High-sensitivity synchronous image encryption based on improved one-dimensional compound sine map

Xingyuan Wang | Maozhen Zhang

School of Information Science and Technology, Dalian Maritime University, Dalian, China

Correspondence
Xingyuan Wang and Maozhen Zhang, School of Information Science and Technology, Dalian Maritime University, Dalian, 116024, China
Email: xywang@dlmu.edu.cn; zhang.maozhen@foxmail.com

Abstract
An improved one-dimensional compound sine map is introduced. The evaluation of this chaotic system shows that it has high sensitivity and random chaotic behaviour. Based on this chaotic system, a new fast image encryption scheme is proposed. Through the cross-processing of multiple chaotic sequences, two high-sensitivity pseudo-random sequences are generated, and the generated high-sensitivity random sequence is used for synchronization cross-processing in four directions. Using the symmetry of the image, the image is divided into four directions, and the processing of each direction is different based on the high-sensitivity sequence and the chaotic value. Through one traversal, the entire image is processed four times differently. The encryption process is uncontrollable and invisible depending on the chaotic sequence. Simulation test results show that the algorithm has good encryption results, is sensitive to the initial secret key and provides satisfactory security capabilities compared with other algorithms.

1 | INTRODUCTION

At present, the Internet is developing rapidly, and people can quickly obtain image information through the Internet. However, the Internet is an open platform. Images are easily stolen or attacked during transmission. Digital image encryption is an important way to protect image information, and it is of great significance to protect the transmission security of digital images. With the development of chaos theory [1–4], due to its pseudo-randomness, initial value sensitivity, parameter sensitivity, ergodicity and unpredictability of chaotic systems, image encryption using chaos theory has achieved novel results [5–7]. Image encryption based on chaos theory has become a hot spot in computer science and cryptography [8–13]. This makes the transmission of digital images on the network more secure, and there are many encryption algorithms to choose from [14–18].

Nowadays, many encryption algorithms of chaotic systems require multiple matrices processing to obtain better encryption effect and security. For example, the encryption method of DNA encoding [19,20] needs to convert the original image into 8-bit binary [21,22] and then encode it, or the encryption method based on bit position, which needs to be processed after the original matrix is transformed into 8 matrices in the encryption process. The algorithm proposed by ref. [23] also needs to process more than 2 rounds to obtain better diffusion effect and safety, which leads to higher time complexity. In addition, some traditional image encryption is executed sequentially [24,25]. The encrypted pixels in this way are related to their neighbouring pixels and chaotic interference values. Random execution can hide the encryption process to a certain extent, using random locations of pixels and chaotic interference values.

For basic chaotic systems such as logistic mapping and sine mapping, there are problems with too small initial value range and control parameter range.[26,27] We propose to combine logical mapping and sine mapping to generate a two-dimensional chaotic system and increase the complexity of the chaotic system to avoid this problem. This paper proposes to
combine logistic mapping and sine mapping to produce a one-dimensional complex chaotic system to expand the range of initial values and control parameters, which is more sensitive than common basic chaotic systems.

Based on the research and comparison of existing technologies, a high-sensitivity synchronization processing algorithm is proposed. First, based on the common logistic mapping and sine mapping, a new chaotic system is proposed, which has a larger control parameter range and initial value range than common chaotic systems. Then a highly sensitive sequence is generated by the proposed new chaotic system improved one-dimensional compound sine (I1DCSL), which is used to locate the position of the pixel to be changed each time. Taking the central axis and the horizontal axis of the image as the symmetry line, the image is scrambled and diffused in four directions at the same time. According to the different operation matrix, the final positions of the pixels in the four directions are not symmetrical, and are randomly distributed according to the chaotic sequence, and the chaotic interference values used for diffusion in different directions are also different. The key sensitivity, histogram analysis, information entropy, number of pixel change rate (NPCR), unified average changed intensity (UACI) and other experiments have proved the safety and reliability of the algorithm.

The rest of this article is organized as follows. Section 2 introduces the basic chaotic system, the new chaotic system called I1DCSL. Section 3 introduces the proposed encryption algorithm. Section 4 presents simulation tests and results. Finally, section 5 is the conclusion of this article.

2  |  BACKGROUND

2.1  |  Logistic map

Logistic chaotic system is one of the basic chaotic systems. The formula is simple but has complicated chaotic behaviour. The formula is shown in Equation (1).

\[ x_{n+1} = f(x_n, u) = ux_n(1-x_n). \]  

(1)

Among them, initial value \( x_0 \in (0, 1) \), when the parameter \( u \in [3.5699456, 4] \) the logistic mapping is a chaotic state, and the generated sequence \( x_n \) is aperiodic, non-convergent and very sensitive to the initial value.

In Figure 1(a), you can see the bifurcation diagram of the logistic chaotic system. It can be seen that when \( u \) belongs to the interval of \([3.5699456, 4]\), the logistic chaotic system in the bifurcation diagram is in a chaotic state, and \( u \) belongs to \([0, 3.57]\), the chaotic system has periodicity and converges to a specific value. It can be seen in Figure 1(c) that the value of Lyapunov exponent (LE) [28] is greater than 0 in the interval where \( u \) belongs to \([3.5699456, 4]\). The LE provides a description of the average separation rate and dispersion of two trajectories with similar initial values. Therefore, a full LE value indicates the presence of chaotic behaviour, and a larger LE value indicates the presence of more obvious chaotic behaviour.

2.2  |  Sine map

The Sine map is also one of the basic chaotic systems. When the control parameters are appropriate, it presents a pseudo-random numerical distribution with an infinite period. The formula is as described in Equation (2).

\[ x_{n+1} = g(x_n, \eta) = \eta \sin(\pi x_n). \]  

(2)

Among them, \( \eta \) is the control parameter, \( \eta \in (0, 1) \), \( x_0 \) is the initial value of the chaotic system, \( x_0 \in (0, 1) \), \( \eta \) is the pseudo-random number generated by the chaotic system and \( x_0 \in (0, 1) \).

Figure 1(b) is the bifurcation diagram of the sine map. It can be seen that the sine map is in a chaotic state when \( \eta \in [0.8, 1] \). Figure 1(d) shows that when \( \eta \) is in this interval, LE is greater than 0, indicating that the chaotic sequence mapped by the sine in this interval is aperiodic and does not converge.

2.3  |  Improved one-dimensional composite sine mapping

After analysing the logistic chaotic system and the sine chaotic system, it can be found that both have the characteristics of complex dynamic behaviour. However, the control parameters that make the chaotic system in a chaotic state are in the range between \([3.7, 4]\) and \([0.8, 1]\), which makes the range of the secret key for image encryption too small and difficult to resist brute force cracking.

Therefore, on the basis of logistic mapping and Sine mapping, a new chaotic system, I1DCSL chaotic system, is proposed. For the convenience of expression, the new chaotic system is called DCSL in the following. The DCSL chaotic system nests logistic into the composite sine map, uses the characteristics of the periodic chaos of the sine map to expand the range of control parameters, and the composite sine map increases the complexity of the chaotic system, and uses the remainder function mod to control the generated sequence \( x_n \in [0, 1] \), to generate a pseudo-random sequence between 0 and 1. The mathematical formula of DCSL chaotic system is defined in Equation (3).

\[ x = \bar{\lambda} \sin(\pi (\sin(\pi (2 - \bar{\lambda}(7 - \bar{\lambda}x) x)))) \mod 1. \]  

(3)

Among them, \( \bar{\lambda} \in [0.4356, +\infty) \), initial value \( x_0 \in (0, +\infty) \), \( x_0 \in (0, 1) \), through the restriction of the remainder function, the chaotic sequence is restricted to \((0, 1)\).

2.3.1  |  Dimensional compound sine’s Lyapunov exponent and bifurcation diagram

In Figure 2, the trajectory of the DCSL chaotic system is drawn with a bifurcation. In Figure 2(a), it can be seen that there is a periodic movement at the beginning, and the period gradually increases with the increase of the control parameter \( \bar{\lambda} \), and the obtained \( x_n \) is a chaotic point between \((0, 1)\). In Figure 2(b), it can be seen that as \( \bar{\lambda} > 0.5 \), the chaotic system has
been in a chaotic state. It shows that with the increase of $\lambda$, the chaotic system evolves into pseudo-random motion, which has the characteristics of pseudo-randomness of the chaotic system.

Figure 3 is the relationship diagram of the Lyapunov exponent (LE) that varies with the control parameter $\lambda$. Through observation, it can be seen in (a) that when the control parameter $\lambda$ is greater than 0.4356, a positive LE is obtained. As the parameter $\lambda$ increases, the LE gradually becomes larger. It can be seen from (b) and (c) that as $\lambda$ increases, Lyapunov presents a positive increase. It shows that with the increase of the control parameter $u$, the LE value of the DCSL chaotic system is gradually greater than 0, while the LE value indicates the presence of chaotic behaviour, and a larger LE value indicates more obvious chaotic behaviour, that is, as $u$ increases, DCSL has obviously chaotic behaviour.

2.3.2 Sample entropy

Sample entropy [29] is a time series complexity evaluation method, which quantitatively describes the self-similarity in the sequence generated by the dynamic system. For time series $\{x_1, x_2, \ldots, x_n\}$ of length $N$, template vector $X_{m}(i) = \{x_i, x_{i+1}, \ldots, x_{i+m-1}\}$, dimension $m$, and acceptance tolerance $r$, The formula of SE is shown in Equation (4).

$$SE(m, r, N) = -\log \frac{A}{B}. \quad (4)$$

Among them, $A$ and $B$ respectively represent the number of vectors $d[X_{m+1}(i), X_{m+1}(j)] < r$ and $d[X_{m}(i), X_{m}(j)] < r$. $d[X_{m}(i), X_{m}(j)]$ is the Chebyshev distance between $X_{m}(i)$ and $X_{m}(j)$. The larger the SE, the lower the regularity, so there is a higher complexity. We set $m = 2$ and $r = 0.2 \times \text{std} [18]$, where $\text{std}$ is the standard deviation of the time series.

Figure 4 is a comparison diagram of the SE results of logistic mapping, sine mapping and DCSL mapping. Figure 4(a) is the SE comparison chart where the control parameter belongs to (0, 4]. (b) is the SE comparison chart where the control parameter belongs to (0, 8], and evenly take points for display. Through comparison, it can be seen that the output generated by DCSL has a higher SE value, which means that the DCSL chaotic system can generate chaotic sequences with higher complexity.
2.3.3 | Sensitivity test

We evaluate the sensitivity of the initial value of DCSL based on the number of iterations. By choosing two initial values with extremely small differences, DCSL is used to generate two chaotic sequences, and the trajectories of the two sequences are compared. Figure 5 shows the trajectory diagrams of the chaotic sequence $Y_1$ and $Y_2$ obtained by 40 iterations with initial values of $x_1 = 0.542142133213167$, $x_2 = 0.542142133213168$, which can be regarded as indicators of initial sensitivity. The results show that the trajectories of the two sequences are completely different after a brief overlap of the chaotic sequences, indicating that the DCSL chaotic system is sensitive to the initial value.

2.3.4 | NIST test

The NIST test includes 15 tests. It is used to estimate the randomness of a binary sequence. Each test is given a significant row level $\alpha$ ($\alpha = 0.001$), and the probability value $(P-value)$ is obtained after the statistical result. If $P-value \geq \alpha$, the sequence is non-random. For random sequences, $P-value$ should be greater than $\alpha$. Table 1 lists the results of the NIST test of the DCSL chaotic system. It is obvious that the DCSL chaotic system has passed these tests, that is, the chaotic system is random.

2.4 | Joseph ring

The Joseph ring used here is a ring sequence that cyclically moves. In this article, two chaotic sequences are added to the ring, which are used to control the starting position, the step length each time, and the forward direction. The calculation process of the three control sequences is shown in Equations (10) and (12). As shown in the Figure 6, there are three sequences respectively, namely Direction, Initial position and Step length, each time through the Joseph ring, a high sensitivity value is obtained. By traversing the entire Joseph ring, a complete high sensitivity sequence is obtained. Figures 7 and 8
FIGURE 3  Lyapunov exponent of dimensional compound sine

(a) LE of $\lambda \in (0, 4]$  
(b) LE of $\lambda \in (4, 8]$  
(c) LE of $\lambda \in (0, 1000]$  

FIGURE 4  Sample entropy of dimensional compound sine

(a) SE of control parameter $\in (0, 4]$  
(b) SE of control parameter $\in (0, 8]$
are clockwise and counter clockwise acquisition of high sensitivity values, the left side is the acquisition process and the right side is the processed Joseph ring. For example, we have an initial Joseph ring, the sequence is shown in Figure 6, and there is a sequence Direction\{0.75, 0.12, …\} that controls the direction, and a sequence Position\{8, 2, …\} that controls the initial position controls, and a control step length sequence Step\{3, 1, …\}. The first high-sensitivity value is obtained through the operation in Figure 7, Direction(1) > 0.5, so the clockwise traversal is performed. The initial position is 8 and the step size is 3. The second high-sensitivity value is obtained through the operation in Figure 8, Direction(2) < 0.5, so the counter clockwise traversal is performed. The initial position is 2 and the step size is 1.

2.5 Improved Fisheryz scrambling diffusion

The traditional Fisher-Yezi scrambling used in image encryption is to obtain random pixels in a row or column in line units to scrambling the image. This article improves the Fishery Scrambling, and expands the scrambling range to the entire image matrix, instead of confining it to one row or one column. When transposing each pixel, the diffusion process is performed at the same time. The transposition of each pixel combines the high-sensitivity random value generated by the Joseph ring to diffuse and transpose the three pixels. And based on the four directions of the image, the entire image is traversed once. As shown in Figures 9 and 10, you can see the process of improving Fisher’s scrambling and diffusion for one pixel.
3 | THE PROPOSED IMAGE ENCRYPTION ALGORITHM

3.1 | Encryption process

Based on the DCSL chaotic system, this paper proposes a secure image encryption algorithm. Different from the traditional algorithm, this algorithm adopts the method of synchronous scrambling diffusion crossover, and introduces the non-linear diffusion value into the encryption process. Get good performance through one-time encryption. The encryption flow chart is shown in Figure 11. The encryption and decryption process are as follows:

Input: The plain image \( P \) which is size of \( M \) rows and \( N \) columns.

Step 1: For the original image \( P \), use SHA-512 to generate a 512-bit key \( K \) and convert every 8 bits of the key \( K \) into a decimal number to obtain \( K' \), which belongs to \([0, 255]\) and the length of the array \( K' \) is 64. The initial value \( x_i \) is obtained by Equation (5), the control parameter \( \lambda_i \) is obtained by Equation (6), the control parameter \( x_0 \) is obtained by Equation (7), and the control parameter \( u_0 \) is obtained by Equation (8).
Among them, $i=1, 2, 3, 4, 5, 6$ and $b=7, 8, 9, 10$.

$$x_i = \frac{\sum_{j=1}^{8} x_j + (i-1) \times 8}{255 \times 8} + \frac{\sum_{j=1}^{8} (i-1) + (i-1) \times 8}{(255 \times 8)^2}, (i=1, 2, 3, 4, 5, 6).$$

\( (5) \)

$$\lambda_i = K'_i (i=1, 2, 3, 4, 5, 6) \prod_{j=1}^{8} (i-1) \times 8 \prod_{k=1}^{8} (i-1) \times 8 \prod_{l=1}^{8} (i-1) \times 8 \prod_{m=1}^{8} (i-1) \times 8$$

\( \prod_{j=1}^{8} (i-1) \times 8 \prod_{k=1}^{8} (i-1) \times 8 \prod_{l=1}^{8} (i-1) \times 8 \prod_{m=1}^{8} (i-1) \times 8 (i=1, 2, 3, 4, 5, 6). \)\( \)\( (6) \)

$$x_t = \frac{\sum_{j=1}^{16+8(i-1) \times 16} x_j + (i-1) \times 8}{255 \times 16}, (b=7, 8, 9, 10).$$

\( (7) \)

$$\lambda_b = \frac{(255 \times 16 \times (1-x_b))}{16} + (1 + x_b) / 3, (b=7, 8, 9, 10).$$

\( (8) \)

Step 2: Substitute the 10 control parameters $\lambda$ and the initial value $x$ into the Equation (3) iterations and discard the results of the first 1000 iterations to avoid transient effects and generate 10 chaotic sequences. The length of the chaotic sequence $J_1, J_2, J_3$ is $M$, and the length of $J_4, J_5, J_6$ is $N$. The length of the chaotic
sequence $A_1, A_2, A_3, A_4$ is $M \times N$, $M$ is the height of the image, and $N$ is the width of the image.

Step 3: Get the sorted sequence by processing $J_1, J_2$ through Equation (9). Processing $J_i$ ($i = 3, 4, 5, 6$) by Equation (10) to obtain a pseudo-random sequence $J'_i$ with a value between [0, 255]. Use Equation (11) to process $A_1, A_2, A_3, A_4$ sequence.

$$J'_i = \text{sort}(J_i), (i = 1, 4).$$  

$$J'_i = \left\lfloor \left( J_i \times 10^8 \right) \text{mod} M + 1 \right\rfloor, (i = 2, 5).$$  

$$J'_i = \left\lfloor \left( J_i \times 10^8 \right) \text{mod} N + 1 \right\rfloor, (i = 3, 6).$$  

$$A_i = \text{reshape}(A_i, M, N), (i = 1, 2, 3, 4).$$  

Where, the $\text{sort()}$ function is a sorting function. In $S = \text{sort}(B)$, the values in $B$ will be sorted, and the sorted sequence will be in ascending order. Store the original position of each value in the sequence after sorting in $S$. And the $\text{reshape()}$ function is a function to resize the matrix. In Equation (11), it adjusts the one-dimensional matrix $A_i$ to a matrix with $M$ rows and $N$ columns.

Step 4: $J'_1, J'_4$ are the original pseudorandom chaotic sequence. Connect the pseudo-random sequence end to end. $J'_2$ and $J'_5$ are respectively used to locate the starting position of pseudorandom chaotic sequence selection. $J'_3$ and $J'_6$ are the step lengths when the coordinates are selected for pseudorandom chaotic sequence selection. And the forward direction is obtained by Equation (12). When $d_i > 0.5$, the pseudorandom chaotic sequence moves clockwise, and when $d_i \leq 0.5$, the pseudorandom chaotic sequence moves counter clockwise, where ($i = 1, 2$). Are the directions of the two pseudorandom chaotic sequences $J'_1$ and $J'_4$ respectively. Take the value of the position after each movement and put it into a one-dimensional matrix. $R, C$ finally produce two matrices, $R$ with length $M$ and $C$ with length $N$.

$$\begin{cases}  
  d_1 = (J_3 \times 10^8) \text{mod} M - (J_3 \times 10^8) \text{mod} M  
  
  d_2 = (J_6 \times 10^8) \text{mod} N - (J_6 \times 10^8) \text{mod} N 
\end{cases}.$$  

Step 5: The image is processed by a multi-directional synchronous cross-processing algorithm, in which highly sensitive random sequences $R$ and $C$ are used simultaneously. The
synchronization starts from the upper left corner, upper right corner, lower left corner, and lower right corner respectively. The image $P$ is processed through Equations (13)–(16). After completing one encryption, the encryption ends and an encrypted image $P'$ is obtained. The synchronous cross-processing algorithm can be seen in Figures 9 and 10.

\[
\begin{align*}
\text{temp} &= (P(m, n) \oplus (A_1(m, n) \times 10^3) \mod 256 \\
&\quad + \lfloor (A_1(R(m), C(n)) \times (1 - A_1(R(m)), C(n))) \times 10^3 \mod 256 \rfloor) \ mod 256 \\
\{P(m, n) = P\left([M \times A_1(m, n) + 1], [N \times A_2(m, n) + 1]\right) \\
&= \left\lfloor \frac{(P(R(m), C(n)) + A_1(R(m), C(n)) \times 10^3 + R(m)}{M} \times X\right\rfloor \mod 256 \\
\{P(R(m), C(n)) = \text{temp} \\
&= (P(m, N - n + 1) \oplus (A_2(m, N - n + 1) \times 10^3) \mod 256 \\
&\quad \lfloor [A_1(R(m), C(N - n + 1)) \times (1 - A_1(R(m), C(N - n + 1))] \times 10^3 \mod 256 \rfloor) \ mod 256 \\
\{P(m, N - n + 1) = P\left([M \times A_3(m, N - n + 1) + 1], [N \times A_4(m, N - n + 1) + 1]\right) \\
&= \left\lfloor \frac{(P(R(m), C(N - n + 1) + A_2(R(m), C(N - n + 1)) \times X + R(m)) / M \times (1 - R(m)) / M \times X^{10^3}}{N} \right\rfloor \mod 256 \\
\{P(R(m), C(N - n + 1)) = \text{temp} \\
&= (P(m - m + 1, n) \oplus (A_3(M - m + 1, n) \times 10^3) \mod 256 \\
&\quad + [A_1(R(m - m + 1), C(n)) \times (1 - A_3(R(M - m + 1), C(n))] \times 10^3 \mod 256 \rfloor) \ mod 256 \\
\{P(M - m + 1, n) = P\left([M \times A_1(M - m + 1, n) + 1], [N \times A_3(M - m + 1, n) + 1]\right) \\
&= \left\lfloor \frac{(P(R(M - m + 1), C(n)) + A_1(R(M - m + 1), C(n)) \times X + R(M - m + 1)) / M \times (1 - R(M - m + 1)) / M \times X^{10^3}}{N} \right\rfloor \mod 256 \\
\{P(R(M - m + 1), C(n)) = \text{temp} \\
&= (P(M - m + 1, N - n + 1) \oplus (A_4(M - m + 1, N - n + 1) \times 10^3) \mod 256 \\
&\quad + [A_4(R(M - m + 1), C(N - n + 1)) \times (1 - A_4(R(M - m + 1), C(N - n + 1))] \times 10^3 \mod 256 \rfloor) \ mod 256 \\
\{P(M - m + 1, N - n + 1) = P\left([M \times A_2(M - m + 1, N - n + 1) + 1], [N \times A_4(M - m + 1, N - n + 1) + 1]\right) \\
&= \left\lfloor \frac{(P(R(M - m + 1), C(N - n + 1)) + A_4(R(M - m + 1), C(N - n + 1)) \times X + R(M - m + 1)) / M \times (1 - R(M - m + 1)) / M \times X^{10^3}}{N} \right\rfloor \mod 256 \\
\{P(R(M - m + 1), C(N - n + 1)) = \text{temp} \\
\end{align*}
\]

In this algorithm, $m$ belongs to $M$, $n$ belongs to $N$, and $m, n$ increases from $1$ to $M$ or $N$ respectively. Taking $m, n$ as the benchmark, each time the row is incremented by $1$, the column is incremented to $N$. Until $m$ is incremented to $M$, the entire image is traversed. At this point, the entire image has been fully encrypted

Output: Encrypted image $P'$.

3.2 Decryption process

Input: Encrypted image $P'$ and secret key $K$.

Step 1–Step 4: Pre-processing the secret key and pseudo-random sequence, the process is the same as the encryption process.

Step 5: Contrary to the steps when encrypting images, use $m$ and $n$ to decrease from $M$ and $N$ to 1 respectively. Perform the inverse operation with the steps starting from the lower right corner, the lower left corner, the upper right corner, and the upper left corner, respectively, as shown in Equations (17)–(20) to get Decrypt the image $P$.

Output: Decrypt image $P$.

\[
\begin{align*}
\text{temp} &= \left\lfloor \frac{P(R(M - m + 1), C(N - n + 1)) + A_4(R(M - m + 1), C(N - n + 1)) \times (1 - A_4(R(M - m + 1), C(N - n + 1))] \times 10^3 \mod 256 \\
&= \left\lfloor \frac{X + R(M - m + 1)) / M \times (1 - R(M - m + 1)) / M \times X^{10^3}}{N} \right\rfloor \mod 256 \\
\end{align*}
\]

\[
\begin{align*}
\{P(M - m + 1, N - n + 1) = P\left([M \times A_2(M - m + 1, N - n + 1) + 1], [N \times A_4(M - m + 1, N - n + 1) + 1]\right) \\
&= \left\lfloor \frac{(P(R(M - m + 1), C(N - n + 1)) + A_4(R(M - m + 1), C(N - n + 1)) \times X + R(M - m + 1)) / M \times (1 - R(M - m + 1)) / M \times X^{10^3}}{N} \right\rfloor \mod 256 \\
\{P(R(M - m + 1), C(N - n + 1)) = \text{temp} \\
\end{align*}
\]
PERFORMANCE ANALYSIS

Simulation results

\begin{align*}
P(R(M - m + 1), C(N - n + 1)) \\
= [P(M \times A_2(M - m + 1, N - n + 1) + 1], \\
[N \times A_3(M - m + 1, N - n + 1) + 1]) \\
- A_1(R(M - m + 1), C(N - n + 1)) \times 10^9 \\
- R(M - m + 1) / M \times (1 - R(M - m + 1) / M) \times 10^9 \\
- C(N - n + 1) / N \times (1 - C(N - n + 1) / N) \\
\times 10^9] \mod 256
\end{align*}

\begin{align*}
\{P(M \times A_2(M - m + 1, N - n + 1) + 1], \\
[N \times A_3(M - m + 1, N - n + 1) + 1]) = P(R(M - m + 1), C(N - n + 1))
\end{align*}

\begin{align*}
\{P(R(M - m + 1), C(N - n + 1)) = temp
\end{align*}

\begin{align*}
\text{temp} = \{[P(R(m), C(n)) \\
- (A_1(R(m), C(n))) \\
\times (1 - A_1(R(m), C(n))) \times 10^5 \mod 256 \}
\end{align*}

\begin{align*}
\{P(M \times A_1(m, n) + 1], \\
[N \times A_2(m, n) + 1]) = P(R(m), C(n))
\end{align*}

\begin{align*}
\{P(R(m), C(n)) = temp.
\end{align*}

\begin{align*}
P(R(M - m + 1), C(n)) \\
= [P(M \times A_1(M - m + 1, n + 1) + 1], \\
[N \times A_3(M - m + 1, n + 1) + 1]) \\
- A_1(R(M - m + 1, C(n)) \times 10^9 \\
- R(M - m + 1) / M \times (1 - R(M - m + 1) / M) \times 10^9 \\
- C(N - n + 1) / N \times (1 - C(N - n + 1) / N) \\
\times 10^9]
\end{align*}

\begin{align*}
\{P(M \times A_1(M - m + 1, n + 1) + 1], \\
[N \times A_3(M - m + 1, n + 1) + 1]) = P(R(M - m + 1), C(n))
\end{align*}

\begin{align*}
\{P(R(M - m + 1), C(n)) = temp.
\end{align*}

4 PERFORMANCE ANALYSIS

This section analyses multiple aspects of encryption algorithms to evaluate performance, such as key space, histogram analysis, sensitivity analysis, correlation analysis, NPCR, UACI, information entropy etc. The analysis results show that the algorithm is safe and reliable. The analysis results of the simulation test are as follows:

4.1 Simulation results

Figure 12 shows the encryption results of the Lena graph and the black graph of 512 × 512 size.

4.2 Key space

The size of the secret key space can be expressed as the ability of the algorithm to resist brute force attacks. The larger the key space, the greater the possibility of resisting brute force attacks. In this algorithm, the secret key K is a 512-bit binary value. Therefore, the size of the key space should be $2^{512}$. Therefore, the algorithm can resist brute force attacks.

4.3 Sensitivity analysis

A good algorithm should be sensitive to secret keys. We randomly change one bit of the secret key K to decrypt the encrypted Lena image, and the decrypted image obtained is shown in Figure 13.
The histogram can be used to describe the distribution of image pixel values. Figure 14 shows the histogram of the original image and its cipher image. It can be observed that the histogram of the cipher image is evenly distributed compared with the histogram of the original image, so the algorithm can resist known cipher attacks.

4.5 Correlation between adjacent pixels

The correlation between adjacent pixels includes horizontal, vertical and diagonal correlation. Low correlation can effectively prevent statistical attacks [30]. This paper randomly selects 10,000 pairs of adjacent pixels from the ordinary image P and the encrypted image, and calculates the correlation coefficient of each pair of pixels through the Equation (21).

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

(21)

Table 2 is a comparison between the correlation of the original image and the correlation coefficient of the image after being encrypted by the encryption algorithm. Table 3 is the
FIGURE 14 Histograms of plain images and ciphered images

comparison between the image correlation coefficients of the encryption algorithm proposed in this paper and other algorithms. It can be seen that the encryption algorithm proposed in this paper fully reduces the correlation between the pixels of the image, and can effectively prevent statistical attacks.

Figure 15(a) shows the pixel value distribution of the Lena map in space. It is obvious that there are many peaks and valleys. Because the pixel distribution has a sense of hierarchy, it carries a lot of useful information. In image encryption, all pixels of the encrypted image should be evenly distributed between [0, 255], as shown in (b), it is not observed that the image carries effective information.

4.6 Differential attack

Differential attacks are effective security attacks. It focuses on establishing the relationship between the two by analysing the
impact of changes in ordinary images on the encryption results. A good encryption algorithm should have diffusion characteristics to resist differential attacks. The diffusion characteristic is the ability to diffuse the changes in the ordinary image to the entire encrypted image. We use the NPCR and UACI to evaluate the diffusibility of encryption algorithms [28]. We use the encrypted image $C$ to evaluate the diffusion of encryption algorithms. We use the entire encrypted image. We use the NPCR and UACI to evaluate the impact of changes in ordinary images on the encryption results. The diffusion characteristic is the ability to diffuse the changes in the ordinary image to the entire encrypted image. We use the NPCR and UACI to evaluate the diffusibility of encryption algorithms [28]. We use the encrypted image $C_1$ and the encrypted image $C_2$ that is encrypted after changing a bit of the original image. The calculation formula is shown in Equation (22).

$$\text{NPCR}(C_1, C_2) = \frac{1}{W \times H} \sum_{i,j} D(i,j) \times 100\%$$

$$\text{UACI}(C_1, C_2) = \frac{1}{W \times H} \left[ \sum_{i,j} \left| \frac{C_1(i,j) - C_2(i,j)}{255} \right| \right] \times 100\%$$

(22)

Where $W \times H$ is the number of pixels and the calculation method of $D$ is shown in Equation (23)

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

(23)

According to ref. [28], if the NPCR value of an image’s encryption scheme is higher than the critical value $N_{\rho}^*$, and its UACI score falls within the critical interval $(U_{\rho}^{\text{min}}, U_{\rho}^{\text{max}})$ of UACI, the image passes the diffusion test. The calculation formula for the significance level $\rho$, $N_{\rho}^*$, $(U_{\rho}^{\text{min}}, U_{\rho}^{\text{max}})$ is as follows:

$$N_{\rho}^* = \frac{F - \Phi^{-1}(\rho)}{\sqrt{M \times N}}$$

$$\begin{cases} U_{\rho}^{\text{min}} = \mu_u - \Phi^{-1}(\rho/2)\sigma_u \\ U_{\rho}^{\text{max}} = \mu_u + \Phi^{-1}(\rho/2)\sigma_u \end{cases}$$

(24)

$$\sigma_u = \frac{(F + 2)(F^2 + 2F + 3)}{18(F + 1)^2}$$

(25)

Where $F$ represents the largest pixel value, $M \times N$ represents the size of the image. $\Phi^{-1}(\rho)$ is the inverse cumulative density function of the standard normal distribution. $\mu_u = (F + 2)/(3F + 3)$, this is the ideal value of UACI.

We tested 25 grayscale images in the USC-SIPI database, six images with a size of $256 \times 256$, sixteen images with a size of $512 \times 512$, and three images with a size of $1024 \times 1024$. According to the theory of ref. [28], set $\rho$ to 0.005. For an image with a size of 256, $N_{\rho}^* = 99.5693\%$, $[U_{\rho}^{\text{min}}, U_{\rho}^{\text{max}}] = [33.2824\%, 33.6447\%]$, for the image of size 512, $N_{\rho}^* = 99.5893\%$, $[U_{\rho}^{\text{min}}, U_{\rho}^{\text{max}}] = [33.3730\%, 33.5541\%]$, for the image of size 1024, $N_{\rho}^* = 99.5994\%$, $[U_{\rho}^{\text{min}}, U_{\rho}^{\text{max}}] = [33.4138\%, 33.5088\%]$. Tables 4 and 5 respectively show the NPCR and UACI results of several encryption schemes. In order to make the result closer to the real situation, we encrypt each image 100 times to calculate the average NPCR and UACI values. Through comparison, we can find that the 25 images encrypted by the encryption algorithm proposed in this article all pass the test, while some of the images in other encryption schemes cannot pass the test. This shows that the algorithm in this paper has good security and ability to resist differential attacks.

### 4.7 Information entropy

Information entropy is used to describe the statistical measure of randomness of disordered information. For grayscale images,
TABLE 3  Comparison of the correlation coefficients of images

| Image      | Proposed | Ref. [31] | Ref. [32] |
|------------|----------|-----------|-----------|
|            | H        | V         | D         | H        | V         | D         |
| Lena 512   | 0.0032   | 0.0021    | −0.0022   | −0.0285  | 0.0014    | 0.0013    | 0.0008    | 0.0021    | 0.0005    |
| Baboon 512 | −0.0013  | 0.0004    | 0.0044    | 0.0013   | −0.0281   | 0.0128    | −         | −         | −         |
| Barb 512   | −0.0016  | 0.0072    | −0.0004   | −0.0206  | −0.0314   | 0.0220    | 0.0016    | 0.0048    | 0.0024    |
| Lena 256   | −0.0006  | −0.0018   | 0.0051    | −         | −         | −         | 0.0019    | 0.0023    | 0.0011    |
| Pepper 256 | −0.0002  | −0.0003   | −0.0015   | −         | −         | −         | 0.0258    | 0.0037    | 0.0079    |
| Cameraman 256 | 0.0006 | 0.0060    | 0.0030    | 0.0139   | 0.0034    | 0.0107    | 0.0132    | 0.0198    | 0.0032    |

TABLE 4  NPCR evaluation

| Image      | Proposed | Cao et al. [33] | Hua and Zhou [34] | Xu et al. [35] | Zhou et al. [36] |
|------------|----------|-----------------|-------------------|----------------|-----------------|
| 5.1.09     | 99.6097  | 99.5975         | 99.6124           | 99.6246        | 99.5575         |
| 5.1.10     | 99.6102  | 99.3588         | 99.5972           | 0.0092         | 99.5544         |
| 5.1.11     | 99.6100  | 99.6033         | 99.5956           | 99.6445        | 99.8123         |
| 5.1.12     | 99.6076  | 99.5651         | 99.6017           | 99.5972        | 99.6109         |
| 5.1.13     | 99.6046  | 99.5789         | 99.6552           | 99.6582        | 99.7421         |
| 5.1.14     | 99.6092  | 99.6765         | 99.6002           | 99.5987        | 99.6933         |
| 5.2.08     | 99.6090  | 99.6037         | 99.6200           | 99.6216        | 99.6101         |
| 5.2.09     | 99.6094  | 99.6029         | 99.6208           | 99.6048        | 99.7025         |
| 5.2.10     | 99.6076  | 99.6124         | 99.3968           | 99.5861        | 99.6120         |
| 7.1.01     | 99.6111  | 99.6082         | 99.6181           | 99.6162        | 99.5190         |
| 7.1.02     | 99.6089  | 99.6174         | 99.6140           | 99.6025        | 99.7200         |
| 7.1.03     | 99.6089  | 99.6120         | 99.6166           | 99.5998        | 99.4072         |
| 7.1.04     | 99.6081  | 99.5911         | 99.6227           | 99.6033        | 99.6037         |
| 7.1.05     | 99.6090  | 99.6178         | 99.596            | 99.6307        | 99.4572         |
| 7.1.06     | 99.6098  | 99.6174         | 99.6212           | 99.6105        | 99.5213         |
| 7.1.07     | 99.6100  | 99.5922         | 99.6113           | 99.6029        | 99.5007         |
| 7.1.08     | 99.6092  | 99.6056         | 99.5914           | 99.6120        | 99.6902         |
| 7.1.09     | 99.6099  | 99.6086         | 99.6067           | 99.6048        | 99.7181         |
| 7.1.10     | 99.6087  | 99.5941         | 99.6056           | 99.6212        | 99.5163         |
| Boat.512   | 99.6075  | 99.6101         | 99.6021           | 99.6067        | 99.7128         |
| Gray21.512 | 99.6104  | 99.6159         | 99.6239           | 99.6094        | 99.6120         |
| Ruler.512  | 99.6095  | 99.6212         | 99.5930           | 99.6113        | 99.3118         |
| 5.3.01     | 99.6094  | 99.6072         | 99.6100           | 99.6116        | 99.6040         |
| 5.3.02     | 99.6095  | 99.6116         | 99.6129           | 99.6223        | 99.4789         |
| 7.2.01     | 99.6095  | 99.6204         | 99.5964           | 99.6042        | 99.7578         |
| Pass       | 25       | 24              | 24                | 23             | 15              |
| Mean       | 99.6091  | 99.6078         | 99.6097           | 99.62285       | 99.6010         |

the information entropy in an ideal state should be 8. The definition formula of information is shown in Equation (26).

\[ H(s) = \sum_{i=0}^{2^{N-1}} p(s_i) \log_2 \frac{1}{p(s_i)}. \]  

Where \( p(s_i) \) represents the probability of pixel \( s_i \). Table 6 is a comparison between the information entropy of the encrypted image of this article and other algorithms. It can be seen that the information entropy of this article is close to the ideal state 8, and the information described by the encrypted image is disordered enough to obtain useful information from it.
TABLE 5 UACI evaluation

| Image     | Proposed | Hua et al. [18] | Hua and Zhou [34] | Liu et al. [39] | Xu et al. [35] |
|-----------|----------|----------------|-------------------|----------------|----------------|
| 5.1.09    | 33.4403  | 33.3651        | 33.4652           | 33.5527        | 33.4425        |
| 5.1.10    | 33.4567  | 33.5240        | 33.4162           | 33.4390        | 33.4844        |
| 5.1.11    | 33.4519  | 33.5065        | 33.2152           | 33.4373        | 33.3609        |
| 5.1.12    | 33.4817  | 33.4875        | 33.5448           | 33.3488        | 33.3039        |
| 5.1.13    | 33.4731  | 33.4172        | 33.4162           | 33.4390        | 33.5133        |
| 5.1.14    | 33.4707  | 33.4973        | 33.5554           | 33.5133        | 33.5008        |
| 5.2.08    | 33.4778  | 33.4575        | 33.4377           | 33.5233        |                |
| 5.2.09    | 33.4622  | 33.4327        | 33.4175           | 33.4939        | 33.4834        |
| 5.2.10    | 33.4600  | 33.4154        | 33.4315           | 33.3888        | 33.4532        |
| 7.1.01    | 33.4667  | 33.4698        | 33.5150           | 33.5553        | 33.3369        |
| 7.1.02    | 33.4491  | 33.4632        | 33.5221           | 33.4342        | 33.4121        |
| 7.1.03    | 33.4751  | 33.4632        | 33.4777           | 33.4585        | 33.4970        |
| 7.1.04    | 33.4722  | 33.4996        | 33.5342           | 33.4830        | 33.4412        |
| 7.1.05    | 33.4583  | 33.4647        | 33.4757           | 33.4393        | 33.4753        |
| 7.1.06    | 33.4764  | 33.4416        | 33.5035           | 33.5634        | 33.4571        |
| 7.1.07    | 33.4709  | 33.3906        | 33.4317           | 33.5241        | 33.3844        |
| 7.1.08    | 33.4532  | 33.4029        | 33.4274           | 33.4251        | 33.3863        |
| 7.1.09    | 33.4422  | 33.4686        | 33.4452           | 33.4606        | 33.3879        |
| 7.1.10    | 33.4708  | 33.4434        | 33.4434           | 33.4119        | 33.4615        |
| Boat.512  | 33.4606  | 33.4472        | 33.4059           | 33.4993        | 33.4589        |
| Gray21.512| 33.4715  | 33.4781        | 33.4554           | 33.4634        | 33.3857        |
| Ruler.512 | 33.4855  | 33.3883        | 33.4795           | 33.5090        | 33.5253        |
| 5.3.01    | 33.4641  | 33.4683        | 33.4516           | 33.4698        | 33.5380        |
| 5.3.02    | 33.4642  | 33.4428        | 33.4579           | 33.4820        | 33.4525        |
| 7.2.01    | 33.4682  | 33.4688        | 33.4718           | 33.4878        | 33.4348        |
| Pass      | 25       | 25             | 24                | 23            | 22             |
| Mean      | 33.4650  | 33.4548        | 33.4563           | 33.4686        | 32.1035        |

TABLE 6 Information entropy of images

| Image     | Plain | Proposed | Ref. [31] |
|-----------|-------|----------|-----------|
| Lena 512  | 7.4475| 7.9994   | 7.9993    |
| Baboon 512| 7.3814| 7.9992   | 7.9993    |
| Barb 512  | 7.6321| 7.9993   | 7.9992    |
| Pepper 512| 7.3967| 7.9994   | –         |
| Lena 256  | 7.3785| 7.9973   | –         |
| Cameraman 256| 7.0097| 7.9976   | 7.9974    |

4.8 Local information entropy

Local information entropy can be used to analyse the uniformity of local image distribution. The definition of local information entropy is shown in Equation (27).

\[ H_{(k,T_B)}(S_k) = \frac{1}{k} \sum_{i=1}^{k} H(S_i) \]  \quad (27)

where \( S_k \) means randomly selecting \( kT_B \) pixels, \( H(S_k) \) from the image to represent the information entropy of \( S_k \) (see Equation (26)). It can be seen from the ref. [37] that when \( T_B = 1936 \), \( k = 30 \) and confidence \( \alpha \) is 0.01, \( H_{(k,T_B)}(S_k) \) should be in the range of \([7.901722822, 7.903215812]\).

Table 7 shows the local information entropy of the encrypted image proposed in this paper. It can be seen that several encrypted images have passed the local information entropy test. The results show that the local image distribution is sufficiently uniform.

4.9 Robustness analysis

During image transmission, it may be lost or subject to cropping attacks and noise attacks. Therefore, the encryption algorithm should have better robustness. Figure 16 shows the decrypted results of the encrypted image processed by the pruning attack and the noise attack. Experimental results show that the algorithm has good anti-cutting ability and anti-noise attack ability, and has good robustness.
### Table 7  |  Local information entropy

| Image          | Lena (512) | Barb (512) | Lena (256) | 5.1.11 | FullBlack (512) |
|----------------|------------|------------|------------|--------|-----------------|
| Local information entropy | 7.902678   | 7.902025   | 7.902540   | 7.902926 | 7.901168        |
| Pass or fail   | Pass       | Pass       | Pass       | Pass   | Pass            |

![Clipping attacks](image_url)

**Figure 16**  |  Comparison of the clipping attacks

### 4.10  |  $\chi^2$ test

Use $\chi^2$ to describe the degree of deviation of the image from the absolute uniform distribution. The formula is defined in Equation (28).

$$\chi^2 = \sum_{i=0}^{255} \frac{(p_i - \bar{p})^2}{\bar{p}}.$$  \hspace{1cm} (28)

where $\bar{p}$ represents the average frequency of all pixels, namely $(M \times N) / 256$, $p_i$ represents the frequency of pixel $i$ in the image. The smaller the $\chi^2$, the more uniform the pixel distribution.

Table 8 shows the $a$ value of normal image and encrypted image. It can be clearly seen that the $a$ value of the ciphertext image is much lower than that of the plaintext image, indicating that the ciphertext image has a higher degree of mixing distribution.

### 4.11  |  The speed performance

The algorithm encryption process only traverses the image once, and processes the image simultaneously in four directions. For a 256 $\times$ 256 Lena image, it takes 0.416718s to complete a complete encryption Table 9 is a comparison between the time consumed by this encryption algorithm and the time consumed by other encryption algorithms.

### Table 8  |  The results of $\chi^2$ test

| Image        | Lena (512) | Baboon (512) | Barb (512) | Lena (256) | Pepper (256) | Cameraman (256) |
|--------------|------------|--------------|------------|------------|--------------|-----------------|
| Plain        | 160001     | 180257       | 95548      | 48084      | 34921        | 110973          |
| Proposed     | 235.2227   | 281.1406     | 261.6660   | 245.8438   | 280.9453     | 217.3203        |
TABLE 9 Speed analysis between our proposed method and other chaotic-cryptosystems

| Algorithm | Encryption time (s) | Platform & system characteristics |
|-----------|---------------------|----------------------------------|
| Our proposed algorithm | 0.416718 s | Matlab R2017a, CPU 2.8GHZ, 8 GB memory |
| Ref. [38] | 2.8716 s | Matlab R2012b, CPU 2.93 GHz, 2 GB memory |
| Ref. [39] | 3.5015 s | – |
| Ref. [35] | 1.32 s | MATLAB 6.5, CPU 2.00 GHz, 2 GB memory |
| Ref. [40] | 1.120248 s | 2.53 GHz, i3/3GB/Windows 7 |

The encryption speed is highly dependent on CPU/GPU performance, RAM size and programming environment. Therefore, for different platform tests, the results only represent approximate comparison results.

5 | CONCLUSION

This paper proposes a compound sine chaotic map called H1DCSL. The evaluation shows the superiority of DCSL compared with other one-dimensional chaotic systems. And further designed a novel image encryption scheme based on DCSL. Make full use of the pseudo-random sequence generated by the chaotic system, and apply the chaotic system to image scrambling and diffusion. The simulation test results show that the encryption scheme can obtain a certain security ciphertext image, and the security evaluation shows that it is sensitive to the secret key and has a certain degree of resistance to reduction attacks and noise attacks. Compared with other schemes, this encryption scheme has better resistance to differential attacks. This work promotes the expansion of chaotic systems and the choice of encryption schemes. In the future, we will try to study the expansion of high-dimensional chaotic systems and design more secure encryption schemes.

ACKNOWLEDGEMENTS

This research is supported by the National Natural Science Foundation of China (No: 61672124), the Password Theory Project of the 13th Five-Year Plan National Cryptography Development Fund (No: MMJJ20170203), Liaoning Province Science and Technology Innovation Leading Talents Program Project (No: XLYC1802013), Key R&D Projects of Liaoning Province (No: 2019020105-JH2/103), Jinan City ‘20 universities’ Funding Projects Introducing Innovation Team Program (No: 2019GXRC031).

ORCID

Xingyuan Wang https://orcid.org/0000-0002-3844-3195
Maochen Zhang https://orcid.org/0000-0002-5796-0956

REFERENCES

1. Lian, S.: A block cipher based on chaotic neural networks. Neurocomputing 72(4), 1296–1301 (2009)
2. Zhang, Y., et al.: A new image encryption algorithm based on multiple chaos system. In 2008 International Symposium on Electronic Commerce and Security, Guangzhou, China, 347–350 (2008)
3. Xu, Q., et al.: An effective image encryption algorithm based on compressive sensing and 2D-SLIM. Opt. Lasers Eng. 134, 106178 (2020)
4. Kaur, M., et al.: Color image encryption using non-dominated sorting genetic algorithm with local chaotic search based 5D chaotic map. Futur. Gen. Comput. Syst. 107, 333–350 (2020)
5. Wang, X., Gao, S.: Image encryption algorithm for synchronously updating Boolean networks based on matrix semi-tensor product theory. Inf. Sci. (Ny) 507, 16–36 (2020)
6. Bouhous, A., Kermih, K.: Novel encryption method based on optical time-delay chaotic system and a wavelet for data transmission. Opt. Laser Technol. 108, 162–169 (2018)
7. Wang, M., et al.: A novel chaotic encryption scheme based on image segmentation and multiple diffusion models. Opt. Laser Technol. 108, 558–573 (2018)
8. Wang, X., Feng, L., Zhao, H.: Fast image encryption algorithm based on parallel computing system. Inf. Sci. (Ny) 486, 340–358 (2019)
9. Wang, M., et al.: Spatiotemporal chaos in cross coupled map lattice with dynamic coupling coefficient and its application in bit-level color image encryption. Chaos, Solitons Fractals 139, 110028 (2020)
10. Xian, Y., Wang, X.: Fractal sorting matrix and its application on chaotic image encryption. Inf. ences 547, 1154–1169 (2021)
11. Wang, X., Gao, S.: Image encryption algorithm based on the matrix semi-tensor product with a compound secret key produced by a Boolean network. Inf. Sci. (Ny) 539, 195–214 (2020)
12. Zhang Hao, Wang Xiao-qing, Wang Xing-yuan, Yan Peng-fei Novel multiple images encryption algorithm using CML system and DNA encoding. IET Image Process. 14(3), 518–529 (2020) . https://doi.org/10.1049/iet-ipr.2019.0771
13. Wang, X., Gu, S., Zhang, Y.: Novel image encryption algorithm based on cycle shift and chaotic system. Opt. Lasers Eng. 68, 126–134 (2015)
14. Yang Feifei, Mou Jun, Sun Kehui, Chu Ran Lossless image compression-encryption algorithm based on BP neural network and chaotic system. Multimed. Tools Appl. 79(27–28), 19963–19992 (2020) . https://doi.org/10.1007/s11042-020-08821-w
15. Chen, C., Sun, K., Xu, Q.: A color image encryption algorithm based on 2D-CIMM chaotic map. China Commun. 17(5), 12–20 (2020)
16. Xu, Q., Sun, K., Zhu, C.: A visually secure asymmetric image encryption scheme based on RSA algorithm and hyperchaotic map. Phys. Scr. 95(3), 35223 (2020)
17. Xu, Q., et al.: A fast image encryption algorithm based on compressive sensing and hyperchaotic map. Opt. Lasers Eng. 121, 203–214 (2019)
18. Hua, Z., Zhou, Y., Huang, H.: Cosine-transform-based chaotic system for image encryption. Inf. Sci. (Ny) 480, 403–419 (2019)
19. Wu, J., Liao, X., Yang, B.: Image encryption using 2D Hénon-Sine map and DNA approach. Signal Processing 153, 11–23 (2018)
20. Liu, H., Wang, X., kadir, A.: Image encryption using DNA complementary rule and chaotic maps. Appl. Soft Comput. 12(5), 1457–1466 (2012)
21. Teng, L., Wang, X.: A bit-level image encryption algorithm based on spatiotemporal chaotic system and self-adaptive. Opt. Commun. 285(20), 4048–4054 (2012)
22. Wang, T., Wang, M.: Hyperchaotic image encryption algorithm based on bit-level permutation and DNA encoding. Opt. Laser Technol. 132, 106355 (2020)
23. Wang, X., Guo, K.: A new image alternate encryption algorithm based on chaotic map. Nonlinear Dyn. 76(4), 1943–1950 (2014)
24. Zhang, Y., Wang, X.: A new image encryption algorithm based on non-adjacent coupled map lattices. Appl. Soft Comput. 26, 10–20 (2015)
25. Zhang, Y., Wang, S.: Spatiotemporal chaos in mixed linear–nonlinear coupled logistic map lattice. Phys. A Stat. Mech. its Appl. 402, 104–118 (2014)
26. Hua, Z., Zhou, Y.: Image encryption using 2D Logistic-adjusted-Sine map. Inf. Sci. (Ny) 339, 237–253 (2016)
27. Hua, Z., et al.: 2D Logistic-sine-coupling map for image encryption. Signal Processing 149, 148–161 (2018)
28. Mansouri, A., Wang, X.: A novel one-dimensional sine powered chaotic map and its application in a new image encryption scheme. Inf. Sci. (Ny) 520, 46–62 (2020)
29. Richman, J., Moorman, J.: Physiological time-series analysis using approximate entropy and sample entropy. Am. J. Physiol. Heart Circ. Physiol. 278, H2039–H2049 (2000)
30. Diaconu, A.-V.: Circular inter–intra pixels bit-level permutation and chaos-based image encryption. Inf. Sci. (Ny) 355–356, 314–327 (2016)
31. Chai, X.: An image encryption algorithm based on bit level Brownian motion and new chaotic systems. Multimed. Tools Appl. 76, 1159–1175 (2017)
32. Enayatifar, R., et al.: Image encryption using a synchronous permutation-diffusion technique. Opt. Lasers Eng. 90, 146–154 (2017)
33. Cao, C., Sun, K., Liu, W.: A novel bit-level image encryption algorithm based on 2D-LICM hyperchaotic map. Signal Processing 143, 122–133 (2018)
34. Hua, Z., Zhou, Y.: Design of image cipher using block-based scrambling and image filtering. Inf. Sci. (Ny) 396, 97–113 (2017)
35. Xu, L., et al.: A novel bit-level image encryption algorithm based on chaotic maps. Opt. Lasers Eng. 78, 17–25 (2016)
36. Zhou, Y., Bao, L., Chen, C.L.P.: A new 1D chaotic system for image encryption. Signal Processing 97, 172–182 (2014)
37. Wu, Y., et al.: Local Shannon entropy measure with statistical tests for image randomness. Inf. Sci. (Ny) 222, 323–342 (2013)
38. Liu, Y., et al.: Image encryption algorithm based on chaotic system and dynamic S-boxes composed of DNA sequences. Multimed. Tools Appl. 75(8), 4363–4382 (2016)
39. Farah, M.A.B., Farah, A., Farah, T.: An image encryption scheme based on a new hybrid chaotic map and optimized substitution box. Nonlinear Dyn. 99(1), 1–24 (2020)
40. Farah M. A. Ben, Guesmi R., Kachouri A., Samet M. A new design of cryptosystem based on S-box and chaotic permutation. Multimed. Tools Appl. 79(27-28), 19129–19150 (2020). https://doi.org/10.1007/s11042-020-08718-8

How to cite this article: Wang X, Zhang M: High-sensitivity synchronous image encryption based on improved one-dimensional compound sine map. IET Image Processing, 2021;1–19. https://doi.org/10.1049/ipr2.12193.