Article

Multi-Image Compression–Encryption Algorithm Based on Compressed Sensing and Optical Encryption

Jingjin Wei, Miao Zhang * and Xiaojun Tong

Computer Science and Technology, Harbin Institute of Technology, Weihai 264200, China; 20s030163@stu.hit.edu.cn (J.W.); tong_xiaojun@hit.edu.cn (X.T.)
* Correspondence: zm@hitwh.edu.cn

Abstract: In order to achieve large-capacity, fast and secure image transmission, a multi-image compression–encryption algorithm based on two-dimensional compressed sensing (2D CS) and optical encryption is proposed in this paper. Firstly, the paper uses compressed sensing to compress and encrypt multiple images simultaneously, and design a new structured measurement matrix. Subsequently, double random phase encoding based on the multi-parameter fractional quaternion Fourier transform is used to encrypt the multiple images for secondary encryption, which improves the security performance of the images. Moreover, a fractional-order chaotic system with more complex chaotic behavior is constructed for image compression and encryption. Experimental results show that the algorithm has strong robustness and security.

Keywords: compressed sensing; fractional-order chaotic system; double random phase encoding; image compression

1. Introduction

As part of the advances in networking and communication technology, extensive research has been conducted on how to transmit images in real-time and securely. In recent years, chaos has been widely used in data transmission and image encryption due to its inherent characteristics, such as sensitivity to initial conditions and inherent randomness. In chaotic mapping, the fractional-order chaotic system, as a generalization of integer-order chaos, has more complex dynamics than the integer-order chaotic system, and thus has more abundant application value in the field of image encryption. For example, Ding et al. [1] described an image encryption scheme using two chaotic systems, including a fractional-order Henon chaotic mapping and a four-dimensional hyperchaotic system. Xu et al. [2] proposed a fractional-order chaotic system based on the Hopfield neural network as a pseudo-random number generator, and constructed a new image encryption algorithm with the multiple hash index chain. The fractional chaotic system in the above paper is mainly applied to image encryption, but it can also be applied to the two processes of image compression and encryption concurrently. Hu et al. [3] used a fractional-order simplest memristive chaotic system and compressive sensing to compress and encrypt images. However, they did not consider applying the fractional-order chaotic system to the image compression part.

While ensuring image security, to save more transmission bandwidth, transmission time and storage space, compressed sensing (CS) has gradually applied to image compression and image encryption. CS theory [4] means that under the condition of a far smaller than Nyquist sampling rate, discrete signal samples under random sampling can be nonlinearly reconstructed into original signals, but they need to rely on the two preconditions of sparsity and incoherence. Since CS was proposed, many compression and encryption algorithms based on CS have appeared. Belyaev et al. [5] studied an iterative threshold-based compressed sensing video restoration algorithm. Huang et al. [6] embedded the encrypted image into the carrier image after SHA-3 and CS compression.
to achieve multi-image visual security. Gan et al. [7] fully combined information entropy and CS for color image compression and encryption. Ye et al. [8] proposed an image compression and encryption algorithm based on the elliptic curve and CS. At present, most of the objects of compression–encryption algorithms are single color images and multiple grayscale images, and how to compress and encrypt multiple color images is also a situation worthy of research.

Optical encryption technology has been widely used in the field of multi-encryption image encryption because of its powerful computing capability, multi-dimensional storage and parallel processing capability. The double random phase encoding [9] with a simple implementation process and high robustness is a very classical optical encryption system, but it cannot resist attacks such as selected plaintext attacks and known-plaintext attacks. Therefore, to improve the security of the encryption system, the Fourier transform domain used in double random phase encoding technology has been extended to other transform fields, such as the fractional Fourier transform and Gyrotor transform domain. In addition, the combination of the optical encryption system, chaos and the CS system can also enhance the performance of image compression–encryption systems. Sun et al. [10] presented a multi-image encryption algorithm based on multi-dimensional chaos and cascade rotator transformation. Huo et al. [11] adopted CS and orthogonal coding to carry out data sampling and the integration of multiple images, proposing an encryption algorithm combining chaos matrix and double random phase encoding. However, the current multi-color image compression and encryption method combining 2D CS and the optical encryption system has been considered less.

Based on the above discussion, to reduce the amount of transmitted data and improve the encryption capacity and security, a multi-image compression–encryption algorithm based on 2D CS and optical encryption technology is proposed, which is suitable for color images and grayscale images. The innovative design of the overall idea of this paper is as follows. The application of quaternion can improve the encryption capability without increasing the complexity of the encryption system, but currently, few algorithms apply it to image compression and encryption. When double random phase encoding technology is applied to image encryption, two parts of data, the real part and the imaginary part, are generated, which means the transmission of the encrypted image data needs to pass twice the data. Similarly, the application of double random phase encoding based on a multi-parameter fractional quaternion Fourier transform (MPFrQFT) to encrypted data can also result in an exponential increase in data volume [12]. However, when multiple images are combined with double random phase encoding based on an MPFrQFT transform for image encryption, the amount of data needed to be transmitted is the same as the original image data. Meanwhile, when image compression and multi-image encryption are combined, the amount of data to be transmitted can be greatly reduced, and the secure and efficient compression and encryption effect can be achieved.

This paper firstly designed a new fractional-order chaotic map, and the generated chaotic sequences were applied to the whole algorithm to improve its security of this algorithm. Secondly, a new deterministic random measurement matrix was constructed for 2D CS, which reduces the complexity of constructing the measurement matrix and makes the image easy to reconstruct. Thirdly, an improved image encryption algorithm was designed. The two-dimensional Joseph scrambling algorithm can confuse the row and column places of all global pixels simultaneously, while the diffusion algorithm can compress and encrypt in the same process, protecting the image information while reducing the data. Fourthly, double random phase coding based on MPFrQFT is used to encrypt multiple images. The algorithm can compress and encrypt multiple images, which improves the encryption capacity and image security.

The rest of this paper is organized as follows. Section 2 introduces some preliminary work. Section 3 gives a description of the proposed multi-image compression and encryption algorithm. Section 4 gives the experimental results and analysis of the algorithm. Finally, Section 5 is the conclusion part.
2. Preliminaries

2.1. Fractional-Order Chaotic System

In the three-dimensional fractional chaotic system, the fractional differential operator is used instead of the standard differential as follows [13,14],

\[
\begin{align*}
\frac{dx}{dt^q} &= ax + by \\
\frac{dy}{dt^q} &= -bx + ay + c \sin(z) \\
\frac{dz}{dt^q} &= dz + e \sin(x)
\end{align*}
\]

(1)

where \(a, b, c, d\) and \(e\) are system parameters, and \(q\) is the fractional order.

If the nonlinear terms \(\sin(x)\) and \(\sin(z)\) in the system (1) are replaced with \(\cos(x)\) and \(\cos(z)\), a new fractional-order chaotic system can be obtained.

\[
\begin{align*}
\frac{dx}{dt^q} &= ax + by \\
\frac{dy}{dt^q} &= -bx + ay + c \cos(z) \\
\frac{dz}{dt^q} &= dz + e \cos(x)
\end{align*}
\]

(2)

In this paper, the fractional-order chaotic system parameters when \(q = 0.8\), \(a = -1\), \(b = 1\), \(c = -80\), \(d = -1\) and \(e = 18\) are selected. Meanwhile, the Lyapunov exponent of the system are 2.3111, \(-0.0037\) and \(-10.4974\), as shown in Figure 1, and the system exhibits chaotic behavior [15]. Figure 2 shows the chaotic attractors of the fractional-order chaotic system on different phase planes.

![Figure 1. Lyapunov exponent.](image)

Different from integer-order chaotic systems, fractional-order chaotic systems have stronger memory characteristics and complex dynamic characteristics, can more accurately describe actual chaotic systems and are more in line with the actual needs of natural engineering applications. For image compression and encryption, it can generate more complex chaotic sequences and larger key spaces.
Figure 2. Fractional-order chaotic attractor. (a) x-y plane. (b) x-z plane. (c) y-z plane. (d) x-y-z plane.

2.2. Compressed Sensing

Compressed sensing is a new signal sampling compression theory, which can realize image compression and encryption simultaneously during the sampling period [4,5]. Two-dimensional compressed sensing is to sample and measure the signal from two directions, which can not only further reduce the size of the compressed image, but also make the reconstructed image obtain better reconstruction quality. Assuming that the two-dimensional signal $x$ is a signal of the size $N \times N$, it can be sparsely expressed in specific domains such as DCT and DWT as,

$$x = \Psi s$$  \hspace{1cm} (3)

where $\Psi$ is the $N \times N$ orthogonal transform base matrix, and $s$ is the sparse coefficient vector under $\Psi$. By linearly projecting $x$ from two directions onto the measurement matrix $\Phi_1$ and $\Phi_2$ of size $M \times N$, the $M \times M$ measurement value matrix $y$ can be obtained.

$$y = \Phi_1 x \Phi_2^T = \Phi_1 \Psi s \Phi_2^T = \Theta s$$  \hspace{1cm} (4)

When the original signal $x$ is to be reconstructed from the measured value $y$, the signal needs to be restored by solving the following convex optimization problem.

$$\min ||s||_1 \text{ s.t. } y = \Theta s$$  \hspace{1cm} (5)

The existing reconstruction algorithms mainly include an orthogonal matching pursuit algorithm, a smooth l0 norm algorithm, a base pursuit algorithm, etc. This paper chooses a 2D projected gradient with an embedding decryption (2DPG-ED) algorithm [16] as the image reconstruction algorithm. Moreover, this paper will use the newly constructed measurement matrix to measure signals from two directions.

2.3. Multi-Parameter Fractional Quaternion Fourier Transform

Quaternion is a super-complex number with one real part and three imaginary parts [17]. In particular, when the quaternion modulus is 1, it is called a unit quater-
nion. When the real number term is 0, the quaternion is a pure quaternion. The quaternion function \( f(x,y) \) of multiple images can be expressed as [18],

\[
f(x,y) = f_1(x,y) + f_2(x,y)i + f_3(x,y)j + f_4(x,y)k
\]  

(6)

where \( f_1(x,y), f_2(x,y), f_3(x,y) \) and \( f_4(x,y) \) are represented as four color images in this paper.

A two-dimensional multi-parameter fractional quaternion Fourier transform model is defined as [12]:

\[
\mathcal{U}_{M_1,M_2,\eta_1,\eta_2}^{\alpha,\mu_1,\mu_2}(f(x,y)) = \sum_{u=0}^{M_1-1} \sum_{v=0}^{M_2-1} F_{u,v}^{\mu_1,\mu_2}(f(x,y)) \omega_u^{\alpha,\mu_1}(\eta_1) \omega_v^{\beta,\mu_2}(\eta_2)
\]  

(7)

where \( u \) and \( v \) are fractional orders, \( \mu_1 \) and \( \mu_2 \) are pure quaternions, \( M_1 \) and \( M_2 \) are arbitrary integers and \( \eta_1 \) and \( \eta_2 \) are real number vectors with the dimensions \( M_1 \) and \( M_2 \). \( F_{u,v}^{\mu_1,\mu_2}(f(x,y)) \) is a two-dimensional fractional quaternion Fourier transform and \( \omega_u^{\alpha,\mu_1} \) and \( \omega_v^{\beta,\mu_2} \) are weights. The coefficient is defined as follows:

\[
\omega_u^{\alpha,\mu_1}(\eta) = \frac{1}{M_1} \sum_{k=0}^{M_1-1} \exp(-2\pi i \mu_1 / M_1) [a(k + \eta M_1) - uk]
\]  

(8)

\[
\omega_v^{\beta,\mu_2}(\eta) = \frac{1}{M_2} \sum_{k=0}^{M_2-1} \exp(-2\pi i \mu_2 / M_2) [\beta(k + \eta M_2) - vk]
\]  

(9)

The inverse transform of the multi-parameter fractional quaternion Fourier transform is \( \mathcal{U}_{M_1,M_2,\eta_1,\eta_2}^{-\alpha,\beta,\mu_1,\mu_2} \).

3. Multiple-Image Compression and Encryption Algorithm Based on 2D CS and Optical Encryption

3.1. Fractional-Order Chaotic Sequence Generation

This paper calculates the initial values of the fractional-order chaotic system by calculating the SHA/MD5 hash value and external key \( K \) of four original images [19]. First, the 256-bit hexadecimal external key \( K \) is randomly generated, in 8-bit decimal format, which can be expressed as \( K = \{ k_1, k_2, \ldots, k_{32} \} \). Secondly, by XOR operation, the foreign key \( K \) is combined with the hash value \( H \) of the four images to obtain \( K' = \{ k'_1, k'_2, \ldots, k'_{32} \} \). Then, the initial value can be calculated according to Equation (10).

\[
\begin{align*}
x_0 &= \frac{k'_1 \oplus k'_2 \oplus k'_3 \oplus k'_4 \oplus k'_5 \oplus k'_6}{256} \\
y_0 &= \frac{k'_7 \oplus k'_8 \oplus k'_9 \oplus k'_{10} \oplus k'_{11} \oplus k'_{12}}{256} \\
z_0 &= \frac{k'_{13} \oplus k'_{14} \oplus k'_{15} \oplus k'_{16} \oplus k'_{17} \oplus k'_{18}}{256}
\end{align*}
\]  

(10)

Finally, the initial value can be input into the new fractional-order chaotic equation to calculate the chaotic sequence.

3.2. Josephus Scrambling

Scrambling is an important step in encrypting images. This paper improves a Josephus scrambling algorithm [20], which makes it better for eliminating the correlation between image pixels. In the algorithm of this paper, the initial parameters are generated by the three-dimensional fractional-order chaotic sequence, which reduces the number of parameters that need to be passed, and the generated sequence is related to the number of cycles, which enhances the confusion of the sequence. The scrambling steps are as follows:

**Step 1**: Generate initial parameters. Assume that the size of the original image is \( M \times N \). Three groups of chaotic sequences are sequentially connected to form \( S \), and the following four parameters are generated through the Formula (11),
\[
\begin{align*}
MI &= 1 + \text{sum}(\text{round}(S(1:M))) \mod M \\
NI &= 1 + \text{sum}(\text{round}(S(1:N))) \mod N \\
M_{\text{step}} &= \text{sum}(\text{round}(S(1:M))) \mod M \\
N_{\text{step}} &= \text{sum}(\text{round}(S(1:N))) \mod N
\end{align*}
\]

(11)

where \(MI\) and \(NI\) represent the position of the starting row and column, respectively, and \(M_{\text{step}}\) and \(N_{\text{step}}\) represent the number of steps moved by each row and column, respectively.

**Step 2:** The row sequence, \(ros\), and column sequence, \(cos\), are generated as follows:
1. Set vector \(x\) to 1 through \(M\) and vector \(y\) to 1 through \(N\). (2) Initialize the first values of \(ros\) and \(cos\) to \(MI\) and \(NI\), respectively, and then delete the \(MI\) and \(NI\) values in vector \(x\) and vector \(y\). (3) Starting from the last deleted position, move the \(M_{\text{step}}\) and \(N_{\text{step}}\) steps in the circular vector \(x\) and \(y\), and send the next value to the second position of \(ros\) and \(cos\). (4) Update the values of \(M_{\text{step}}\) and \(N_{\text{step}}\) and add them to the number of rounds of this cycle, respectively. (5) Repeat the first two steps with rows \(M\) times and columns \(N\) times so that all values in vectors \(x\) and \(y\) have been offered to \(ros\) and \(cos\).

**Step 3:** In the \(i\)-th row of the image, where \(i\) belongs to 1 and to \(M\), the pixel points are sequentially replaced with pixel values \(\{(ros(\cos(1)) + i, \cos(1)), (ros(\cos(2)) + i, \cos(2)), \ldots, (ros(\cos(N)) + i, \cos(N)\}\}\) in the following order. If the value of \(ros(\cos(k)) + i (k = 1 \sim N)\) exceeds \(M\), the value of \(ros(\cos(k)) + i \mod M\) is taken as the replacement position instead of \(ros(\cos(k)) + i\).

**Step 4:** Repeat the previous step \(M-1\) times. In each round, the \(NI\) value is updated with the value of \(\cos(N)\), and a new \(cos\) column sequence is generated again in the way of step 2.

### 3.3. Measurement Matrix

In order to enhance the randomness of the chaotic sequence, discard the first 1000 values of each chaotic sequence and sample at equal intervals to ensure that the ratio of the values in the measurement matrix from the three sequences is one-third each. Simplify the total sequence to form a diagonal matrix \(z_i\), and then construct the total measurement matrix \(\Phi\) as follows:

\[
\Phi = \begin{bmatrix}
z_1 & z_1 & z_1 & z_1 & \ldots & z_l \\
z_1 & -z_1 & z_1 & -z_1 & \ldots & -z_l \\
\end{bmatrix}
\]

(12)

Among them, the size of the diagonal matrix \(z_i\) is \(m/2 \times m/2\). Extract \(N\) columns from matrix \(\Phi\) to form a measurement matrix of size \(m \times N\), as shown in Algorithm 1.

**Algorithm 1 Generation of measurement matrix**

Input: chaotic sequence \(x, y, z\) compression ratio \(m/N\)
Output: Measurement matrix \(\Phi\)

1: \(X = x(1001:m/2 + 1000)\);  
2: \(Y = y(1001:m/2 + 1000)\);  
3: \(Z = z(1001:m/2 + 1000)\);  
4: \(W = [X Y Z]\);  
5: \(j = 1:3; \text{length}(W)\);  
6: \(\Phi_{t1} = \text{sqrt}(2/m)*W(j)\);  
7: for \(i = 1:m/2\)  
8: if \(\Phi_{t1}(i) < 0\)  
9: \(\Phi_{t2}(i) = -1\);  
10: else  
11: \(\Phi_{t2}(i) = 1\);  
12: end  
13: \(\Phi_{t3} = \text{diag}(\Phi_{t2})\);  
14: \(\Phi_{t4} = \begin{bmatrix}
\Phi_{t3} & \Phi_{t3} & \Phi_{t3} & \Phi_{t3} & \ldots & \Phi_{t3} & \Phi_{t3} \\
-\Phi_{t3} & \Phi_{t3} & -\Phi_{t3} & \Phi_{t3} & \ldots & -\Phi_{t3} & \Phi_{t3} \\
\end{bmatrix}\);  
15: \(\Phi = \Phi_{t4}(1:m,1:N)\);
3.4. Image Compression and Encryption Algorithm

The process of image encryption and compression is shown in Figure 3. It is mainly composed of two parts: (1) 2D CS is used to encrypt and compress four images. (2) In order to achieve better security performance, double random phase encoding based on MPFrQFT is used to further encrypt the compressed and encrypted image. Among them, the chaotic sequence of the fractional-order chaotic system is used in many places in the whole algorithm.

![Multi-image compression and encryption flow chart.](image)

### 3.4.1. Image Compression and Encryption Based on 2D CS

Assuming that the sizes of the four images $P_1$, $P_2$, $P_3$ and $P_4$ are all $N \times N$, the compression steps are as follows:

**Step 1:** According to Section 3.1, generate the initial values of the fractional-order chaotic equation and the chaotic sequences.

**Step 2:** Extract the R, G and B components of the four images, respectively. According to Section 3.2, perform two-dimensional Josephus scrambling and linear random transformation (LRT) on the 12 components to obtain the encrypted image set $C_1$.

The process of LRT transformation can be expressed as,

$$C(i, j) = \begin{cases} \frac{C(i, j)}{C_1(i, j)}, & S(i, j) \mod 0 = 0 \\ P_{max} - C(i, j), & S(i, j) \mod 2 = 1 \end{cases}$$  \hspace{1cm} (13)$$

where $C$ is the encrypted image, $S$ is the chaotic matrix generated by the fractional-order chaotic equation and $P_{max}$ is the maximum pixel value of the encrypted image.

**Step 3:** Generate the $M \times N$ measurement matrices $\Phi_1$ and $\Phi_2$ described in Section 3.3. Perform gray mapping on the image set $C_1$ first to reduce the dynamic range of CS sampling and save the sampling bit width [16]. Then, according to formula (4), using the measurement matrices $\Phi_1$ and $\Phi_2$, the images are compressed and measured from two directions to obtain the compressed image set $C_2$.

### 3.4.2. Image Encryption with Double Random Phase Encoding Based on MPFrQFT

After completing the 2D CS, the paper further encrypts the image set $C_2$ as follows.

**Step 1:** Generate the quaternion $Q_{P_1}$ according to formula (14) on the four compressed and encrypted images that were originally R components in the image set $C_2$. In the same way, the G and B components can also generate quaternions $Q_{P_2}$ and $Q_{P_3}$.
which is the reverse operation of the encryption process. It is worth noting that the 2DPG-ED algorithm is used to reconstruct the image during the decryption process. The specific process is as follows.

3.5. Image Decryption and Reconstruction Algorithm

The decryption process is similar to the encryption process, as shown in Figure 4, which is the reverse operation of the encryption process. It is worth noting that the 2DPG-ED algorithm is used to reconstruct the image during the decryption process. The specific process is as follows.
3.5.1. Image Decryption with Double Random Phase Encoding Based on MPFrQFT

**Step1:** According to the external key K and Hash value, generate the initial value of the fractional-order chaotic equation and the chaotic sequences.

**Step2:** Perform an inverse diffusion operation on each layer and all layers of the color image C to obtain the encrypted image C4.

**Step3:** Perform inverse quantization operation on all layers to get color image C5.

**Step4:** Generate the quaternion Q_D1 according to formula (18) on the four compressed and encrypted images that were originally R components in the image set C5. In the same way, the G component and the B component can also generate quaternions Q_D2 and Q_D3.

\[
\begin{align*}
Q_{D1} &= R_1(x, y) + R_2(x, y)i + R_3(x, y)j + R_4(x, y)k \\
Q_{D2} &= G_1(x, y) + G_2(x, y)i + G_3(x, y)j + G_4(x, y)k \\
Q_{D3} &= B_1(x, y) + B_2(x, y)i + B_3(x, y)j + B_4(x, y)k \\
\end{align*}
\] (18)

where \(R_k, G_k\) and \(B_k\) \((k = 1, 2, 3, 4)\) are the R component, G component and B component of the four images to be decrypted, respectively.

**Step5:** Perform MPFrQFT transformation on \(Q_{D1}, Q_{D2}\) and \(Q_{D3}\) to obtain \(Q_{D4}\), \(Q_{D5}\) and \(Q_{D6}\), where the parameters are \(a_2, \beta_2, M_3, M_4, \eta_3, \eta_4\) and \(\mu_4\).

**Step6:** Right-multiply \(Q_{D4}, Q_{D5}\) and \(Q_{D6}\) by the quaternion random phase mask \(e^{-\mu_52\pi(b(u,v))}\) to obtain quaternion \(Q_{D7}, Q_{D8}\) and \(Q_{D9}\).

\[
\begin{align*}
Q_{D7} &= Q_{D4} \times e^{-\mu_52\pi(b(u,v))} \\
Q_{D8} &= Q_{D5} \times e^{-\mu_52\pi(b(u,v))} \\
Q_{D9} &= Q_{D6} \times e^{-\mu_52\pi(b(u,v))} \\
\end{align*}
\] (19)

**Step7:** Perform IMPFrQFT transformation on \(Q_{D7}, Q_{D8}\) and \(Q_{D9}\) to obtain \(Q_{D10}, Q_{D11}\) and \(Q_{D12}\), where the parameters are \(a_1, \beta_1, M_1, M_2, \eta_1, \eta_2\) and \(\mu_2\).

**Step8:** Right-multiply \(Q_{D10}, Q_{D11}\) and \(Q_{D12}\) by the quaternion random phase mask \(e^{-\mu_12\pi(a(x,y))}\) to obtain quaternion \(Q_{D13}, Q_{D14}\) and \(Q_{D15}\).

\[
\begin{align*}
Q_{D13} &= Q_{D10} \times e^{-\mu_12\pi(a(x,y))} \\
Q_{D14} &= Q_{D11} \times e^{-\mu_12\pi(a(x,y))} \\
Q_{D15} &= Q_{D12} \times e^{-\mu_12\pi(a(x,y))} \\
\end{align*}
\] (20)

3.5.2. 2D CS Image Reconstruction

**Step1:** Extract the real part and three imaginary parts in \(Q_{D13}, Q_{D14}\) and \(Q_{D15}\), respectively, and arrange them in order to form a \(2M \times 2M\) color image C6.

**Step2:** Generate measurement matrices \(\Phi_1\) and \(\Phi_2\) again.

**Figure 4.** Multi-image decryption and reconstruction flowchart.
Step3: First, divide the R, G and B layers of color image C6 into 12 layers by the size of the original image, then perform grayscale mapping, and then reconstruct the image according to the 2DPG-ED algorithm. The reconstruction algorithm includes the inverse operation of two-dimensional Josephus scrambling and LRT transformation.

Step4: Turn the reconstructed 12 layers back to the original four images according to the original encryption order of the image.

4. Simulation Results and Analysis

In order to verify the feasibility of the proposed compression and encryption algorithm, a series of numerical simulations were carried out on a computer equipped with CPU @ 2.10 GHz, 16G RAM and MATLAB R2019b. The initial parameters of the fractional-order chaotic system are set to $x_0 = 0.1$, $y_0 = 0.1$ and $z_0 = 0.1$. The compression ratio (CR) is 0.5. The parameters of the double random phase encoding part are set as: period $M_1 = 23$, $M_2 = 29$, $M_3 = 25$ and $M_4 = 27$, the vector parameter $\eta_s$ ($s = 1,2,3,4$) are $M_s$-dimensional random arbitrary real vectors and the unit pure quaternion arrays $\mu_1$, $\mu_2$, $\mu_3$ and $\mu_4$ are $\mu_1 = (i+j+k)/3$, $\mu_2 = i$, $\mu_3 = j$ and $\mu_4 = k$. Set $a_1 = 3.8287$, $a_2 = 3.2011$, $b_1 = 1.9415$ and $b_2 = 0.5675$. Furthermore, the algorithm is also suitable for the multi-image compression and encryption of grayscale images. This paper realizes double images compression and encryption by setting two imaginary parts equal to zero, and the decrypted image is obtained by the corresponding non-zero imaginary part.

4.1. Experimental Results

The test images include $512 \times 512$ color images: Lena, Peppers, Lake, Airplane, and $256 \times 256$ grayscale images: Lena, Cameraman. To distinguish the images, the specific image is indicated by the combination of the image name and the image size, as shown in Figure 5. The last three of the color images are from the USC-SIPI image library, while Lena and Cameraman are commonly used images. The image encryption and decryption results are shown in Figures 6 and 7.

![Original Image](image1.png)

(a) Lena512. (b) Peppers512. (c) Lake512. (d) Airplane512. (e) Lena256. (f) Cameraman256.

Figure 5. Original image. (a) Lena512. (b) Peppers512. (c) Lake512. (d) Airplane512. (e) Lena256. (f) Cameraman256.
It can be seen from Figures 6 and 7 that both color images and grayscale images can be effectively compressed and encrypted by the algorithm in this paper, and the decrypted image is visually very similar to the original image.

4.2. Compression Performance Analysis

In this paper, 2D CS is used to compress and encrypt multiple images simultaneously. The encrypted image is compressed into different sizes according to CR. The larger the CR, the better the reconstruction quality of the decrypted image. Both the peak signal-to-noise ratio (PSNR) and average structural similarity (MSSIM) are used to evaluate the compression performance of the algorithm. PSNR is a common indicator for appraising the quality of decrypted images. The calculation method is as follows:

\[
PSNR = 10 \log \frac{255^2}{(1/3MN) \sum_{i=1}^{M} \sum_{j=1}^{N} [P(i,j) - D(i,j)]^2}
\]  (21)

where \( M \) and \( N \) are the sizes of the image, and \( P(i,j) \) and \( D(i,j) \) are the pixel values at row \( i \) and column \( j \) in the original image and the decrypted image, respectively. The larger the PSNR, the smaller the image distortion.
MSSIM is an indicator of similarity, and its distribution range is between 0 and 1. The larger the test value, the stronger the similarity between images. It is defined as:

\[
SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{\mu_x^2 + \mu_y^2 + C_1}(\sigma_x^2 + \sigma_y^2 + C_2)
\]  

(22)

where \(C_1 = (k_1L)^2\), \(C_2 = (k_2L)^2\), \(k_1 = 0.01\), \(k_2 = 0.03\) and \(L = 255\). \(\mu_x\), \(\mu_y\), \(\sigma_x\), \(\sigma_y\) and \(\sigma_{xy}\) represent the mean, variance and covariance of the original image and the decrypted image, respectively. \(x_j\) and \(y_j\) represent the two images of the \(j\)-th window and \(N\) is the total number of windows, where \(N = 64\). Table 1 shows the PSNR and MSSIM of the decryption results of the algorithm in this paper under different CRs.

Table 1. PSNR and MSSIM of plaintext image and decrypted image.

| Images    | CR  | PSNR     | MSSIM   |
|-----------|-----|----------|---------|
| Lena512   | 0.75| 40.0126  | 0.9929  |
|           | 0.5 | 36.5825  | 0.9844  |
|           | 0.25| 33.6496  | 0.9704  |
| Peppers512| 0.75| 38.4215  | 0.9909  |
|           | 0.5 | 35.1721  | 0.9806  |
|           | 0.25| 32.7130  | 0.9671  |
| Lake512   | 0.75| 34.9974  | 0.9776  |
|           | 0.5 | 31.6709  | 0.9593  |
|           | 0.25| 28.8113  | 0.9317  |
| Airplane512| 0.75| 42.0901  | 0.9832  |
|           | 0.5 | 38.6556  | 0.9662  |
|           | 0.25| 34.9643  | 0.9371  |
| Lena256   | 0.75| 41.0852  | 0.9881  |
|           | 0.5 | 36.7034  | 0.9947  |
|           | 0.25| 32.2037  | 0.9850  |
| Cameraman256| 0.75| 38.8366  | 0.9966  |
|           | 0.5 | 34.0923  | 0.9904  |
|           | 0.25| 30.0109  | 0.9737  |

From Table 1, we can see that both the PSNR and MSSIM of the decrypted image are in a range with better results. As the CR value increases, the value of MSSIM also increases. The higher the similarity between the decrypted image and the original image, the better the image reconstruction effect. The results in Table 1 show that this algorithm can compress and encrypt images in a diversified manner.

Table 2 shows the PSNR results of different images under different compression ratios, as well as comparisons with references. From Table 2, we can see that our reconstruction effect is significantly better than the reference [7,21,22].

Table 2. Compression performance of different algorithms.

| Images     | CR  | PSNR     | Ref. [21] | Ref. [22] | Ref. [7] |
|------------|-----|----------|-----------|-----------|---------|
| Peppers512 | 0.75| 38.4215  | 31.7121   | -         | <30     |
|            | 0.5 | 35.1721  | 26.4965   | -         | <30     |
|            | 0.25| 32.7130  | 14.0119   | -         | >25     |
| Lena256    | 0.75| 41.0852  | -         | 36.1415   | -       |
|            | 0.5 | 36.7034  | -         | 32.1471   | -       |
|            | 0.25| 32.2037  | -         | 28.0615   | -       |
4.3. Statistical Analysis

4.3.1. Histogram

The histogram reflects the relationship between the frequency of each gray level pixel in the image and the gray level, and it is one of the important criteria for evaluating the security performance of an image encryption scheme. If the histograms of encrypted images are relatively evenly distributed, the encryption effect is good. Figure 8 is the histogram of the respective red, green and blue components of the color images and the histogram of the total encrypted text image of the four images. Figure 9 is the histogram of the two grayscale images and the total cipher text image.

![Histograms of color image and ciphertext](image1)

**Figure 8.** Histograms of color image and ciphertext. (a) Histogram of Lena512 R. (b) Histogram of Lena512 G. (c) Histogram of Lena512 B. (d) Histogram of Peppers512 R. (e) Histogram of Peppers512 G. (f) Histogram of Peppers512 B. (g) Histogram of Lake512 R. (h) Histogram of Lake512 G. (i) Histogram of Lake512 B. (j) Histogram of Airplane512 R. (k) Histogram of Airplane512 G. (l) Histogram of Airplane512 B. (m) Histogram of encrypted image R. (n) Histogram of encrypted image G. (o) Histogram of encrypted image B.
of Lake512 B. (j) Histogram of Airplane512 R. (k) Histogram of Airplane512 G. (l) Histogram of Airplane512 B. (m) Histogram of encrypted image R. (n) Histogram of encrypted image G. (o) Histogram of encrypted image B.

Figure 9. Histograms of grayscale image and ciphertext. (a) Histogram of Lena512. (b) Histogram of Cameraman256. (c) Histogram of encrypted image.

From Figures 8 and 9, it can be found that the histograms of the color ciphertext image and the grayscale ciphertext image are very uniform. This means that any statistical information of plaintext cannot be obtained from the histogram of the ciphertext. This shows that the multi-image compression–encryption algorithm can effectively resist statistical analysis attacks.

4.3.2. Correlation between Adjacent Pixels

The correlation between adjacent pixels can also be used to obtain information about the original image. Strongly correlated pixels exist in the original image, and image encryption can destroy this correlation. The correlation coefficient $\rho_{xy}$ is defined as follows:

$$\rho_{xy} = \frac{E((x - E(x))(y - E(y)))}{\sqrt{D(x)D(y)}}$$

(24)

where $E(x)$ and $D(x)$ are the mean and variance of $x$, respectively, and the same is true for $E(y)$ and $D(y)$. The value of $\rho_{xy}$ is in the range of 0 to 1, and the closer to 1, the stronger the correlation. Table 3 shows the correlation coefficients between the proposed multi-image compression–encryption algorithm and references. Figure 10 shows the relative distribution of adjacent pixels of the “Peppers512” image and its ciphertext image.

Table 3. The adjacent pixel correlation coefficient of original image and ciphertext image.

| Images          | Horizonta | Vertical | Diagonal |
|-----------------|-----------|----------|----------|
| Lena512         | Plain image | 0.9881  | 0.9775   | 0.9698   |
|                 | Cipher image | −0.0014| −0.0013 | −0.0030 |
|                 | Ref. [23]   | 0.0019  | 0.0184   | −0.0129  |
| Peppers512      | Plain image | 0.9822  | 0.9786   | 0.9693   |
|                 | Cipher image | 0.0017 | −0.0008 | −0.0014 |
|                 | Ref. [24]   | 0.0004  | −0.0002  | 0.0019   |
| Lake512         | Plain image | 0.9879  | 0.9774   | 0.9696   |
|                 | Cipher image | −0.0018| −0.0009 | −0.0013 |
|                 | Ref. [23]   | 0.0047  | −0.0021  | −0.0093  |
| Airplane512     | Plain image | 0.9588  | 0.9586   | 0.9246   |
|                 | Cipher image | −0.0016| 0.0003   | −0.0006  |
|                 | Ref. [24]   | −0.0008| −0.0042  | −0.0030  |
| Lena256         | Plain image | 0.9703  | 0.9413   | 0.9152   |
|                 | Cipher image | 0.0037 | −0.0024 | 0.0038   |
|                 | Ref. [25]   | −0.0018| −0.0070  | −0.0024  |
| Cameraman256    | Plain image | 0.9595  | 0.9334   | 0.9084   |
|                 | Cipher image | 0.0082 | −0.0011 | −0.0004 |
|                 | Ref. [25]   | −0.0096| 0.0082   | −0.0043  |
Figure 10. The relative distribution of the original image “Peppers512” and the encrypted image in three directions. (a) Correlation distribution in the horizontal direction of the Peppers512. (b) Correlation distribution in the vertical direction of the Peppers512. (c) Correlation distribution in the diagonal direction of the Peppers512. (d) Correlation distribution in the horizontal direction of the ciphertext image. (e) Correlation distribution in the vertical direction of the ciphertext image. (f) Correlation distribution in the diagonal direction of the ciphertext image.

It can be seen from Table 3 that the correlation coefficients of the ciphertext image of the color image and the grayscale image are close to 0, and are basically better than the values in references [23–25], indicating that our encryption method can resist statistics attacks and have higher security.

Figure 10 shows the pixel distribution of “Peppers512” in different directions. Compared to the diagonal pixel distribution of the original image, the ciphertext image is uniformly distributed in the entire coordinate space.

4.3.3. Information Entropy

Information entropy is used to evaluate the randomness and unpredictability of the image, which is calculated as follows:

\[
H(m) = \sum_{i=1}^{N} p(m_i) \log_2 \frac{1}{p(m_i)}
\]  
(25)

where \( p(m_i) \) is the probability of the occurrence of image pixel gray value \( m_i \), and \( N \) is the total number of \( m_i \). Among the values of information entropy, eight is an ideal value. The information entropy values of color images, grayscale images and their encrypted images are shown in Tables 4 and 5.

Table 4. Information entropy of color images and ciphertext image.

| Images       | R Component | G Component | B Component |
|--------------|-------------|-------------|-------------|
| Lena512      | Plain image | 7.2531      | 7.5940      | 6.9684      |
|              | Cipher image| 7.9987      | 7.9985      | 7.9987      |
|              | Ref. [26]   | 7.9970      | 7.9972      | 7.9967      |
| Peppers512   | Plain image | 7.3388      | 7.4963      | 7.0583      |
|              | Cipher image| 7.9985      | 7.9987      | 7.9986      |
|              | Ref. [26]   | 7.9974      | 7.9972      | 7.9971      |
| Lake512      | Plain image | 7.3124      | 7.6429      | 7.2136      |
|              | Cipher image| 7.9986      | 7.9986      | 7.9987      |
|              | Ref. [26]   | 7.9970      | 7.9967      | 7.9969      |
| Airplane512  | Plain image | 6.7178      | 6.7990      | 6.2138      |
|              | Cipher image| 7.9988      | 7.9986      | 7.9986      |
|              | Ref. [26]   | 7.9974      | 7.9972      | 7.9980      |
Table 5. Information entropy of grayscale image and ciphertext image.

| Images      | Plain Image | Cipher Image |
|-------------|-------------|--------------|
| Lena256     | 7.4443      | 7.9952       |
| Cameraman256| 7.0097      | 7.9939       |

Obviously, it can be seen from Tables 4 and 5 that the information entropy of the R component, G component and B component of the color ciphertext image exceeds 7.99, which is better than the information entropy of the reference [26]. The information entropy of the grayscale ciphertext image is also around eight. This means that the image encrypted by this algorithm has good randomness.

Local Shannon entropy is an index that quantitatively describes the randomness of an image from a local perspective, which can be written as,

$$H_{k,T_B}(S) = \sum_{i=1}^{k} \frac{H(S_i)}{k}$$

(26)

where $S_1\sim S_k$ are non-overlapping image blocks, $k$ is the number of selected blocks, $T_B$ represents the number of pixels in each selected image block and $H(S_i)$ is the information entropy of the selected image block. If the value of local Shannon entropy is in the interval $(h_{left}^*, h_{right}^*)$, the image will pass the local Shannon entropy test. In this paper, $k$ and $T_B$ are selected as 30 and 1936. Table 6 shows the local Shannon entropy of the color ciphertext image.

Table 6. The local Shannon entropy of the color ciphertext image.

| Image      | Lena512 | Peppers512 | Lake512 | Airplane512 | Mean          |
|------------|---------|------------|---------|-------------|---------------|
| Local Shannon entropy | 7.901517 | 7.903376   | 7.901792 | 7.901701    | 7.902096      |

According to Table 6, it can be known that the color ciphertext images have passed the test, and the local Shannon entropy of the total ciphertext image is also within the range of values. This means that the ciphertext image of the algorithm in the paper has good randomness.

4.4. Key Space Analysis

To resist brute force attacks, the key space of a secure image compression–encryption algorithm is considered to be at least $2^{100}$. In this algorithm, the key is mainly composed of the following two parts: the 256-bit initial key K and the 256-bit hash value of the original image. Multiple parameters in MPFrQFT can also be used as a key during the transfer process. By simple addition, it can be seen that the key space of the algorithm is at least $2^{512}$, which is greater than $2^{100}$. Table 7 gives the comparison results of key space with other algorithms.

Table 7. Comparison of key space of the proposed method with other methods.

| Algorithm | Ours   | Ref. [25] | Ref. [27] | Ref. [28] | Ref. [7]   |
|-----------|--------|-----------|-----------|-----------|-----------|
| Key Space | $>2^{512}$ | $\approx2^{572}$ | $\approx2^{210}$ | $\approx2^{233}$ | $\approx2^{294}$ |

As can be seen in Table 7, the proposed scheme has the largest key space than other encryption schemes. So the algorithm can meet the security requirements of the key space and resist brute force attacks.
4.5. Key Sensitivity Analysis

Key sensitivity requires that the encryption algorithm can produce completely different encryption results due to the slight change of key. This means that only a unique and correct key can recover the plaintext. In this paper, two keys, $K_a$ and $K_b$, are randomly selected during verification, but only 1-bit difference is guaranteed between the two keys.

$$K_a = '4ffDc1EB6bFfC9ac5Abe63a7BBe20c6EF6BDdDEcAc09eb8ECC8dDEe4e18eEBaA'$$

$$K_b = '4ffDc1EB6bFfC9ac5Abe63a7BBe20c6EF6BDdDEcAc09eb8ECC8dDEe4e18eEBaB'$$

In the simulation, the results of encrypted and decrypted four color images using keys $K_a$ and $K_b$ are shown in Figure 11.

![Image](image-url)

**Figure 11.** Key sensitivity analysis. (a) Image $C_a$ encrypted with $K_a$. (b) Image $C_b$ encrypted with $K_b$. (c) Differences between encrypted images. (d) Use $K_b$ to decrypt the Lena512 after ciphertext $C_a$. (e) Use $K_b$ to decrypt the Peppers512 after ciphertext $C_a$. (f) Use $K_b$ to decrypt the Lake512 after ciphertext $C_a$. (g) Use $K_b$ to decrypt the Airplane512 after ciphertext $C_a$.

From Figure 11a–c, we can see that with the original image encrypted with $K_a$ and $K_b$, two visually secure encrypted results can be obtained, and the difference between them is larger. Moreover, as shown in Figure 11d–g, using the key that is 1-bit different from the correct key for decryption, none of the four original images can be reconstructed correctly. Therefore, the proposed multi-image compression–encryption algorithm is key sensitive.

4.6. Differential Attack Analysis

The differential attack is an effective and common attack against security. After adding slight changes to the original image, the attacker can obtain usable information by analyzing the differences between the two encrypted images. The pixel change rate (NPCR), uniform average change intensity (UACI) and avalanche effect are three common indicators to evaluate the impact of differential attacks. If the plaintext pixel value changes slightly, the ciphertext pixel value can change significantly after encryption, which shows the algorithm has good randomization characteristics. The calculation methods of NPCR and UACI are as follows:
\[
\text{NPCR} = \frac{\sum_{i,j} D(i,j)}{W \times H} \times 100\% \quad (27)
\]
\[
\text{UACI} = \frac{1}{W \times H} \left[ \sum |d_1(i,j) - d_2(i,j)| \right] \times 100\% \quad (28)
\]

where \( W \) and \( H \) represent the width and height of the image, respectively, and \( d_1 \) and \( d_2 \) are the two ciphertext images before and after the plaintext image changes one-pixel value. If \( d_1(i,j) = d_2(i,j) \), then \( D(i,j) = 0 \), otherwise, \( D(i,j) = 1 \). The ideal expected values of the NPCR, UACI and avalanche effects are 99.6094\%, 33.4635\% and 50\%, respectively. Table 8 shows the NPCR, UACI and avalanche effect values, as well as the comparison results with other algorithms.

### Table 8. NPCR, UACI and avalanche effect values of ciphertext images.

| Images   | Algorithm | NPCR    | UACI    | Avalanche Effect |
|----------|-----------|---------|---------|------------------|
| Lena512  | Ours      | 99.6078 | 33.4741 | 50.0094          |
|          | Ref. [29] | 99.7470 | 36.7368 | -                |
| Peppers512 | Ours     | 99.6088 | 33.4719 | 50.0022          |
|          | Ref. [29] | 99.7509 | 34.7577 | -                |
| Lake512  | Ours      | 99.6080 | 33.4716 | 49.9982          |
|          | Ref. [29] | -       | -       | -                |
| Airplane512 | Ours   | 99.6106 | 33.4777 | 50.0072          |
|          | Ref. [29] | 99.7548 | 46.6056 | -                |
| Lena256  | Ours      | 99.6087 | 33.4924 | 50.0151          |
|          | Ref. [22] | 99.6101 | 33.4654 | -                |
| Cameraman256 | Ours   | 99.6075 | 33.4671 | 49.9867          |
|          | Ref. [22] | 99.6066 | 33.4679 | -                |

The results in Table 8 show that the values of the NPCR, UACI and avalanche effect of ciphertext images are close to their theoretical values and are better than the algorithms in reference [25]. Therefore, this algorithm has a good performance of resistance to differential attacks.

### 4.7. Robustness Analysis

#### 4.7.1. Analysis of Noise Attacks

During the transmission process, the ciphertext image will inevitably be affected by noise and interference. To evaluate the impact of noise and interference on the decrypted image, salt and pepper noise (SPN), speckle noise (SN) and Gaussian noise (GN) were added to the encrypted image to measure the ability of the algorithm to resist noise attacks. Table 9 shows the decrypted results after adding different types and degrees of noise to the ciphertext image, as well as the comparison with other algorithms. Figure 12 is a decrypted image of a noise image.

Table 9 shows that our algorithm can resist \(10^{-5}\) SPN, \(10^{-6}\) GN and SN. Compared with reference [23], the anti-noise effect of our algorithm has obvious advantages when the noise is greater.

Figure 12 shows the effect of decrypting images under higher levels of noise. It can be found that when the noise is \(10^{-3}\) GN and SN, there are obvious spots in the decrypted image, but the original image can still be easily recognized. When the noise is \(10^{-3}\) SPN, the quality of the decrypted image is still good. It shows the algorithm can resist at least \(10^{-5}\) noise attacks, and the ability to resist salt and pepper noise is stronger than that of other types of noise.
Table 9. PSNR of the decrypted image of the noisy ciphertext image.

| Noise Type | Noise Intensity | Lena512 Ours | Lena512 [23] | Peppers512 | Lake512 | Airplane512 |
|------------|----------------|-------------|--------------|------------|---------|-------------|
| SPN        | $1 \times 10^{-5}$ | 36.2751     | 32.2248      | 34.9380    | 31.5531 | 38.1682     |
|            | $3 \times 10^{-5}$ | 35.9222     | 31.8791      | 34.6932    | 31.3934 | 37.6614     |
|            | $5 \times 10^{-5}$ | 35.4256     | 29.6256      | 34.3170    | 31.1725 | 36.9469     |
| SN         | $1 \times 10^{-6}$ | 36.5825     | 32.6759      | 35.1716    | 31.6857 | 38.6497     |
|            | $3 \times 10^{-6}$ | 35.9578     | 17.2882      | 34.7033    | 31.4104 | 37.6926     |
|            | $5 \times 10^{-6}$ | 35.0968     | 13.5636      | 34.0699    | 31.0119 | 36.4843     |
| GN         | $1 \times 10^{-6}$ | 35.9662     | 18.0331      | 34.7121    | 31.3983 | 37.7041     |
|            | $3 \times 10^{-6}$ | 34.4031     | 13.1115      | 33.5063    | 30.6553 | 35.4992     |
|            | $5 \times 10^{-6}$ | 33.5791     | 9.2767       | 32.8993    | 30.2160 | 34.4459     |

Figure 12. Noise attack results with different noise intensities. (a) $10^{-5}$ SPN. (b) $10^{-4}$ SPN. (c) $10^{-3}$ SPN. (d) $10^{-3}$ SN. (e) $10^{-6}$ SN. (f) $10^{-4}$ GN. (g) $10^{-3}$ SN. (h) $10^{-3}$ SN. (i) $10^{-4}$ GN. (j) $10^{-3}$ GN. (k) $10^{-3}$ GN. (l) $10^{-3}$ GN.

4.7.2. Analysis of Shear Attack

The ciphertext may also be subject to clipping attacks during transmission, so the compression and encryption algorithm needs to have the ability to resist clipping attacks. Figure 13 shows the decrypted image recovery result after the ciphertext image has been subjected to different cutting methods and varying degrees of data loss.
Figure 13 shows that the main information of the original image can still be roughly represented after being subjected to different degrees of shearing attacks. As the amount of data loss increases, the recovery effect is gradually declining. As far as this algorithm is concerned, the cutting position has little effect on the image restoration effect. All in all, our algorithm can resist shearing attacks to a certain degree.

5. Conclusions

This paper proposes a multi-image encryption and compression algorithm based on 2D CS and optical encryption. This paper first uses 2D CS to compress and encrypt multiple images simultaneously, and then uses MPFrQFT-based double random phase encoding
to encrypt the images twice, which enhances the encryption effect. Among them, a newly designed structured measurement matrix is used in 2D CS, which can effectively reduce the transmission load, and so the encryption of multiple images is realized in MPFrQFT. In addition, the chaotic sequence generated by the fractional chaotic system is used in the sampling process of 2D CS and the double random phase encoding process. This algorithm combines the advantages of compression–encryption and multi-image encryption, which can further reduce the amount of data transmission and key transmission consumption, while increasing the encryption capacity and ensuring the security of the image. Experimental results show that this compression–encryption algorithm has good robustness and compression performance, can resist at least $10^{-3}$ noise attacks and shear attacks with 20% data loss and can obtain better recovery results under different compression ratios, which is better than the current reference data. In terms of security performance, it has 2512 key space and can resist a statistical analysis attack and differential attack.

**Author Contributions:** Writing—original draft, J.W., M.Z. and X.T. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the following projects and foundations: the National Natural Science Foundation of China (grant number 61902091), the Shandong Provincial Natural Science Foundation (ZR2019MF054), the National Key R&D Program of China (2021YFB2012400).

**Informed Consent Statement:** Informed consent was obtained from all subjects involved in the study.

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

**References**

1. Ding, L.; Ding, Q. A Novel Image Encryption Scheme Based on 2D Fractional-order chaotic Map, DWT and 4D Hyper-chaos. *Electronics* **2020**, *9*, 1280. [CrossRef]
2. Xu, S.; Wang, X.; Ye, X. A new fractional-order chaos system of Hopfield neural network and its application in image encryption. *Chaos Solitons Fractals* **2022**, *157*, 111889. [CrossRef]
3. Hu, H.; Cao, Y.; Xu, J.; Ma, C.; Yan, H. An Image Compression and Encryption Algorithm Based on the Fractional-Order Simplest Chaotic Circuit. *IEEE Access* **2021**, *9*, 22141–22155. [CrossRef]
4. Donoho, D.L. Compressed sensing. *IEEE Trans. Inf. Theory* **2006**, *52*, 1289–1306. [CrossRef]
5. Belyaev, E.; Codreanu, M.; Junitt, M.; Egiazarian, K. Compressive Sensed Video Recovery via Iterative Thresholding with Random Transforms. *IET Image Process.* **2020**, *14*, 1187–1199. [CrossRef]
6. Huang, X.; Dong, Y.; Zhu, H.; Ye, G. Visually asymmetric image encryption algorithm based on SHA-3 and compressive sensing by embedding encrypted image. *Alex. Eng. J.* **2022**, *61*, 7637–7647. [CrossRef]
7. Gan, Z.; Bi, J.; Ding, W.; Chai, X. Exploiting 2D compressed sensing and information entropy for secure color image compression and encryption. *Neural Comput. Appl.* **2021**, *33*, 12845–12867. [CrossRef]
8. Ye, G.; Liu, M.; Wu, M. Double image encryption algorithm based on compressive sensing and elliptic curve. *Alex. Eng. J.* **2021**, *61*, 6785–6795. [CrossRef]
9. Refregier, P.; Javidi, B. Optical image encryption based on input plane and Fourier plane random encoding. *Opt. Lett.* **1995**, *20*, 767–769. [CrossRef] [PubMed]
10. Sun, X.; Shao, Z.; Shang, Y.; Liang, M.; Yang, F. Multiple-image encryption based on cascaded gyrator transforms and high-dimensional chaotic system. *Multimed. Tools Appl.* **2021**, *80*, 15825–15848. [CrossRef]
11. Huo, D.; Zhou, X.; Zhang, L.; Zhou, Y.; Li, H.; Yi, S. Multiple-image encryption scheme via compressive sensing and orthogonal encoding based on double random phase encoding. *J. Mod. Opt.* **2018**, *65*, 2093–2102. [CrossRef]
12. Chen, B.; Yu, M.; Tian, Y.; Li, L.; Wang, D.; Sun, X. Multiple-parameter fractional quaternion Fourier transform and its application in colour image encryption. *IET Image Process.* **2018**, *12*, 2238–2249. [CrossRef]
13. Zhang, H.; Liu, X.; Shen, X.; Liu, J. Chaos Entanglement: A New Approach to Generate Chaos. *Int. J. Bifurc. Chaos* **2013**, *23*, 30014. [CrossRef]
14. Eshaghi, S.; Ghaziani, R.K.; Ansari, A. Hopf bifurcation, chaos control and synchronization of a chaotic fractional-order system with chaos entanglement function. *Math. Comput. Simul.* **2020**, *172*, 321–340. [CrossRef]
15. Danca, M.-F.; Kuznetsov, N. Matlab code for Lyapunov exponents of fractional order systems. *Int. J. Bifurc. Chaos* **2018**, *28*, 1850067. [CrossRef]
16. Zhang, B.; Xiao, D.; Xiang, Y. Robust Coding of Encrypted Images via 2D Compressed Sensing. *IEEE Trans. Multimed.* **2020**, *23*, 2656–2671. [CrossRef]
17. Hamilton, W.R. Elements of quaternions. *Nature* **1969**, *2*, 564.
18. Yu, C.; Li, J.; Li, X.; Ren, X.; Gupta, B.B. Four-image encryption scheme based on quaternion Fresnel transform, chaos and computer generated hologram. *Multimed. Tools Appl.* **2018**, *77*, 4585–4608. [CrossRef]
19. Zefreh, E.Z. An image encryption scheme based on a hybrid model of DNA computing, chaotic systems and hash functions. *Multimed. Tools Appl.* **2020**, *79*, 24993–25022. [CrossRef]
20. Hua, Z.; Xu, B.; Jin, F.; Huang, H. Image Encryption Using Josephus Problem and Filtering Diffusion. *IEEE Access* **2019**, *7*, 8660–8674. [CrossRef]
21. Jiang, X.; Xiao, Y.; Xie, Y.; Liu, B.; Ye, Y.; Song, T.; Chai, J.; Liu, Y. Exploiting optical chaos for double images encryption with compressive sensing and double random phase encoding. *Opt. Commun.* **2021**, *484*, 126683. [CrossRef]
22. Nan, S.X.; Feng, X.F.; Wu, Y.F.; Zhang, H. Remote sensing image compression and encryption based on block compressive sensing and 2D-LCCCM. *Nonlinear Dyn.* **2022**, *108*, 2705–2729. [CrossRef]
23. Chai, X.; Bi, J.; Gan, Z.; Liu, X.; Zhang, Y.; Chen, Y. Color image compression and encryption scheme based on compressive sensing and double random encryption strategy. *Signal Process.* **2020**, *176*, 107684. [CrossRef]
24. Wang, X.; Su, Y. Color image encryption based on chaotic compressed sensing and two-dimensional fractional Fourier transform. *Sci. Rep.* **2020**, *10*, 18556. [CrossRef]
25. Fu, J.; Gan, Z.; Chai, X.; Lu, Y. Cloud-decryption-assisted image compression and encryption based on compressed sensing. *Multimed. Tools Appl.* **2022**, *81*, 17401–17436. [CrossRef]
26. Dong, H.; Bai, E.; Jiang, X.Q.; Wu, Y. Color Image Compression-Encryption Using Fractional-Order Hyperchaotic System and DNA Coding. *IEEE Access* **2020**, *8*, 163524–163540. [CrossRef]
27. Gan, Z.; Chai, X.; Bi, J.; Chen, X. Content-adaptive image compression and encryption via optimized compressive sensing with double random phase encoding driven by chaos. *Complex Intell. Syst.* **2022**, *8*, 2291–2309. [CrossRef]
28. Liu, J.; Zhang, M.; Tong, X.; Wang, Z. Image compression and encryption algorithm based on 2D compressive sensing and hyperchaotic system. *Multimed. Syst.* **2021**, *28*, 595–610. [CrossRef]
29. Es, A.; Rw, B.; Aks, B. Securing color image transmission using compression-encryption model with dynamic key generator and efficient symmetric key distribution. *Digit. Commun. Netw.* **2020**, *6*, 486–503.