A Study of An Image Encryption Model Based on Tent-Ushiki Chaotic Fusion

Jun Li[1,*] and Weijun Huang[2]

[1]*School of Intelligent Manufacturing, Shunde Polytechnic, Foshan, Guangdong, 528300, China*
[2]*Shanghai Communications Polytechnic, Shanghai, 200030, China*
E-mail: gdlj1981@sohu.com
[*]*Corresponding Author*

Received 01 November 2023; Accepted 29 November 2023;
Publication 09 April 2024

Abstract

Aiming at the shortcomings of current image encryption, such as simple structure, few parameters and small key space, we propose the Tent-Ushiki chaotic mapping image encryption method based on the improved firefly optimization algorithm. First, a chaotic mapping model based on the fusion of Tent and Ushiki is proposed. Second, for the lack of parameter optimization in chaotic models in general, we introduce the firefly optimization algorithm and optimize the algorithm for the shortcomings in terms of the adaptive step size, the adjustment factor and the inertia weights. Finally, the improved firefly algorithm is used for the fusion of the chaotic parameters of Tent and Ushiki. In the simulation experiments, this paper's algorithm performs well in the statistical analysis and adjacent element correlation of the classical image test and significantly outperforms the comparative algorithms in terms of information entropy and anti-attack, which demonstrates that the algorithm is able to optimize the image encryption effect better.

Keywords: Image encryption, chaotic mapping, firefly algorithm.

Journal of Cyber Security and Mobility, Vol. 13_3, 489–516.
doi: 10.13052/jcsm2245-1439.1337

1 Introduction

The security of network transmission has always been a hot topic of concern. The transmission data in the network easily have the possibility of being intercepted or even intercepted. Therefore, how to ensure the security of data transmitted in the network has been the direction of research by scholars in various countries [1]. Common data encryption algorithms [2–4] can no longer be applied to current digital image encryption. At present, common digital image security protection methods are mainly divided into information hiding techniques [5] and information encryption techniques [6]. The former mainly includes steganography [7] and digital watermarking techniques [8]. The latter adopts the idea of active defense, which transforms plaintext images into ciphertext, thus affecting the protection of the images; these techniques mainly include chaos theory [10, 11], compression awareness [12, 13], frequency domain transformation [14, 15], DNA theory [16, 17] and evolutionary algorithms [18–20]. Based on the research results of existing image encryption algorithms, we propose an optimized Tent-Ushiki chaotic image encryption model based on the firefly algorithm. The main contributions are as follows: (1) we propose a chaotic mapping model based on the fusion of Tent and Ushiki, which can effectively improve the chaotic structure and make the security of chaos effectively improved; (2) to address the problem that the firefly algorithm has a fast convergence speed and easily falls into a local optimum, we carry out three aspects of adaptive step size, adjustment factor and inertia weight optimization, and the simulation results show that the performance of the algorithm is significantly improved; and (3) we address the lack of optimization in parameters of chaotic mapping, which affects the effect of chaotic encryption. We use the improved firefly algorithm for Tent-Ushik parameter optimization, which improves the encryption effect of the chaotic algorithm.

This paper is organized as follows: Section 1 describes the development of image encryption, Section 2 describes the related research work, Section 3 describes the firefly algorithm, the Tent and Ushiki chaotic model, Section 4 describes the process of the firefly algorithm used for the parameters of the Tent-Ushiki model as well as the whole process of encryption, Section 5 conducts simulation experiments with different contents for the encryption effect of the model, and Section 6 concludes the paper.

2 Related Work

A wide range of scholars have conducted different degrees of image encryption research using chaos theory. (1) In terms of chaotic mapping functions,

literature [21] uses one-dimensional tent mapping to obtain key sequences fully applied to image obfuscation and diffusion, using logical operations of iso-or and circular shift to spread weak changes in a single pixel over many pixels and increase the resistance of the algorithm to differential attacks; literature [22] uses the Chen hyperchaotic system to generate the dislocation needed for row and column random sequences, and the proposed image encryption algorithm has strong robustness due to the sufficiently large key space and high sensitivity to the key. However, due to the high complexity and low utilization of high-dimensional chaotic mappings and the simple structure, fewer parameters and lower security of low-dimensional chaotic mappings, an increasing number of scholars have started to improve the one-dimensional chaotic mappings to enhance the randomness of chaotic mappings and thus improve the security of chaotic mapping-based image encryption algorithms. Among them, literature [23] constructs segmentation of Tent and Tent mapping functions by the literature [23] obtains the parameters needed for Arnold mapping by constructing segmentation functions to Arnold mapping functions to obtain the transformation functions needed for image scrambling. The literature [24] scourges the Tent mapping with sine mapping to obtain a chaotic one-dimensional sinusoidal driven chaotic system with high sensitivity and randomness. The literature [25] introduces nonlinear terms in Henon chaotic mapping to improve the performance of chaotic mapping functions. In addition to improving low-dimensional chaotic mappings, some scholars also try to combine chaotic mappings with other randomness factors in a nonlinear way. The literature [26] nonlinearly mixes the improved tent mappings with sequences generated by quantum random wandering random number generators to obtain a mixed key sequence. (2) For image encryption: Literature [27] proposed a new chaotic image encryption method that uses the substitution and replacement of a single substitution box to solve the problems of contemporary image encryption algorithms. Literature [28] proposed a new image encryption method based on a logical chaotic system and deep autoencoder, and experiments illustrated that the algorithm has excellent encryption performance and can effectively resist attacks and improve the security of the image. Literature [29] proposed a chaotic image encryption scheme based on the sine-cosine algorithm. cosine algorithm-based chaotic image encryption scheme, and the proposed scheme can resist various attacks and has good robustness. Reference [30] proposed a multichannel orthogonal Gegenbauer moment with fractional order in Cartesian coordinates. The simulation shows that this algorithm has excellent encryption performance and can effectively resist attacks and improve the security of images. Reference [30] proposed a new image encryption method

based on the cosine algorithm. coordinates, simulation experiments show that the method has a wide range of key spaces, high key sensitivity and good encryption effects. Reference [31] proposes an image encryption algorithm based on two-dimensional Tent-Gaussian hyperchaos, and the results show that the algorithm has a high level of robustness and effectiveness and can resist different security attacks and data loss. Reference [32] proposes an image encryption algorithm based on the Tent quantum chaos. For the quantum chaos-based image encryption scheme, the simulation results show that the scheme has a larger key space and stronger key sensitivity. The literature [33] proposed a new piecewise-Tent-Sine map, and the results show that it has better chaotic behavior and low time complexity. The literature [34] proposed a color image encryption algorithm combining the KAA map and multiple chaotic maps, and the results show that the algorithm has high robustness and effectiveness against different security attacks and data loss. The literature [34] proposed a combination of the KAA map and multiple chaotic map color image encryption algorithm, which adopts Shannon's theorem, and simulation illustrates that the proposed algorithm has better results in multiple metrics. The literature [35] proposes an efficient image encryption method based on a mixture of watermarking and encryption techniques, and simulation experiments illustrate that it has better results in differential attack, statistical attack, and noise attack.

In the above studies, we found that most scholars have gradually shifted from the study of one-dimensional chaotic image encryption to the study of multidimensional chaotic encryption and achieved certain results. Based on this, we propose the Tent-Ushiki chaotic mapping encryption method and optimize the parameters of this chaotic mapping using the improved firefly algorithm. The simulation experiments show that this method has a good statistical analysis performance and correlates the adjacent elements, especially for information entropy and anti-attacks.

3 Basic Algorithm

3.1 Firefly Algorithm

The firefly algorithm (FA) is a swarm intelligence optimization algorithm created by Xin Sheng Yang [36], which simulates the behavior of firefly populations at night; the algorithm simulates how an entire firefly population catches food and attracts the opposite sex, with luminescence being an important part of the interconnection between the individual fireflies. The idea is

that movements are determined by the strength of the fluorescence emitted by the fireflies, i.e., individuals with weak fluorescence move toward the individuals with strong fluorescence and continuously update the position to obtain the optimal solution to the algorithm.

(1) Relative fluorescence brightness

In the range of the search space, there are two firefly individuals i, j. When the brightness of firefly individual i is stronger than that of firefly individual j, the phenomenon of individual i moving toward individual j will occur. If this mutual attraction has a greater effect, then the relative fluorescence brightness between them is stronger, and vice versa. Thus, the expression for this relative fluorescence brightness is as follows:

$$I(r_{ij}) = I_0 e^{-\gamma r_{ij}^2} \tag{1}$$

$$r_{ij} = ||X_i - X_j|| = \sqrt{\sum_{k=1}^{d} (X_{i,k} - X_{j,k})} \tag{2}$$

In Equation (1), I_0 denotes the starting value of the fluorescence brightness, which depends on the target function value of the location of the individual fluorescent insect. It is usually considered that there is a better target function value for an individual with strong luminosity. γ denotes the attenuation coefficient of fluorescence brightness because brightness has a certain attenuation in the air. In Equation (2), r_{ij} denotes the Euclidean distance between the individual fireflies, i and j, and k denotes the dimension in which the individual is located.

(2) Degree of attraction

The relative brightness of individual firefly i to individual firefly j is proportional, and the attraction of individual firefly i to j is defined as

$$\beta(r_{ij}) = \beta_0 e^{-\gamma r_{ij}^2} \tag{3}$$

In Equation (3), β_0 denotes the attractiveness of the firefly.

(3) Individual location update

$$X_i(t+1) = X_i(t) + \beta(r_{ij})\{X_j(t) - X_i(t)\} + \alpha \times (rand - 1/2) \tag{4}$$

$$X_i(t+1) = X_i(t) + \alpha \times (rand - 1/2) \tag{5}$$

In Equations (4)–(5), $rand$ is a random number between (0,1), and α is the step size. At time t, two firefly individuals, i, j, are substituted into the relative firefly brightness formula of Equation (1), and when the relative brightness of firefly individual i is less than the relative brightness of firefly individual j, Equation (4) is used for individual position updates, and vice versa.

3.2 Tent Chaos Mapping

Tent chaotic mapping is commonly used in image encryption algorithms because of its simple structure, easy platform implementation and complex chaotic dynamics and is mainly expressed as follows:

$$x_{n+1} = \begin{cases} \mu x_n & x_n < \dfrac{1}{2} \\ \mu(1 - x_n) & x_n >= \dfrac{1}{2} \end{cases} \tag{6}$$

In Equation (6), x_n denotes the mapping function, and μ denotes the parameter.

3.3 Ushiki Chaotic Mappings

Ushiki mappings are typical two-dimensional nonlinear discrete systems that are widely used in chaotic sequences. The main expression form is as follows:

$$\begin{cases} x_{n+1} = (3.7 - x_n - 0.1 \times y_n) \times x_n \\ y_{n+1} = (3.7 - 0.15 \times x_n - y_n) \times y_n \end{cases} \tag{7}$$

In Equation (7), x_n and y_n denote the mapping functions.

4 Optimized Tent-Ushiki Mapping Image Encryption Based on an Improved Firefly Algorithm

The traditional tent image encryption algorithm has the disadvantages of a simple chaotic structure, few parameters and a small key space, which affects the security of image encryption. Two-dimensional chaotic systems are used in encryption systems because they have a better complex structure. In this paper, we construct a Tent-Ushiki two-dimensional chaotic mapping for image encryption and use the improved firefly algorithm to optimize

the parameters of the two-dimensional chaotic system to obtain the optimal chaotic encryption effect.

4.1 Tent-Ushiki Chaos Mapping

In this paper, we use Tent and Ushiki chaos mappings as the chaos model. The idea is to use the output sequence of a tent chaos mapping to adjust the input of Ushiki chaos mapping and then use the modulo operation to fix the result to a certain range of values, which can compress the obtained sequence to a certain plane to obtain a bounded sequence. By adding some unknown parameters to the chaotic system, a new Tent-Ushiki chaotic system is obtained with the following mathematical expressions:

$$\begin{cases} x_{n+1} = \mathrm{mod}(3.7 - a1\mu x_n(1-x) - 0.1 \times a2\mu y_n(1-y)) \times y_n, 1) \\ y_{n+1} = \mathrm{mod}(3.7 - 0.15 \times a3\mu x_n(1-x) - a4\mu y_n(1-y)) \times y_n, 1) \end{cases} \tag{8}$$

In Equation (8), $a_i(i = 1, 2, 3, 4)$ is the unknown parameter of Tent-Ushiki, and μ is the control parameter.

Most images are encrypted using chaos theory to obtain better results. Chaos theory has the characteristics of sensitivity to initial conditions, control parameters, and ergodicity, and these characteristics are similar to the current cryptographic system of dislocation and diffusion; thus, the introduction of chaos theory in images can provide better encryption results. However, the following two problems must be considered:

(1) The strength of image pixel dislocation depends on the sensitivity of the initial value of the chaotic mapping and its traversal because the higher the sensitivity of the initial value of the chaotic mapping is, the smaller the correlation between the adjacent pixels of the dislocated image, and vice versa. Thus, the randomness of the dislocation is stronger.
(2) The higher the number of iterations of the chaotic mapping in the process of pixel dislocation and substitution, the higher the encryption strength, which increases the computational complexity of the encryption process.

To consider these two problems in more detail, we use a fractional-order Fourier transform based on the image pixel replacement matrix, a one-dimensional Tent chaos algorithm to reduce the randomness of the image pixels, and a sine chaos-based optimization diffusion algorithm to reduce the computational complexity.

4.2 Improved Firefly Algorithm

The firefly algorithm suffers from premature convergence and slow iteration in solving practical application problems, mainly because the algorithm is unable to find a balance between the global optimum and the local optimum process. To better optimize the parameters in the Tent-Ushiki chaotic mapping, we optimize the algorithm in three aspects to improve the algorithm performance.

(1) Adaptive step size optimization

A firefly's individual position is very important for the accuracy of the optimal solution. In the existing firefly algorithm, when the value of the step size is large, the algorithm oscillates at a later stage, which reduces the operation speed of the algorithm and leads to a decrease in the accuracy of the algorithm; conversely, when the value of the step size is small, the search speed rises, and the accuracy of the solution increases. Therefore, the setting of the step size is very important. To improve the performance of the algorithm, we propose an adaptive step size strategy in this paper. In the early stage of the algorithm, a larger step size is used to prompt an individual firefly to approach the local optimum rapidly, which improves the convergence speed of the algorithm. When the number of iterations increases, the step size value gradually decreases, which makes the individual firefly continuously search for the optimal solution around itself, which makes the individual algorithm expand the search range, improve the accuracy of the algorithm and optimize the quality of the individual solution. Therefore, the step size is set as follows:

$$a = a_0 \times \sin\left(\frac{t}{t_{\max}}\right) \tag{9}$$

In Equation (9), a_0 is set as the initial value of the step size, t is the current number of iterations, and $t_{\max}$ is the maximum number of iterations.

(2) Introduction of the adjustment factor

In the firefly algorithm, the mutual attraction between individuals is one of the important factors affecting the quality of the solution. Since β_0 is set to a fixed value in the algorithm, this can lead to a significant effect on the attraction parameter when the distance between individuals is too large or

too small. To avoid this situation, we set β_0 by introducing a moderating factor η. η is expressed as shown in Equation (10), and β_0 is set as shown in Equation (11):

$$\eta = (iter_{\max} - iter)/iter_{\max} \tag{10}$$

$$\beta(r_{ij}) = \eta \times \beta_0 e^{-\gamma r_{ij}^2} \tag{11}$$

In Equations (10)–(11), $iter_{\max}$ denotes the maximum number of iterations, and $iter$ denotes the current number of iterations. We find that by such iterations, the firefly individuals move in larger steps, which helps the algorithm to reach the optimal solution quickly. In the beginning, the number of iterations is small, which makes the attraction parameter $\beta(r_{ij})$ larger and moves more individuals toward the local optimal solution, and as the iterations go deeper and deeper, the attraction parameter $\beta(r_{ij})$ decreases gradually when the individuals are close to the optimal solution, which makes the individuals of the solution obtain higher search accuracy.

(3) Introduction of inertia weights

In the late stage of the firefly algorithm, most firefly individuals easily reach or approach the optimal extremum point, which makes the attraction between individuals increase, making individuals oscillate near the local extremum point, making firefly individuals unable to appear in the optimal position, resulting in the algorithm taking a long time and decreasing the accuracy. To avoid this situation, we introduced an inertia weight factor for algorithm optimization. When the inertia weight value is larger, the current position has a greater impact on the next position of the fireflies, which reduces the degree of attraction between firefly individuals, enhancing the algorithm's optimization abilities and decreasing the local optimization abilities. Conversely, when the weight value is smaller, the current position has a smaller impact on the next position of fireflies. In contrast, when the weight value is small, the current position has less influence on the next position of the fireflies, which increases the attraction between individuals and enhances the algorithm's local optimization abilities, thus ensuring that the algorithm has a better optimization ability in the early stage. In the later stage, when it is close to the optimal value of the population, the movement speed of the firefly individuals decreases, which improves the algorithm's optimization seeking ability and ensures the convergence accuracy of the algorithm. The inertia

weighting factor is shown in Equation (12).

$$w(t) = 1 - \frac{iter \times (w_{\max} - w_{\min})}{iter_{\max}} \tag{12}$$

$$X_i(t+1) = X_i(t) + \alpha \times w(t) \times (rand - 1/2) \tag{13}$$

In Equations (12)–(13), $w_{\max}$ and $w_{\min}$ denote the maximum and minimum values of the weight factor between (0,1), respectively, $iter_{\max}$ is the maximum number of iterations, and $iter$ is the current number of iterations.

4.3 Parameter Optimization Based on the Improved Firefly Algorithm

The Tent-Ushik chaotic mapping contains $a_i(i = 1, 2, 3, 4)$ total of four unknown parameter variables, which are optimized using the improved artificial firefly algorithm. The number of feasible solutions in the population is set to N, so the population is represented as $\{X_1, X_2, \ldots, X_N\}$. The process is as follows:

Step 1: Perform the initialization of the firefly population by generating a one-dimensional array containing four unknown parameter variables representing one set of feasible solutions at a time, and randomly generate N sets of feasible solutions to construct a new initial population;

Step 2: Substitute this N set of feasible solutions into Equation (8) for iteration to obtain the N set of chaotic sequences $X(i)$. Turn this chaotic sequence into a single objective function using information entropy and Lyapunov, and locate this objective function as an individual fitness function;

Step 3: Optimize the update of the individual position of the firefly algorithm according to the adaptive step size, the adjustment factor and the inertia weight and compare the current individual fitness value with the global fitness value;

Step 4: When the maximum number of iterations or accuracy is reached, the algorithm turns to Step 5; otherwise, it turns to Step 3;

Step 5: Output the optimal 4 unknown participation variables.

4.4 Encryption Algorithms

We optimize Tent-Ushiki mapping encryption based on the improved Firefly optimization algorithm into three elements based on the study of encryption

models elaborated in the literature [36]: (1) key generation, (2) Tent-Ushiki-based bit-plane disruption and (3) Tent-Ushiki-based chunking diffusion.

4.4.1 Generation of the key

(1) Hash value in the bit plane position

We use SHA-128 to calculate the plaintext to obtain a set of 128 bit binary numbers a. The bit plane dislocation operation process needs to dislocate each bit plane by 8 groups of chaotic sequences. Therefore, 4 groups of initial values and 4 chaotic control parameters according to b of Equation (14) are needed to obtain the first group of initial values c and the other initial values from the generated first group of chaotic sequences to be obtained, enhancing each group of connections between chaos. The control parameter d is calculated by Equations (15)–(16).

$$\begin{cases} x_1 = \dfrac{\sum_{i=1}^{32} key(i) \times 2^{i-1}}{2^{32}} \\ y_1 = \dfrac{\sum_{i=33}^{64} key(i) \times 2^{i-33}}{2^{32}} \end{cases} \tag{14}$$

$$\mu_j = \sum_{i=95+33*(j-1)}^{95+33*j-1} \frac{key(i) \times 2^{i-33(j+1)+1}}{2^{33}} \ (j = 1, 2, 3, 4) \tag{15}$$

$$\mu_j = floor(\mu_j \times 50) \tag{16}$$

Two chaotic sequences, X_1, Y_1, are generated by substituting (x_1, y_1) and the computational control parameter μ_j into Equation (15). The $M \times N$th value of these two chaotic sequences is used as the initial value of the second chaotic sequence, and the initial values of the remaining two groups are obtained in the same way. The method is shown in Equation (17).

$$\begin{cases} x_{\text{initialize}k} = X_{k-1}(M \times N) \\ y_{\text{initialize}k} = Y_{k-1}(M \times N) \end{cases} (k = 2, 3, 4) \tag{17}$$

In the formula, $(x_{\text{initialize}k}, y_{\text{initialize}k})$ denotes the initial value of the sequence of group k, and X_{k-1} and Y_{k-1} denote the chaotic sequence of group $k-1$.

(2) The key value for the diffusion of the scrambled ciphertext image.

The disrupted image is divided into 4 blocks, each of size $M/2 \times N/2$. The SHA-224 hash algorithm is used to calculate the hash value of

each row of each block after blocking, and the key is obtained as in Equation (18).

$$key_{ij} = \{k_{ij}(1), k_{ij}(2), \ldots, k_{ij}(224)\}\ i \in [1, M/2], j \in [1, 4] \tag{18}$$

where key_{ij} denotes the key in row i of block j.

$$\begin{cases} x_{ij} = \dfrac{\sum_{m=1}^{108} key_{ij}(m) \times 2^{m-1}}{2^{108}} \\ y_{ij} = \dfrac{\sum_{m=109}^{216} key_{ij}(m) \times 2^{m-109}}{2^{108}} \\ \mu_{ij} = floor\left(\dfrac{\sum_{m=217}^{224} key_{ij}(m) \times 2^{m-218}}{2^{108}} \times 50\right) \end{cases} \tag{19}$$

This is transformed sequentially into the initial values and control parameters of the chaotic system used for the diffusion operation by Equation (19). The subsequent diffusion operation is conducted by generating three different sets of chaotic sequences for the heterogeneous operation with the plaintext image. By keeping the initial value conditions of each set of chaotic sequences the same and changing the control parameters of Tent-Ushiki, different chaotic sequences can be generated. The remaining two sets of parameter values are generated with the initial values and control parameters obtained by the above method, with the following computational aspects:

$$\begin{cases} \mu_{ij1} = \mu_{ij} + x_{ij} \\ \mu_{ij2} = \mu_{ij} + y_{ij} \end{cases} \tag{20}$$

4.4.2 Tent-Ushiki-based bit-plane dislocation algorithm

Step 1: Input the plaintext image and calculate the initial values and control parameters of the desired chaotic sequences according to Section 3.3.1;

Step 2: Substitute the obtained initial values and control parameters into Equation (1) for iteration to generate 8 different sets of chaotic sequences $\{Y_1, Y_2, \ldots, Y_8\}$;

Step 3: Implement the bit-plane decomposition operation on the plain-text images to obtain bit-plane images of different bit levels $\{P_1, P_2, \ldots, P_8\}$;

Step 4: The eight bit-plane images are processed as follows: the $i(i \in [1, 8])$ bit-plane is arranged into a one-dimensional vector A in a column-first

manner, and the columns of the chaotic sequence are arranged to generate a sequence T for recording the positions of the elements of the sorted sequence in the original sequence. The one-dimensional vector A is rearranged according to the sequence T to obtain a new one-dimensional vector D after the transformation, and then the size of D is adjusted to match the size of the plaintext image so that it is the same size as the plaintext image;

Step 5: The eight scrambled bit planes are merged, and finally, the scrambled image P_{con} is obtained.

4.4.3 Tent-Ushiki-based chunking diffusion algorithm

Step 1: Divide P_{con} equally into 4 subblock images of the same size, each of size $M/2 \times N/2$.

Step 2: Calculate the hash value of each row of each block after chunking by the SHA-384 hashing algorithm and calculate the initial values and control parameters needed in the diffusion process of each row of each block according to Equation (19).

Step 3: The first block of the first row is used as an example for the diffusion operation. (x_{11}, y_{11}) and μ_{11} are obtained according to Equation (19). μ_{111} and μ_{112} are obtained according to Equation (20), and the three sets of data (x_{11}, y_{11}) and μ_{11}, (x_{11}, y_{11}) and μ_{111}, (x_{11}, y_{11}) and μ_{112} are substituted into Equation (8) and iterated $M/2 \times N/2$ times to obtain three different chaotic sequences μ_1, μ_2 and μ_3. The chaotic sequences are quantized as integers between 0 and 255 according to Equation (21).

$$\mu_k = round(u_k \times 255)(k = 1, 2, 3) \tag{21}$$

Step 4: Transform the subblock image into a $M/2$ one-dimensional a-column row vector. The obtained three chaotic sequences and all the image elements of the first block are subjected to an aliasing operation based on Equation (22), and the ciphertext image of the vector size obtained after adjusting the aliasing will be the same size as the subblock image. $a(i)$ is each subblock image, and the $C\_a(i)$ subblock corresponds to the encrypted image.

$$C\_a(i) = \mu_1 \oplus (\mu_2 \oplus (\mu_3 \oplus a(i)))i \in [1, 4] \tag{22}$$

Step 5: Repeat Steps 3 and 4 to complete the operation of the first line of the remaining subblocks for the diffusion operation, according to Formula (23),

to merge the four subblocks to obtain a diffusion of the image.

$$img(j) = \begin{bmatrix} C\_a(1) & C\_a(2) \\ C\_a(3) & C\_a(4) \end{bmatrix},\ j \in [1, M/2] \tag{23}$$

Step 6: Repeat Steps 3–5 to complete the diffusion operation of all the subblocks in row $M/2$ and generate the $M/2$ diffusion image.

5 Simulation Experiments

5.1 Experimental Configuration

To further verify the performance of the algorithm, we use a Core I5 CPU, 32 GB of RAM, a 1T hard disk, a GeForce RTX2060 graphics card, Windows 10 software, and MATLAB 2012a as the system simulation software.

5.2 Algorithm Performance Validation

To further illustrate the performance of the firefly algorithm after optimization for the parameters of the chaotic model, we chose four benchmark functions for testing. The comparison indexes are the minimum value, the maximum value, the average value and the variance. The comparison algorithms are the FA algorithm, discrete FA (DFA) [38] and binary FA (BFA) [39] and this paper's algorithm (IFA) for comparison.

Based on the data provided in Tables 2–5, it is clear that IFA outperforms the other algorithms in terms of minimum, maximum, mean and standard

Table 1 Four benchmark functions

No	Benchmark Function
F1	$\sum_{i=1}^{n-1}[100(x_{i+1}-x_i^2)^2+(x_i-1)^2]$
F2	$20\exp\left(-\frac{1}{5}\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right)-\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right)$
F3	$\sum_{i=1}^{n}(x_i^2-10\cos(2\pi x_i)+10)$
F4	$\sum_{i=1}^{n}([x_i+0.5])^2$

Table 2 F1 benchmark function

Algorithm	Dim	Min-Value	Max-Value	Mean	St-Deviation
FA	2	2.1327	9.8024	12.4262	15.8121
	10	5.9712	9.8614	18.9317	28.6322
	30	30.1962	60.2364	40.2136	27.5831
	50	89.0273	110.2243	90.1222	46.2023
DFA	2	1.9132	3.1263	3.1321	2.9218
	10	4.3135	9.5134	5.9218	4.8927
	30	26.1382	38.4127	28.5513	26.2671
	50	78.9134	92.2162	83.3126	86.2452
BFA	2	0.1873	1.2463	2.0138	2.0541
	10	3.1575	6.3156	2.8127	2.8144
	30	10.2928	29.5221	18.7521	14.6272
	50	56.8494	90.2061	78.7821	76.9253
IFA	2	9.2752E-01	3.4149E-01	3.1972E-01	5.2602E-01
	10	3.9172E-01	8.3212E-01	4.7318E-01	3.2187E-01
	30	2.8913E+01	8.9212E+01	7.3914E+01	3.2413E+01
	50	7.2753E+01	2.9172E+01	2.3453E+01	3.2826E+01

Table 3 F2 benchmark function

Algorithm	Dim	Min-Value	Max-Value	Mean	St-Deviation
FA	2	2.1702	9.7318	4.8324	2.2613
	10	7.9243	10.8942	10.9132	16.2142
	30	21.8393	36.2289	25.4136	33.2712
	50	76.1912	98.7345	85.2192	78.5342
DFA	2	1.1923	8.9215	5.7235	2.6143
	10	10.8622	18.1739	15.8251	16.2344
	30	28.2827	47.3528	37.3278	42.2152
	50	73.8124	88.4136	85.1312	72.5341
BFA	2	1.9123	7.3328	1.3823	3.2816
	10	9.5382	11.1732	8.2146	6.1834
	30	17.7923	24.1763	19.4018	17.6228
	50	26.4152	38.2574	31.1728	34.3623
IFA	2	8.2352E-01	3.7334E-01	8.9462E-01	3.7912E-01
	10	9.2375E-01	7.8152E-01	3.7162E-01	3.3139E-01
	30	6.3214E+01	3.2816E+01	3.3268E+01	4.6148E+01
	50	3.6571E+01	9.8316E+01	3.9126E+01	3.8103E+01

Table 4 F3 benchmark function

Algorithm	Dim	Min-Value	Max-Value	Mean	St-Deviation
FA	2	2.7327	8.3921	8.5127	6.3495
	10	5.3418	9.8361	6.8314	3.9213
	30	10.7821	18.0137	14.1732	18.7638
	50	26.3134	39.7162	32.3138	33.4213
DFA	2	2.1437	8.9213	6.4812	4.1436
	10	2.7145	5.2419	3.4143	3.7424
	30	9.2143	13.2517	11.7132	11.9341
	50	20.2182	26.8123	26.9213	26.3712
BFA	2	1.1991	2.0924	2.1923	2.7214
	10	4.1273	16.7914	13.1724	10.3218
	30	8.1934	12.8245	13.8721	15.7214
	50	19.8762	24.1285	22.2413	27.9515
IFA	2	0	12.9172E-01	8.4741E-01	3.9631E-01
	10	1.9317E-01	4.2818E-01	5.4719E-01	5.8748E-01
	30	1.3715E+01	3.1578E+01	6.3741E+01	3.3192E+01
	50	2.3142E+01	2.3842E+01	5.2839E+01	3.4724E+01

Table 5 F4 benchmark function

Algorithm	Dim	Min-Value	Max-Value	Mean	St-Deviation
FA	2	3.7434	6.3193	5.3284	4.2315
	10	9.2772	17.2865	12.6187	10.9254
	30	19.4179	23.8442	19.6941	26.2873
	50	38.4276	64.3291	51.7426	39.3582
DFA	2	2.7374	5.2413	4.3251	3.2695
	10	8.1492	12.3361	10.7217	14.9351
	30	14.1419	20.7162	16.9132	22.0923
	50	32.4216	56.9191	44.7881	35.2136
BFA	2	1.0631	3.1922	2.4116	2.4133
	10	5.8254	8.1842	7.3312	6.8415
	30	12.8691	18.3148	36.2169	19.4193
	50	21.8016	45.7832	26.7142	29.4158
IFA	2	2.6902E-01	2.9191E-01	4.1671E-01	6.5912E-01
	10	2.3062E-01	3.1942E-01	6.8238E-01	4.3187E-01
	30	9.8328E+01	7.8256E+01	4.1713E+01	3.3429E+01
	50	0	4.2375E+01	3.7139E+01	8.1925E+01

value results for the four functions studied. The superiority of IFA compared to FA is clearly demonstrated in these tables. In addition, IFOA shows a clear performance advantage compared to DFA and BFA. Particularly noteworthy is the fact that in the F3 and F4 functions, IFA minimizes to 0 when the dimensions are 2 or 50, while IFA consistently provides favorable results in all four functions when the dimensions are 10 and 30. This indicates that the overall performance of the IFA is greatly improved by strategies such as adaptive step size, moderating factors, and inertia weighting aspects. A foundation is laid for subsequent optimization of the chaotic model.

5.3 Encryption Effect Verification

In this paper, Lena, Baboon and Camerman with pixel sizes of 256*256 are selected as image encryption objects. We compare them in terms of statistical analysis, adjacent pixel correlation, information entropy comparison and anti-differential attack analysis metrics. The algorithms from the literature [27], literature [28] and literature [30] are selected for analysis.

In the results of Figures 1–2, the information of the original image is hidden after encryption, and no information can be obtained by the naked eye alone. To verify the security and reliability of the proposed encryption algorithm, the following analysis will be conducted from a statistical analysis and a correlation of the adjacent elements.

(1) Statistical analysis

Figure 3 shows the histograms of the algorithm in this paper before and after encryption, Figure 3(a)(c)(e) shows the statistical analysis values of the Lena, Baboon and Camerman images before encryption, and Figure 3(b)(d)(f)

Fig. 1(a) Lena Fig. 1(b) Baboon Fig. 1(c) Camerman

Figure 1 Original images.

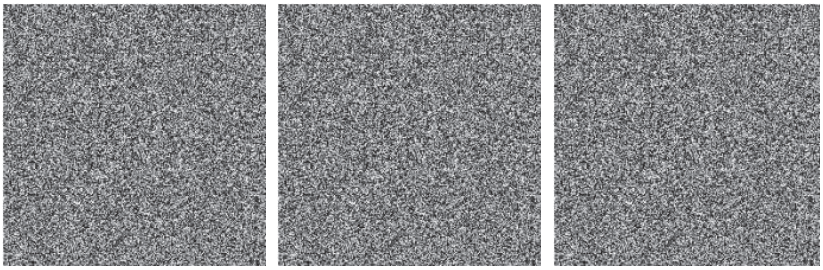

Fig. 2(a) Lena Fig. 2(b) Baboon Fig. 2(c) Camerman

Figure 2 Encrypted image.

shows the statistical analysis values of the Lena, Baboon and Camerman images after encryption, from which we find that the grayscale histogram of the images before encryption is not very evenly distributed, while the distribution of the encrypted images is very even, which indicates that the effect of the algorithm after statistical analysis is good.

(2) Adjacent pixel correlation

To illustrate the effect of encryption, we select 100 sets of adjacent pixels in Figures 1 and 2 and calculate the pixel correlation in the horizontal, vertical and diagonal directions according to Equations (24)–(27).

$$E(x) = \frac{1}{N}\sum_{k=1}^{N} x_k \tag{24}$$

$$D(x) = \frac{1}{N}\sum_{k=1}^{N}(x_k - E(x)) \tag{25}$$

$$Cov(x,y) = \frac{1}{N}\sum_{k=1}^{N}(x_k - E(x))(y_k - E(y)) \tag{26}$$

$$r(x,y) = \frac{|Cov(x,y)|}{\sqrt{D(x)\sqrt{D(y)}}} \tag{27}$$

where Cov in Equation (26) denotes the covariance, (x, y) denotes the gray value of the adjacent pixel points in the image, and N is the number of pixels picked. Tables 6 and 7 show the results of the correlation of neighboring

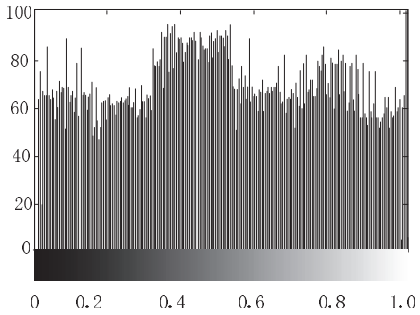

Fig. 3(a) Histogram of the unencrypted Lena images

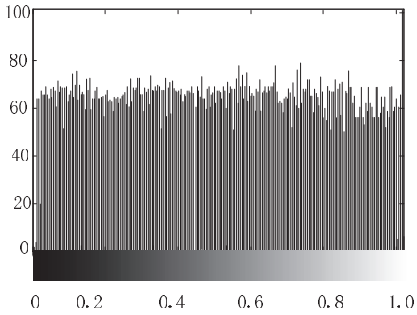

Fig. 3(b) Histogram of the encrypted Lena images

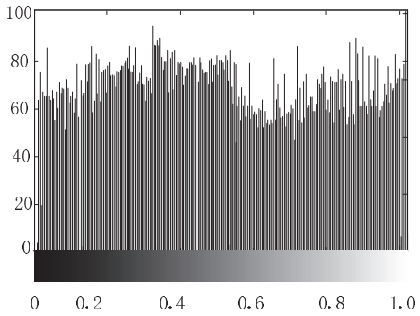

Fig. 3(c) Histogram of the unencrypted Baboon images

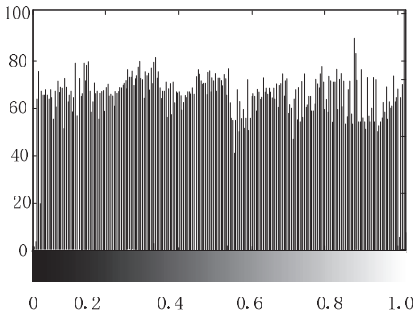

Fig. 3(d) Histogram of the encrypted Baboon images

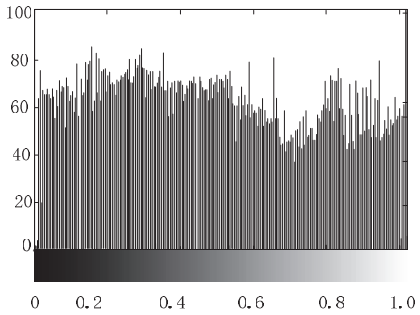

Fig. 3(e) Histogram of the unencrypted Camerman images

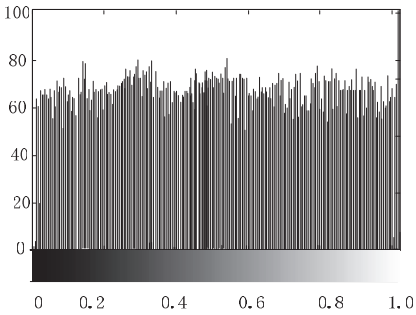

Fig. 3(f) Histogram of the encrypted Camerman images

Figure 3 Histograms of the three test images.

elements in three directions of the three images, before and after encryption. From the data results, the comparison data results of the three images in three directions differ greatly, which shows that the encrypted image retains the main pixel features of the original image, and the comparison results

Table 6 Two-adjacent pixel correlation of the three images

Direction	Fig. 1(a)	Fig. 1(b)	Fig. 1(b)	Fig. 2(b)	Fig. 1(c)	Fig. 2(c)
Horizontal Direction	0.8742	0.0091	0.9172	0.0083	0.8324	0.0061
Vertical Direction	0.8521	0.0094	0.9012	0.0083	0.8516	0.0068
Diagonal direction	0.8424	0.0088	0.8142	0.0084	0.8632	0.0072

Table 7 Time complexity of the three images (%)

Direction	Fig. 1(a)	Fig. 1(b)	Fig. 1(b)	Fig. 2(b)	Fig. 1(c)	Fig. 2(c)
Horizontal Direction	87.23	34.15	87.42	48.25	88.21	36.27
Vertical Direction	85.72	51.27	80.43	53.63	86.37	54.65
Diagonal direction	94.43	45.18	85.37	52.42	94.35	47.28

of time complexity through Table 7 show that the image complexity is reduced after encryption but can still retain the encrypted information of the images well. Figures 4–6 show the comparison results of the three images in the diagonal, horizontal and vertical directions. From the comparison, the neighboring pixel values of the original images in each direction are generally concentrated around the center region; however, the overall distribution of the pixels in the encrypted three images shows a random distribution, which attains a better encryption effect.

(3) Information entropy comparison

Table 8 shows the comparison of this algorithm with literature [27], literature [28], and literature [30] in terms of information entropy and attack resistance. The results in Table 8 show that the information entropy of the encrypted image (maximum value of 8) increases, the pixel distribution in the ciphertext is uniform, and the randomness is enhanced. The encryption scheme proposed in this paper has greater information entropy than the other three algorithms, and the attacker can obtain very little useful information from the ciphertext, so it is more secure, less likely to leak information, and has the ability to resist statistical analysis.

(4) Resistant Attack Analysis

Table 9 shows the effectiveness of this paper's algorithm with respect to the literature [27], literature [28], and literature [30] for anti-differential attack analysis. From Table 4, it is found that the algorithm in this paper is significantly better than the comparison algorithms in terms of the NPCR (99.6093%) and UACI (33.4635%), which also indicates that the proposed algorithm in this paper has a certain ability to resist differential attacks.

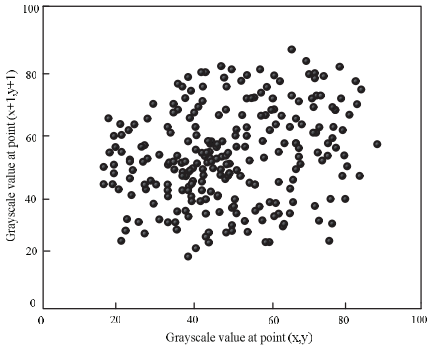

Fig. 4(a) Diagonal direction of Figure 1(a)

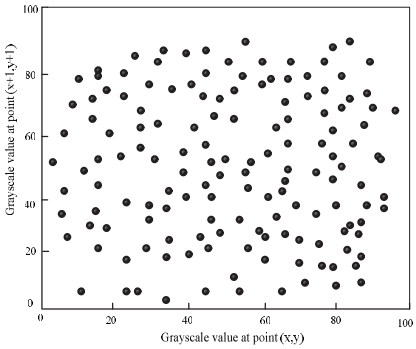

Fig. 4(b) Diagonal direction of Figure 2(a)

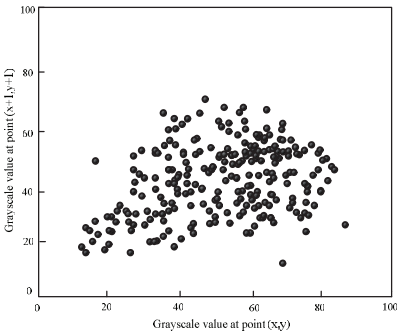

Fig. 4(c) Vertical direction of Figure 1(a)

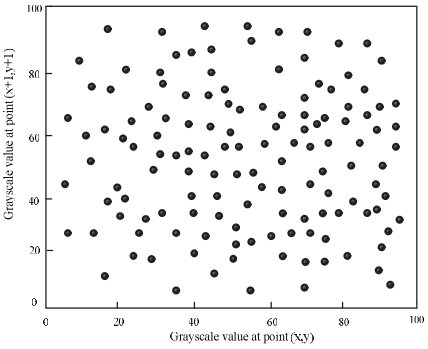

Fig. 4(d) Vertical direction of Figure 2(a)

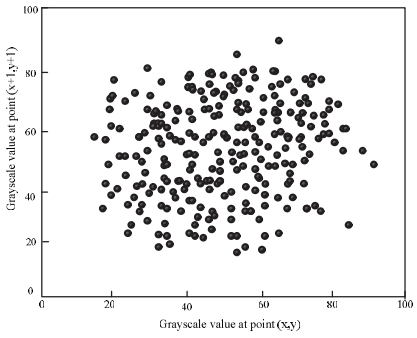

Fig. 4(e) Horizontal direction of Figure 1(a)

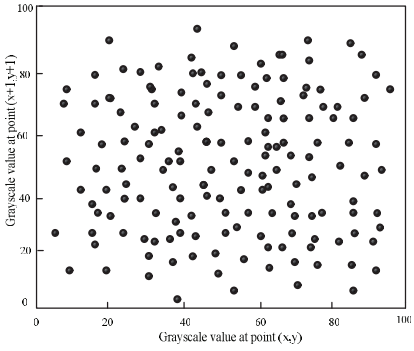

Fig. 4(f) Horizontal direction of Figure 2(a)

Figure 4 Lena image adjacent element correlation.

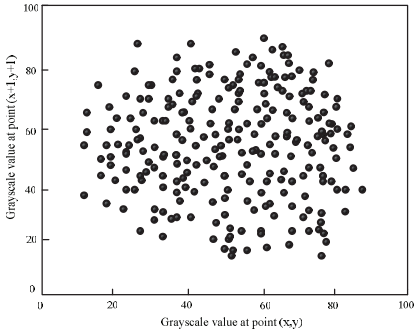

Fig. 5(a) Diagonal direction of Figure 1(b)

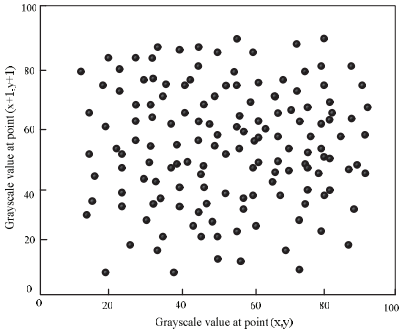

Fig. 5(b) Diagonal direction of Figure 2(b)

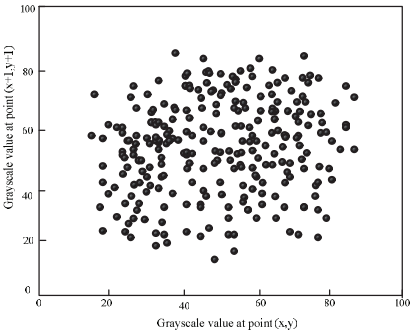

Fig. 5(c) Vertical direction of Figure 1(b)

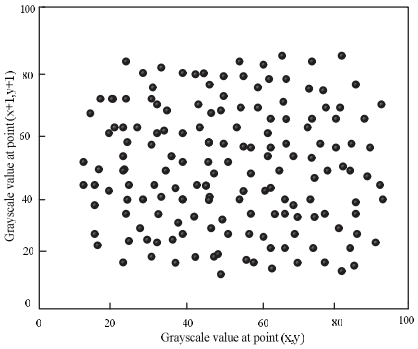

Fig. 5(d) Vertical direction of Figure 2(a)

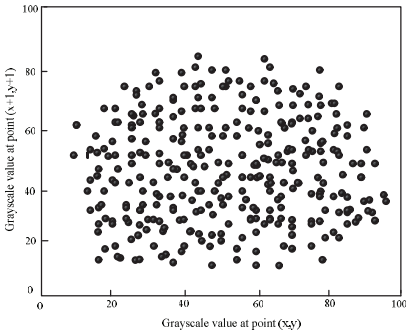

Fig. 5(e) Horizontal direction of Figure 1(b)

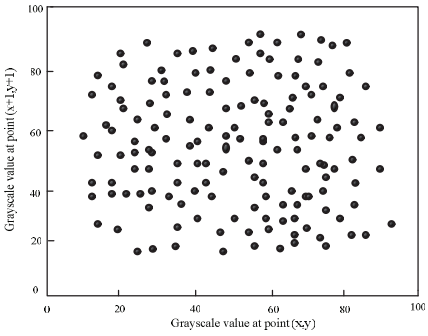

Fig. 5 (f) Horizontal direction of Figure 2(b)

Figure 5 Camerman image adjacent element correlation.

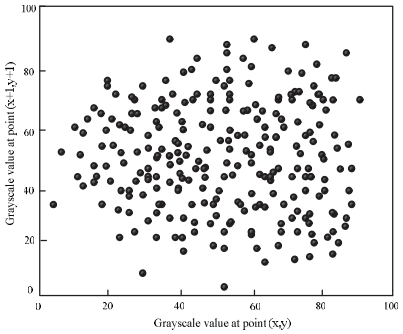

Fig. 6(a) Diagonal direction of Figure 1(c)

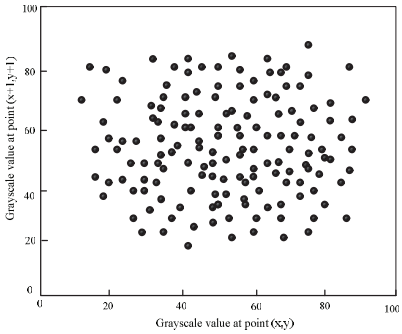

Fig. 6(b) Diagonal direction of Figure 2(c)

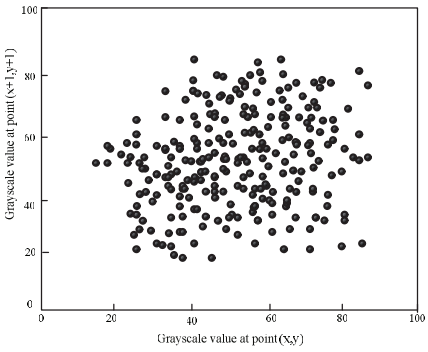

Fig. 6(c) Vertical direction of Figure 1(c)

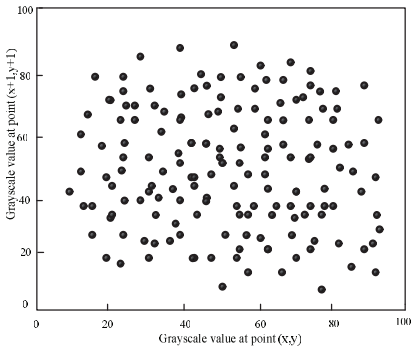

Fig. 6(d) Vertical direction of Figure 2(a)

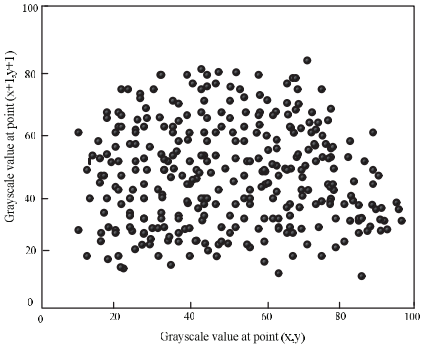

Fig. 6(e) Horizontal direction of Figure 1(c)

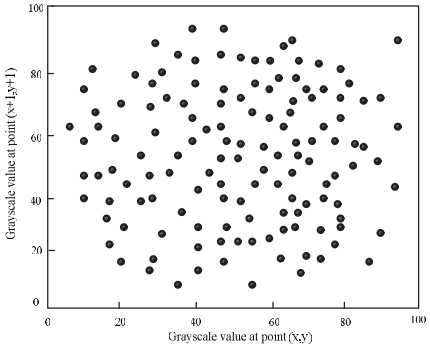

Fig. 6 (f) Horizontal direction of Figure 2(c)

Figure 6 Baboon image adjacent element correlation.

Table 8 Comparison of the information entropy of the four algorithms

Image Name	Original Image	Literature [27] Algorithm	Literature [28] Algorithm	Literature [30] Algorithm	Algorithm of This Paper
Lena	7.3793	7.9901	7.9892	7.9902	7.9965
Camerman	7.2304	7.9889	7.9901	7.9887	7.9912
Baboon	7.5029	7.9831	7.9328	7.9902	7.9956

Table 9 Comparison of anti-differential attack analysis

Indicators	Literature [27] Algorithm	Literature [28] Algorithm	Literature [30] Algorithm	Algorithm of This Paper
NPCR	99.3011%	99.4319%	99.4908%	99.5022%
UACI	33.3014%	33.2918%	33.3409%	33.3929%

6 Conclusions

We proposed a Tent-Ushiki chaotic mapping image encryption scheme based on the improved firefly algorithm. To address the lack of optimization of the chaotic mapping parameters, we first improve and optimize the firefly algorithm in terms of the adaptive step size, adjustment factor and inertia weights, and second, we use the optimized algorithm for the optimization of chaotic parameters. In the simulation experiments, the algorithm in this paper performs well in a statistical analysis of classical image tests and in the correlation of adjacent elements and significantly outperforms the comparison algorithms in terms of information entropy and anti-attacks, with better results. The future development of image encryption technology will be a comprehensive process involving the improvement of traditional encryption methods combined with emerging technologies such as quantum computing and deep learning to meet evolving security challenges and diverse application scenarios.

References

[1] Liu S, Guo C, Sheridan J T. A review of optical image encryption techniques[J]. Optics & Laser Technology, 2014, 57: 327–342.

[2] Davis R. The data encryption standard in perspective[J]. IEEE Communications Society Magazine, 1978, 16(6): 5–9.

[3] Nechvatal J, Barker E, Bassham L, et al. Report on the development of the Advanced Encryption Standard (AES)[J]. Journal of research of the National Institute of Standards and Technology, 2001, 106(3): 511–576

[4] Zimmermann R, Curiger A, Bonnenberg H, et al. A 177 Mb/s VLSI implementation of the international data encryption algorithm[J]. IEEE Journal of Solid-State Circuits, 1994, 29(3): 303–307.
[5] Ye G, Pan C, Huang X, et al. An efficient pixel-level chaotic image encryption algorithm[J]. Nonlinear Dynamics, 2018, 94: 745–756.
[6] Ye G, Huang X. An efficient symmetric image encryption algorithm based on an intertwining logistic map[J]. Neurocomputing, 2017, 251: 45–53.
[7] Artz D. Digital steganography: hiding data within data[J]. IEEE Internet computing, 2001, 5(3): 75–80.
[8] Cox I, Miller M, Bloom J, et al. Digital watermarking[J]. Journal of Electronic Imaging, 2002, 11(3): 414–414.
[9] Çavuşoğlu Ü, Kaçar S. A novel parallel image encryption algorithm based on chaos[J]. Cluster Computing, 2019, 22: 1211–1223.
[10] Wang X, Liu L, Zhang Y. A novel chaotic block image encryption algorithm based on dynamic random growth technique[J]. Optics and Lasers in Engineering, 2015, 66: 10–18.
[11] Wang R, Deng G Q, Duan X F. An image encryption scheme based on double chaotic cyclic shift and Josephus problem[J]. Journal of Information Security and Applications, 2021, 58: 102699.
[12] Huang X, Ye G, Chai H, et al. Compression and encryption for remote sensing image using chaotic system[J]. Security and Communication Networks, 2015, 8(18): 3659–3666.
[13] Huang R, Rhee K H, Uchida S. A parallel image encryption method based on compressive sensing[J]. Multimedia tools and applications, 2014, 72: 71–93.
[14] Yao L, Yuan C, Qiang J, et al. An asymmetric color image encryption method by using deduced gyrator transform[J]. Optics and Lasers in Engineering, 2017, 89: 72–79.
[15] Kanso A, Ghebleh M. An algorithm for encryption of secret images into meaningful images[J]. Optics and lasers in engineering, 2017, 90: 196–208.
[16] Chai X, Chen Y, Broyde L. A novel chaos-based image encryption algorithm using DNA sequence operations[J]. Optics and Lasers in engineering, 2017, 88: 197–213.
[17] Zhang Q, Han J, Ye Y. Image encryption algorithm based on image hashing, improved chaotic mapping and DNA coding[J]. IET Image Processing, 2019, 13(14): 2905–2915.

[18] Mozaffari S. Parallel image encryption with bitplane decomposition and genetic algorithm[J]. Multimedia Tools and Applications, 2018, 77: 25799–25819.
[19] Suri S, Vijay R. A biobjective genetic algorithm optimization of chaos-DNA based hybrid approach[J]. Journal of Intelligent Systems, 2019, 28(2): 333–346.
[20] Liu X, Tong X, Wang Z, et al. Uniform nondegeneracy discrete chaotic system and its application in image encryption[J]. Nonlinear Dynamics, 2022, 108(1): 653–682.
[21] Yavuz E, Yazıcı R, Kasapbaşı M C, et al. A chaos-based image encryption algorithm with simple logical functions[J]. Computers & Electrical Engineering, 2016, 54: 471–483.
[22] Yu S S, Zhou N R, Gong L H, et al. Optical image encryption algorithm based on phase-truncated short-time fractional Fourier transform and hyperchaotic system[J]. Optics and Lasers in Engineering, 2020, 124: 105816.
[23] Wang X Y, Li Z M. A color image encryption algorithm based on Hopfield chaotic neural network[J]. Optics and Lasers in Engineering, 2019, 115: 107–118.
[24] Mansouri A, Wang X. A novel one-dimensional sine powered chaotic map and its application in a new image encryption scheme[J]. Information Sciences, 2020, 520: 46–62.
[25] Luo H, Ge B. Image encryption based on Henon chaotic system with nonlinear term[J]. Multimedia Tools and Applications, 2019, 78: 34323–34352.
[26] Ge B, Luo H B. Image encryption application of chaotic sequences incorporating quantum keys[J]. International Journal of Automation and Computing, 2020, 17(1): 123–138.
[27] Arif J, Khan M A, Ghaleb B, et al. A novel chaotic permutation-substitution image encryption scheme based on logistic map and random substitution[J]. IEEE Access, 2022, 10: 12966–12982.
[28] Sang Y, Sang J, Alam M S. Image encryption based on logistic chaotic systems and deep autoencoder[J]. Pattern Recognition Letters, 2022, 153: 59–66.
[29] Daoui A, Karmouni H, Sayyouri M, et al. Robust image encryption and zero-watermarking scheme using SCA and modified logistic map[J]. Expert Systems with Applications, 2022, 190: 116193.

[30] Hosny K M, Kamal S T, Darwish M M. A novel color image encryption based on fractional shifted Gegenbauer moments and 2D logistic-sine map[J]. The Visual Computer, 2023, 39(3): 1027–1044.
[31] Lai Q, Hu G, Erkan U, et al. High-efficiency medical image encryption method based on 2D Logistic-Gaussian hyperchaotic map[J]. Applied Mathematics and Computation, 2023, 442: 127738.
[32] Wang Y, Chen L, Yu K, et al. An image encryption scheme based on logistic quantum chaos[J]. Entropy, 2022, 24(2): 251–272.
[33] Shao S, Li J, Shao P, et al. Chaotic Image Encryption Using Piecewise-Logistic-Sine Map[J]. IEEE Access, 2023, 11: 27477–27488.
[34] Alexan W, Elkandoz M, Mashaly M, et al. Color Image Encryption Through Chaos and KAA Map[J]. IEEE Access, 2023, 11: 11541–11554.
[35] Gupta M, Singh V P, Gupta K K, et al. An efficient image encryption technique based on two-level security for internet of things[J]. Multimedia Tools and Applications, 2023, 82(4): 5091–5111.
[36] Yang X S, He X. Firefly algorithm: recent advances and applications[J]. International journal of swarm intelligence, 2013, 1(1): 36–50.
[37] Zhou Y Q. Research on Chaotic Parameter Optimization and Image Encryption Algorithm Based on Artificial Bee Colony Algorithm[D]. Haerbin: Heilongjiang University, 2022: 37–39.
[38] Balaji K, Sai Kiran P, Sunil Kumar M. Power aware virtual machine placement in IaaS cloud using discrete firefly algorithm[J]. Applied Nanoscience, 2023, 13(3): 2003–2011.
[39] Xie W, Wang L, Yu K, et al. Improved multilayer binary firefly algorithm for optimizing feature selection and classification of microarray data[J]. Biomedical Signal Processing and Control, 2023, 79: 104080.

Biographies

Jun Li received her Bachelor's degree in Computer Science and Technology from Wuhan University in 2003 and a Master's degree in Computer Application Technology from Wuhan University in 2005. She is currently a lecturer in Shunde Polytechnic, with research interests in computer algorithms, big data, and cloud computing.

Weijun Huang is an senior lecturer at Shanghai Communications Polytechnic. He received B.S. degree in Industrial Electrification and Automation from Hunan University of Science and Technology in 1988. His research interests include Information Technology and algorithm design