symmetry

MDPI

Article

A Novel Fractional-Order Memristive Chaotic Circuit with Coexisting Double-Layout Four-Scroll Attractors and Its Application in Visually Meaningful Image Encryption

Yuebo Wu, Duansong Wang, Tan Zhang, Jinzhong Zhang and Jian Zhou *

School of Electrical and Opto-Electronic Engineering, West Anhui University, Lu'an 237012, China; wybyj1980@126.com (Y.W.); dswangsd@126.com (D.W.); 42000028@wxc.edu.cn (T.Z.); zhangjinzhongz@126.com (J.Z.)
* Correspondence: 42000015@wxc.edu.cn

Abstract: This paper proposes a fractional-order chaotic system using a tri-stable locally active memristor. The characteristics of the memristor, dynamic mechanism of oscillation, and behaviors of the proposed system were analyzed, and then a visually meaningful image encryption scheme was designed based on the chaotic system, DNA encoding, and integer wavelet transform (IWT). Firstly, the mathematical model of the memristor was designed, which was nonvolatile, locally active, and tri-stable. Secondly, the stability, dynamic mechanism of oscillation, bifurcation behaviors, and complexity of the fractional-order memristive chaotic system were investigated and the conditions of stability were obtained. Thirdly, the largest Lyapunov exponent, bifurcation diagram, and complexity of the novel system were calculated and the coexisting bifurcation, coexisting attractors, spectral entropy, and so on are shown. Finally, a visually meaningful image encryption scheme based on the proposed system was designed, and its security was assessed by statistical analysis and different attacks. Numerical simulation demonstrated the effectiveness of the theoretical analysis and high security of the proposed image encryption scheme.

Keywords: mechanism of oscillation; dynamic behavior; fractional-order; visually meaningful cipher image; locally-active memristor

Citation: Wu, Y.; Wang, D.; Zhang, T.; Zhang, J.; Zhou, J. A Novel Fractional-Order Memristive Chaotic Circuit with Coexisting Double-Layout Four-Scroll Attractors and Its Application in Visually Meaningful Image Encryption. *Symmetry* **2023**, *15*, 1398. https://doi.org/10.3390/sym15071398

Academic Editors: Shaobo He, Quan Xu and Chunlai Li

Received: 14 May 2023
Revised: 20 June 2023
Accepted: 22 June 2023
Published: 11 July 2023

1. Introduction

Prof. Chua predicted that there should be four fundamental elements of the theory of completeness in nature, namely resistance, capacitance, inductance, and memristance [1]. As a possible component, initial research on memristors was slow and very few researchers were interested because there was no physical device to support memristance. With the development of technology, HP Labs first reported a physical device of a nano-size memristor in 2008, which was of groundbreaking significance [2]. From then on, research on memristors has attracted a huge amount of attention in industry and academia. Their natural nonlinear and non-volatile characteristics give memristors great potential for data storage and computation [3,4], artificial neural networks [5–7], nonlinear circuits [8–10], secure communication [11], image encryption [12–14], and other fields.

Chua proposed that local activity is the origin of complexity in 2005 [15]. The first locally active memristor was designed and called the Chua corsage memristor [16]. In order to analyze the characteristics of the memristor, some new methods such as DC V-I Plot, Power-Off Plot (POP), and Dynamic Route Map (DRM) were proposed. These methods can analyze the locally active and non-volatile characteristics of the memristor. Research showed that the memristor has negative differential memristance or memductance in some regions, called locally active memristors [17]. According to the conservation of energy, a memristive system can produce and maintain continuous self-excited oscillation in the locally active regions of the memristor, resulting in rich dynamic behaviors.

Recently, research on locally active memristors and their applications are going through a blowout period. In 2013, a NbO_2 Mott locally active memristor was first realized by Williams et al. [18]. Later, in 2021, the Williams team used a resistor, capacitor, and NbO_2 Mott locally active memristor to build a neuron chaotic system and analyzed the dynamic behaviors of the neuron system. Their research indicated that the physical locally active memristor could greatly improve the efficiency and accuracy of the computation when it was applied in a Hopfield neural network [19]. In 2016, Mannan et al. demonstrated a nonautonomous system based on the Chua corsage memristor and analyzed the edge of chaos and the Hopf bifurcation of the system. The results showed that the system would oscillate near locally active operating points located on the memristor's DC V-I curve [16]. In 2017, Mannan et al. further studied the Chua corsage memristor and found that the memristor has two asymptotically stable equilibrium points and two distinct coexisting pinched hysteresis loops [20]. In 2018, Mannan et al. designed 4-lobe Chua corsage memristor and a 6-lobe Chua corsage memristor and analyzed their characteristics [21,22]. In 2020, Mannan et al. reviewed the nonlinear dynamic attributes, switching kinetics, bifurcation analysis, and physical realization of a family of Chua corsage memristors and analyzed the nonautonomous system based on the Chua corsage memristors, finding unique stable limit cycles from a supercritical Hopf bifurcation along with static attractors [23]. We find that all systems based on locally active memristors only produce periodic oscillations and no chaotic oscillations. In 2020, Dong et al. designed a bistable nonvolatile locally active memristor and added an inductor to the periodic oscillating circuit of the memristor. The coexisting chaotic attractors and various complex dynamic phenomena were reported in [24]. Tan et al. designed a simple locally active memristor and established HR neurons on the memristor. Complicated firing behaviors and coexisting position symmetry for different attractors were found in [25]. All of these studies showed that the locally active memristor has extensive application prospects and potential in chaotic oscillations.

As we all know, almost all systems are fractional-order systems in nature, and the integer-order system is a special case of fractional-order systems. Fractional calculus has the same historical memory function as a memristor; therefore, all memristive systems can be extended to fractional order. However, research on fractional-order chaotic systems based on locally active memristors is still in its infancy. Xie et al. proposed a fractional-order chaotic system based on a designed locally active memristor model, and dynamics analysis indicated that the system not only had diverse nonlinear dynamics, such as infinitely many discrete equilibrium points, multistability, and anti-monotonicity, but it also produced two new phenomena [26]. Ding et al. designed a fractional-order Hopfield neural network system based on a coupled locally active memristor, and the simulation results displayed that the fractional-order system not only had rich dynamic phenomena but also had some special transient transition process [27]. Yang et al. devised a fractional-order simplest chaotic system based on a bi-stable locally active memristor. Rich coexisting phenomena were found and some transient transition behaviors were analyzed [28]. It can be found that fractional-order systems based on locally-active memristors can generate richer dynamic behaviors and some new phenomena that have not been found in other chaotic systems.

Nowadays, digital images are ubiquitous, and yet the security of their transmission and storage causes anxiety. Research shows that there is higher security for image encryption in a chaotic system, which has better security than other methods because of the initial value sensitivity as well as the pseudo-random and aperiodic characteristics of chaotic systems. In recent years, many image encryption schemes based on chaotic systems have been proposed; however, these ciphertext images are noise-like or texture-like and easily distinguished from normal visually meaningful images. These visually meaningless noise-like images can only protect the security of image data but they fail to protect the security of image visual data. In pursuit of dual security of image data and visual data, the noise-like or texture-like ciphertext image should be converted into a visually meaningful image after the plain image is encrypted by existing encryption algorithms.

In order to achieve visual security, Bao et al. first proposed an image encryption scheme to encrypt plain images into visually meaningful ciphertext images [29]. This scheme added an embedding process on the basis of the existing image encryption algorithm. When the visually meaningful cipher image emerged in an insecure channel, it was difficult to catch the attention of attackers; therefore, the carrier image and the original image were more secure. Chai et al. proposed a novel visually secure image encryption scheme based on compressive sensing. Simulation results and performance analyses both demonstrated high sensitivity to the plain image and availability to known- and chosen-plaintext attacks [30]. Wang et al. proposed a visually meaningful scheme by employing the parallel compressive sensing method and embedding technique. Simulation analysis demonstrated that the cipher image exhibited excellent security and the reconstructed image had higher quality [31]. Ping et al. designed a visually meaningful image encryption scheme using the compressive sensing and partial block pairing substitution technique. Experimental results demonstrated that the scheme had higher quality and higher security [32]. It can be observed that the style of image decomposition is discrete wavelet transform (DWT); however, the DWT is irreversible, and the quality of the final reconstructed image will be poor during decryption. To solve this problem, the integer wavelet transform (IWT) is used to decompose and reconstruct the carrier image, which is fully reversible. Owing to the fact that IWT and inverse IWT are completely reversible, the obtained image is exactly the same as the original carrier image after the carrier image undergoes IWT and inverse IWT transformation. Based on this, we used this method to complete the embedding and extracting processes of the secret image.

Based on the above discussions, this study proposed a new continuous nonlinear tri-bistable locally active memristor model, and we studied its nonlinear characteristics. Then, we analyzed the features of the locally active memristor, including time-domain waveforms, three coexisting pinched hysteresis loops, Power-Off Plot, and DC V-I Curve. Next, we designed a fractional-order simple circuit system using our memristor, a linear passive inductor, and a linear passive capacitor in series. We observed that the system could produce oscillation under some conditions and had abundant dynamic behaviors. Finally, we used the fractional-order memristive chaotic system to design an image encryption scheme and analyzed its security. The contributions of this paper are listed as follows: (1) A tri-stable locally active memristor was designed and analyzed. (2) A fractional-order chaotic system was built based on the proposed memristor, and we discovered its rich coexisting dynamic behaviors. In order to apply the chaotic system, we analyzed the complexity when some parameters were changed. (3) A visually secure image encryption scheme was proposed based on the fractional-order memristive chaotic system and integer wavelet transform (IWT) embedding. The analysis results pointed out that our algorithm not had only higher security, but it also enabled dual protection of data security and visual security.

The remainder of this paper is organized as follows. The tri-stable locally-active memristor model is built in Section 2. Section 3 shows the dynamic characteristics of the designed memristor. The new fractional-order chaotic system based on the tri-stable locally-active memristor is presented, and the dynamic behavior of the chaotic system is analyzed in Section 4. Section 5 designs a visually meaningful image encryption scheme to use the chaotic system. Finally, Section 6 is the conclusion.

2. New Locally Active Memristor

Based on Chua's definition and classification of memristors, this paper designed a new tri-stable locally-active memristor model, which can be described by the voltage–current relationship and the state equation as follows:

$$\frac{dx}{dt} = k_1(ax + bx|x| + cx^3 + dxv^2 + ev) \tag{1}$$

$$i = k_2(x^2 - m)v \tag{2}$$

where v and i are the input and output of the memristor, respectively. x is the state variable of the memristor. k_1, k_2, e, and m are parameters.

According to Equations (1) and (2), when the parameters are set as $k_1 = 1$, $k_2 = 1$, $a = -4.7$, $b = 10$, $c = -4.7$, $d = 0.1$, $m = 10$, and $e = 10$, the color POP of Equation (1) with arrow heads is shown in Figure 1. It can be observed that there are five intersections with $dx/dt = 0$, namely E_1, E_2, E_3, E_4, and E_5. By adjusting the dynamic route of the five points, we can judge that the equilibrium points E_1, E_3, and E_5 are asymptotically stable, whereas the equilibrium points E_2 and E_4 are unstable. The green lines represent the negative slope ranges, while the blue lines represent the positive slope ranges. The arrow above the x = 0 axis faces right, and the curve arrow below the x = 0 axis faces left.

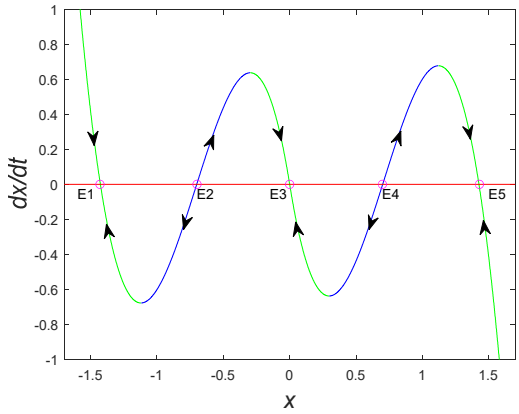


Figure 1. POP of Equation (1).

2.1. Pinched Hysteresis Loops

When the memristor is driven by a periodic signal source with amplitude A and frequency ω, the $v - i$ curves of the memristor are pinched hysteresis loops passing through the origin. While the parameters of the memristor and the periodic signal source take different values, and the initial values of Equation (1) also take different values, the memristor will exhibit a variety of characteristics. The parameters of Equations (1) and (2) are set as $k_1 = 1$, $k_2 = 1$, $a = -4.7, b = 10$, $c = -4.7$, $d = 0.1$, $m = 10$, and $e = 10$.

Let $A = 4, \omega = 10\text{rad/s}$, and the parameter d is changed. When $d \in [-0.048, 0.17]$, the dynamic trajectory displays three coexisting pinched hysteresis loops, as shown in Figure 2b, and the corresponding time-domain waveforms are shown in Figure 2a.

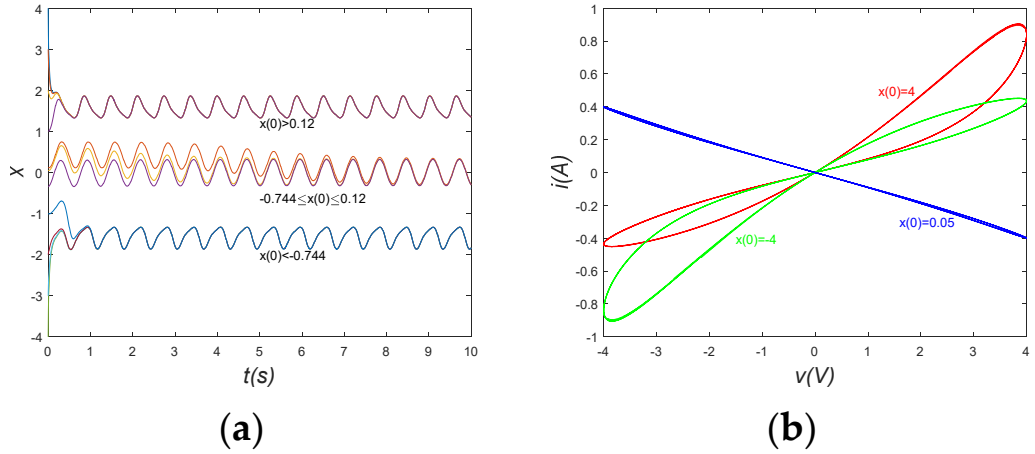


Figure 2. The dynamic trajectory of the memristor when $d = 0.1$: (**a**) time-domain waveforms; (**b**) three coexisting pinched hysteresis loops.

It can be observed from Figure 2 that if the initial value $x(0) > 0.12$, the pinched hysteresis loop is the red curve of Figure 2b; if the initial value $-0.744 \leq x(0) \leq 0.12$, the pinched hysteresis loop is the blue curve of Figure 2b; if the initial value $x(0) < -0.744$, the pinched hysteresis loop is the green curve of Figure 2b. The red and green curves are symmetrical hysteresis loops about the origin. When $d > 0.17$, the dynamic trajectories display double coexisting pinched hysteresis loops, as shown in Figure 3b, and the corresponding time-domain waveforms are shown in Figure 3a. It can be observed from Figure 3 that if the initial value $x(0) > -0.32$, the pinched hysteresis loop is the red curve of Figure 3b; if the initial value $x(0) < -0.32$, the pinched hysteresis loop is the green curve of Figure 3b. The red and blue curves are symmetrical hysteresis loops about the origin, and they have two pinched points. When $d < -0.048$, the dynamic trajectory displays only one pinched hysteresis loop, as shown in Figure 4b, and the corresponding time-domain waveforms are shown in Figure 4a. It can be observed that the dynamic trajectories have three stable states at the beginning, and as time goes on, the final curves are independent of the initial values and eventually converge to the same curve.

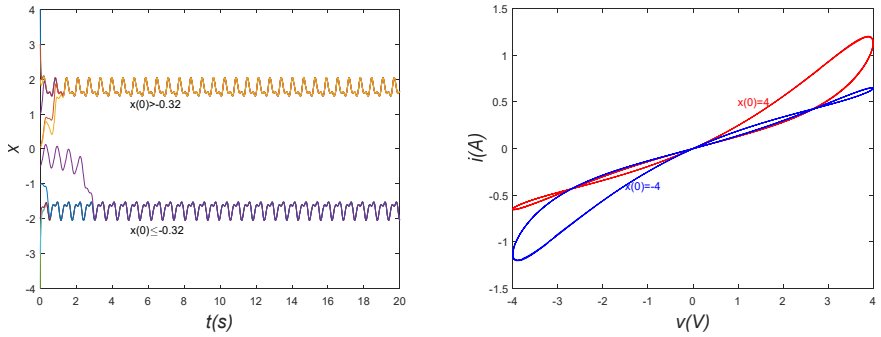


Figure 3. The dynamic trajectory of the memristor when $d = 0.15$: (**a**) time-domain waveforms; (**b**) double coexisting pinched hysteresis loops.

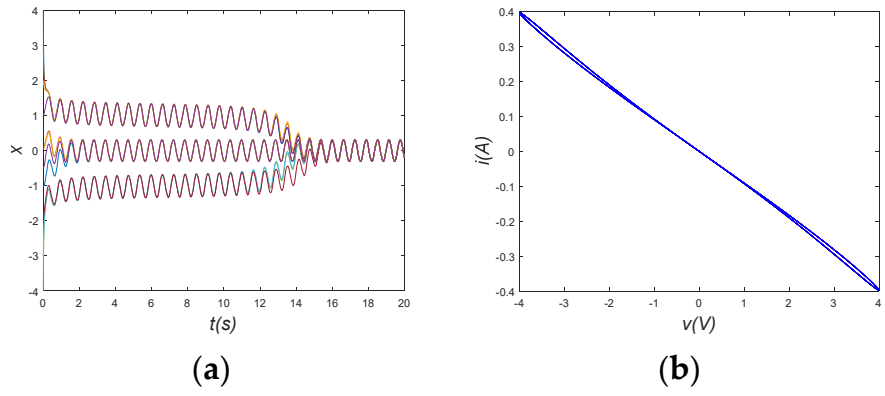


Figure 4. The dynamic trajectory of the memristor when $d = -0.05$: (**a**) time-domain waveforms; (**b**) double coexisting pinched hysteresis loops.

Let the amplitude $A = 4$ V, $\omega = 10$ rad/s, and the parameter e is changed. When $e \in [-1, 0.958]$, the dynamic trajectory displays three coexisting pinched hysteresis loops, as shown in Figure 5b, and the corresponding time-domain waveforms are shown in Figure 5a. It can be observed from Figure 5 that if the initial value $x(0) > 0.077$, the pinched hysteresis loop is the red curve of Figure 5b; if the initial value $-0.745 \leq x(0) \leq 0.077$, the pinched hysteresis loop is the blue curve of Figure 5b; if the initial value $x(0) < -0.745$, the pinched hysteresis loop is the green curve of Figure 5b. The red and green curves are symmetrical hysteresis loops about the origin. When $e > 0.958$ and $-3.6 \leq e < -1$,

the dynamic trajectories display double coexisting pinched hysteresis loops, as shown in Figure 5d, and the corresponding time-domain waveforms are shown in Figure 5c. It can be observed that if the initial value $x(0) > 0.6$, the pinched hysteresis loop is the red curve of Figure 5b; if the initial value $x(0) \leq 0.6$, the pinched hysteresis loop is the green curve of Figure 5b. The red and blue curves are symmetrical hysteresis loops about the origin, and they have two pinched points. When $e > -3.6$, the dynamic trajectory displays only one pinched hysteresis loop, as shown in Figure 5f, and the corresponding time-domain waveforms are shown in Figure 5e. It can be observed that the pinched hysteresis loop has three pinched points and is symmetric about the origin.

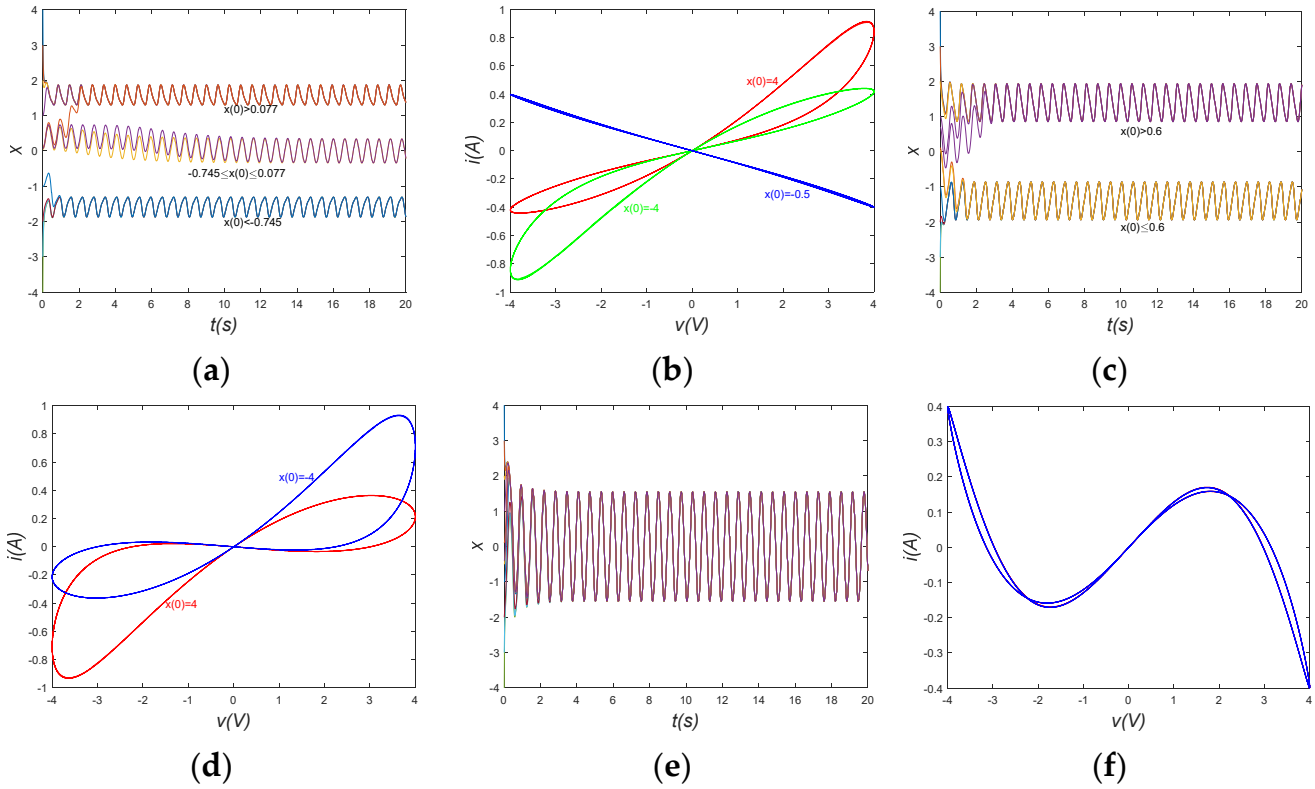
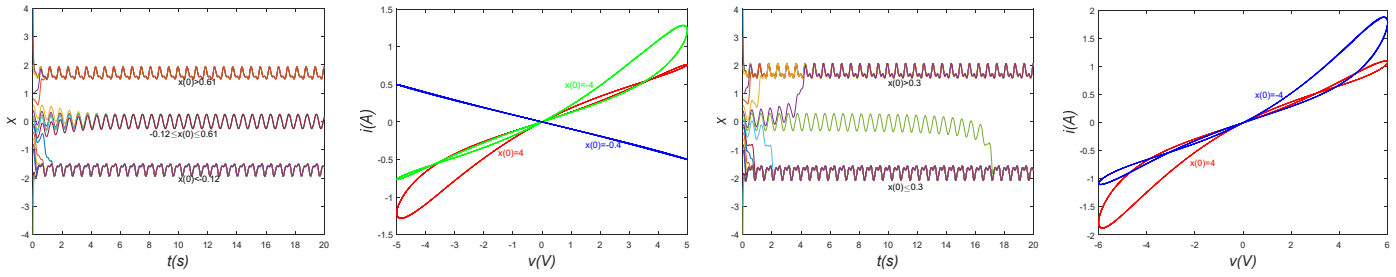


Figure 5. The dynamic trajectory of the memristor: (**a**) time-domain waveforms when $e = 0.5$; (**b**) double coexisting pinched hysteresis loops when $e = 0.5$; (**c**) when $e = 1.5$; (**d**) when $e = 1.5$; (**e**) when $e = -3.9$; (**f**) when $e = -3.9$.

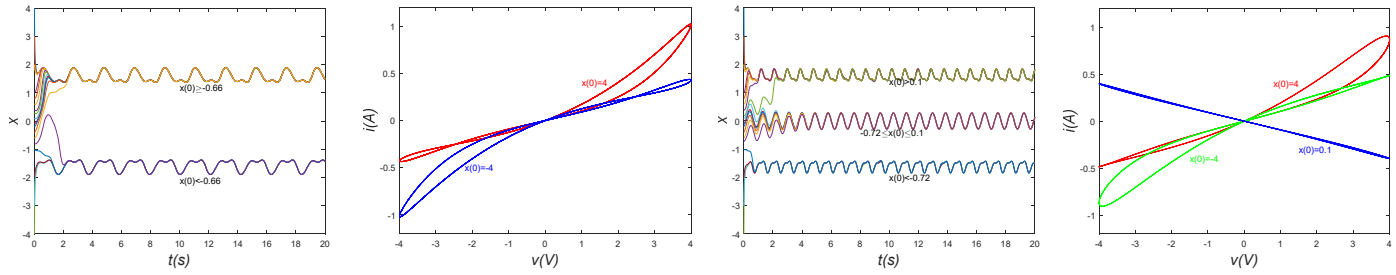
Let the amplitude $e = 0.5$, $d = 0.1$, $\omega = 10$ rad/s, and the parameter A is changed. When $A \leq 5.7$, the dynamic trajectory displays three coexisting pinched hysteresis loops, as shown in Figure 6b, and the corresponding time-domain waveforms are shown in Figure 6a. It can be seen that if the initial value $x(0) > 0.61$, the pinched hysteresis loop is the red curve of Figure 6b; if the initial value $-0.12 \leq x(0) \leq 0.61$, the pinched hysteresis loop is the blue curve of Figure 6b; if the initial value $x(0) < -0.12$, the pinched hysteresis loop is the green curve of Figure 6b. The red and green curves are symmetrical hysteresis loops about the origin. When $A > 5.7$, the dynamic trajectories display double coexisting pinched hysteresis loops, as shown in Figure 6d, and the corresponding time-domain waveforms are shown in Figure 6c. It can be observed that if the initial value $x(0) > 0.3$, the pinched hysteresis loop is the red curve of Figure 6d; if the initial value $x(0) \leq 0.3$, the pinched hysteresis loop is the green curve of Figure 6d. The red and blue curves are symmetrical hysteresis loops about the origin and they have two pinched points.



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

Figure 6. The dynamic trajectory of the memristor: (**a**) time-domain waveforms when A = 5; (**b**) double coexisting pinched hysteresis loops when A = 5; (**c**) when A = 6; (**d**) when A = 6.

Let the amplitude e = 0.5, d = 0.1, A = 4 V, and the parameter ω is changed. When $\omega \leq 5$, the dynamic trajectory displays double coexisting pinched hysteresis loops, as shown in Figure 7b, and the corresponding time-domain waveforms are shown in Figure 7a. It can be observed that if the initial value $x(0) \geq -0.66$, the pinched hysteresis loop is the red curve of Figure 7b; if the initial value $x(0) < -0.66$, the pinched hysteresis loop is the blue curve of Figure 7b. The red and green curves are symmetrical hysteresis loops about the origin. When $\omega > 5$, the dynamic trajectories display three coexisting pinched hysteresis loops, as shown in Figure 7d, and the corresponding time-domain waveforms are shown in Figure 7c. It can be observed that if the initial value $x(0) > 0.1$, the pinched hysteresis loop is the red curve of Figure 7d; if the initial value $-0.72 \leq x(0) \leq 0.1$, the pinched hysteresis loop is the blue curve of Figure 7d; if the initial value $x(0) < -0.72$, the pinched hysteresis loop is the green curve of Figure 7d. The red and blue curves are symmetrical hysteresis loops about the origin. It can be seen from the above analysis that the memristor presented different stable states with changes in parameters and input signals.



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

Figure 7. The dynamic trajectory of the memristor: (**a**) time-domain waveforms when $\omega = 3$ rad/s; (**b**) double coexisting pinched hysteresis loops when $\omega = 3$ rad/s; (**c**) when $\omega = 7$ rad/s; (**d**) when $\omega = 7$ rad/s.

2.2. Local Activity

The DC V-I plot reflects Ohm's law for the memristor, which describes the DC characteristics of the memristor and can analyze the intrinsic locally active features of the memristor. Let $x = X, dx/dt|_{x=X} = 0$ in Equation (1), we can obtain the relationship between voltage *V* and state variable *X*:

$$aX + bX|X| + cX^3 + dXV^2 + eV = 0 \tag{3}$$

Then, setting a = −4.7, b = 10, c = −4.7, d = 0.1, and e = 0.5. Based on Equation (3), we can obtain a function between the state variable X and the DC voltage V:

$$V_1 = \frac{-5 + \sqrt{25 + 4X(47X - 100X|X| + 47X^3)}}{2X} \tag{4}$$

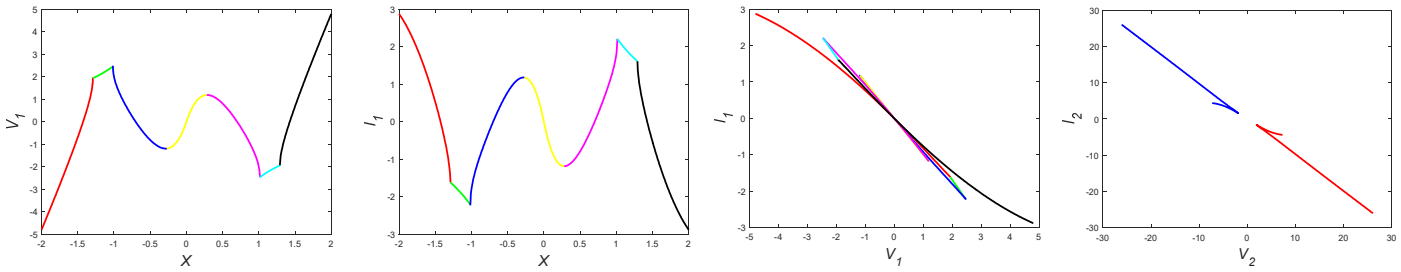
$$V_2 = \frac{-5 - \sqrt{25 + 4X(47X - 100X|X| + 47X^3)}}{2X} \tag{5}$$

Substituting Equations (4) and (5) into Equation (2), the DC current I can be calculated, respectively:

$$I_1 = (X^2 - m)V_1 \tag{6}$$

$$I_2 = (X^2 - m)V_2 \tag{7}$$

Let $m = 10$, based on Equations (4) and (6), when $-2 < X < 2$, we drew points (X,V_1), (X,I_1), and (V_1,I_1) in the $X - V_1$, $X-I_1$ and $V_1 - I_1$ planes, respectively, and obtained the DC $V_1 - I_1$ plots shown in Figure 8a–c. Based on Equations (5) and (7), we drew points (V_2,I_2) in the $V_2 - I_2$ plane and obtained the DC $V_2 - I_2$ plot shown in Figure 8d in the same way. It was observed that the slopes of all of the curves had negative values; therefore, we determined that the memristor was a locally active memristor.



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

Figure 8. The DC plots of the memristor. (**a**) $X-V_1$; (**b**) $X-I_1$; (**c**) $V_1 - I_1$; (**d**) $V_2 - I_2$.

Comparison with the existing memristor model is shown in Table 1. It was found that the model proposed in this paper had more parameters and equilibrium points. In order to verify the correctness of the analysis, we designed a circuit on the proposed memristor, as shown in Figure 9. When we placed a sinusoidal excitation at both ends of the memristor, pinched hysteresis loops were observed, as shown in Figure 10. It could be concluded that the analysis was correct by comparison.

Table 1. Comparison with the memristor model.

Memristor Models	Number of Parameters	Number of Equilibrium Points
Ref. [17]	1	3
Ref. [25]	1	3
Ref. [33]	2	infinite
Ref. [34]	1	5
Ours	7	5

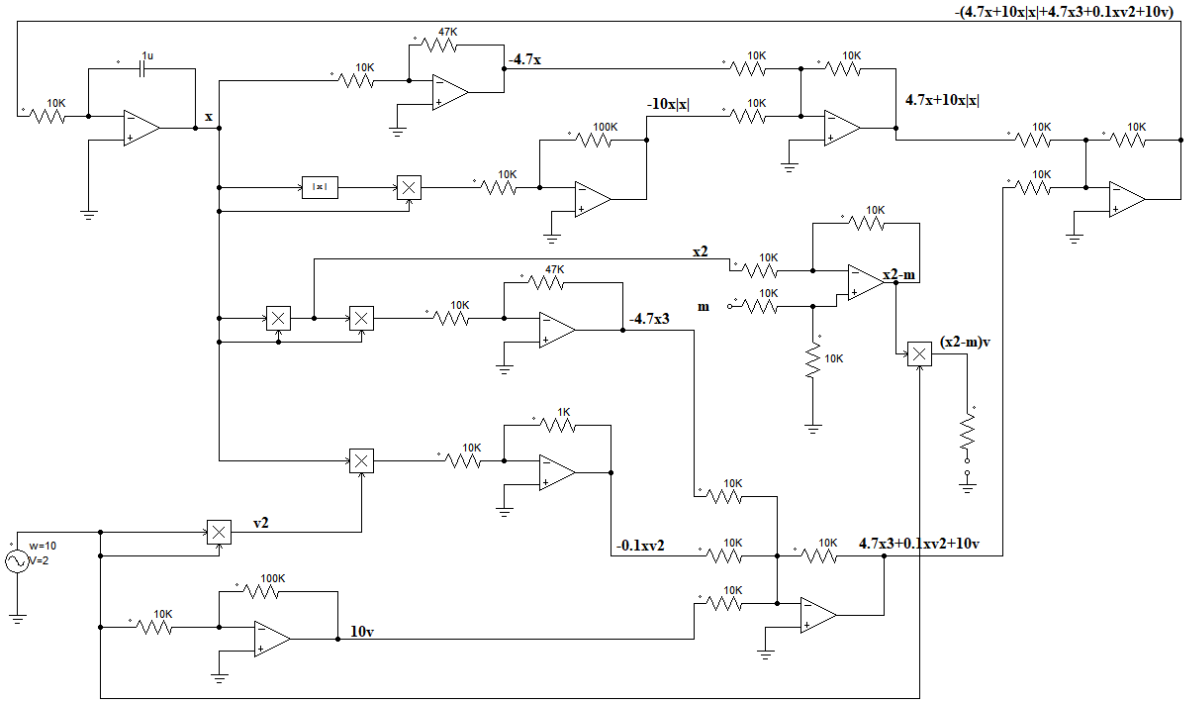
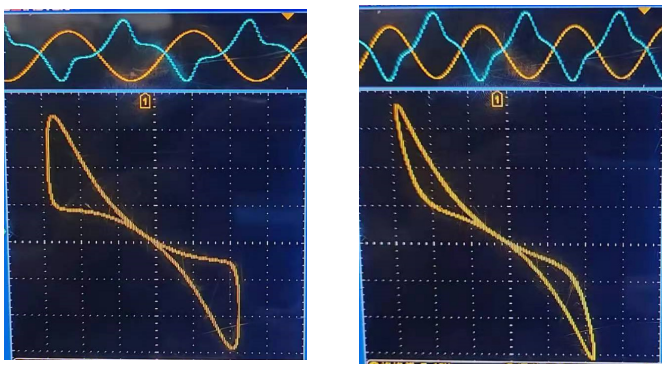


Figure 9. The circuit diagram of the proposed memristor.



(**a**) (**b**)

Figure 10. The pinched hysteresis loops of the memristor. (**a**) $\omega = 10$; (**b**) $\omega = 14$.

3. Fractional-Order Memristive Chaotic System

A fractional-order capacitor, fractional-order inductance, and our proposed fractional-order memristor were connected in series to construct a third-order fractional-order circuit, as shown in Figure 11. The new 3D fractional-order memristive system may generate chaotic oscillation because it contained a locally active memristor.

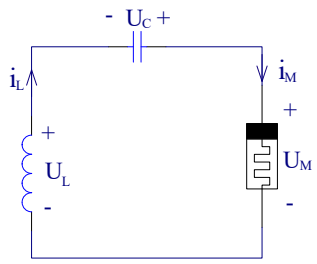


Figure 11. The memristive chaotic circuit.

According to Kirchhoff's laws, we could obtain the fractional-order differential equations as follows:

$$\begin{cases} \frac{d^q x}{dt^q} = k\left(ax + bx|x| + cx^3 + exy^2 + fy\right) \\ \frac{d^q y}{dt^q} = -\frac{\left(x^2 - m\right)y + z}{L} \\ \frac{d^q z}{dt^q} = \frac{y}{C} \end{cases} \tag{8}$$

where a, b, c, e, and f are the parameters of the designed system, L represents the inductance of an inductor, C represents the capacitance of the capacitor, q represents the order of the system, x is the internal state variable of the memristor, y is the current value through the inductor L, and z is the voltage value of the capacitor C.

3.1. Fractional Calculus

Definition 1 ([35])**.** *The Caputo fractional derivation definition of fractional-order α is:*

$$D_t^\alpha f(t) = \frac{1}{\Gamma(m-\alpha)} \int_0^t (t-\tau)^{m-\alpha-1} f^{(m)}(\tau) d\tau,\ m-1 < \alpha < m \tag{9}$$

where $\Gamma(\cdot)$ *is the Gamma function, and* $\alpha \in R, m \in Z^+$*. When* $0 < \alpha < 1$, $D_t^\alpha = \frac{1}{\Gamma(1-\alpha)} \int_0^t \frac{f'(\tau)}{(t-\tau)^\alpha} d\tau, 0 < \alpha < 1$.

In this paper, we used the Adomian Decomposition Method (ADM) to solve the proposed fractional-order differential equations.

3.2. The Stability of the System

Let the right side of Equation (8) be equal to zero:

$$\begin{cases} \frac{d^q x}{dt^q} = 0 \\ \frac{d^q y}{dt^q} = 0 \\ \frac{d^q z}{dt^q} = 0 \end{cases} \tag{10}$$

and the equilibrium points $\left(x_e, y_e, z_e\right)$ of Equation (8) can be obtained as $e_1(-1.4268, 0, 0)$, $e_2(-0.7009, 0, 0)$, $e_3(0, 0, 0)$, $e_4(0.7009, 0, 0)$, and $e_5(1.4268, 0, 0)$. The Jacobian matrix of system (8) at $\left(x_e, y_e, z_e\right) = \left(x^*, 0, 0\right)$ is:

$$J_E = \begin{pmatrix} f(x^*) & 0 & 0 \\ 0 & \frac{m - x^{*2}}{L} & -\frac{1}{L} \\ 0 & \frac{1}{C} & 0 \end{pmatrix} \tag{11}$$

where $f\left(x^*\right) = k\left(a + b\left|x^*\right| + \frac{bx^{*2}}{\left|x^*\right|} + 3cx^{*2}\right)$. The corresponding characteristic equation is:

$$\left[\lambda - f(x^*)\right]\left[\lambda^2 - \frac{m - x^{*2}}{L}\lambda + \frac{1}{LC}\right] = 0 \tag{12}$$

Based on Equation (12), we could calculate its three eigenvalues as follows:

$$\lambda_1 = f(x^*) \tag{13}$$

$$\lambda_{2,3} = \frac{\left(m - x^{*2}\right)/L \pm \sqrt{\left(\left(m - x^{*2}\right)/L\right)^2 - 4/LC}}{2} \tag{14}$$

In order to analyze the stability of the system (8), we discuss the characteristic roots at the equilibrium points, respectively, and we obtained the conditions of stability of the system.

Obviously, when $L > 0, C > 0$, then $L/C > 0$. When the equilibrium point was set as (x*,0,0), then $\lambda_1 = f(x^*)$, and $\lambda_{2,3} = m \pm \sqrt{m^2 - 4L/C}/2L$. If $f(x^*) < 0, m < 0$, then all of the characteristic roots had negative real parts, namely $Re(\lambda_1) < 0, Re(\lambda_{2,3}) < 0$. Clearly, the system (8) was asymptotically stable at the equilibrium point. If $f(x^*) > 0$ or $m > 0$, then at least one of the eigenvalues had a positive real part, namely $Re(\lambda_1) > 0$ or $Re(\lambda_{2,3}) > 0$. Clearly, the system (8) was unstable and chaotic oscillation may occur.

3.3. Mechanism of Chaotic Oscillation

We set the parameters as $a = -4.7, b = 10, c = -4.7, k = 20{,}000, m = 160$, $e = 10, f = 10, L = 1e-7, C = 1e-7$, and $q = 0.9$, and we set the initial value as $(x, y, z) = (12.5, -2, 0.1)$. The phase diagram of chaotic oscillation of system (8) is shown in Figure 9d. In order to explore the mechanism of chaotic oscillation of system (8), we took several values of discretized voltage $V = V_1, V_2, \cdots, V_k$, and plotted dynamic routes in the dx/dt vs.x phase plane. Part of the routes of the dynamic switching process for values of discretized voltage V are shown in Figure 12a, where the step length was 1us. In Figure 12a, different colors indicate different dynamic routes obtained by the different voltages. The red points and short blue lines indicate the dynamic evolution when the state variable x was changed. In order to observe the results more clearly, we enlarged the local area in the black box of Figure 12a, as shown in Figure 12b. The corresponding phase diagram is shown in Figure 12c. When we set the initial values as $(x, y, z) = (12.5, -0.53, 0.3)$ and $(x, y, z) = (-12.5, -0.53, 0.3)$, double coexisting routes of the dynamic switching process could be observed in Figure 12d. The phase diagram corresponding to Figure 12d is shown in Figure 12e. When we took smaller step lengths and more points, a smooth dynamic trajectory and chaotic attractor were generated, which possessed a double layout and four scrolls, as shown in Figure 12f.

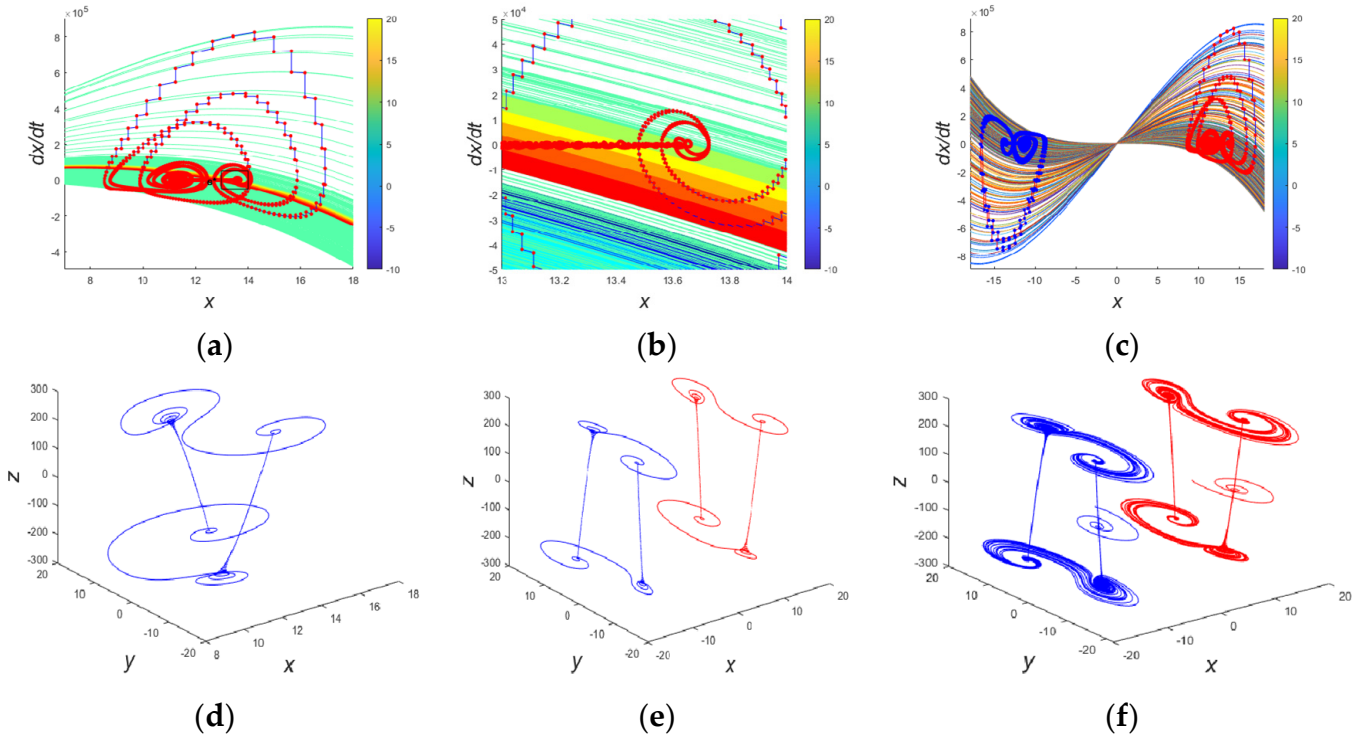


Figure 12. Mechanism analysis of chaotic oscillation: (**a**) the dynamic process with 1 us step length; (**b**) an enlarged view of the dynamic process in the black box; (**c**) the part of the phase diagram corresponding to (**a**); (**d**) the coexisting dynamic process with 1us step length; (**e**) the part of the coexisting phase diagram corresponding to (**d**); (**f**) co-existing attractors.

3.4. Dynamic Behaviors

In order to clearly show the dynamic behaviors of the system (8) when some relevant parameters were changed in a certain range, a phase diagram, bifurcation diagram, and the largest Lyapunov exponents [36] should first be drawn. In this paper, we used the QR algorithm to calculate the Lyapunov exponents.

3.4.1. Dynamic Depending on C

There were nine parameters in the system (8); therefore, we could only select some different types of parameters to study the impact on the dynamic behaviors of the system. Firstly, we set initial values as $(x_0, y_0, z_0) = (12.5, -0.5, 0.3)$ and $(x_0, y_0, z_0) = (-12.5, -0.5, 0.3)$, and some parameters as $a = -4.7, b = 10, c = -4.7, k = 20{,}000, m = 160$, $e = 10, f = 10, L = 1e-7$, and $q = 0.9$. When the parameter C was changed, the resulting bifurcation diagram and the largest Lyapunov exponent are shown in Figure 13.

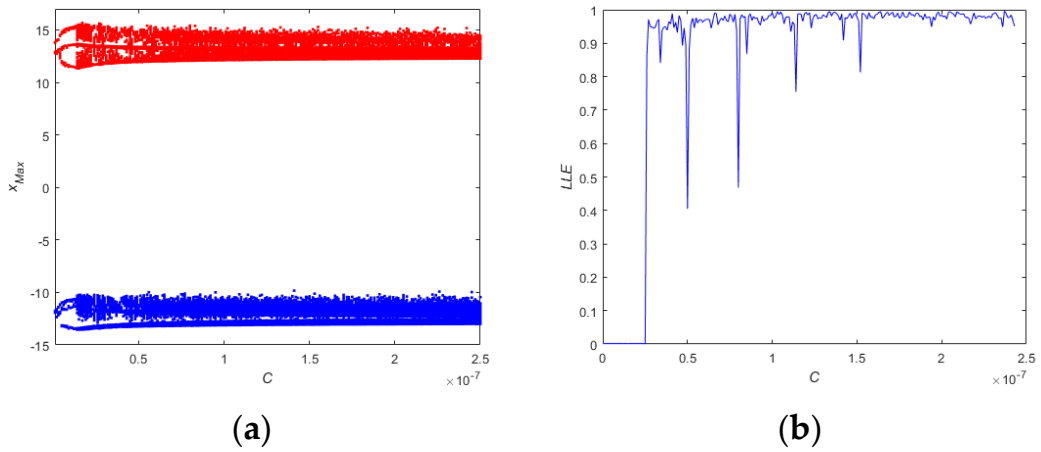


Figure 13. Dynamic behavior depending on C: (**a**) coexisting bifurcation diagram; (**b**) largest Lyapunov exponent.

It can be seen that the system (8) was in a periodic state when $C < 13$ nF, while it was in a chaotic state when $C > 13$ nF. The LLE diagram on the right and the bifurcation diagram on the left strictly corresponded. They were symmetrical and separated when the parameter C was changed from 1 to 243 nF. To further illustrate the status of the system, we selected different values of the parameter C and the same initial conditions, and we obtained the coexisting attractors of the system in Figure 11. In Figure 11, the three-dimensional coexisting attractors and their projection in the $x-y$, $x-z$, and $y-z$ coordinate axes can be clearly observed. Figure 14a,b shows the phase diagrams of different periodic states. Figure 14c,d shows the phase diagrams of the chaotic state, which possessed a double layout and four scrolls, but their oscillation amplitudes and frequencies were different. As the parameter C was increased, the system (8) shifted from periodic to chaotic.

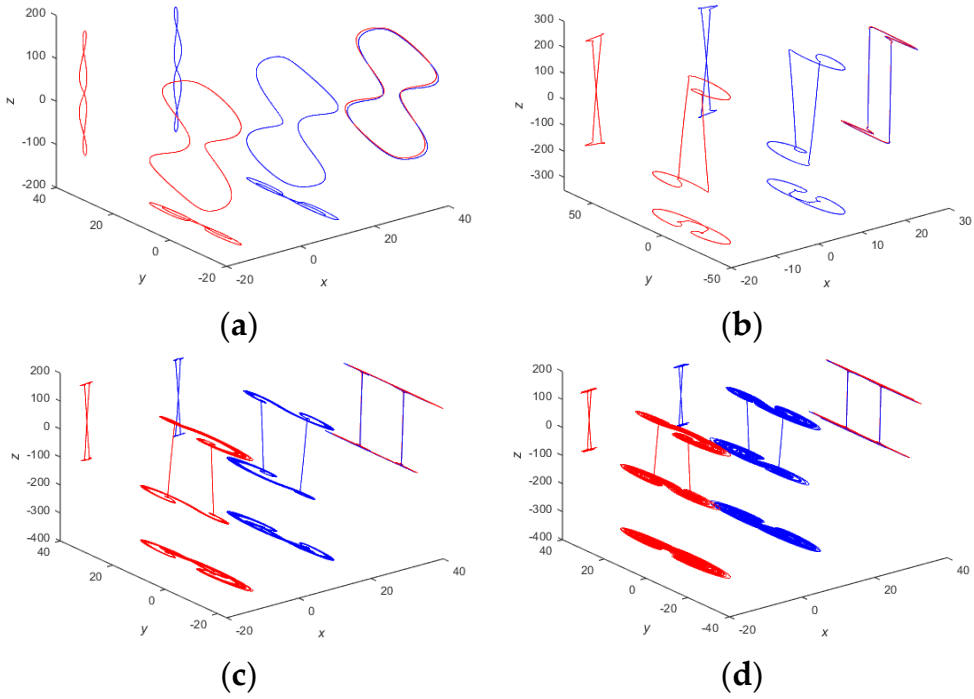


Figure 14. Coexisting attractors: (**a**) C = 1 nF; (**b**) C = 10 nF; (**c**) C = 50 nF; (**d**) C = 100 nF.

3.4.2. Dynamic Depending on L

Setting the initial values as $(x_0, y_0, z_0) = (12.5, -0.5, 0.3)$ and $(x_0, y_0, z_0) = (-12.5, -0.5, 0.3)$, some parameters were confirmed as $a = -4.7, b = 10, c = -4.7$, $k = 20{,}000, m = 160, e = 10, f = 10, C = 1e-7$, and $q = 0.9$. When the parameter L was changed, the resulting bifurcation diagram and the largest Lyapunov exponent are shown in Figure 15.

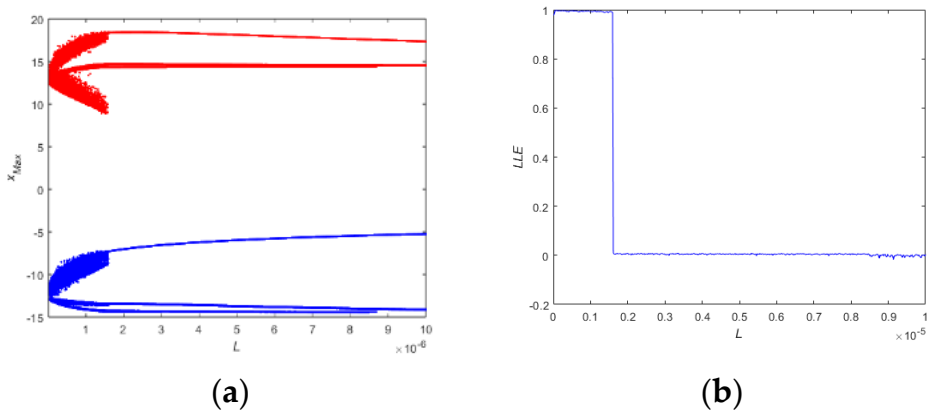


Figure 15. Dynamic behavior depending on *L*: (**a**) coexisting bifurcation diagram; (**b**) largest Lyapunov exponent.

It can be seen that the system (8) was in a chaotic state when L < 1.590 uH, while it was in three cycle state when L < 8.678 uH and in a two cycle state when L ≥ 8.678 uH. The LLE diagram on the right and the bifurcation diagram on the left strictly corresponded. To further illustrate the status of the system, we selected several values of the parameter C and the same initial conditions, and we obtained the coexisting attractors of the system in Figure 16. In Figure 16, three-dimensional coexisting attractors and their projection in the $x - y$, $x - z$, and $y - z$ coordinate axes can be clearly seen. Figure 16a,b shows the phase diagrams of the chaotic state. When the parameter L was increased, the oscillation

amplitude of system (8) increased. Figure 16c,d shows the phase diagrams of different cycle states, but the numbers of periods were different. As the parameter L was increased, the system (8) shifted from chaotic to periodic, which was the opposite result of changes in the parameter C.

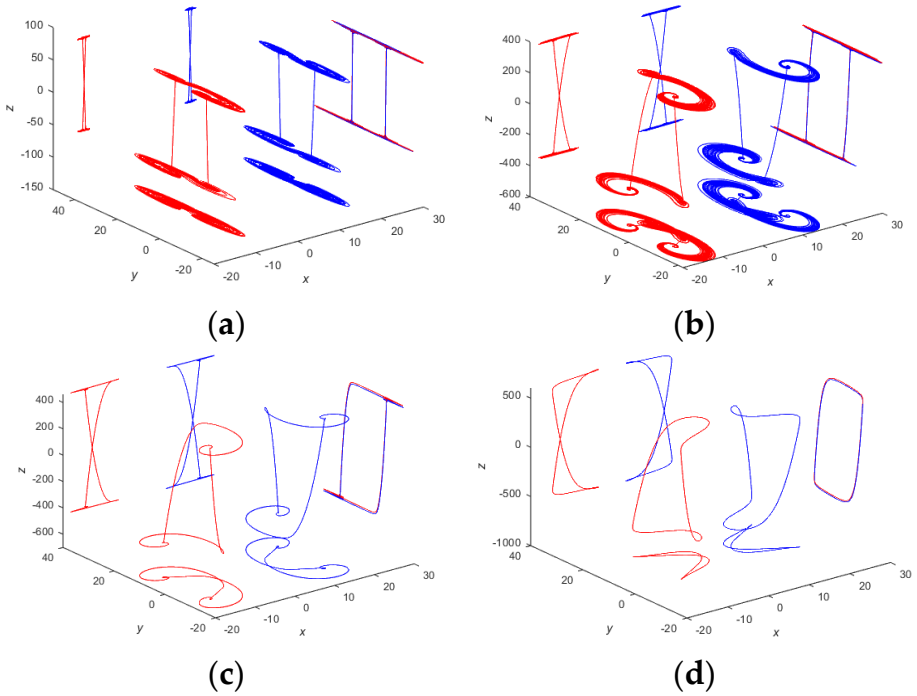


Figure 16. Coexisting attractors: (**a**) L = 5 nH; (**b**) L = 100.0 nH; (**c**) L = 5000 nH; (**d**) L = 10,000 nH.

3.4.3. Dynamic Depending on k

Setting the initial values as $(x_0, y_0, z_0) = (12.5, -0.5, 0.3)$ and $(x_0, y_0, z_0) = (-12.5, -0.5, 0.3)$, some parameters were confirmed as $a = -4.7, b = 10, c = -4.7$, $m = 160, e = 10, f = 10, L = 1e-7, C = 1e-7$, and $q = 0.9$. When the parameter k was change, the resulting bifurcation diagram and the largest Lyapunov exponent are shown in Figure 17.

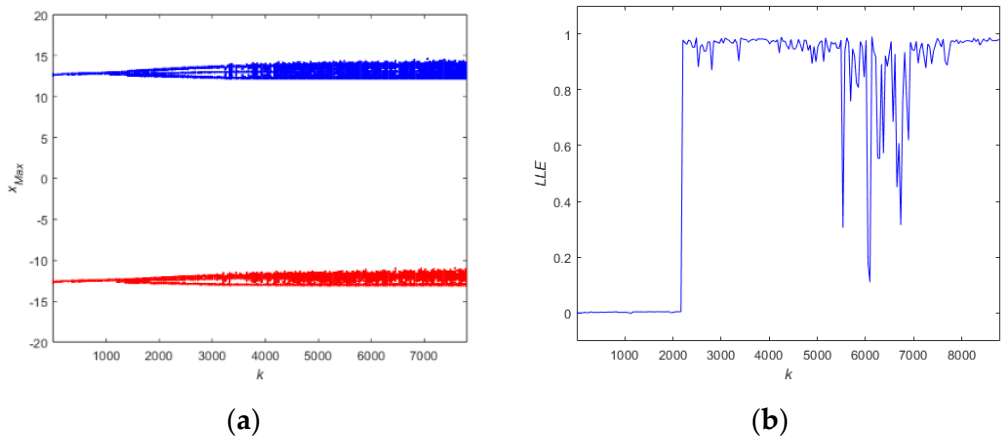


Figure 17. Dynamic behavior depending on k: (**a**) coexisting bifurcation diagram; (**b**) largest Lyapunov exponent.

In Figure 18, the three-dimensional coexisting attractors and their projection in the $x-y$, $x-z$, and $y-z$ coordinate axes can be clearly seen. Figure 18a–c shows the phase

diagrams of cycle states. When the parameter k was increased, the oscillation amplitude and number of cycles of the system (8) increased. Figure 18d shows the phase diagrams of the chaotic state. As the parameter k was increased, the system (8) shifted from periodic to chaotic.

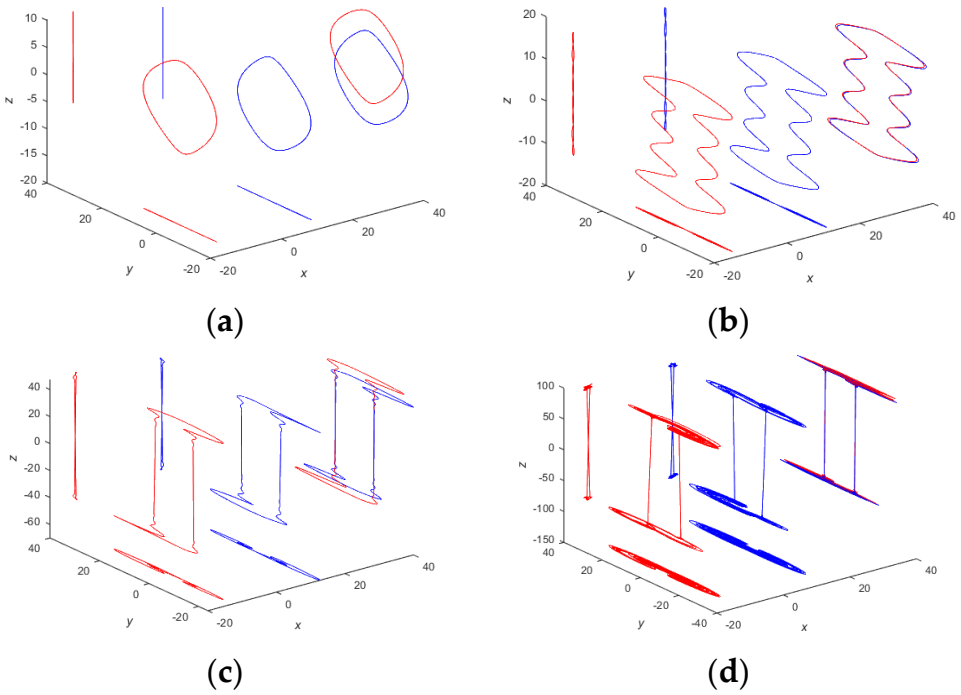


Figure 18. Coexisting attractors: (**a**) k = 100; (**b**) k = 1000; (**c**) k = 2000; (**d**) k = 6000.

3.4.4. Dynamic Depending on m

Setting the initial values as $(x_0, y_0, z_0) = (12.5, -0.5, 0.3)$ and $(x_0, y_0, z_0) = (-12.5, -0.5, 0.3)$, some parameters were confirmed as $a = -4.7, b = 10, c = -4.7$, $k = 20,000, e = 10, f = 10, L = 1e-7, C = 1e-7$, and $q = 0.9$. When the parameters m was changed, the resulting bifurcation diagram and the largest Lyapunov exponent are shown in Figure 19.

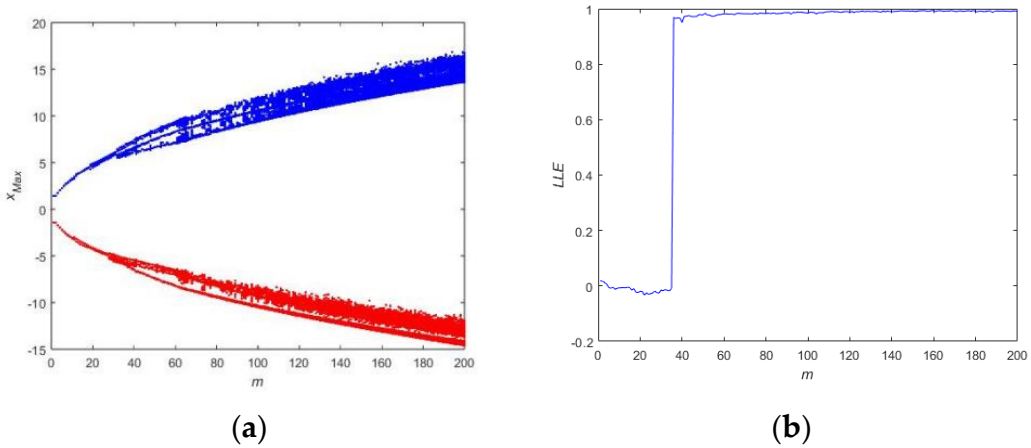


Figure 19. Dynamic behavior depending on m: (**a**) coexisting bifurcation diagram; (**b**) largest Lyapunov exponent.

It can be seen that the system (8) was in a periodic state when m < 38, while it was in a chaotic state when m ≥ 38. The LLE diagram on the right and the bifurcation

diagram on the left strictly corresponded. To further illustrate the status of the system, we selected several values of the parameter m and the same initial conditions, and we obtained the coexisting attractors of the system in Figure 20. In Figure 20, the three-dimensional coexisting attractors and their projection in the $x-y$, $x-z$, and $y-z$ coordinate axes can be clearly observed. Figure 20a–c shows the phase diagrams of different periodic states. Figure 20d shows the phase diagram of the chaotic state. As the parameter m was increased, the system (8) shifted from periodic to chaotic.

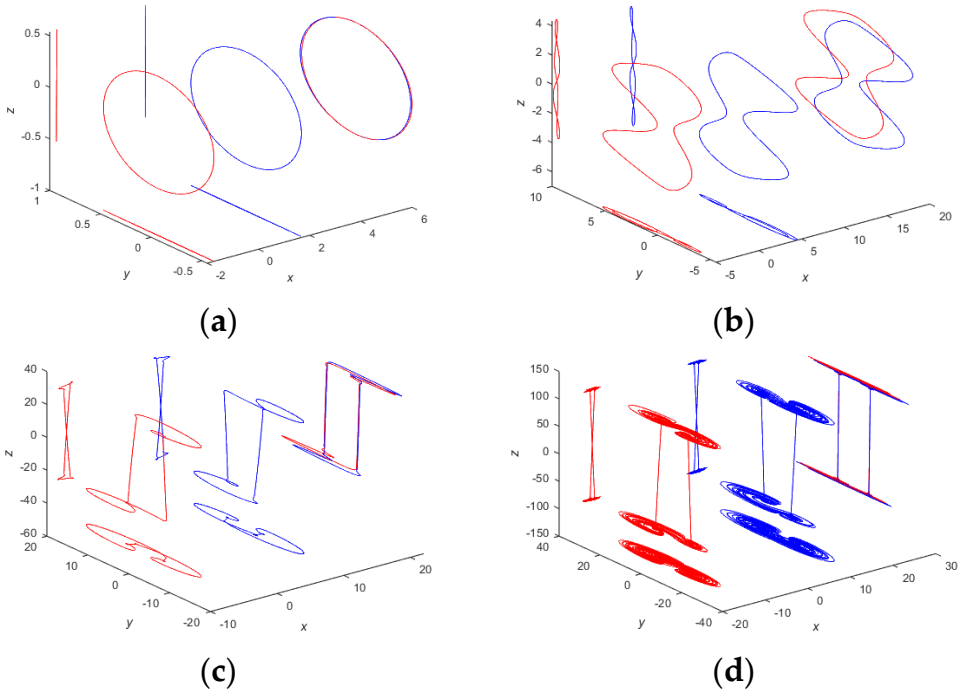


Figure 20. Coexisting attractors: (**a**) m = 3; (**b**) m = 20; (**c**) m = 50; (**d**) m = 150.

3.4.5. Dynamic Depending on q

Setting the initial values as $(x_0, y_0, z_0) = (12.5, -0.5, 0.3)$ and $(x_0, y_0, z_0) = (-12.5, -0.5, 0.3)$, some parameters were confirmed as $a = -4.7, b = 10, c = -4.7$, $k = 20{,}000, m = 160, e = 10, f = 10, C = 1e-7$, and $L = 1e-7$. When the parameter q was changed, the resulting coexisting bifurcation diagram is shown in Figure 21a. It can be seen that no matter how q changed, the system (8) was always in a chaotic state. Although the state of the system had not changed, the frequency of system oscillation underwent significant changes from the time-domain diagram in Figure 21b–d. When q was increased, the frequency of the system rapidly decreased.

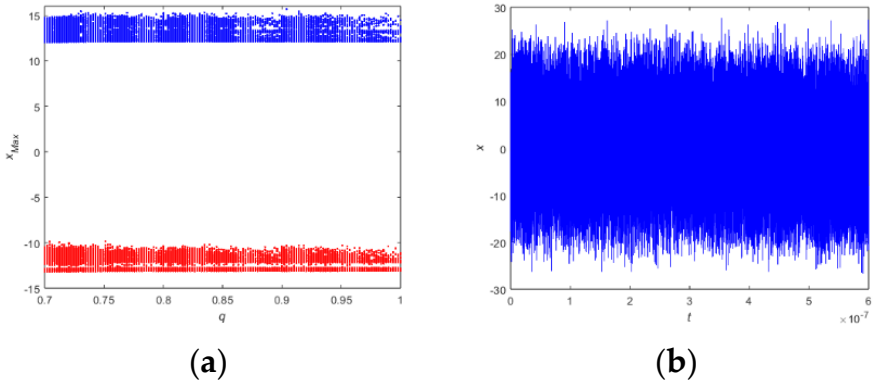
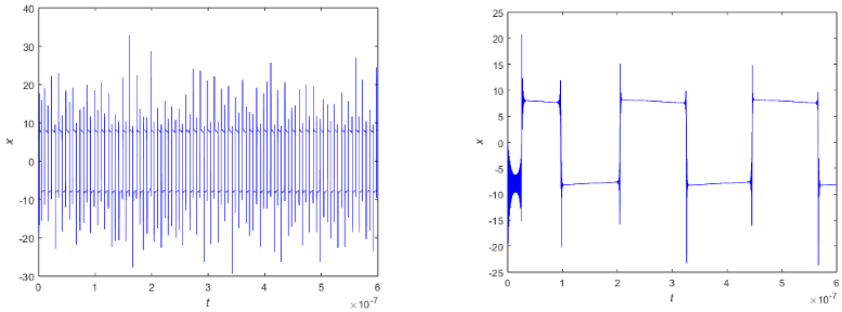


Figure 21. *Cont.*



(**c**) (**d**)

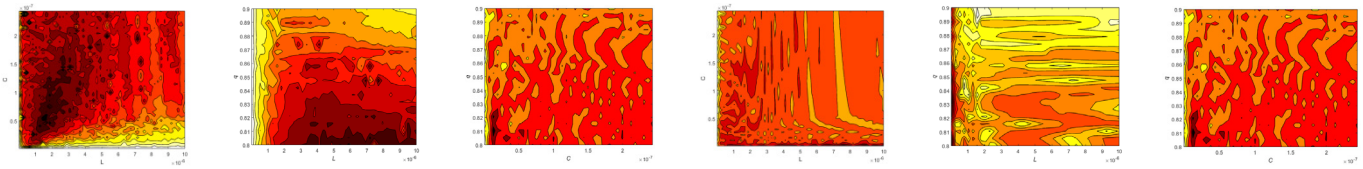
Figure 21. Dynamic behavior depending on q: (**a**) coexisting bifurcation diagram; (**b**) time-domain diagram when q = 0.7; (**c**) time-domain diagram when q = 0.8; (**d**) time-domain diagram when q = 0.9.

3.5. Complexity Analysis

Complexity analysis is an important aspect of chaotic dynamics research, which includes two aspects: behavioral complexity and structural complexity. The complexity of a chaotic system is a measurement of the degree of a chaotic sequence approaching a random sequence. The closer the sequence is to a random sequence, the higher the complexity, and the higher the corresponding security. In this paper, we mainly studied the structural complexity of the system (8), which consisted of spectral entropy (SE) and C0 complexity.

Spectral entropy is a measure of disorder applied to the power spectrum of periods of time series data [36]. The principle of C0 complexity is as follows: Calculate the amplitude spectrum of the signal and its mean value, and keep the amplitude spectrum components unchanged if their amplitude values are no less than the mean value while replacing all the other components with zero. Calculate the inverse FFT of the new spectra and obtain a new signal. The ratio of the area of the original signal to the area of the new signal and its mean value over the area between the original signal and its mean value is defined as the C0 complexity of the original system [37].

The chaos diagrams of the system (8) based on the C0 algorithm and the spectral entropy algorithm are shown in Figure 22. From the diagrams, it can be seen that the boundaries were more obvious between high complexity regions and low complexity regions. Comparing the C0 algorithm and the spectral entropy algorithm, the C0 algorithm performed better.



(**a**) (**b**) (**c**) (**d**) (**e**) (**f**)

Figure 22. Complexity diagrams of the system (8): (**a**) C0 of L-C plane; (**b**) C0 of L-q plane; (**c**) C0 of C-q plane; (**d**) SE of L-C plane; (**e**) SE of L-q plane; (**f**) SE of C-q plane.

4. Visually Meaningful Image Encryption and Decryption Scheme

In this section, the visually meaningful image encryption and decryption scheme is described. First, the fractional-order memristive chaotic sequences were generated. Second, image encryption based on DNA coding is described in detail, and the flow chart of the proposed scheme is shown as Figure 23a. Third, the embedding process of the encrypted images by IWT is described, and the flow chart of the proposed scheme is shown as Figure 23b. Finally, the simulation results and performance analysis are presented.

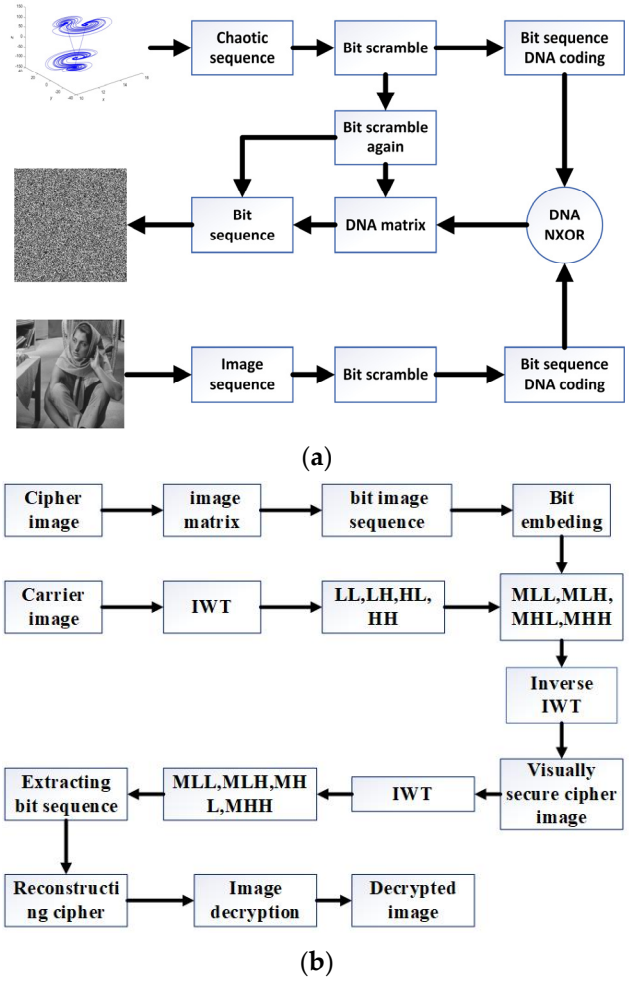


Figure 23. Flow chart of the proposed scheme: (**a**) encryption scheme; (**b**) embedding and extracting schemes of a cipher image.

4.1. Chaotic Sequence Generation

The fractional-order memristive chaotic system could stably and continuously generate chaotic sequences, which had strong pseudo-random and statistical characteristics, thus providing a solid foundation for their application. The detailed process is as follows.

Step 1: In order to increase security, we selected the region with the highest complexity, as shown in Figure 22, and removed the initial N_0 iteration data. Then, four different sets of initial values were used in Equation (8), generating four sets of chaotic sequences: $x_1, y_1, z_1; x_2, y_2, z_2; x_3, y_3, z_3$; and x_4, y_4, z_4. Next, we chose $s_1(i,j) = \{x_j(i),\ i = 1,2,\cdots,mn,\ j = 1,2,3,4\}$, and $s_2(i,j) = \{y_j(i),\ i = 1,2,\cdots,mn,\ j = 1,2,3,4\}$ as two four-dimensional sequences.

Step 2: The two new sequences s_1 and s_2 were used to generate two special sequences $s_1'(i,j) \in [0,255]$, $j = 1,2,3,4$ and $s_2'(i,j) \in [0,255]$, $j = 1,2,3,4$, respectively. Their transformation process is as follows:

$$s_1'(i,j) = \mathrm{mod}\left\{\mathrm{floor}\left(\mathrm{abs}\left(x_j(i)\right) - \mathrm{floor}\left(\mathrm{abs}\left(x_j(i)\right)\right)\right) \times 10^6, 256\right\},$$

$$i = 1,2,\cdots,mn, j = 1,2,3,4$$

$$s_2'(i,j) = \mathrm{mod}\left\{\mathrm{floor}\left(\mathrm{abs}\left(y_j(i)\right) - \mathrm{floor}\left(\mathrm{abs}\left(y_j(i)\right)\right)\right) \times 10^6, 256\right\}$$

$$i = 1,2,\cdots,mn, j = 1,2,3,4$$

where mod$\{\cdot\}$ denotes the modulo operation, floor$(\cdot)$ denotes the flooring operation, and abs$(\cdot)$ denotes the absolute value operation. Then, we merged s'_1 and s'_2 as vector *ss* :

$$ss(i) = [s'_1(i,1), s'_2(i,1), s'_1(i,2), s'_2(i,2), s'_1(i,3), s'_2(i,3), s'_1(i,4), s'_2(i,4)]$$

Step 3: Then, *ss* was composed for a sequence of k, and every element of k was translated into a binary number of 8 bits. The sequence of k can be represented as:

$$k = [ss(1), ss(3), \cdots, ss(2), ss(4), \cdots, ss(mn)]$$

4.2. Image Encryption

The methods of DNA encoding [38] and bit confusion are used to encrypt images, thereby resulting in changes in pixel values and pixel positions. Based on DNA encoding rules, the combination of two binary bits stands for a nucleotide, e.g., A-00, C-01, G-10, and T-11. There are eight combinations that are frequently used in all combinations, as shown in Table 2. If one wants to encode an image using the DNA rule, all pixels of the image need to be converted into 8-bit binary sequences, and then encoded using DNA rules. For instance, the 135-pixel value is defined as a 10000111 binary array and it can be encoded as 01111000 by utilizing rule 8 in Table 2. It can be seen that the DNA encoding rules can change the pixel values of the image and have an effect on encryption. The specific scheme can be divided into five steps:

Table 2. DNA coding rule.

Rule	1	2	3	4	5	6	7	8
00	A	C	C	A	G	T	T	G
01	C	T	A	G	A	G	C	T
10	G	A	T	C	T	C	G	A
11	T	G	G	T	C	A	A	C

Step 1: Transform the plaintext image P to binary matrix P_{bm} with the size of $m \times 8n$. The pixel values of the image P are confused by bits, and the pixel positions are also simultaneously changed to reduce the correlation between adjacent pixels. The special process of the scheme is as follows: First, transform the two-dimension binary matrix P_{bm} into the one-dimension binary sequence s_{bin}. Second, arrange the binary sequence of k in special order, and acquire a new sequence k_{new}. Third, use the sequence k_{new} to scramble the one-dimension sequence s_{bin}, and obtain binary sequence $s'_{bin} = s_{bin}(k_{new})$.

Step 2: First, encode the binary sequence s'_{bin} as DNA sequence s_{DNA} using the final DNA encoding rule (Table 2). Second, transform sequence s'_2 into a binary sequence s'_{2bin}, then encode binary sequence s'_{2bin} as DNA sequence s_{2DNA} using the first DNA encoding rule. Third, conduct the DNA XNOR operation (Table 3) on DNA sequences s_{DNA} and s_{2DNA}, and obtain a new DNA sequence s'_{DNA}.

$$s'_{DNA}(i) = s_{2DNA}(i) \odot s_{DNA}(i), i \in [1, 4mn]$$

where $\odot$ denotes the DNA XNOR operation.

Step 3: Extract the odd terms of sequence k_{new}, and obtaining a new sequence $k^{odd}_{new} = [k_{new}(1), k_{new}(3), \cdots, k_{new}(8mn-1)]$. Then, transform sequence s'_{DNA} into the sequence s''_{DNA} on the value of k^{odd}_{new}: if $0 < k^{odd}_{new}/255 < 0.5$, the corresponding $s'_{DNA}(i)$ is performed in reverse operation, otherwise it remains unchanged.

Step 4: The fourth DNA encoding rule is used to decode the DNA sequence s''_{DNA} to obtain the binary sequence s''_{bin}.

Step 5: The binary sequence s''_{bin} is converted to the corresponding cipher image P_{cipher}.

Table 3. DNA XNOR.

$\odot$	**A**	**G**	**C**	**T**
A	T	C	G	A
G	C	T	A	G
C	G	A	T	C
T	A	G	C	T

According to the five steps above, we summarize them into the following Algorithm 1.

Algorithm 1: The image encryption process.

(1) Convert P into a binary matrix P_{bm}
(2) Convert P_{bm} into a binary sequence s_{bin}
(3) Arrange the k sequence order, and acquire a new sequence k_{new}
(4) for i = 1:mn $\times$ 8 do

$$S_{bin}(i) = s_{bin}(k_{new}(i))$$

End for
(5) Encode the sequence S_{bin} as a DNA sequence s_{DNA} using the eighth DNA encoding rule
(6) Transform the sequence s'_2 into a binary sequence s'_{2bin}
(7) Encode the sequence s'_{2bin} as a DNA sequence s_{2DNA} using the first DNA encoding rule
(8) for i = 1:mn $\times$ 4 do

$$s'_{DNA}(i) = s_{2DNA}(i) \odot s_{DNA}(i)$$

End for
(9) Extract the odd term of k sequence as a new sequence k_{new}^{odd}
(10) Transform the sequence s'_{DNA} into the sequence s''_{DNA} on the value of k_{new}^{odd}
(11) Decode the sequence s''_{DNA} to a binary sequence s''_{bin} using the fourth DNA encoding rule
(12) Convert binary sequence s''_{bin} into decimal matrix P_{cipher}

In order to elaborate on this encryption process, an illustrative example for a 5 $\times$ 5 pixel sample of the Barbara image is presented in Figure 24 on Algorithm 1.

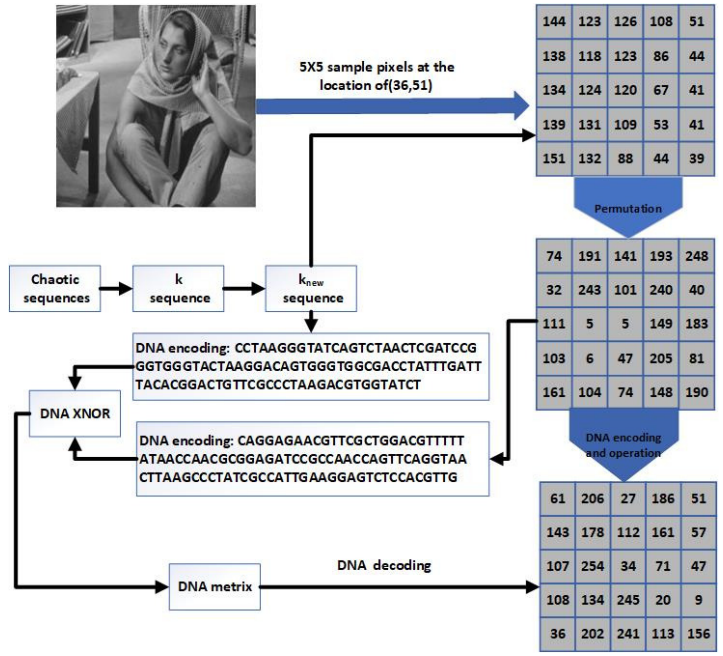


Figure 24. Processes of the proposed encryption scheme with an illustrative example.

4.3. Cipher Image Embedding Process

In order to obtain a visually meaningful cipher image, we embedded the cipher image P_{cipher} into the carrier image $P_{carrier}$ using the IWT and Least Significant Bit (LSB) methods. LSB replacement is a well-known steganographic scheme, in which the LSBs of the carrier image are replaced by cipher image data bits to obtain the stego image. The method has the advantages of sufficient payload, good visual and statistical imperceptibility, and ease of implementation. The specific steps of the embedding process are as follows:

Step 1: Decompose the carrier image $P_{carrier}$ to obtain the wavelet coefficient matrices LL, LH, HL, and HH by IWT, and then transform them into one-dimensional sequences, labeled as s_{LL}, s_{LH}, s_{HL}, and s_{HH}, respectively.

Step 2: Transform s_{LL}, s_{LH}, s_{HL}, and s_{HH} as 8-bit binary sequences $s_{binLL} = \{ll_i, i = 1, 2, \cdots, mn\}$, $s_{binLH} = \{lh_i, i = 1, 2, \cdots, mn\}$, $s_{binHL} = \{hl_i, i = 1, 2, \cdots, mn\}$, and $s_{binHH} = \{hh_i, i = 1, 2, \cdots, mn\}$, respectively.

Step 3: Convert the secret image P_{cipher} into a one-dimensional sequence $s_{P_{cipher}} = \{y_i, i = 1, 2, \cdots, mn\}$, and then transform the sequence $s_{P_{cipher}}$ into 8-bit binary sequence $s_{binP_{cipher}}$.

Step 4: Replace the lowest 2 bits of s_{binLL}, s_{binLH}, s_{binHL}, and s_{binHH} with the bits of $s_{binP_{cipher}}$ as follows:

$$ll_{i1}ll_{i2} = y_{i1}y_{i8},\ lh_{i1}lh_{i2} = y_{i2}y_{i7},$$

$$hl_{i1}hl_{i2} = y_{i3}y_{i6},\ hh_{i1}hh_{i2} = y_{i3}y_{i4}.$$

Step 5: Obtain four new binary sequences s'_{binLL}, s'_{binLH}, s'_{binHL}, and s'_{binHH}, which contain the secret image. Transform the binary sequences s'_{binLL}, s'_{binLH}, s'_{binHL}, and s'_{binHH} into decimal sequences s'_{LL}, s'_{LH}, s'_{HL}, and s'_{HH}, respectively, and then arrange the sequences s'_{LL}, s'_{LH}, s'_{HL}, and s'_{HH} into two-dimensional forms. Finally, use inverse IWT to convert them into a visually meaningful cipher image.

According to the five steps above, we summarize them into the following Algorithm 2.

Algorithm 2: The cipher embedding process.

(1) Decompose the carrier image $P_{carrier}$ by IWT, and then transform it into the sequences s_{LL}, s_{LH}, s_{HL}, and s_{HH}
(2) Transform s_{LL}, s_{LH}, s_{HL}, and s_{HH} as 8-bit binary sequences s_{binLL}, s_{binLH}, s_{binHL}, and s_{binHH}
(3) Convert the secret image P_{cipher} into 8-bit binary sequence $s_{binP_{cipher}}$
(4) Replace the lowest 2 bits of s_{binLL}, s_{binLH}, s_{binHL}, and s_{binHH} with the bits of $s_{binP_{cipher}}$
(5) $s'_{binLL} = [s_{binLL}(:,1{:}6), s_{binP_{cipher}}(:,1{:}2)]$
(6) $s'_{binLH} = [s_{binLH}(:,1{:}6), s_{binP_{cipher}}(:,3{:}4)]$
(7) $s'_{binHL} = [s_{binHL}(:,1{:}6), s_{binP_{cipher}}(:,5{:}6)]$
(8) $s'_{binHH} = [s_{binHH}(:,1{:}6), s_{binP_{cipher}}(:,7{:}8)]$
(9) Transform the sequences s'_{binLL}, s'_{binLH}, s'_{binHL}, and s'_{binHH} into decimal sequences s'_{LL}, s'_{LH}, s'_{HL}, and s'_{HH}
(10) Use inverse IWT to convert s'_{LL}, s'_{LH}, s'_{HL}, and s'_{HH} into a visually meaningful cipher image

In order to elaborate on this embedding process, an illustrative example is presented for a 3×3 pixel sample of the cipher image and a 6×6 pixel sample of the carrier image in Figure 25 on Algorithm 2.

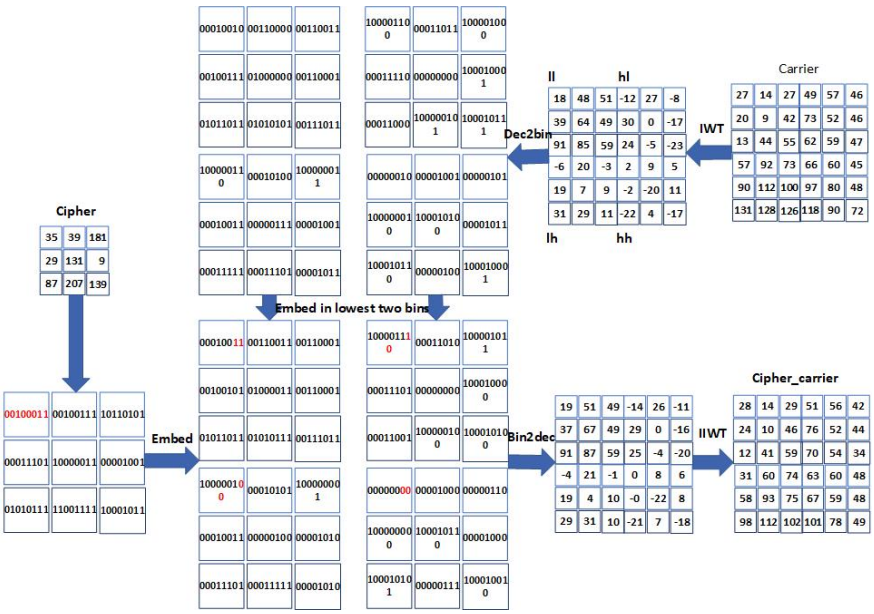


Figure 25. Processes of the proposed embedding scheme with an illustrative example.

4.4. Image Decryption Scheme

The image decryption and image encryption processes are opposite. First, extract the secret image from the visually secure cipher image, and then recover the plain image from the secret image. The flow chart of the image decryption scheme is shown in Figure 26.

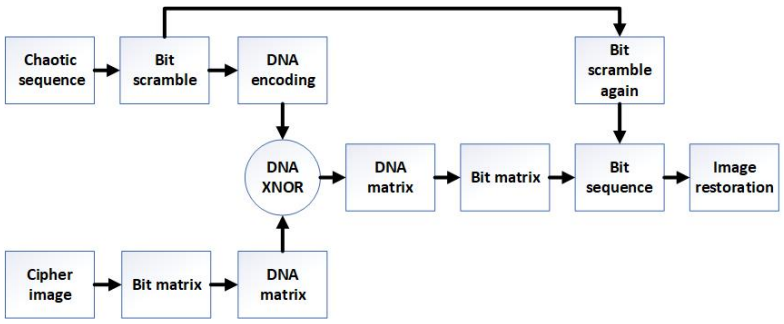


Figure 26. Flow chart of the decryption scheme.

4.5. Extracting the Cipher Image

Step 1: Decompose the visually meaningful cipher image by IWT, and obtain four modified wavelet coefficient matrices: MLL, MLH, MHL, and MHH.

Step 2: Transform the four modified wavelet coefficient matrices MLL, MLH, MHL, and MHH into one-dimensional sequences s_{MLL}, s_{MLH}, s_{MHL}, and s_{MHH}, and then decompose them into 8-bit binary sequences s_{binMLL}, s_{binMLH}, s_{binMHL}, and s_{binMHH}.

Step 3: Extract the final 2 bits of every pixel as follows:

$$y_{i1}y_{i8} = ll_{i1}ll_{i2},\ y_{i2}y_{i7} = lh_{i1}lh_{i2},$$

$$y_{i3}y_{i6} = hl_{i1}hl_{i2},\ y_{i3}y_{i4} = hh_{i1}hh_{i2}.$$

Step 4: Compose the bit values and transform them into decimals, and then arrange them in two-dimensions to obtain the secret image.

5. Numerical Simulation and Analysis

The simulation test results and performance analysis of the proposed visually meaningful image encryption algorithm are described in this section. Some classic metrics such as histograms, correlation coefficients, secret key sensitivity, NPCR, UACI, noise attacks, and cropping attacks were measured. The size of the plain image was 256×256, and the size of the carrier image was 512×512.

5.1. Simulated Results

The visual security performance of our encryption scheme was tested, the results are displayed in Figure 27. The first row shows four plain images, and the second row shows the visually secret cipher images corresponding to the four plain images of the first row. The third row shows the carrier images. It can be seen that the carrier images look like the normal images, and therefore they are visually safe and would not motivate an attacker. The fourth row shows the corresponding cipher images extracted from the carrier images. The final row shows the corresponding decryption images.

Figure 27. Encryption and decryption results for carrier images: from the first row to the fifth row: plain images, secret images, carrier images, cipher images, and reconstructed images, respectively.

It can be seen from Figure 27 that the decryption images were almost the same as the plain images, which indicated that the quality of the reconstructed images was pretty good. Peak Signal-to-Noise Ratio (PSNR) and Mean Structure Similarity (MSSIM) [39] are often used to evaluate the performance of proposed schemes, which are defined as

$$\mathrm{PSNR} = 10\log\frac{255^2}{\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}[R(i,j) - P(i,j)]^2} \tag{15}$$

$$\begin{cases} \mathrm{SSIM}(D_i, I_i) = \frac{2\mu_D^i \mu_I^i + (0.01 \times 255)^2}{(\mu_D^i)^2 + (\mu_I^i)^2 + (0.01 \times 255)^2} \\ \quad \times \frac{2\sigma_D^i \sigma_I^i + (0.03 \times 255)^2}{(\sigma_D^i)^2 + (\sigma_I^i)^2 + (0.03 \times 255)^2} \\ \quad \times \frac{2\sigma_{DI}^i + (0.03 \times 255)^2}{2\sigma_D^i \sigma_I^i} \\ \mathrm{MSSIM}(D, I) = \frac{1}{L} \sum_{i=1}^{L} \mathrm{SSIM}(D_i, I_i) \end{cases} \tag{16}$$

where R and P refer to the reconstructed image and the plain image, respectively. L is the gross of the selected image subblocks, μ_I^i and σ_I^i are the mean value and variance of the i-th subblock selected in image I, respectively.

The MSSIM value quantitatively describes the similarity of two images in terms of brightness, contrast, and structure. The experimental results are listed in Table 4, where the symbol "p-r" represents the value between the plain image and the reconstructed image, and "c-v" represents the value between the carrier image and the visually meaningful cipher image. The numerical results indicated that the similarities among the plain images and the reconstructed images were very high. Additionally, the mean values of PSNRc-v and MSSIMc-v were 43.8008 dB and 0.9956, respectively. In short, the above numerical values implied that the proposed scheme could provide good visual security and data security.

Table 4. The values of PSNR and MSSIM.

Plain Image	Carrier Image	$\mathbf{PSNR_{p\text{-}r}}$	$\mathbf{MSSIM_{p\text{-}r}}$	$\mathbf{PSNR_{c\text{-}v}}$	$\mathbf{MSSIM_{c\text{-}v}}$
Barbara	Peppers	37.9526	0.9523	43.9256	0.9962
Cameraman	Pirate	38.8541	0.9567	43.2873	0.9952
Circuit	Living room	39.6512	0.9612	43.9638	0.9937
House	Walk bridge	39.0246	0.9528	43.4846	0.9990
Average		38.8706	0.95575	43.6653	0.9960

5.2. Performance Analyses

In order to analyze the security and robustness of the proposed scheme, some quantitative indexes were defined, calculated, and then compared with existing visually meaningful image encryption schemes.

5.2.1. Key Security Analysis

Key sensitivity is an important parameter to evaluate the security of an image encryption algorithm. In this subsection, the plain image is labeled as P, and the carrier image is labeled as $P_{carrier}$. In the simulation, the key parameters were fixed as $a = -4.7$, $b = 10, c = -4.7, k = 20,000, m = 160, e = 10, f = 10, C = 1e-7, L = 1e-7$, and $q = 0.9$. When we added 10^{-16} to one of the parameters and the others were constant, the cipher image was incorrectly decrypted, as shown in Figure 28d. It can be seen that the decrypted image was completely different from the plain image. When we used the correct key to decrypt the cipher, it was almost the same as the plain image. In addition, we can observe that the embedded images with the correct key were almost the same as those with the modified keys, and their similar histograms are shown in Figure 28g,h. Obviously, our encryption algorithm was extremely sensitive to keys, yet modifying the keys had little effect on the carrier image. When the modified keys were used in the decryption process, the recovered images were in noisy-like form, and the change in their pixel values was more than 99% between the wrongly decrypted image and the plain image. These results plainly evidenced that the proposed encryption method had high sensitivity to secret keys in the decryption process.

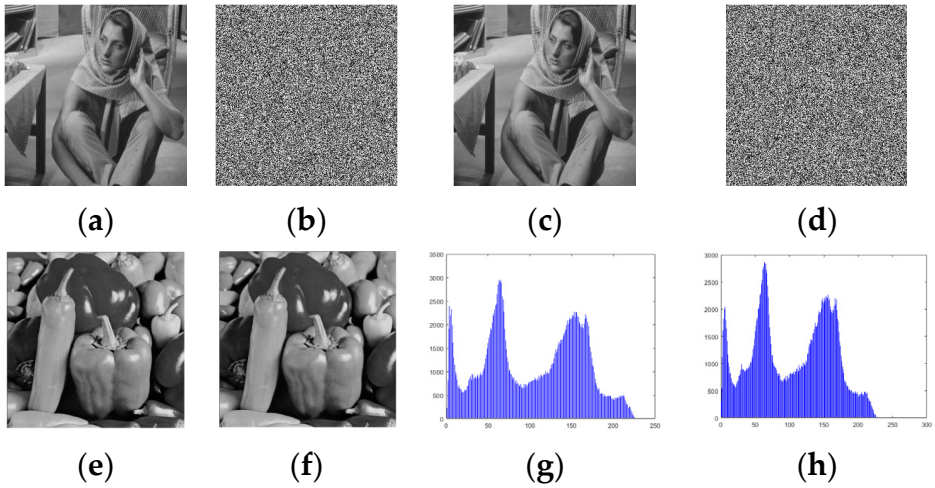


Figure 28. Key sensitivity test: (**a**) plain image; (**b**) cipher image; (**c**) decrypted image; (**d**) erroneously decrypted image; (**e**) P_{cipher} with the correct key; (**f**) P_{cipher} with the modified keys; (**g**) the histogram of (**e**); (**h**) the histogram of (**f**).

5.2.2. Histogram Analysis

The histogram of an image reflects the frequency of image pixel values. Traditional image encryption algorithms can generate a secret image with a uniform histogram to resist statistical attack. When we used our algorithm to generate a visually meaningful cipher image, the histogram of the encrypted image was uniform, as shown in Figure 29b,d, and the histogram of the embedded image was similar to that of the carrier image, as shown in Figure 29e–l. The histograms of the plain images (Barbara and cameraman) and their cipher images are shown in Figure 29a–d. The histograms of the carrier images (third row) and four cipher images (fourth row) in Figure 27 are shown in the second and third row of Figure 29. It can be seen that the histograms of the encrypted images were completely different from those of the plain images, and the histograms of the embedded images and the carrier images were nearly similar, which showed that the attacker would be unable to obtain relevant information by analyzing the histograms of the encrypted images and the embedded images. These results implied that the pixel value distributions of the carrier images were effectively preserved by the embedding method, and the pixel value distributions of the plain images were effectively hidden by the encryption method.

5.2.3. Correlation Analysis

As we all know, there is strong correlation between adjacent pixels in a natural image. In order to quantitatively measure the correlation, we randomly selected three sets of pixel pairs (x_i, y_i) from the plain image, carrier image, and cipher image, where i is from 0 to 3000, and then calculated in the horizontal, vertical, and diagonal directions according to:

$$r_{xy} = \frac{\sum_{i=1}^{N}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \overline{x})^2 \times \sum_{i=1}^{N}(y_i - \overline{y})^2}} \tag{17}$$

where x_i and y_i are the values of two adjacent pixels in the image, $\overline{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$ and $\overline{y} = \frac{1}{N}\sum_{i=1}^{N} y_i$ are the mean values, respectively.

Here, the plain image "Barbara" and the carrier image "peppers" were employed to evaluate the correlation coefficients between two adjacent pixels in the horizontal, vertical, and diagonal directions. The plain image 'Barbara' (first row and first column in Figure 27), carrier image 'peppers' (second row and first column in Figure 27), and the cipher image 'peppers' (second row and fifth column in Figure 27) were calculated using Equation (15), and the corresponding correlation coefficients in the horizontal direction are plotted in Figure 30. From these results one can see that the correlation distributions of the plain

image and encrypted image were completely different, and the correlation distribution of the visually meaningful cipher image was very similar to that of the carrier image but was quite different from those of the plain image and encrypted image. All of these results indicated that it was difficult to distinguish between the visually meaningful cipher image and the carrier image through correlation analysis, and the visual security of our proposed scheme was achieved. Moreover, an attacker could not seek out the connection between the cipher image and plain image through correlation analysis.

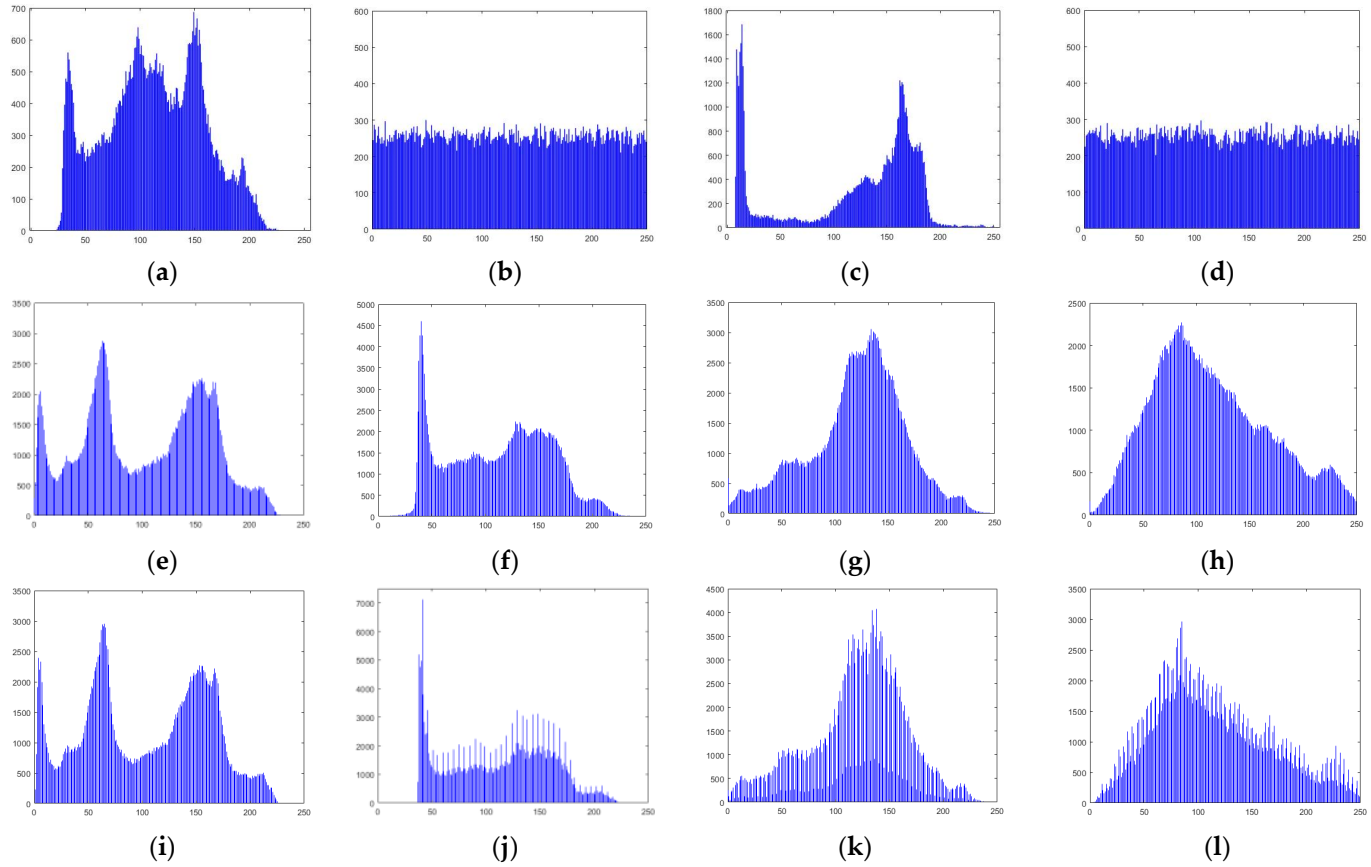


Figure 29. Histograms: (**a**) Barbara; (**b**) encrypted (**a**); (**c**) cameraman; (**d**) encrypted (**c**); (**e**–**h**) carrier images (third row of Figure 27); (**i**–**l**) embedded images (fourth row of Figure 27).

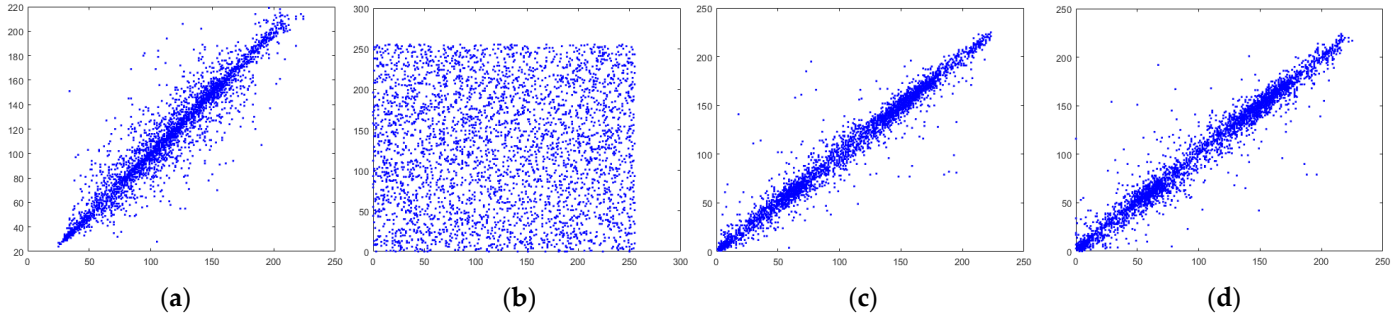


Figure 30. The correlation distributions of two adjacent pixels in the horizontal direction: (**a**) plain image 'Barbara' (first row and first column in Figure 27); (**b**) encrypted image 'Barbara' (**c**) carrier image 'peppers'; (**d**) visually meaningful cipher image 'peppers' (second row and fifth column in Figure 27).

5.3. Various Attacks

In this subsection, the abilities of the proposed scheme to resist noise attack, cropping attack, differential attack, and chosen-plaintext attack are evaluated.

5.3.1. Noise Attack

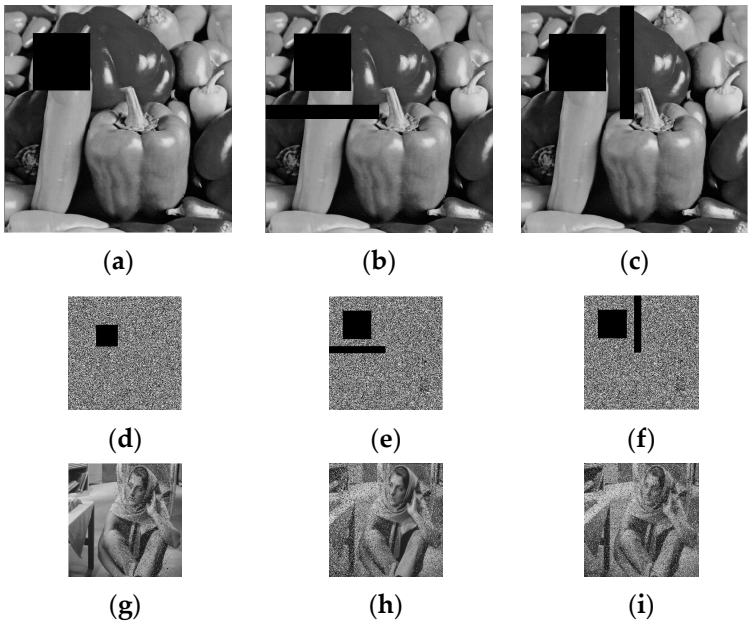
When the cipher image is transferred in real image communication, noise will inevitably be added to the cipher image. In order to test the ability of our scheme to defend against noise attack, salt-and-pepper noise (SPN) was added to the cipher image. The resulting images are shown in Figure 31. It can be seen that SPN had different impacts on different noise densities, and the recovered images are shown in Figure 31a–e. The PSNR values were 25.13, 23.52, 19.41, 18.71, and 17.78 dB, respectively. In a word, the proposed encryption scheme had a strong capability for resisting SPN.

(**a**) (**b**) (**c**) (**d**) (**e**)

Figure 31. The resistance capability to different SPN densities: (**a**) d = 0.04; (**b**) d = 0.08; (**c**) d = 0.12; (**d**) d= 0.16; (**e**) d = 0.2.

5.3.2. Cropping Attack

When the image is transferred over the internet, data loss of the visually meaningful cipher image can greatly affect the recovered image. Cropping attack is a factor that causes data loss. We randomly cut a few regions in these images and recovered the plain images using our decryption algorithm. The visually meaningful cipher images with three manners of data loss are depicted in Figure 32a–c, and the extracted cipher images and the recovered images are shown in Figure 32d–i. It can be seen that our scheme had higher robustness. The results implied that the proposed scheme could resist large-scale cropping attack to a certain extent.



(**a**) (**b**) (**c**)

(**d**) (**e**) (**f**)

(**g**) (**h**) (**i**)

Figure 32. Results of resistance against cropping attack: (**a**–**c**) schematic of cropping in the visually meaningful cipher images; (**d**–**f**) cipher images extracted from the visually meaningful cipher images; (**g**–**i**) decrypted results of the proposed scheme.

5.3.3. Differential Attack

Hackers often use differential attacks to attack some target images. By changing a pixel value in the plain image and analyzing the difference between two cipher images, attackers attempt to seek out the relationship between the plain image and the cipher image and then reconstruct the original image without secret keys. In the proposed scheme, we first assumed the cipher images were obtained from the carrier images. Then we tested the ability of the encryption algorithm to defend against differential attack. First, suppose P_1 and P_2 are two plain images with one bit difference. Second, encrypt P_1 and P_2 using the same chaotic sequence. Third, obtain the corresponding secret images S_1 and S_2. Finally, calculate the NPCR (Number of Pixels Change Rate) and UACI (Unified Average Changing Intensity) values of S_1 and S_2. NPCR and UACI are usually applied to check the performance of the system against differential attack, and they are calculated as follows:

$$\mathrm{NPCR} = \frac{1}{W \times H} \sum_{i=1}^{W} \sum_{j=1}^{H} D(i,j) \times 100\% \tag{18}$$

$$\mathrm{UACI} = \frac{1}{W \times H} \sum_{i=1}^{W} \sum_{j=1}^{H} \frac{|C_1(i,j) - C_2(i,j)|}{255} \times 100\% \tag{19}$$

where $C_1(i,j)$ and $C_2(i,j)$ are cipher images, both of which are $W \times H$ in size. If $C_1(i,j) = C_2(i,j)$, then $D(i,j) = 0$, otherwise $D(i,j) = 1$. The results are shown in Table 5.

Table 5. The NPCR and UACI values of the plain images changed by one bit.

Item	Barbara	Cameraman	Circuit	House	Average	Ref. [40]
NPCR (%)	99.63	99.57	99.69	99.48	99.592	99.59
UACI (%)	33.69	33.58	33.62	33.63	33.63	33.5

Table 5 shows the results for four different plain images. One can observe that when the plain image changed by one bit, the NPCR value of the corresponding cipher image was about 99.592% and the UACI value was 33.65%. Comparing the results reported in [40] with our results, it could be seen that the values of NPCR and UACI were higher. This meant that the cipher image remained almost unchanged and the proposed encryption algorithm had a strong resistance capability against differential attack.

5.3.4. Chosen-Plain Attack

Chosen-plain attack can break a variety of image encryption schemes. An encryption scheme should have the ability to resist such an attack. For an entirely black image, as shown in Figure 33a, its secret image was obtained, as shown in Figure 33b. It can be seen that there was no obvious difference between the secret image and the secret images of other plain images. The visually meaningful cipher image obtained through the embedding process is shown in Figure 33e, and its appearance was similar to that of the carrier image, as shown in Figure 33d. Finally, the noise-like cipher image in Figure 33e was extracted, as shown in Figure 33c. Therefore, it would be difficult for the attacker to obtain any useful information through this method, which implied that the encryption scheme could better resist the chosen-plain attack.

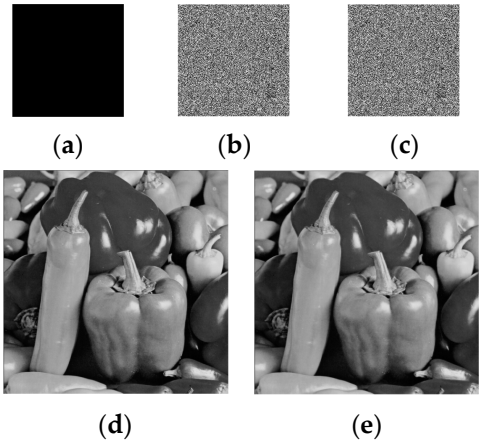


Figure 33. Chosen-plain attack scheme: (**a**) all-zero image; (**b**) all-zero image encrypted; (**c**) extracted cipher image from the visually meaning cipher image; (**d**) carrier image; (**e**) visually meaningful cipher image.

5.4. Comparison with Existing Schemes

An excellent scheme should have good comprehensive performance. In this section, we compare our scheme with existing image encryption schemes in terms of safety and time efficiency. For this purpose, we list some data to highlight the advantages of our algorithm in Tables 6 and 7, which mainly include the following two aspects: various attacks and execution efficiency.

Table 6. Comparison of the capability to resist SPN and cropping attacks.

Noise Type	Attack Intensity	PSNR			
		Ref. [14]	Ref. [31]	Ref. [41]	Ours
SPN	0.0001	33.44	31.56	28.18	28.01
	0.0003	33.26	30.22	28.18	27.77
	0.0005	33.02	30.02	28.17	27.58
Cropping	32×32	24.92	27.21	30.18	30.35
	48×48	20.13	23.96	29.01	30.10

Table 7. Running time (Unit: s).

Algorithm	Encryption	Embedding	Extraction	Reconstruction
Ref. [41]	0.1262	0.1995	0.0833	0.4374
ours	0.1352	0.1024	0.0811	0.1578

In Table 6, we compare some of the latest encryption schemes with our scheme in terms of robustness. It can be seen that our scheme had stronger resistance to attacks under the same conditions. In Table 7, we compare another scheme with our scheme in terms of running time. Obviously, our scheme had a shorter running time and higher efficiency. In a word, our scheme was robust and efficient.

6. Conclusions

In this paper, a tri-bistable locally active memristor model was proposed, and its nonlinear characteristics were studied, including time-domain waveforms, three coexisting pinched hysteresis loops, Power-Off Plot, and DC V-I Curve. Then, a fractional-order chaotic system was built based on the proposed memristor. Research results showed that the system could generate abundant chaotic dynamic behaviors, including coexisting attractors with four scrolls, two channels, and complexity. Finally, a visually secure image

encryption scheme was proposed, which consisted of two parts: the pre-encryption process based on DNA coding, and the embedding process based on IWT. In the pre-encryption process, bit scrambling, bit transformation, and DNA coding were used and a secret image was obtained, which protected the data security of the plain image. In the embedding process, the carrier image was decomposed by IWT, the secret image was embedded into the carrier image by bits, and finally a visually meaningful cipher image was obtained, which protected the visual security of the plain image. The simulation experiment and attack analysis showed that the proposed visually secure image encryption scheme could resist statistical analysis attack, chosen-plain attack, and differential attack.

Author Contributions: Validation, T.Z., D.W. and J.Z. (Jinzhong Zhang); writing—original draft preparation, Y.W.; writing—review and editing, J.Z. (Jian Zhou). All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Research Foundation for Advanced Talents of West Anhui University (grant number WGKQ2021050) and the Anhui Provincial Natural Science Foundation (grant number 2008085MA20).

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Chua, L.O. Memristor-The missing circuit element. *IEEE Trans. Circuit Theory* **1971**, *18*, 159–182. [CrossRef]
2. Strukov, D.B.; Snider, G.S.; Stewart, D.R.; Williams, R.S. The missing memristor found. *Nature* **2008**, *453*, 80–83. [CrossRef]
3. Zhong, Y.; Tang, J.; Li, X.; Gao, B.; Qian, H.; Wu, H. Dynamic memristor-based reservoir computing for high-efficiency temporal signal processing. *Nat. Commun.* **2021**, *12*, 408. [CrossRef]
4. Gul, F. Addressing the sneak-path problem in crossbar RRAM devices using memristor-based one schottky diode-one resistor array. *Results Phys.* **2019**, *12*, 1091–1096. [CrossRef]
5. Bao, H.; Zhu, D.; Liu, W.; Xu, Q.; Chen, M.; Bao, B. Memristor synapse-based morris-lecar model: Bifurcation analyses and FPGA-based validations for periodic and chaotic bursting/spiking firings. *Int. J. Bifurc. Chaos* **2020**, *30*, 2050045. [CrossRef]
6. Zhang, S.; Zheng, J.; Wang, X.; Zeng, Z.; He, S. Initial offset boosting coexisting attractors in memristive multi-double-scroll hopfield neural network. *Nonlinear Dyn.* **2020**, *102*, 2821–2841. [CrossRef]
7. Zhou, P.; Yao, Z.; Ma, J.; Zhu, Z. A piezoelectric sensing neuron and resonance synchronization between auditory neurons under stimulus. *Chaos Solit. Fract.* **2021**, *145*, 110751. [CrossRef]
8. Wu, X.; He, S.; Tan, W.; Wang, H. From Memristor-Modeled Jerk System to the Nonlinear Systems with Memristor. *Symmetry* **2022**, *14*, 659. [CrossRef]
9. Lei, T.; Zhou, Y.; Fu, H.; Huang, L.; Zang, H. Multistability Dynamics Analysis and Digital Circuit Implementation of Entanglement-Chaos Symmetrical Memristive System. *Symmetry* **2022**, *14*, 2586. [CrossRef]
10. Yang, B.; Wang, Z.; Tian, H.; Liu, J. Symplectic Dynamics and Simultaneous Resonance Analysis of Memristor Circuit Based on Its van der Pol Oscillator. *Symmetry* **2022**, *14*, 1251. [CrossRef]
11. Dai, W.; Xu, X.; Song, X.; Li, G. Audio Encryption Algorithm Based on Chen Memristor Chaotic System. *Symmetry* **2022**, *14*, 17. [CrossRef]
12. Rajagopal, K.; Kacar, S.; Wei, Z.; Duraisamy, P.; Kifle, T.; Karthikeyan, A. Dynamical investigation and chaotic associated behaviors of memristor chua's circuit with a non-ideal voltage-controlled memristor and its application to voice encryption. *AEU-Int. J. Electron. Commun.* **2019**, *107*, 183–191. [CrossRef]
13. Chen, J.; Yan, D.-W.; Duan, S.-K.; Wang, L.-D. Memristor-based hyper-chaotic circuit for image encryption. *Chin. Phys. B* **2020**, *29*, 110504. [CrossRef]
14. Zhu, L.; Jiang, D.; Ni, J.; Wang, X.; Rong, X.; Ahmad, M. A visually secure image encryption scheme using adaptive thresholding sparsification compression sensing model and newly-designed memristive chaotic map. *Inf. Sci.* **2022**, *607*, 1001–1022. [CrossRef]
15. Tsafack, N.; Iliyasu, A.M.; De Dieu, N.J.; Zeric, N.T.; Kengne, J.; Abd-El-Atty, B.; Belazi, A.; Abd EL-Latif, A.S. A memristive RLC oscillator dynamics applied to image encryption. *J. Info. Sec. App.* **2021**, *61*, 102944. [CrossRef]
16. Chua, L.O. Local activity is the origin of complexity. *Int. J. Bifur. Chaos* **2005**, *15*, 3435–13456. [CrossRef]
17. Mannan, Z.I.; Choi, H.; Kim, H. Chua corsage memristor oscillator via Hopf bifurcation. *Int. J. Bifur. Chaos.* **2016**, *26*, 1630009. [CrossRef]
18. Chua, L.O. If it's pinched it's a memristor. *Semicond. Sci. Technol.* **2014**, *29*, 104001. [CrossRef]
19. Williams, R.S.; Pickett, M.D. *The Art and Science of Constructing a Memristor Model: Memristors and Memristive Systems*; Springer: New York, NY, USA, 2013; pp. 93–104.

20. Messaris, I.; Brown, T.D.; Demirkol, A.S.; Ascoli, A.; Al Chawa, M.M.; Williams, R.S.; Tetzlaff, R.; Chua, L.O. NbO_2-Mott memristor: A circuit-theoretic investigation. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2021**, *68*, 4979–4992. [CrossRef]
21. Mannan, Z.I.; Choi, H.; Rajamani, V.; Kim, H.; Chua, L. Chua corsage memristor: Phase portraits, basin of attraction, and coexisting pinched hysteresis loops. *Int. J. Bifur. Chaos* **2017**, *27*, 1730011. [CrossRef]
22. Mannan, Z.I.; Choi, H.; Rajamani, V. Oscillation with 4-lobe Chua corsage memristor. *IEEE Circuits Syst. Mag.* **2018**, *18*, 14–27. [CrossRef]
23. Mannan, Z.I.; Yang, C.; Adhikari, S.P.; Kim, H. Exact analysis and physical realization of the 6-lobe Chua corsage memristor. *Complexity* **2018**, *2018*, 1–12. [CrossRef]
24. Dong, Y.; Wang, G.; Chen, G.; Shen, Y.; Ying, J. A bistable nonvolatile locally-active memristor and its complex dynamics. *Commun. Nonlinear Sci.* **2020**, *84*, 105203. [CrossRef]
25. Tan, Y.; Wang, C. A simple locally active memristor and its application in HR neurons. *Chaos* **2020**, *30*, 053118. [CrossRef]
26. Xie, W.; Wang, C.; Lin, H. A fractional-order multistable locally active memristor and its chaotic system with transient transition, state jump. *Nonlinear Dyn.* **2021**, *104*, 4523–4541. [CrossRef]
27. Ding, D.; Xiao, H.; Yang, Z.; Luo, H.; Hu, Y.; Zhang, X.; Liu, Y. Coexisting multi-stability of Hopfield neural network based on coupled fractional-order locally active memristor and its application in image encryption. *Nonlinear Dyn.* **2022**, *108*, 4433–4458. [CrossRef]
28. Yang, Z.-L.; Liang, D.; Ding, D.-W.; Hu, Y.-B.; Li, H. Transient transition behaviors of fractional-order simplest chaotic circuit with bi-stable locally-active memristor and its ARM-based implementation. *Chin. Phy. B.* **2021**, *30*, 120515. [CrossRef]
29. Bao, L.; Zhou, Y. Image encryption: Generating visually meaningful encrypted images. *Info. Sci.* **2015**, *324*, 197–207. [CrossRef]
30. Chai, X.; Gan, Z.; Chen, Y.; Zhang, Y. A visually secure image encryption scheme based on compressive sensing. *Signal Process.* **2017**, *134*, 35–51. [CrossRef]
31. Wang, H.; Xiao, D.; Li, M.; Xiang, Y.; Li, X. A visually secure image encryption scheme based on parallel compressive sensing. *Signal Process.* **2019**, *155*, 218–232. [CrossRef]
32. Ping, P.; Yang, X.; Zhang, X.; Mao, Y.; Khalid, H. Generating visually secure encrypted images by partial block pairing-substitution and semi-tensor product compressed sensing. *Digit. Signal Process.* **2022**, *120*, 103263. [CrossRef]
33. Li, R.H.; Dong, E.Z.; Tong, J.G. A Novel Multiscroll Memristive Hopfield Neural Network. *Int. J. Bifur. Chaos.* **2022**, *32*, 2250130. [CrossRef]
34. Ying, J.; Liang, Y.; Wang, J.; Dong, Y.; Wang, G.; Gu, M. A tristable locally-active memristor and its complex dynamics. *Chaos Solitons Fractals* **2022**, *160*, 112241. [CrossRef]
35. Carpinteri, A.; Mainardi, F. *Fractals and Fractional Calculus in Continuum Mechanics*; International Centre for Mechanical Sciences; Springer: Berlin/Heidelberg, Germany, 1997; Volume 378.
36. Bremen, H.F.V.; Udwadia, F.E.; Proskurowski, W. An efficient QR based method for the computation of Lyapunov exponents. *Physica D* **1997**, *101*, 1–16. [CrossRef]
37. Shen, E.H.; Cai, Z.J.; Gu, F.J. Mathematical foundation of a new complexity measure. *Appl. Math. Mech.* **2005**, *26*, 1188–1196.
38. Erkan, U.; Toktas, A.; Enginoğlu, S.; Akbacak, E.; Thanh, D.N.H. An image encryption scheme based on chaotic logarithmic map and key generation using deep CNN. *Multimed. Tools Appl.* **2022**, *81*, 7365–7391. [CrossRef]
39. 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]
40. Wang, X.Y.; Liu, C.; Jiang, D.H. Visually meaningful image encryption scheme based on new-designed chaotic map and random scrambling diffusion strategy. *Chaos Solitons Fractals* **2022**, *164*, 112625. [CrossRef]
41. Zhu, L.; Song, H.; Zhang, X.; Yan, M.; Zhang, T.; Wang, X.; Xu, J. A robust meaningful image encryption scheme based on block compressive sensing and SVD embedding. *Signal Process.* **2020**, *175*, 107629. [CrossRef]