Research Article

Dynamical System in Chaotic Neurons with Time Delay Self-Feedback and Its Application in Color Image Encryption

Yao-Qun Xu, Xin-Xin Zhen, and Meng Tang

1School of Computer and Information Engineering, Harbin University of Commerce, Harbin 150028, China
2Institute of System Engineering, Harbin University of Commerce, Harbin 150028, China

Correspondence should be addressed to Yao-Qun Xu; xuyaoqun@sina.com

Received 17 December 2021; Revised 5 May 2022; Accepted 18 May 2022; Published 1 July 2022

Copyright © 2022 Yao-Qun Xu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The time delay caused by transmission in neurons is often ignored, but it is demonstrated by theories and practices that time delay is unavoidable. A new chaotic neuron model with time delay self-feedback is proposed based on Chen’s chaotic neuron. The bifurcation diagram and Lyapunov exponential diagram are used to analyze the chaotic characteristics of neurons in the model when they receive the output signals at different times. The experimental results exhibit that it has a rich dynamic behavior. In addition, the randomness of chaotic series generated by chaotic neurons with time delay self-feedback under different conditions is verified. In order to investigate the application of this model in image encryption, an image encryption scheme is proposed. The security analysis of the simulation results shows that the encryption algorithm has an excellent anti-attack ability. Therefore, it is necessary and practical to study chaotic neurons with time delay self-feedback.

1. Introduction

The Hopfield neural network is a typical dynamic neural network with abundant dynamic characteristics [1–3]. It is considered as the basic network structure for studying a chaotic neural network, mainly because nonlinear dynamic behavior is introduced into the Hopfield neural network, and chaos is the typical embodiment of this behavior. Therefore, many chaotic neural networks are based on the Hopfield neural network. A chaotic neural network effectively solves the problem that the Hopfield neural network is easy to fall into local minimum solution and infeasible solution in the process of optimization [4].

Chen’s chaotic neural network is a typical chaotic neural network [5, 6]. A self-feedback term is introduced into discrete Hopfield neural network. In the initial stage, the ergodic property of chaos is used to optimize rough search in a large range to avoid falling into local minimum [7, 8]. In order to improve the optimization performance of chaotic neural networks, many scholars have proposed innovative research [9–15]. Based on Chen’s chaotic neural network, Xu et al. proposed the wavelet chaotic neural network using the Morlet wavelet function as an excitation function [10]. Ye and Yang presented the Legendre chaotic neural network by combining the polynomial of Legendre function with the Sigmoid function as the excitation function [11, 14]. These chaotic neural network models have rich dynamic characteristics. The study of the chaotic system has attracted the attention of scholars. In [16], the synchronization problem of chaotic systems using integral-type sliding mode control for a category of hyperchaotic systems is considered, controlling the nonlinear systems and chaotic behaviors for achievement of finite-time synchronization. In [17], an adaptive nonsingular terminal sliding mode control method based on obstacle function is proposed to solve the problem of robust stability of disturbed nonlinear systems. However, for the convenience of analysis and application, these research studies do not consider the delay in signal transmission.

Time delay refers to the fact that the effect of signal transport is always slower compared with the time at which a physical event occurs [18–20]. Time delay is universal and inevitable [21, 22]. The time delay can improve the dynamic characteristics of the original system and can be better
applied to the fields such as encryption [23–25] and control [26]. System analysis with time delay has become a research hotspot [27–33]. Some one-dimensional chaotic mappings were studied in [27–29]. A projection neural network with two-time delay is proposed for solving quadratic programming problems subject to linear constraints in [30]. Wang investigated the stability analysis of the fractional-order Hopfield neural networks with time delay [31]. The study shows that the introduction of time delay makes the dynamic behavior of the model more complicated.

A chaotic neuron is the basic unit of the chaotic neural network, so it is the focus of our research. Grasping the chaotic characteristics of a single chaotic neuron can also provide a necessary premise for understanding the corresponding chaotic neural network [34]. There is a delay in signal transmission within the neuron. Input signals and output signals closely cooperate and influence each other. Therefore, introducing time delay into chaotic neurons is more suitable for signal transmission in chaotic neurons.

In this paper, a time delay self-feedback chaotic neuron model is proposed by introducing the time delay self-feedback into Chen’s chaotic neuron model. The Lyapunov exponential diagram and bifurcation diagram are used to analyze the chaotic characteristics of time delay self-feedback chaotic neuron considering the input signals and parameter allocation. Test suites such as NIST SP800-22 and TestU01 are used to evaluate the randomness of the sequences generated by the tests.

Currently, the encryption method based on the chaos theory is one of the main streams in the field of image encryption. Chaotic time series generated by time delay self-feedback chaotic neuron are used in image encryption combined with a classical permutation diffusion encryption structure. Simulation experiments are conducted to analyze the performance indexes of the image encryption algorithm.

The structure of the paper is as follows. Section 2 introduces the time delay self-feedback chaotic neuron model, analyzing dynamical behavior and the randomness of time series of time delay self-feedback chaotic neuron under different parameters and different number of neuron connections. Section 3 proposes a color image encryption algorithm based on time series generated by time delay self-feedback chaotic neurons. Section 4 uses the encryption algorithm to carry on the simulation experiment and the security analysis. The conclusion is drawn in Section 5.

2. Chaotic Neuron with Time Delay Self-Feedback

The model of chaotic neuron with time delayed self-feedback is as follows:

\[ x(t) = f(y(t)) = 1/(1 + \exp(-ry(t))), \]

\[ y(t + 1) = ky(t) - z(t) \left( \sum_{i=1}^{m} a_i x(t - i + 1) \right) - h, \]

\[ z(t + 1) = (1 - \beta)z(t), \]

where \( x(t) \) is the output of the neuron at time \( t \), \( y(t) \) is the internal state of the neuron at time \( t \), \( r \) is the steepness parameter of Sigmoid function, \( k \) represents the ability of neuron to retain internal state ranging from \([0, 1]\), \( m \) is the number of output signal received, \( a_i \) is the weight of output signals, \( z(t) \) is the self-feedback connection weight, and \( \beta \) is the annealing speed parameter.

The hardware implementations of neurons can be performed from discretized equations using specific numerical methods [35–37]. The chaotic neuron with time delay self-feedback model proposed in this paper is a small dynamic system; the structure of chaotic neurons with time delay self-feedback is shown in Figure 1, which is based on Equations (1)–(4). The output of neuron at \( t - m + 1 \) is delayed to \( t \) to provide data for the iteration of neuron at \( t \).

This paper studies the dynamical system of chaotic neurons with time delay self-feedback when \( m = 4 \) the cases in Table 1. Table 1 shows several situations of the chaotic neuron with different time delays and influence weights.

### 2.1. Chaotic Neurons with No Time Delay

2.1.1. Dynamical Behaviour in Chaotic Neurons with No Time Delay

When \( a_1 = 1, a_2 = a_3 = a_4 = 0 \), the single neuron has no time delay, which is Chen’s chaotic neuron. The single neuron model can be described as follows:

\[ x(t) = f(y(t)) = 1/(1 + \exp(-ry(t))), \]

\[ y'(t + 1) = ky(t) - z(t)x(t) - h, \]

\[ z(t + 1) = (1 - \beta)z(t). \]

The state bifurcation diagram and the maximal Lyapunov exponential time evolution diagram can reflect the
chaotic searching ability of the chaotic neuron to some extent [38, 39]. According to the model, the dynamic behavior of chaotic neuron is affected by four parameters. In order to analyze the influence of each parameter on the dynamic behavior of a single neuron, the remaining parameters are fixed and the bifurcation diagrams and Lyapunov exponent diagrams of the chaotic neuron are observed by changing one parameter.

The selection of a parameter is a result of experience and continuous testing. At the beginning, we set the iteration step of the parameter to a larger size so that the parameter can change quickly. The speed of training can be guaranteed by controlling the number of iterations of chaotic neuron with time delayed self-feedback. According to the bifurcation diagram and the maximal Lyapunov exponential diagram, the approximate range of the parameter is determined. Within the rough range of the parameter, the iteration step size of the parameter is reduced and the selected parameter is slightly adjusted. We observe the bifurcation diagram and the maximal Lyapunov exponential diagram and select the optimal parameter solution of the chaotic neuron with time delay self-feedback.

Set the parameters as follows: \( r = 250, z = -0.1, k = 0.6, h = 0.01 \). The bifurcation diagrams and Lyapunov exponent diagrams of one parameter are drawn. The simulation results of parameters \( r \) and \( k \) varied are shown in Figures 2 and 3. For parameters \( z \) and \( h \), as shown in Figure 4 and Figure 5. The simulation results indicate that this neuron model has transient chaotic dynamic behavior. The bifurcation diagrams have an obvious correspondence with the Lyapunov exponent diagrams. When the Lyapunov exponent is greater than 0, the neuron is in a chaotic state.

2.1.2. Sensitivity to Initial Condition. An important feature of chaotic neural networks is that they are sensitive to initial conditions. That is, small differences in initial conditions will generate different sequences [40]. In order to detect the

Figure 2: Single-parameter bifurcation diagram (a) and the maximal Lyapunov exponent diagram (b) of \( y \) variation.

Figure 3: Single-parameter bifurcation diagram (a) and the maximal Lyapunov exponent diagram (b) of \( k \) variation.
sensitivity of chaotic neurons with no time delay self-feedback to the initial value, Equation (8) is made to process the output of \( x \), with the purpose of expanding the output of \( x \) to \([0, 1]\) [41]. We set the parameters as follows: \( r = 250 \), \( z = -0.1 \), \( k = 0.5 \), \( h = 0.001 \). We make a tiny change to each parameter or initial value to observe the output value of \( x \) under different circumstances. The simulation results are shown in Figure 6. It can be seen that after 15 iterations, the output result of \( x \) has a great change.

\[
x(t) = x(t) \times 2^{14} - \text{floor}(x(t) \times 2^{14}). \tag{8}
\]

2.1.3. NIST SP800-22 Test. Random numbers play an important role in the field of image encryption. SP800-22 is used to detect the randomness of time series generated by chaotic neuron with no time delay self-feedback. SP800 is a series of guidelines on information security issued by National Institute of Standards and Technology (NIST) [42]. The time series generated by chaotic neurons with no time delay self-feedback processed by Equations (9) and (10) after Equation (8).

\[
s_i = (10^{10} \times x_i) \text{mod} 1, \tag{9}
\]

\[
y_i = \begin{cases} 0, & 0 \leq s_i < 0.5, \\ 1, & 0.5 \leq s_i < 1, \end{cases} \tag{10}
\]

NIST conducts 15 randomized tests and outputs a \( p \) value at the end of the tests. When \( p \) value is greater than 0.01, it indicates that the randomness detection of the data set is successful. This article tests \( 10^8 \) bits of binary data, and the simulation results are shown in Table 2.

2.1.4. TestU01 Test. TestU01 test suite provides a set of utilities for detecting random number generators [43]. The test standard of this suite is stricter than the NIST test. In the experiment, three predefined algorithms, Rabbit, Alphabit,
Figure 6: Continued.
and BlockAlphabet, are used to evaluate the randomness of chaotic sequences generated by chaotic neurons with no time delay self-feedback. Each algorithm is tested with sequences of 10^7 bit. The simulation results are shown in Table 3. This shows that the random numbers generated by the neuron model have good randomness.

### 2.2. Chaotic Neurons with One-Time Delay Self-Feedback

#### 2.2.1. Dynamical Behaviour in Chaotic Neurons with One-Time Delay Self-Feedback

When \( a_2 = 1, \ a_3 = a_4 = 0 \), the output of chaotic neuron time \((t - 1)\) is delayed to time \(t\), and the single neuron model can be described as follows.

\[
x(t) = f(y(t)) = 1/(1 + \exp(-ry(t)), \quad (11)
\]

\[
y(t + 1) = ky(t) - z(t)x(t - 1) - h, \quad (12)
\]

\[
z(t + 1) = (1 - \beta)z(t). \quad (13)
\]

We set the parameters as follows: \( r = 250, \ z = -0.1, \ k = 0.6, \ h = 0.01 \). The bifurcation graph and Lyapunov exponential graph of one parameter are drawn. The simulation results of parameters \( r \) and \( k \) are shown in Figures 7 and 8. For parameters \( z \) and \( h \), it is shown in Figures 9 and 10.

Compared to the dynamical behavior of chaotic neuron with no time delay, the chaotic interval of chaotic neuron with one time delay self-feedback is not continuous, \( x \) has a wider range of output values in parameters \( k \) and \( z \).

#### 2.2.2. Sensitivity to Initial Condition

We analyze the sensitivity of the model to the initial values. We set the parameters as follows: \( r = 125, \ z = -0.15, \ k = 0.9, \ h = 0.05 \). It can be seen from Figure 11 that after 12 iterations, the output result of \( x \) has a great change.

The sensitivity of the model to the initial value is slightly improved. Compared the chaotic neuron with no time delay, the chaotic neuron with one-time delay self-feedback has fewer iterations when the same changes are made to the initial value.
2.2.3. NIST SP800-22 Test. The time series generated by the chaotic neuron with one-time delay self-feedback processed by Equations (9) and (10) after Equation (8). This article tests 100000000 bits of binary data and the results are shown in Table 4. The results show that all p-values were greater than 0.01, indicating that the randomness test of the data set was successful. Compared with the chaotic neuron with no time delay, the randomness of time series generated by chaotic neurons with one-time delay self-feedback is improve.

2.2.4. TestU01 Test. TestU01 test suite is the most rigorous test to evaluate the statistical properties of generated sequence. Three predefined algorithms, Rabbit, Alphabit, and BlockAlphabit, are used to evaluate the randomness of the sequence generated by the chaotic neuron with one-time delay self-feedback, the sequence length is 10^7 bit. The results are shown in Table 5. We prove that the generated binary sequence is random by performing TestU01 statistical test.

2.3. Chaotic Neurons with Two-Time Delay Self-Feedback

2.3.1. Dynamical Behaviour in Chaotic Neurons with Two-Time Delay Self-Feedback. When \(a_1 = 1, a_2 = a_3 = 1/2, a_4 = 0\), we set the output of chaotic neuron time \((t-1)\) and \((t-2)\) are delayed to time \(t\) and the weight is 1/2. The chaotic neuron with two-time delay self-feedback can be described as follows:

\[
x(t) = f(y(t)) = 1/(1 + \exp(-ry(t)),
\]

\[
y(t + 1) = ky(t) - z(t)\left(\frac{1}{2}x(t - 1) + \frac{1}{2}x(t - 2)\right) - h,
\]

\[
z(t + 1) = (1 - \beta)z(t).
\]

We set the parameters as follows: \(r = 250, z = -0.1, k = 0.6, h = 0.01\). The bifurcation graph and Lyapunov exponential graph of one parameter are drawn. The simulation results of parameters \(r\) and \(k\) are shown in Figures 12 and 13. For parameters \(z\) and \(h\), it is shown in Figures 14 and 15.

The dynamics of the network is changed by increasing the duration of the time delay and decreasing the proportion of each delayed neuron in the output signal. Compared with the chaotic neuron with one-time delay self-feedback, the chaotic intervals of \(r, k, h\) parameters increase obviously and the interval of Lyapunov exponent greater than 0 expands.

2.3.2. Sensitivity to Initial Condition. We analyze the sensitivity of the model to the initial values. We set the parameters as follows: \(r = 150, z = -0.02, k = 0.17, h = 0.001\). It can be seen from Figure 16 that after 10 iterations, the output result of \(x\) has a great change.

2.3.3. NIST SP800-22 Test. The time series generated by chaotic neuron with two-time delay self-feedback processed by Equations (9) and (10) after Equation (8). This article tests 1000000000 bits of binary data and the results are shown in Table 6. Compared with chaotic neuron with no time delay, the randomness of time series generated by the chaotic neuron with two-time delay self-feedback is improved.

2.3.4. TestU01 Test. TestU01 test is performed on sequence generated by the chaotic neuron with two-time delay self-feedback. The results are shown in Table 7. We prove that the generated binary sequence is random by performing TestU01 statistical test.

2.4. Chaotic Neurons with Three-Time Delay Self-Feedback

2.4.1. Dynamical Behaviour in Chaotic Neurons with Three-Time Delay Self-Feedback. When \(a_1 = 0, a_2 = a_3 = a_4 = 1/3\), we set the output of chaotic neuron at time \((t-1), (t-2),\) and \((t-3)\) are delayed to time \(t\) and the weight is 1/3. The chaotic neuron with three-time delay self-feedback can be described as follows:

\[
x(t) = f(y(t)) = 1/(1 + \exp(-ry(t)),
\]

\[
y(t + 1) = ky(t) - z(t)\left(\frac{1}{3}x(t - 1) + \frac{1}{3}x(t - 2) + \frac{1}{3}x(t - 3)\right) - h,
\]

\[
z(t + 1) = (1 - \beta)z(t).
\]

We set the parameters as follows: \(r = 250, z = -0.1, k = 0.6, h = 0.01\). The bifurcation graph and Lyapunov exponential graph of one parameter are drawn. The simulation results of parameters \(r\) and \(k\) varied are shown in Figures 17 and 18. For parameters \(z\) and \(h\), it is shown in Figure 19 and 20. It can be seen from Figure 17-20 that the maximum value of Lyapunov is increased compared with chaotic neuron with no time delay.

2.4.2. Sensitivity to Initial Condition. We analyze the sensitivity of the model to the initial values. We set the parameters as follows: \(r = 150, z = -0.02, k = 0.17, h = 0.001\). It can be seen from Figure 21 that after 11 iterations, the output result of \(x\) has a great change. The sensitivity of the model to the initial value is slightly improved.
Figure 7: Single-parameter bifurcation diagram (a) and the maximal Lyapunov exponent diagram (b) of γ variation.

Figure 8: Single-parameter bifurcation diagram (a) and the maximal Lyapunov exponent diagram (b) of k variation.

Figure 9: Single-parameter bifurcation diagram (a) and the maximal Lyapunov exponent diagram (b) of z variation.
2.4.3. NIST SP800-22 Test. The time series generated by the chaotic neuron with three-time delay self-feedback processed by Equations (9) and (10) after Equation (8). This article tests 1,000,000,000 bits of binary data and the results are shown in Table 8. Compared with the chaotic neuron with no time delay, the randomness of time series generated by chaotic neurons with three-time delay self-feedback is improved.

2.4.4. TestU01 Test. TestU01 test is performed on a sequence generated by the chaotic neuron with two-time delay self-feedback. The results are shown in Table 9. We prove that the generated binary sequence is random by performing TestU01 statistical test.

2.5. Chaotic Neurons with Three-Time Delay and Four-Output

2.5.1. Dynamical Behaviour in Chaotic Neurons with Three-Time Delay and Four-Output. When \( a_1 = a_2 = a_3 = 1/4 \), we set the output of the chaotic neuron at time \( t - 1 \), \( t - 2 \), \( t - 3 \) are delayed to time \( t \), including the output of time \( t \), with the weight is 1/4. The chaotic neuron with three-time delay and four-output can be described as follows:

\[
x(t) = f(y(t)) = 1/(1 + \exp(-ry(t))),
\]

\[
y(t + 1) = ky(t) - z(t) \left[ \frac{1}{4} (x(t) + x(t - 1) + x(t - 2) + x(t - 3)) \right] - h,
\]

\[
z(t + 1) = (1 - \beta)z(t).
\]

Table 4: Test result of chaotic neuron with one-time delay self-feedback.

<table>
<thead>
<tr>
<th>Number</th>
<th>Subset</th>
<th>( p ) value</th>
<th>Proportion</th>
<th>Test result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Frequency</td>
<td>0.779188</td>
<td>99/100</td>
<td>Random</td>
</tr>
<tr>
<td>2</td>
<td>Block frequency</td>
<td>0.719747</td>
<td>98/100</td>
<td>Random</td>
</tr>
<tr>
<td>3</td>
<td>Cumulative sums</td>
<td>0.236810</td>
<td>99/100</td>
<td>Random</td>
</tr>
<tr>
<td>4</td>
<td>Runs</td>
<td>0.383827</td>
<td>99/100</td>
<td>Random</td>
</tr>
<tr>
<td>5</td>
<td>Longest run</td>
<td>0.202268</td>
<td>99/100</td>
<td>Random</td>
</tr>
<tr>
<td>6</td>
<td>Rank</td>
<td>0.419021</td>
<td>99/100</td>
<td>Random</td>
</tr>
<tr>
<td>7</td>
<td>FFT</td>
<td>0.678686</td>
<td>99/100</td>
<td>Random</td>
</tr>
<tr>
<td>8</td>
<td>Nonoverlapping template</td>
<td>0.946308</td>
<td>99/100</td>
<td>Random</td>
</tr>
<tr>
<td>9</td>
<td>Overlapping template</td>
<td>0.935716</td>
<td>100/100</td>
<td>Random</td>
</tr>
<tr>
<td>10</td>
<td>Universal Approximate entropy</td>
<td>0.554020</td>
<td>99/100</td>
<td>Random</td>
</tr>
<tr>
<td>11</td>
<td>Approximate entropy</td>
<td>0.224821</td>
<td>99/100</td>
<td>Random</td>
</tr>
<tr>
<td>12</td>
<td>Random excursions</td>
<td>0.401199</td>
<td>100/100</td>
<td>Random</td>
</tr>
<tr>
<td>13</td>
<td>Random excursions variant</td>
<td>0.249284</td>
<td>100/100</td>
<td>Random</td>
</tr>
<tr>
<td>14</td>
<td>Serial</td>
<td>0.678686</td>
<td>100/100</td>
<td>Random</td>
</tr>
<tr>
<td>15</td>
<td>Linear complexity</td>
<td>0.637119</td>
<td>100/100</td>
<td>Random</td>
</tr>
</tbody>
</table>

Table 5: Test result of TestU01.

<table>
<thead>
<tr>
<th>Sequence length</th>
<th>Test result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rabbit 10⁷</td>
<td>Pass</td>
</tr>
<tr>
<td>Alphabit 10⁷</td>
<td>Pass</td>
</tr>
<tr>
<td>BlockAlphabit 10⁷</td>
<td>Pass</td>
</tr>
</tbody>
</table>

Figure 10: Single-parameter bifurcation diagram (a) and the maximal Lyapunov exponent diagram (b) of \( h \) variation.
Figure 11: Sensitivity to initial condition of (a) varied initial value. (b) Varied \( r \). (c) Varied \( k \). (d) Varied \( z \). (e) Varied \( h \).
Figure 12: Single-parameter bifurcation diagram (a) and the maximal Lyapunov exponent diagram (b) of $y$ variation.

Figure 13: Single-parameter bifurcation diagram (a) and the maximal Lyapunov exponent diagram (b) of $k$ variation.

Figure 14: Single-parameter bifurcation diagram (a) and the maximal Lyapunov exponent diagram (b) of $z$ variation.
We set the parameters as follows: \( r = 250, z = -0.1, k = 0.8, h = 0.01 \). The bifurcation graph and Lyapunov exponential graph of one parameter are drawn. The simulation results of parameters \( r \) and \( k \) are shown in Figures 22 and 23. The dynamic behavior of the neuron shows a distinct spiral shape, and the chaos range is obviously expanded in parameters \( z \) and \( h \) (Figures 24 and 25).

### 2.5.2. Sensitivity to Initial Condition

We analyze the sensitivity of the model to the initial values. We set the parameters as follows: \( r = 150, z = -0.02, k = 0.17, h = 0.001 \). It can be seen from Figure 26 that after 8 iterations, the output result of \( x \) has a great change. The sensitivity to initial values is significantly improved. Therefore, the chaotic neuron with three-time delay and four-output is sensitive to initial values and can ensure the security of encryption.

### 2.5.3. NIST SP800-22 Test

The time series generated by the chaotic neuron with three-time delay and four-output processed by Equations (9) and (10) after Equation (8). This article tests 100000000 bits of binary data and the results are shown in Table 10. The results show that the time series generated by this model has good randomness and can guarantee the security of encryption.

### 2.5.4. TestU01 Test

TestU01 test is performed on a sequence generated by the chaotic neuron with two-time delay self-feedback. The results are shown in Table 11. We prove that the generated binary sequence is random by performing TestU01 statistical test.

<table>
<thead>
<tr>
<th>Test number</th>
<th>Subset</th>
<th>p-Value</th>
<th>Proportion</th>
<th>Test result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Frequency</td>
<td>0.616305</td>
<td>99/100</td>
<td>Random</td>
</tr>
<tr>
<td>2</td>
<td>Block frequency</td>
<td>0.867692</td>
<td>98/100</td>
<td>Random</td>
</tr>
<tr>
<td>3</td>
<td>Cumulative sums</td>
<td>0.494392</td>
<td>100/100</td>
<td>Random</td>
</tr>
<tr>
<td>4</td>
<td>Runs</td>
<td>0.964295</td>
<td>100/100</td>
<td>Random</td>
</tr>
<tr>
<td>5</td>
<td>Longest run</td>
<td>0.759756</td>
<td>98/100</td>
<td>Random</td>
</tr>
<tr>
<td>6</td>
<td>Rank</td>
<td>0.883171</td>
<td>99/100</td>
<td>Random</td>
</tr>
<tr>
<td>7</td>
<td>FFT</td>
<td>0.779188</td>
<td>98/100</td>
<td>Random</td>
</tr>
<tr>
<td>8</td>
<td>Nonoverlapping template</td>
<td>0.964295</td>
<td>100/100</td>
<td>Random</td>
</tr>
<tr>
<td>9</td>
<td>Overlapping template</td>
<td>0.191687</td>
<td>100/100</td>
<td>Random</td>
</tr>
<tr>
<td>10</td>
<td>Universal</td>
<td>0.025193</td>
<td>98/100</td>
<td>Random</td>
</tr>
<tr>
<td>11</td>
<td>Approximate entropy</td>
<td>0.401199</td>
<td>99/100</td>
<td>Random</td>
</tr>
<tr>
<td>12</td>
<td>Random excursions</td>
<td>0.654467</td>
<td>100/100</td>
<td>Random</td>
</tr>
<tr>
<td>13</td>
<td>Random excursions variant</td>
<td>0.848588</td>
<td>100/100</td>
<td>Random</td>
</tr>
<tr>
<td>14</td>
<td>Serial</td>
<td>0.851383</td>
<td>99/100</td>
<td>Random</td>
</tr>
<tr>
<td>15</td>
<td>Linear complexity</td>
<td>0.224821</td>
<td>100/100</td>
<td>Random</td>
</tr>
</tbody>
</table>

**Table 6: Test result of the chaotic neuron with two-time delay self-feedback.**

**Table 7: Test result of TestU01.**

<table>
<thead>
<tr>
<th>Sequence length</th>
<th>Test result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rabbit</td>
<td>10^7</td>
</tr>
<tr>
<td>Alphabit</td>
<td>10^7</td>
</tr>
<tr>
<td>BlockAlphabit</td>
<td>10^7</td>
</tr>
</tbody>
</table>

**Figure 15:** Single-parameter bifurcation diagram (a) and the maximal Lyapunov exponent diagram (b) of \( h \) variation.
Figure 16: Sensitivity to initial condition of (a) varied initial value. (b) Varied $r$. (c) Varied $k$. (d) Varied $z$. (e) Varied $h$. 
Figure 17: Single-parameter bifurcation diagram (a) and the maximal Lyapunov exponent diagram (b) of $y$ variation.

Figure 18: Single-parameter bifurcation diagram (a) and the maximal Lyapunov exponent diagram (b) of $k$ variation.

Figure 19: Single-parameter bifurcation diagram (a) and the maximal Lyapunov exponent diagram (b) of $z$ variation.
3. Encryption Process

This paper proposes a color encryption algorithm based on the chaotic neuron with one-time delay self-feedback and the chaotic neuron with three-time delay and four output. The process is as follows:

Step 1: Read the original color image $A$, with the size of $M \times N \times 3M$ and $N$ are the row and column of image $A$, respectively.

Step 2: Image preprocessing, divided $A$ into three individual gray images of $R$, $G$ and $B$ channels, respectively, possessing the same size of $M \times N$.

Step 3: Equation (21) iterates $M$ and $N$ times separately to produce two chaotic sequences $\{x_M\}$ and $\{y_N\}$ with initial value, respectively.

Step 4: The two sequences are processed as Equations (23) and (24) to get new sequences $A$ and $B$, ranging from $(0, 255)$.

$$a = \text{floor}(x,M) + 1,$$

$$b = \text{floor}(y,N) + 1.$$  

Step 5: These two sequences are used as the row address and column address of the permutating matrix, which are used to replace the row and column positions of image pixels. The permutation process is shown in Figure 27.

Step 6: Equation (12) iterates $M \times N$ times to produce two chaotic sequences $\{x_{M\times N}\}$ and $\{y_{M\times N}\}$.

Step 7: Take a point in the permutated image in order. If the ordinal number of the point is odd, the secret key is
Figure 21: Sensitivity to initial condition of (a) varied initial value. (b) Varied $r$. (c) Varied $k$. (d) Varied $z$. (e) Varied $h$. 
constructed from \( \{x_{M \times N}\} \) with Equation (25); if the ordinal number of the point is even, the secret key is constructed from \( \{y_{M \times N}\} \) with Equation (26).

\[
key(n) = \text{mod}(\text{floor}(x(n) \times 10^{15}), 256), \tag{25}
\]

\[
key(n) = \text{mod}(\text{floor}(y(n) \times 10^{15}), 256). \tag{26}
\]

Step 8: Xor the pixel gray value \( R(i, j) \) in the permuted image and the secret key generated by Step 6 to obtain the encrypted pixel value \( R' (i, j) \).

Step 9: Repeat until all the pixels have been encrypted to obtain the encrypted \( R \) channel image.

Step 10: The same operation is performed for \( G(i, j) \), \( B(i, j) \) to obtain the encrypted image \( G'(i, j) \) and \( B'(i, j) \).

Step 11: Combine the three encrypted images to produce a colored encrypted image with the size of \( M \times N \times 3 \).

The encryption process is shown in Figure 28. The decryption process is the reverse of the encryption process. The original image can be restored by the reverse operation of the encryption process under the correct secret key.

4. Security Analysis

4.1. Simulation Result. In the experiment, three different images are used for simulation under the environment of MATLABR2018b. The obtained encrypted images are shown in Figure 29. As can be seen from Figure 29, after the encryption process, the outline of the plain images cannot be
Figure 24: Single-parameter bifurcation diagram (a) and the maximal Lyapunov exponent diagram (b) of $z$ variation.

Figure 25: Single-parameter bifurcation diagram (a) and the maximal Lyapunov exponent diagram (b) of $h$ variation.

Table 10: Test result of chaotic neuron with three-time delay and four-output.

<table>
<thead>
<tr>
<th>Test number</th>
<th>Subset</th>
<th>p-value</th>
<th>Proportion</th>
<th>Test result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Frequency</td>
<td>0.985639</td>
<td>100/100</td>
<td>Random</td>
</tr>
<tr>
<td>2</td>
<td>Block frequency</td>
<td>0.816537</td>
<td>99/100</td>
<td>Random</td>
</tr>
<tr>
<td>3</td>
<td>Cumulative sums</td>
<td>0.976801</td>
<td>100/100</td>
<td>Random</td>
</tr>
<tr>
<td>4</td>
<td>Runs</td>
<td>0.759756</td>
<td>100/100</td>
<td>Random</td>
</tr>
<tr>
<td>5</td>
<td>Longest run</td>
<td>0.554420</td>
<td>98/100</td>
<td>Random</td>
</tr>
<tr>
<td>6</td>
<td>Rank</td>
<td>0.924076</td>
<td>100/100</td>
<td>Random</td>
</tr>
<tr>
<td>7</td>
<td>FFT</td>
<td>0.867692</td>
<td>99/100</td>
<td>Random</td>
</tr>
<tr>
<td>8</td>
<td>Nonoverlapping template</td>
<td>0.834308</td>
<td>100/100</td>
<td>Random</td>
</tr>
<tr>
<td>9</td>
<td>Overlapping template</td>
<td>0.779188</td>
<td>100/100</td>
<td>Random</td>
</tr>
<tr>
<td>10</td>
<td>Universal</td>
<td>0.616305</td>
<td>100/100</td>
<td>Random</td>
</tr>
<tr>
<td>11</td>
<td>Approximate entropy</td>
<td>0.224821</td>
<td>98/100</td>
<td>Random</td>
</tr>
<tr>
<td>12</td>
<td>Random excursions</td>
<td>0.706149</td>
<td>100/100</td>
<td>Random</td>
</tr>
<tr>
<td>13</td>
<td>Random excursions variant</td>
<td>0.862344</td>
<td>100/100</td>
<td>Random</td>
</tr>
<tr>
<td>14</td>
<td>Serial</td>
<td>0.719747</td>
<td>99/100</td>
<td>Random</td>
</tr>
<tr>
<td>15</td>
<td>Linear complexity</td>
<td>0.678686</td>
<td>100/100</td>
<td>Random</td>
</tr>
</tbody>
</table>
Figure 26: Sensitivity to initial condition of (a) varied initial value. (b) Varied $r$. (c) Varied $k$. (d) Varied $z$. (e) Varied $h$. 
seen from the final encrypted images, which achieve the goal of encryption.

4.2. Information Entropy. The uncertainty of digital images is analyzed by the information entropy [44]. The larger the information entropy is, the more uniform the pixel value distribution of the image is, and the stronger the anti-attack ability of the image is. The calculation formula of information entropy is as follows.

\[
H(x) = - \sum_{i=0}^{n} p(x_i) \log_2 p(x_i).
\]

Among them, \(n\) represents the gray level of an image and \(p(x_i)\) represents the probability of the gray value \(A\). For a completely random image with a gray level of 256, the theoretical value of \(H(x)\) is 8. The closer the information entropy is to the theoretical value; the less useful information is Table 12 expressed in the image.

Table 12 records the information entropy of plain images and encrypted images and compares it with other algorithms. The results show that the image encrypted by the scheme proposed in this paper can better conceal information and resist statistical analysis.

4.3. Histogram Analysis. Image histogram shows the distribution of pixel values in the image, the abscissa represents the pixel value, and the ordinate represents the number of times each pixel value appears in the image [51, 52]. The histogram of the encrypted image should be uniformly distributed, meaning that the histogram of the encrypted image has little statistical information about the pixel distribution of the original image.

Figure 30 shows the histogram of the RGB channel of plaintext images is ups and downs, while the histogram distribution of the RGB channel of encrypted images is relatively flat, and the distribution of pixel values is uniform. Therefore, this encryption system can resist statistical attacks.

4.4. Correlation Analysis. The correlation value of adjacent pixels can reflect the degree of correlation between adjacent pixels in images. The higher the correlation between adjacent pixels, the easier it is to obtain image information. Therefore, to resist statistical attacks, the correlation between adjacent pixels must be reduced [53, 54]. The correlation calculation formula is as follows.

\[
\begin{align*}
E(x) &= \frac{1}{N} \sum_{i=1}^{N} x_i, \\
D(x) &= \frac{1}{N} \sum_{i=1}^{N} (x_i - E(x_i))^2, \\
\text{cov}(x, y) &= \frac{1}{N} \sum_{i=1}^{N} (x_i - E(x_i))(y_i - E(y_i)), \\
\rho_{xy} &= \frac{\text{cov}(x, y)}{\sqrt{D(x)} \sqrt{D(y)}}.
\end{align*}
\]

where \(x\) and \(y\) are the gray values of adjacent pixels, and \(\rho_{xy}\) represents the correlation value between adjacent pixels. The correlation diagram among adjacent pixels at horizontal, vertical, and diagonal directions of G channel in Lena image is shown in Figure 31 and the correlation coefficients according to the direction is shown in Table 13.
**Figure 28:** Encryption process.

**Figure 29:** Simulation results. (a) Plain image of Lena. (b) Encrypted image of Lena. (c) Decrypted image of Lena. (d) Plain image of Pepper. (e) Encrypted image of Pepper. (f) Decrypted image of Pepper. (g) Plain image of Baboon. (h) Encrypted image of Baboon. (i) Decrypted image of Baboon.
4.5. Differential Attack. The number of pixels change rate (NPCR) and unified average changing intensity (UACI) are used to analyze the antidifferential attack capability of the encryption algorithm. The original image is randomly changed by one pixel and the changed image is encrypted. The two encrypted images are compared to analyze the sensitivity of the algorithm. The definitions of NPCR and UACI are as follows [57, 58].

\[
NPCR = \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{D(i, j)}{M \times N} \times 100\%,
\]

\[
UACI_{RGB} = \sum_{i=1}^{M} \sum_{j=1}^{N} \left| \frac{C_1(i, j) - C_2(i, j)}{M \times N \times 255} \right| \times 100\%,
\]

\[
D(i, j) = \begin{cases} 
0 & C_1(i, j) = C_2(i, j), \\
1 & C_1(i, j) \neq C_2(i, j), 
\end{cases}
\]

where \( M \) and \( N \) are the width and length of the image, respectively. \( C_1 \) and \( C_2 \) are encrypted images obtained by encrypting two original images with only one pixel difference. The theoretical values of NPCR and UACI were 99.6094% and 33.4635%, respectively. It can be seen from Table 14 that the NPCR and UACI of the image are close to the theoretical values, indicating that the algorithm can resist differential attacks well.

4.6. Secret Key Sensitivity. Secret key sensitivity refers to the decryption of the same encrypted image using two secret keys with extremely slight differences [63]. The plain image can only be retrieved with the correct secret key. The secret keys of the algorithm in this paper include \( r, z, k, h \), and \( y \); the initial value of chaotic neuron with one-time delay self-feedback set \( y_1 = 0.20000000000000001 \), and \( y_2 \) = 0.2, respectively. The rest of secret keys remain unchanged, respectively, to decrypt Figure 29(g), and two completely different decrypted images are obtained as shown in Figure 32. Therefore, the encryption algorithm in this paper is very sensitive to the change of secret key; subtle changes in secret key will result in unrecognizable decrypted image.

4.7. Data Loss. In the process of storage and transmission, the encrypted image is vulnerable to intercept and attack, resulting in the loss of information in the encrypted image, so that the decrypted image cannot be obtained [64]. In order to analyze the anticlipping attack ability of the algorithm in this paper, a matrix with the size of \( \beta \times M \times N \) is
selected from the encrypted image to simulate data loss. The pixel value of the matrix is set to 0, and the $\beta$ value is 15%, 25%, and 50%, respectively. The processed image is decrypted as shown in Figure 33.

As can be seen from Figure 33, most of the image information of the original image can be decrypted under different degrees of data loss of the encrypted image. With the increase of data loss, the decrypted image gradually began to blur. However, even if half of the image information is lost, some features of the original image can be recognized, which is enough to prove that the algorithm has a good ability to resist cropping attack.

4.8. Noise Attack. In the transmission process of the encrypted image, it will inevitably be affected by various factors, which will have an impact on the decryption of the image, so the image encryption algorithm must have enough robustness [65, 66]. In the simulation experiment, different degrees of "salt and pepper" noise are added to the encrypted
Table 14: NPCR and UACI of encrypted images.

<table>
<thead>
<tr>
<th>Image</th>
<th>R (NPCR %)</th>
<th>G (NPCR %)</th>
<th>B (NPCR %)</th>
<th>R (UACI %)</th>
<th>G (UACI %)</th>
<th>B (UACI %)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>99.6113</td>
<td>99.609</td>
<td>99.6147</td>
<td>33.4916</td>
<td>33.5064</td>
<td>33.5142</td>
</tr>
<tr>
<td>Peppers</td>
<td>99.6136</td>
<td>99.588</td>
<td>99.6136</td>
<td>33.4601</td>
<td>33.4625</td>
<td>33.4625</td>
</tr>
<tr>
<td>Baboon</td>
<td>99.612</td>
<td>99.6181</td>
<td>99.6136</td>
<td>33.4766</td>
<td>33.45</td>
<td>33.4607</td>
</tr>
</tbody>
</table>

Figure 32: (a) Decrypted image of the correct secret key. (b) Decrypted image of the changed secret key.

Figure 33: Data loss. (a) Encrypted image with 15% data loss. (b) Encrypted image with 25% data loss. (c) Encrypted image with 25% data loss. (d) Encrypted image with 50% data loss. (e) Decrypted image of (a). (f) Decrypted image of (b). (g) Decrypted image of (c). (h) Decrypted image of (d).
image, and then the encrypted image with noise are decrypted. The decryption results are shown in Figure 34.

As can be seen from Figure 34, most of the image information of the original image can be decrypted by the encrypted image under different degrees of noise attack. With the increase of the noise level, the decrypted image gradually becomes blurred, but some features of the original image can still be recognized, which is enough to prove that the algorithm has a good ability of antinoise attack.

4.9. Speed Analysis. The encryption efficiency can reflect the advantages and disadvantages of image encryption algorithm. Table 15 shows the average time to encrypt and decrypt images of different sizes for ten times. We can see that the execution time of encryption algorithm in this paper is short, which has advantages in terms of time.

Time complexity is an important performance index to evaluate encryption algorithms. Assuming that the size of the image is $M \times N$, the time complexity of the algorithm mainly depends on the permutation and diffusion operation of the image. After calculation, it can be concluded that the time complexity of the encryption algorithm used in this paper is $O(18M \times N)$. Table 16 shows the time complexity of the encryption algorithm used in this paper and compares it with other schemes.

4.10. Secret Key Space. Secret key space directly affects the feasibility of exhaustive attack [69]. Large secret key space can increase the difficulty of cracking encryption algorithms and improve security. In the encryption algorithm proposed in this paper, the secret key space is composed of 10 parameters, which are 5 parameters of the chaotic neuron with one-time delay self-feedback and 5 parameters of the chaotic neuron with three-time delay and four-output, respectively. The value range of $y$ is $(0, 500)$, $h \in (0, 0.1)$, $k \in (0, 1)$, $z \in (-1, 0)$, and $y \in (-\infty, +\infty)$. Because double-precision float can accurately represent 14 digits after the decimal point, the secret key space is as follows:

$$K = 5 \times 10^{16} \times 2 \times 10^{14} \times 10^{13} \times 2^6 \times 2 \approx 10^{153}. \quad (32)$$

The secret key space of the algorithm can resist exhaustive attack.

5. Conclusions

In this paper, time delay self-feedback is introduced into Chen’s chaotic neuron and a chaotic neuron model with time delay self-feedback is proposed. The model delays the signal output before $t$ time to $t$ time and analyzes the chaotic
neuron at different delay times. The results show that the sensitivity of the chaotic neuron with time delay self-feedback to initial value and the randomness of generated time series are improved obviously. Time delay can make the chaotic neuron withdraw from chaos and enter chaos periodically, which makes the chaos control process more flexible. In order to verify the practicability of the chaotic neuron model with time delay self-feedback, we design a color image encryption algorithm based on the model. Simulation results show that the color image encryption algorithm has high security and can resist security attacks effectively.

Data Availability

The data and code used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was funded by Nature Science Foundation of Heilongjiang, grant number (LH2021F035).

References


[64] F.-X. Zhao, M. Z. Liu, K. Wang, and H. Zhang, “Color image encryption via Hénon-zigzag map and chaotic restricted


