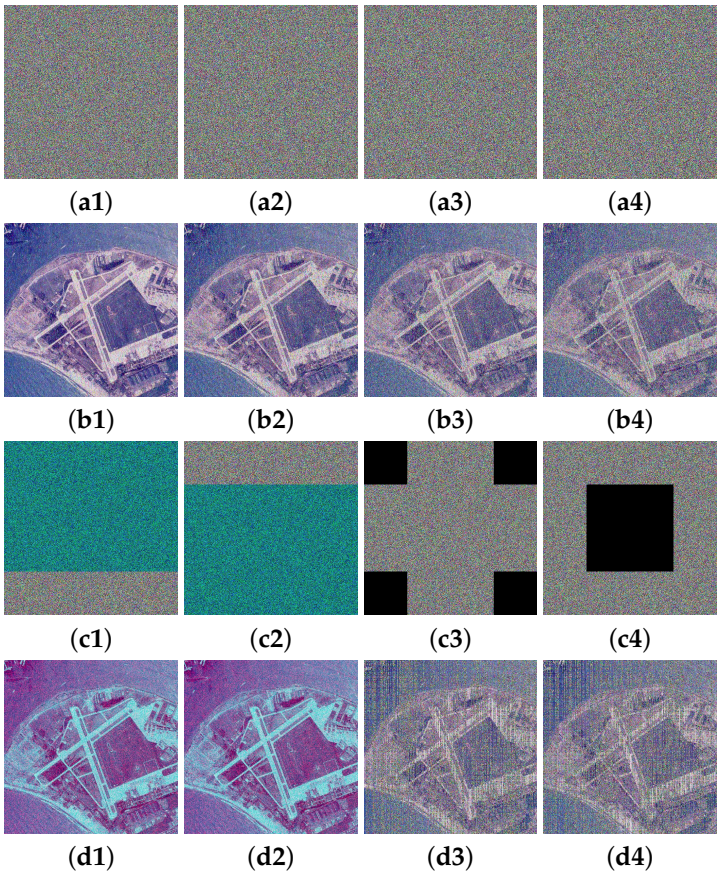


Figure 14. Robustness analysis for MIEA-FCSM: (**a1**–**a4**) are encrypted images contaminated by varying intensities of noise; (**b1**–**b4**) are decrypted images of (**a1**–**a4**); (**c1**–**c4**) are encrypted images with 25% missing pixels at different positions; (**d1**–**d4**) are decrypted images of (**c1**–**c4**).

Table 12. PSNR and SSIM values under different noise intensities.

Noise Intensity	Red Channel		Green Channel		Blue Channel	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
0.02	19.7562	0.7419	20.0190	0.7539	20.1633	0.6462
0.04	16.8391	0.6155	17.0988	0.6210	17.3881	0.4880
0.06	15.1894	0.5370	15.5601	0.5403	15.6278	0.4015
0.08	14.0505	0.4732	14.4138	0.4722	14.5452	0.3376
0.10	13.2139	0.4246	13.5473	0.4224	13.7136	0.2933
0.12	12.5858	0.3815	12.9287	0.3707	13.0371	0.2555
0.14	12.0616	0.3432	12.4292	0.3366	12.5424	0.2206
0.16	11.6114	0.3095	11.9224	0.3056	12.0879	0.2018

Table 13. PSNR and SSIM values under different data losses.

Data Loss	Red Channel		Green Channel		Blue Channel	
	PSNR	**SSIM**	**PSNR**	**SSIM**	**PSNR**	**SSIM**
$8 \times 8 \times 3$ pixels	37.8487	0.9922	38.9434	0.9937	39.6916	0.9917
$8 \times 16 \times 3$ pixels	35.3148	0.9876	35.9973	0.9889	36.5064	0.9840
$16 \times 16 \times 3$ pixels	32.4402	0.9763	33.1772	0.9798	33.4375	0.9700
$16 \times 32 \times 3$ pixels	29.6289	0.9618	30.2890	0.9664	30.4249	0.9508
$32 \times 32 \times 3$ pixels	27.0253	0.9358	27.6840	0.9418	28.0802	0.9215
$32 \times 64 \times 3$ pixels	24.0892	0.8974	24.6410	0.9054	24.9010	0.8691
$64 \times 64 \times 3$ pixels	21.4835	0.8434	21.8936	0.8489	22.2412	0.8024
$64 \times 128 \times 3$ pixels	18.7302	0.7676	19.1784	0.7745	19.5178	0.7187
$128 \times 128 \times 3$ pixels	16.3065	0.6941	16.7271	0.6901	16.9790	0.6479

5.9. Randomness Test

To more comprehensively assess the randomness performance of MIEA-FCSM, we also employed the NIST SP800-22 random test suite to conduct numerous experiments on the ciphertext images generated by MIEA-FCSM. The experiment results we obtained are presented in Table 14. Clearly, for all ciphertext images, the obtained p values are significantly greater than the threshold of 0.01. Thus, MIEA-FCSM indeed features outstanding randomness performance.

Table 14. NIST test outcomes for MIEA-FCSM.

Item Name	p Value (Ciphertext)						Result
	2.1.01	**2.1.02**	**2.1.05**	**2.1.06**	**2.2.01**	**2.2.08**	
Frequency (Monobit)	0.938984	0.455468	0.636900	0.271176	0.125786	0.959300	Random
Frequency (Block)	0.377102	0.695136	0.486114	0.919744	0.626776	0.986615	Random
Runs	0.848245	0.123654	0.657268	0.983728	0.224505	0.301428	Random
Longest Runs	0.554931	0.637710	0.364070	0.558170	0.326202	0.318947	Random
Matrix Rank	0.389840	0.116814	0.209644	0.247903	0.934658	0.998876	Random
Discrete Fourier Transform	0.200888	0.119452	0.853707	0.523441	0.763061	0.515849	Random
Non-Overlapping Template	0.674401	0.265663	0.126927	0.299549	0.222029	0.709035	Random
Overlapping Template	0.549943	0.995207	0.218516	0.353877	0.598382	0.301744	Random
Universal	0.905732	0.975949	0.486118	0.855998	0.293810	0.458157	Random
Linear Complexity	0.167901	0.223193	0.168184	0.564220	0.979454	0.534705	Random
Serial 1	0.887276	0.492728	0.496104	0.177181	0.925572	0.629930	Random
Serial 2	0.664458	0.394830	0.220287	0.827247	0.859346	0.503489	Random
Approximate Entropy	0.126032	0.443620	0.903944	0.459428	0.581134	0.873983	Random
Cumulative Sums (Forward)	0.649861	0.533670	0.476382	0.213114	0.149726	0.956529	Random
Cummulative Sums (Reverse)	0.721430	0.406060	0.892509	0.457783	0.876487	0.977343	Random
Random Excursions ($x = -1$)	0.070625	0.272277	0.210229	0.121351	0.550582	0.176783	Random
Random Excursions ($x = +1$)	0.407268	0.371559	0.122416	0.299785	0.438516	0.302316	Random
Random Excursions Variant ($x = -1$)	0.943201	0.331413	0.829856	0.105968	0.134844	0.083536	Random
Random Excursions Variant ($x = +1$)	0.412591	0.197920	0.067770	0.355611	0.252079	0.554221	Random

5.10. Efficiency Analysis

Given that current digital image applications possess salient characteristics such as large data volume and high throughput, it is essential for a suggested image encryption algorithm to exhibit extremely high encryption efficiency. Otherwise, the suggested algorithm cannot cater to the demands of practical applications.

Actually, several targeted designs have been introduced in our MIEA-FCSM to ensure the attainment of incredibly high encryption efficiency. Firstly, we developed and adopted 2D-SCPM, which possesses a simple structure yet exhibits excellent chaotic performance. Secondly, we optimized the strategy for generating and utilizing chaotic sequences. Thirdly,

we implemented the multi-channel fusion technique on the input image, resulting in a significant decrease in the computational amount of subsequent encryption steps. Lastly, all encryption operations are conducted at the vector level without compromising security. In comparison to pixel-level or bit-level encryption strategies, this can also considerably boost encryption efficiency.

To validate and demonstrate the superior efficiency of MIEA-FCSM, we conducted extensive experiments on six encryption algorithms using the same microcomputer. As can be seen from Table 15, the encryption efficiency of MIEA-FCSM is significantly higher than that of other recently reported algorithms. Even for inputs with a size of 1024×1024 and up to six channels (two 1024×1024 color images), MIEA-FCSM only took 0.3372 seconds on average to complete encryption, and the average throughput achieved is as high as 168.5608 Mbps. This indicates that MIEA-FCSM does have extremely high encryption efficiency, and can well meet the needs of practical applications. For instance, in medical applications, MIEA-FCSM can be employed to encrypt medical images, ensuring the safeguarding of patient privacy. Similarly, in high-data-throughput social applications, individuals can utilize MIEA-FCSM to encrypt different social-related images, thereby protecting trade secrets and personal privacy.

Note that we obtained the average encryption throughput by calculating the average of the encryption throughputs at four different input sizes. For each input size, we calculated the encryption throughput of MIEA-FCSM as follows:

$$\text{Throughput} = \frac{\text{The total number of bits in the input images (Mb)}}{\text{Encryption time (Seconds)}}. \tag{31}$$

For example, if the input size is $512 \times 512 \times 6$, we can use the above equation to obtain the encryption throughput, which is $\frac{(512 \times 512 \times 6 \times 8/2^{20})}{0.0676} \approx 177.5148$ Mbps.

Table 15. Average times (sec.) required and throughputs (Mbps) achieved by six algorithms.

Algorithm	Unit	Time (sec.) and Throughput (Mbps)				Average
		$512 \times 512 \times 3$	$512 \times 512 \times 6$	$1024 \times 1024 \times 3$	$1024 \times 1024 \times 6$	
[33]	sec.	1.1642	2.3685	5.3076	11.5125	–
	Mbps	5.1538	5.0665	4.5218	4.1694	4.7279
[37]	sec.	0.9373	1.9829	4.1355	8.3301	–
	Mbps	6.4014	6.0517	5.8034	5.7622	6.0047
[38]	sec.	1.8495	3.7407	7.4928	15.3363	–
	Mbps	3.2441	3.2080	3.2031	3.1298	3.1962
[39]	sec.	0.2691	0.5594	1.1221	2.3076	–
	Mbps	22.2965	21.4516	21.3885	20.8008	21.4843
[44]	sec.	1.6118	4.0054	10.3403	23.2284	–
	Mbps	3.7225	2.9960	2.3210	2.0664	2.7765
MIEA-FCSM	sec.	0.0308	0.0676	0.1504	0.3372	–
	Mbps	194.8052	177.5148	159.5745	142.3488	168.5608

6. Conclusions

To solve the problems of existing image encryption algorithms and better ensure the security of images, we first constructed a new fractional-order 3D Lorenz chaotic system and a robust hyper-chaotic map named 2D-SCPM. Then, we further developed a highly efficient multi-image encryption algorithm named MIEA-FCSM by exploiting the fractional-order 3D Lorenz chaotic system and 2D-SCPM. The introduction of the fractional-order 3D Lorenz chaotic system not only expands the key space but also strengthens the security of our proposed MIEA-FCSM. In comparison to other chaotic maps currently available, our proposed 2D-SCPM not only boasts a simpler structure but also exhibits superior chaotic performance. As revealed by our chaotic performance experiments, 2D-SCPM possesses a broad and continuous hyper-chaotic range and exceptionally rapid trajectory divergence

speeds. Moreover, 2D-SCPM also demonstrates highly uniform trajectory distributions and excellent trajectory unpredictability, randomness, and complexity. All of these factors make 2D-SCPM better suited for image encryption.

Our proposed MIEA-FCSM consists of four parts, which are the generation of chaotic sequences, multi-channel fusion, generation of plaintext-related parameters, and two rounds of plaintext-related substitution, diffusion, and scrambling. Considering the salient characteristics of images, all of these encryption steps are specifically devised to enhance encryption efficiency while guaranteeing a high level of security. Firstly, the improved chaotic sequence generation process ensures the reusability of chaotic sequences. Secondly, multi-channel fusion significantly reduces the computational workload of subsequent encryption operations to only one-sixth of the original amount. Finally, compared to existing bit-level, DNA-level, and pixel-level encryption operations, full vector-level plaintext-related substitution, diffusion, and scrambling can also significantly improve encryption efficiency. According to numerous experiments and analyses, MIEA-FCSM has excellent security, which is comparable to or even superior to the current leading image encryption algorithms. More importantly, MIEA-FCSM offers significant efficiency advantages. It can encrypt an image of size $1024 \times 1024 \times 3$ in just 0.1504 seconds on average, and its average encryption throughput is as high as 168.5608 Mbps. Therefore, in contrast to existing image encryption algorithms, MIEA-FCSM can better fulfill the requirements of practical applications.

In the future, we will proceed to introduce additional methods or technologies so as to further enhance the encryption efficiency of MIEA-FCSM. For instance, compressive sensing technology can be employed to pre-compress images to be encrypted.

Author Contributions: Conceptualization, W.F., Q.W., H.L. and S.Z.; methodology, W.F., Y.R. and K.Q.; software, W.F., Q.W., H.L. and K.Q.; validation, W.F., J.Z. and K.Q.; formal analysis, Y.R. and S.Z.; resources, Y.R., J.Z. and S.Z.; writing—original draft preparation, W.F., Q.W., H.L. and K.Q.; writing—review and editing, W.F., K.Q. and H.W.; supervision, W.F.; project administration, J.Z. and K.Q.; funding acquisition, W.F., K.Q. and H.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Scientific Research Projects of Hunan Provincial Department of Education (Grant No. 20B273), the Guangdong Basic and Applied Basic Research Foundation (Grant No. 2023A1515011717), the Special Projects for Key Fields of the Education Department of Guangdong Province (Grant No. 2023ZDZX1041), the Guiding Science and Technology Plan Project of Panzhihua City (Grant No. 2020ZD-S-40), the Innovation and Entrepreneurship Project for Chinese University Students (Grant Nos. 202211360021, S202211360072, 202311360019, and 2023cxcy162), and the Project for Zhongshan Science and Technology (Grant No. 2021B2062).

Data Availability Statement: Data will be made available on request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ren, H.; Niu, S.; Chen, J.; Li, M.; Yue, Z. A Visually Secure Image Encryption Based on the Fractional Lorenz System and Compressive Sensing. *Fractal Fract.* **2022**, *6*, 302. [CrossRef]
2. Chang, H.; Wang, E.; Liu, J. Research on Image Encryption Based on Fractional Seed Chaos Generator and Fractal Theory. *Fractal Fract.* **2023**, *7*, 221. [CrossRef]
3. Ye, G.; Wu, H.; Liu, M.; Shi, Y. Image encryption scheme based on blind signature and an improved Lorenz system. *Expert Syst. Appl.* **2022**, *205*, 117709. [CrossRef]
4. Özkaynak, F. Brief review on application of nonlinear dynamics in image encryption. *Nonlinear Dyn.* **2018**, *92*, 305–313. [CrossRef]
5. Wen, H.; Wu, J.; Ma, L.; Liu, Z.; Lin, Y.; Zhou, L.; Jian, H.; Lin, W.; Liu, L.; Zheng, T.; et al. Secure Optical Image Communication Using Double Random Transformation and Memristive Chaos. *IEEE Photonics J.* **2023**, *15*, 7900111. [CrossRef]
6. Si, Y.; Liu, H.; Chen, Y. Constructing keyed strong S-Box using an enhanced quadratic map. *Int. J. Bifurc. Chaos* **2021**, *31*, 2150146. [CrossRef]
7. Huang, P.; Li, D.; Wang, Y.; Zhao, H.; Deng, W. A Novel Color Image Encryption Algorithm Using Coupled Map Lattice with Polymorphic Mapping. *Electronics* **2022**, *11*, 3436. [CrossRef]

8. Wang, J.; Song, X.; El-Latif, A.A.A. Single-Objective Particle Swarm Optimization-Based Chaotic Image Encryption Scheme. *Electronics* **2022**, *11*, 2628. [CrossRef]
9. Xu, S.; Wang, X.; Ye, X. A new fractional-order chaos system of Hopfield neural network and its application in image encryption. *Chaos Solitons Fractals* **2022**, *157*, 111889. [CrossRef]
10. Wang, X.; Liu, C.; Jiang, D. A novel visually meaningful image encryption algorithm based on parallel compressive sensing and adaptive embedding. *Expert Syst. Appl.* **2022**, *209*, 118426. [CrossRef]
11. Zhu, L.; Jiang, D.; Ni, J.; Wang, X.; Rong, X.; Ahmad, M.; Chen, Y. A stable meaningful image encryption scheme using the newly-designed 2D discrete fractional-order chaotic map and Bayesian compressive sensing. *Signal Process.* **2022**, *195*, 108489. [CrossRef]
12. Alawida, M.; Teh, J.S.; Alshoura, W.H. A New Image Encryption Algorithm Based on DNA State Machine for UAV Data Encryption. *Drones* **2023**, *7*, 38. [CrossRef]
13. Zhang, X.; Yan, X. Adaptive Chaotic Image Encryption Algorithm Based on RNA and Pixel Depth. *Electronics* **2021**, *10*, 1770. [CrossRef]
14. Abd-El-Atty, B. Quaternion with quantum walks for designing a novel color image cryptosystem. *J. Inf. Secur. Appl.* **2022**, *71*, 103367. [CrossRef]
15. Janani, T.; Brindha, M. A secure medical image transmission scheme aided by quantum representation. *J. Inf. Secur. Appl.* **2021**, *59*, 102832. [CrossRef]
16. Qian, K.; Xiao, Y.; Wei, Y.; Liu, D.; Wang, Q.; Feng, W. A Robust Memristor-Enhanced Polynomial Hyper-Chaotic Map and Its Multi-Channel Image Encryption Application. *Micromachines* **2023**, *14*, 2090. [CrossRef] [PubMed]
17. Tuli, R.; Soneji, H.N.; Churi, P. PixAdapt: A novel approach to adaptive image encryption. *Chaos Solitons Fractals* **2022**, *164*, 112628. [CrossRef]
18. Yu, J.; Xie, W.; Zhong, Z.; Wang, H. Image encryption algorithm based on hyperchaotic system and a new DNA sequence operation. *Chaos Solitons Fractals* **2022**, *162*, 112456. [CrossRef]
19. Nan, S.; Feng, X.; Wu, Y.; Zhang, H. Remote sensing image compression and encryption based on block compressive sensing and 2D-LCCCM. *Nonlinear Dyn.* **2022**, *108*, 2705–2729. [CrossRef]
20. Liu, H.; Zhao, B.; Huang, L. A remote-sensing image encryption scheme using DNA bases probability and two-dimensional logistic map. *IEEE Access* **2019**, *7*, 65450–65459. [CrossRef]
21. Masood, F.; Boulila, W.; Ahmad, J.; Arshad.; Sankar, S.; Rubaiee, S.; Buchanan, W.J. A Novel Privacy Approach of Digital Aerial Images Based on Mersenne Twister Method with DNA Genetic Encoding and Chaos. *Remote Sens.* **2020**, *12*, 1893. [CrossRef]
22. Wen, H.; Lin, Y. Cryptanalysis of an image encryption algorithm using quantum chaotic map and DNA coding. *Expert Syst. Appl.* **2023**, *237*, 121514. [CrossRef]
23. Zhang, C.; Chen, J.; Chen, D. Cryptanalysis of an Image Encryption Algorithm Based on a 2D Hyperchaotic Map. *Entropy* **2022**, *24*, 1551. [CrossRef] [PubMed]
24. Chen, L.; Li, C.; Li, C. Security Measurement of a Medical Image Communication Scheme based on Chaos and DNA. *J. Vis. Commun. Image Represent.* **2022**, *83*, 103424. [CrossRef]
25. Diethelm, K.; Freed, A.D. The FracPECE subroutine for the numerical solution of differential equations of fractional order. *Forsch. Und Wiss. Rechn.* **1998**, *1999*, 57–71.
26. Danca, M.F.; Kuznetsov, N. Matlab code for Lyapunov exponents of fractional-order systems. *Int. J. Bifurc. Chaos* **2018**, *28*, 1850067. [CrossRef]
27. Sun, J. 2D-SCMCI Hyperchaotic Map for Image Encryption Algorithm. *IEEE Access* **2021**, *9*, 59313–59327. [CrossRef]
28. Hua, Z.; Zhu, Z.; Chen, Y.; Li, Y. Color image encryption using orthogonal Latin squares and a new 2D chaotic system. *Nonlinear Dyn.* **2021**, *104*, 4505–4522. [CrossRef]
29. Cao, W.; Cai, H.; Hua, Z. n-Dimensional Chaotic Map with application in secure communication. *Chaos Solitons Fractals* **2022**, *163*, 112519. [CrossRef]
30. Hua, Z.; Zhou, Y. Image encryption using 2D Logistic-adjusted-Sine map. *Inf. Sci.* **2016**, *339*, 237–253. [CrossRef]
31. Grassberger, P.; Procaccia, I. Estimation of the Kolmogorov entropy from a chaotic signal. *Phys. Rev. A* **1983**, *28*, 2591. [CrossRef]
32. Hua, Z.; Zhu, Z.; Yi, S.; Zhang, Z.; Huang, H. Cross-plane colour image encryption using a two-dimensional logistic tent modular map. *Inf. Sci.* **2021**, *546*, 1063–1083. [CrossRef]
33. Hua, Z.; Zhou, Y.; Huang, H. Cosine-transform-based chaotic system for image encryption. *Inf. Sci.* **2019**, *480*, 403–419. [CrossRef]
34. Qian, K.; Feng, W.; Qin, Z.; Zhang, J.; Luo, X.; Zhu, Z. A novel image encryption scheme based on memristive chaotic system and combining bidirectional bit-level cyclic shift and dynamic DNA-level diffusion. *Front. Phys.* **2022**, *10*, 718. [CrossRef]
35. Feng, W.; Zhao, X.; Zhang, J.; Qin, Z.; Zhang, J.; He, Y. Image Encryption Algorithm Based on Plane-Level Image Filtering and Discrete Logarithmic Transform. *Mathematics* **2022**, *10*, 2751. [CrossRef]
36. Zhu, H.; Zhao, Y.; Song, Y. 2D Logistic-Modulated-Sine-Coupling-Logistic Chaotic Map for Image Encryption. *IEEE Access* **2019**, *7*, 14081–14098. [CrossRef]
37. Li, H.; Li, T.; Feng, W.; Zhang, J.; Zhang, J.; Gan, L.; Li, C. A novel image encryption scheme based on non-adjacent parallelable permutation and dynamic DNA-level two-way diffusion. *J. Inf. Secur. Appl.* **2021**, *61*, 102844. [CrossRef]
38. Zefreh, E.Z. An image encryption scheme based on a hybrid model of DNA computing, chaotic systems and hash functions. *Multimed. Tools Appl.* **2020**, *79*, 24993–25022. [CrossRef]

39. Li, H.; Yu, S.; Feng, W.; Chen, Y.; Zhang, J.; Qin, Z.; Zhu, Z.; Wozniak, M. Exploiting Dynamic Vector-Level Operations and a 2D-Enhanced Logistic Modular Map for Efficient Chaotic Image Encryption. *Entropy* **2023**, *25*, 1147. [CrossRef]
40. Wang, X.; Liu, L.; Song, M. Remote sensing image and multi-type image joint encryption based on NCCS. *Nonlinear Dyn.* **2023**, *111*, 14537–14563. [CrossRef]
41. Wu, Y.; Zhou, Y.; Saveriades, G.; Agaian, S.; Noonan, J.P.; Natarajan, P. Local Shannon entropy measure with statistical tests for image randomness. *Inf. Sci.* **2013**, *222*, 323–342. [CrossRef]
42. Luo, Y.; Wang, F.; Xu, S.; Zhang, S.; Li, L.; Su, M.; Liu, J. CONCEAL: A robust dual-color image watermarking scheme. *Expert Syst. Appl.* **2022**, *208*, 118133. [CrossRef]
43. Wang, Z.; Bovik, A.; Sheikh, H.; Simoncelli, E. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [CrossRef] [PubMed]
44. Hua, Z.; Jin, F.; Xu, B.; Huang, H. 2D Logistic-Sine-coupling map for image encryption. *Signal Process.* **2018**, *149*, 148–161. [CrossRef]