An Image Encryption Scheme Based on Logistic Quantum Chaos

Yu Wang 1, Liquan Chen 1,2,*, Kunliang Yu 1, Yuan Gao 1 and Yang Ma 1

1 School of Cyber Science and Engineering, Southeast University, Nanjing 210096, China; 230198994@seu.edu.cn (Y.W.); kunliang_yu@seu.edu.cn (K.Y.); 230209298@seu.edu.cn (Y.G.); mayang@jscas.gov.cn (Y.M.)
2 Purple Mountain Laboratories, Nanjing 211100, China
* Correspondence: lqchen@seu.edu.cn

Abstract: This paper proposes an image encryption scheme based on logistic quantum chaos. Firstly, we use compressive sensing algorithms to compress plaintext images and quantum logistic and Hadamard matrix to generate the measurement matrix. Secondly, the improved flexible representation of the quantum images (FRQI) encoding method is utilized for encoding the compressed image. The pixel value scrambling operation of the encoded image is realized by rotating the qubit around the axis. Finally, the quantum pixel is encoded into the pixel value in the classical computer, and the bit-level diffusion and scrambling are performed on it. Numerical analysis and simulation results show that our proposed scheme has the large keyspace and strong key sensitivity. The proposed scheme can also resist standard attack methods such as differential attacks and statistical analysis.

Keywords: quantum computation; quantum logistic; quantum image processing; quantum image representation; image encryption

1. Introduction

Digital images are important information carriers and are widely used in various fields, such as social networks, education and medical systems [1,2]. Images can carry much redundant information, and it is easy to cause inadvertent privacy leakage in the transmission process, which eventually brings huge economic loss to users [3]. Especially in recent years, there have been many private information leakage incidents, which have aroused widespread discussion and concern among the public. Therefore, the research on image security is receiving more and more attention from scholars [1,4,5]. Image encryption can protect users’ private information from being leaked. However, traditional encryption methods such as AES, DES, 3DES, RC and BlowFish cannot protect image security because these algorithms cannot eliminate the characteristics of adjacent pixel correlation and information redundancy of ciphertext images [6]. Because the image has a high correlation between adjacent pixels and data redundancy, these characteristics lead to the unsatisfactory results of traditional encryption methods and are easy to be attacked [7,8]. Given the limitations of traditional encryption methods, many scholars have also proposed corresponding image encryption algorithms [9–12]. Image encryption algorithms based on chaotic systems have become a research hotspot.

The chaotic system is a nonlinear dynamic system and is difficult to predict. Meanwhile, it is sensitive to initial parameters and ergodic. Based on the above characteristics, Matthews first proposed an encryption algorithm based on chaos in 1989 [13]. Hua et al. proposed a high efficiency scrambling and diffusion image encryption algorithm for the problem of the privacy leaks of medical images [1]. The robustness of the algorithm is excellent, and the encrypted medical images can effectively resist image noise attacks. Liu et al. used Markov chains to improve the chaotic sequence distribution of the coupled lattice mapping (CML) [14], making the improved sequence distribution uniform. The improved sequence in this scheme can pass 15 test suites of NIST 800. However, the time complexity in this work increases with the introduction of nodes, and the efficiency may
entropy exponentially. Goggin et al. used the recoil rotor model to measure the classical logistic system in 1990 and obtained the three-dimensional quantum logistic mapping [15]. Many scholars have proposed an image encryption algorithm based on quantum chaos based on Goggin's work [16,17].

With the introduction of quantum computers [18], quantum image representation, as a critical information format, has also attracted widespread attention from researchers in recent years [16,19–21]. Quantum computers are pretty different from traditional computers in storage and calculation methods. In terms of calculation methods, taking 2D digital images as an example, the calculation of images by classic computers is matrix operations, and the calculation of images by quantum computers is the evolution of unitary operators. Therefore, many researchers have proposed corresponding algorithms to represent images on quantum computers [21–24]. Yao et al. proposed an algorithm to encode 2D classical images into pure quantum states [23]. They assumed that the quantum state's probability amplitude represents the pixel value of the image on the quantum computer, and the quantum ground state represents the pixel's position. In 2003, Venegas-Andraca et al. first proposed the image quantum encoding algorithm based on qubit lattice [19,25]. They assume that a qubit stores each pixel value of the input 2D image. Therefore, at least 2n bits are required to store the pixel value for the input image. However, this storage method does not realize the quantum acceleration function. In 2011, Le et al. designed a flexible representation of quantum images (FRQI) based on the qubit lattice theory [21]. They took the pixel value and position in the form of a quantum state tensor product. The qubits required in the encoding process are reduced from 2n bits to n bits, which improves storage efficiency. Zhang et al. also proposed the NEQR quantum image encoding method. Both NEQR and FRQI use the base state |k⟩ to represent the pixel position, but it is worth noting that FRQI takes the pixel value during the encoding process [26]. The information is stored in the probability range, while NEQR stores the pixel value information in the ground state. Moreover, because NEQR's quantum state is orthogonal and distinguishable in each ground state, the NEQR has a secondary acceleration in the quantum image preparation process, and the compression rate of the quantum image is 1.5 times higher than FRQI. The researchers propose the corresponding quantum image processing technology and machine learning algorithm for quantum images based on the quantum image representation algorithm. In [27], Ma et al. encode image information using qubits and implement image dilation and erosion operations in quantum computers. In Reference [28], the authors use competitive learning and quantum computing to develop a new classification algorithm that can classify inputs even when they are incomplete. Zhou et al. used novel enhanced quantum representation (NEQR) to encode images and performed Arnold transformation on the pixel values of quantum images to achieve pixel-level scrambling, but they did not eliminate the security threat of Arnold transformation [16].

Compressed sensing can break through the limitations of the Nyquist sampling theorem and use sparse data to restore the original signal [29–32]. Researchers have applied it to image encryption and machine learning [29,33]. Combining chaotic systems and compressed sensing theory to propose image encryption algorithms is a field worthy of in-depth study. In the image encryption process, the length of the encryption time largely depends on the data dimension. If the data dimension can be reduced, the encryption time of the encryption system can be decreased. At present, many researchers have proposed chaos theory and compressive sensing algorithms. For example, Chai et al. used 2D compressed sensing algorithms to encrypt images into cryptographic images with visual significance [29]. Wang et al. used a parallel compressed sensing algorithm to encrypt images, in encryption time and excellent encryption effect [4].

In order to ensure that the quantum image can be securely transmitted and used in the future quantum computer era, we convert the classical image into a quantum state and then encrypt it. This paper proposes an image encryption scheme based on logistic quantum chaos. The contributions of this paper are summarized as follows:
1. We use quantum logistic mapping and the Hadamard matrix to generate the measurement matrix. This new matrix can improve encryption scheme security in the compressed sensing process in this paper.

2. We use the FRQI quantum encoding method to perform quantum encoding on the compressed image, which can significantly reduce the amount of encoded data without affecting the encryption result.

3. We combined the Bloch spherical surface to propose a new pixel value scrambling method. This new operation can reduce time complexity and improve system security in the encryption process.

4. We use the improved FRQI to encode the classical image. In the new quantum image representation scheme, the pixels of the classical image are only related to one rotation angle of the Bloch sphere. On the premise that the other rotation angle is unchanged, it is easier to encrypt the quantum image, and the complexity of the new quantum representation scheme is the same as that of FRQI.

The structure of this paper is as follows: In Section 2, we give the corresponding essential work. In Section 3, we describe the encryption and decryption process of the proposed algorithm. Section 4 analyzes the image security after encryption, and Section 5 concludes this paper.

2. Related Work
2.1. Compressive Sensing (CS)

Suppose the $N \times 1$ dimensional signal $f$ is a non-sparse signal [31]. We need to find an $N \times N$ dimensional orthogonal basis transform base matrix $\Psi_{N \times N}$ so that $f$ has a sparse representation in the transform domain $\Psi$. Namely:

$$f = \Psi x.$$ \hfill (1)

Therefore, the CS process can be expressed as:

$$y = \Phi \Psi x = \Phi f = Ax.$$ \hfill (2)

The $\Phi$ is the measurement matrix. $A$ represents the sensing matrix, which is an $M \times N$ dimensional matrix.

Since the dimension $M$ of the measurement vector $y$ is much lower than the dimension $N$ of the observed signal $x$, (1) is an underdetermined system, making $y$ impossible to accurately calculate the target signal $f$. However, CS theory proves that if the perception matrix $A$ can meet the Restricted Isometry Property (RIP) condition [32], the original signal can be recovered from the measured signal.

Therefore, the measurement matrix needs to meet:

$$(1 - \delta_k)||x||^2_2 \leq ||Ax||^2_2 \leq (1 + \delta_k)||x||^2_2.$$ \hfill (3)

The $\delta_k$ represents isometry constant. Reconstruct the original problem accurately by transforming it into an $l_1$ norm problem.

$$\min ||x||_1 \text{ s.t. } y = \Phi \Psi x.$$ \hfill (4)

The $||x||_1$ represents the $l_1$ norm of the vector $x$.

The image reconstruction algorithm can use orthogonal matching pursuit (OMP) [4,33,34], matching pursuit (MP) [35], subspace tracking (ST) or basis tracking algorithm. In order to effectively restore the original image from the measured matrix, in this paper, we use the OMP method to reconstruct the compressed image.
2.2. Quantum Logistic Mapping

In 1990, Goggin [15,17] used the recoil rotor model to quantify the Logistic chaotic system. The Goggin method uses the coupling dissipative quantum system and the harmonic oscillator path to produce the new quantum system. Finally, Goggin obtained the 3D quantum modified logistic system corresponding to \( \langle \delta a \delta a \rangle \) according to the Heisenberg dynamic motion equation \( \delta a \) [17]. The mathematical definition of the three-dimensional quantum logistic system is as follows:

\[
\begin{align*}
x_{n+1} &= r(x_n - |x_n|^2) - ry_n \\
y_{n+1} &= -y_ne^{-2\beta} + re^{-\beta}(2 - x_n - x_n^*)y_n - x_n z_n^* - x_n^* z_n \\
z_{n+1} &= -z_ne^{-2\beta} + re^{-\beta}(2(1 - x_n^*)z_n - 2x_n y_n - x_n) \\
\end{align*}
\]

\[ (5) \]

In (5), \( x = \langle a \rangle, y = \langle \delta a^+ \delta a \rangle, z = \langle \delta a \delta a \rangle \), \( \delta a \) describes the quantum fluctuations about \( \langle a \rangle \) and \( \delta a \to 0 \). The \( r \) represents the control parameter, \( \beta \) is the dissipation parameter and \( x_n^*, z_n^* \) is the conjugate complex number of \( x_n, z_n \), and in the chaotic state when \( x_n \in (0, 1), y_n \in (0, 0.1), z_n \in (0, 0.2), \beta \in [0, +\infty], r \in 4 \).

2.3. Quantum Encoding of Images

This section will use the quantum encoding method proposed in reference [22] to encode images and continue to encrypt images after encoding. The specific encoding method is as follows:

\[ |I(\theta, \varphi)\rangle = \frac{1}{2^n} \sum_{k=0}^{2^n-1} |c_k\rangle. \]

\[ (6) \]

\[
\begin{align*}
|c_k\rangle &= \cos \frac{\theta_k}{2} |0\rangle + e^{i\varphi_k} \sin \frac{\theta_k}{2} |1\rangle \\
\theta_k &= \frac{2n\pi}{2^n}, 0 \leq \theta_k \leq \pi \\
\varphi_k &= 2\pi x_k, 0 \leq \varphi_k \leq 2\pi \\
\end{align*}
\]

\[ (7) \]

The \( \theta \) and \( \varphi \) are the Bloch vector angles. The position of the \( k \)-th \( (k = 0, 1, \ldots, 2^n-1) \) pixel is described as \( |k\rangle \). The gray value is \( |c_k\rangle \). The \( x_k \) is a pseudo-random number in \([0, 1]\). In this article, we use the quantum logistic equation to generate \( x_k \).

The quantum state of the encoded image satisfies the normalization condition [22]:

\[
||I(\theta, \varphi)|| = \frac{1}{2^n} \sqrt{\sum_{k=0}^{2^n-1} \left( \cos \frac{\theta_k}{2} \right)^2 + \left| e^{i\varphi_k} \sin \frac{\theta_k}{2} \right|^2} = 1.
\]

\[ (8) \]

At the same time, reference [22] gives a fast implementation of image quantum coding, the specific description is as follows:

Step 1: Define quantum revolving door \( R_k(\theta_k, \varphi_k) \):

\[
R_k(\theta_k, \varphi_k) = \begin{pmatrix}
\cos \frac{\theta_k}{2} & -e^{i\varphi_k} \sin \frac{\theta_k}{2} \\
e^{i\varphi_k} \sin \frac{\theta_k}{2} & \cos \frac{\theta_k}{2}
\end{pmatrix}.
\]

\[ (9) \]

Step 2: Define a controlled revolving door \( CR_k(\theta_k, \varphi_k) \):

\[
CR_k(\theta_k, \varphi_k) = \left( \sum_{j=0}^{2^n-1} |j\rangle \langle j| \right) \otimes I + |k\rangle \langle k| \otimes R_k(\theta_k, \varphi_k).
\]

\[ (10) \]

Step 3: Prepare the initial state of the qubit \( |0\rangle^{\otimes 2^n+1} \).

Step 4: To calculate the intermediate state, use the two-dimensional Hadamard matrix and the identity matrix \( |H\rangle \).
\[ |H \rangle = \left( \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \otimes 2^n \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right) |0 \rangle \otimes 2^{n+1} = \frac{1}{2^n} \left( \sum_{k=0}^{2^n-1} |k \rangle \right) |0 \rangle. \] (11)

Step 5: Perform \( CR_k(\theta_k, \varphi_k) \) on the intermediate state:

\[ CR_k(\theta_k, \varphi_k) |H \rangle = \frac{1}{2^n} \left( \sum_{j=0, j \neq k}^{2^n-1} |j \rangle \otimes |0 \rangle + |k \rangle \otimes |c_k \rangle \right). \] (12)

Step 6: In order to obtain the encoded quantum image, perform \( CR_k(\theta_k, \varphi_k) \) continuously on the intermediate state:

\[ I(\theta, \varphi) = \left( \prod_{k=0}^{2^n-1} CR_k(\theta_k, \varphi_k) \right) |H \rangle. \] (13)

3. Algorithm Description

In this section, we select Lena.bmp as plaintext, in which \( M = 256 \) and \( N = 256 \), where \( M \) and \( N \) are the length and width of plaintext, respectively. Quantum logistic function’s parameters are: \( x = 0.5, y = 0.0032, z = 0.12, r = 3.99, \beta = 7 \). The iterative number of quantum logistic is related to the size of plaintext images. Orthogonal matching pursuit (OMP) algorithm is used to reconstruct the image after compressive sensing.

3.1. Encryption Process

3.1.1. Compressed Sensing Process

In the encryption process, the plaintext image is represented by \( I \).

Step 1: The two-dimensional wavelet transforms to the plaintext image \( I \), transformation model is haar transform model. After the two-dimensional wavelet transforms, we can obtain the decomposition coefficients \( c \) and length \( s \) for each layer.

\[
\begin{align*}
[c, s] &= \text{wavedec2}(I, N, 'haar') \\
\text{end} &= [A(N)|H(N)|V(N)|D(N)|H(N-1)|...|H(1)|V(1)|D(1)]^T.
\end{align*}
\] (14)

\( N \) is the layer of the two-dimensional wavelet transform, and we set \( N = 4 \) in step 1. \( H, V \) and \( D \) are three directions of plaintext image. The \( A \) and \( H \) are low frequency and the high frequency of each layer, respectively. The low frequency contains the feature of the plaintext image, and the high frequency contains the details of the plaintext image. To ensure the quality of the reconstructed image, we only compress the high-frequency data, and the low-frequency data of the plaintext image will not be compressed.

Step 1.1: Use QR decomposition to low-frequency data \( cA \),

\[
\begin{cases}
\{ \begin{array}{l} cA = c(A(N)) \\ [q, r] = QR(cA) \end{array} \}
\end{cases}
\] (15)

\( N \) is the two-dimensional wavelet transform layer number and \( N = 1, 2, 3, 4 \). Both \( q \) and \( r \) are matrices.

Step 1.2: Extract the diagonal data in the \( r \) and append it to the end of matrix \( q \) as one of the decryption keys. The matrix \( r \) after being extracted data replaces the \( cA \) as a part of the plaintext image to encrypt.

\[
\begin{cases}
\{ \begin{array}{l} temp(i) = r(i, i) \\ r(i, i) = 0 \end{array} \}
\end{cases}
\] (16)

The vector \( temp \) is a temporary vector, and \( i \) is the position index of the matrix \( r \).
Step 1.3: Grouping high-frequency data:

\[
\begin{align*}
    c_{Bh}(i) &= c(H(i)) \\
    c_{Bv}(i) &= c(V(i)), \ i = 1, 2, 3, \ldots, N. \\
    c_{Bd}(i) &= c(D(i))
\end{align*}
\]  

Equation (17) can be used to obtain high-frequency data of each layer \(c_B\), and then \(c_B\) is transformed to the matrix according to the length of the decomposition coefficient \(s\) for each layer. Namely,

\[
    c_{B_{h,v,d}}(i) = \text{reshape}(c_{B_{h,v,d}}(i), s_{h,v,d}).
\]  

Step 2: To ensure the high reconstruction quality of the plaintext image, the high-frequency data after wavelet decomposition needs to be sparse operation. We use Algorithm 1 to sparse the high-frequency data matrix.

**Algorithm 1 Calculating the sparsity threshold**

**Input:** \(c_B = c_{B_{h,v,d}} + r\).

**Output:** sparsity matrix \(STD\).

1. Set sparse rate: \(0.10 \leq iSTD \leq 0.15\);
2. Obtain length and width of the matrix: \(s_{h,v,d}\).
3. Sparsify each row of the matrix,
   \[
   \text{std} = 0; \quad // \text{sparsity threshold} \\
   iSTD = \text{sparsity}(c_B, \text{std}); \quad // \text{sparsity rate}
   \]
4. while \(\text{std} < 0.1\) and \(0.10 \leq iSTD \leq 0.15\)

   \[
   \text{std} = \text{std} + 1 \times 10^{-4} \\
   iSTD = \text{sparsity}(c_B, \text{std});
   \]
5. end
6. pos = find(abs(cB) < std);
7. \(c_B(pos) = 0;\)

Step 3: After sparsification, the overall sparsity may reach the \(iSTD\), but some data may be concentrated within a particular region. Therefore, it is necessary to displace the data and distribute these pixel points evenly throughout the sparse matrix. Use Equation (19) to confuse the matrix,

\[
[STD_{i,j}, \text{index}] = \text{josephTraverse}(STD_{i,j}, \text{start}, \text{space}).
\]  

Both \(\text{start}\) and \(\text{space}\) are the confusing start and end positions, respectively. The \(\text{index}\) represents the position of the original matrix, which is a part of the decryption key. The \text{josephTraverse} is the confusion function.

Step 4: Building the measurement matrix \(\Phi\).

Repeat Algorithm 2 until each layer’s high-frequency and low-frequency data have corresponding Hadamard measurement matrices \(\Phi_{ij}^{M \times N/2^r}\).

Step 5: Use Equations (20) and (21) to compressive high-frequency and low-frequency data,

\[
    y_{ij} = \Phi_{ij}^{s(1,1) \times s(1,2)/2}STD_{ij}, i \in (1, N), j \in (1, 3), \quad (20)
\]

\[
    r = \Phi_{1,1}^{s(1,1) \times s(1,2)/2^r}r. \quad (21)
\]

Splice all matrices into the compressed matrix \(x\) after compressive sensing.

Step 6: Quantizing the compressed matrix \(x\) can facilitate bit-level encryption and quantum pixel conversion operations.

\[
    x = \text{uint8}(\frac{x - \text{min}(x)}{\text{max}(x) - \text{min}(x)}).
\]  

(22)
Algorithm 2 Building Hadamard measurement matrix based on quantum logistic system

Input: sampling rate: sample = 0.5 and parameters of the quantum logistic system.

Output: Measurement matrix \( \Phi \).

1: Obtain size of STDi, marked as \([m,n]\).
   \([m,n] = \text{size(STDi)}; \)
2: The Hadamard matrix \( H \) was calculated according to the sample rate.
   \( H = \text{Hadamard}(m, n \times \text{sample}); \)
3: The iterative quantum logistic mapping generates the confusion vector \( cs \). To ensure the security of the confusion vector, we discard the data of the first 200 iterations and take the values from 201 to 201 + \((m \times n/2)\) as the confusion vector of the Hadamard matrix. The values are transposed into a matrix and denoted as \( cs' \).
   \( cs(201: 201 + (m \times n/2)) = \text{quantum}_\text{logistic}(\text{parameters}); \)
   \( cs' = \text{reshape}(cs, m, n/2); \)
4: To sort row and col of confusion matrix \( cs' \), denote the index after sorted as \( \text{index} \).
   \( \text{while } i = \text{row} \text{ and col } \)
   \( \[, \text{index} \] = \text{sort}(cs' (i)); \)
   \( cs'(i) = \text{index}; \)
   \( \text{end} \)
5: Confuse the Hadamard matrix,
   \[ \Phi = \text{Hadamard}(\text{index}); \] \( (23) \)

3.1.2. Quantum Pixel-Level Encryption

Step 7: Use Equations (9)–(13) to quantum code the image \( x \) after compressive sensing, the size of \( x \) is \( M/2 \times N \) and the coding result is recorded as \( I(\theta, \varphi) \).

Step 8: Scrambling operations in quantum pixel level. From Section 2.3, we know that each pixel value is only related to \( \theta_k \) and not related to \( \varphi \). Therefore, we can let \( \varphi \) remain unchanged and rotate \( \theta_k \) to change the gray value.

Calculate the rotation angle \( \Delta \theta_k \):
\[
\theta_1 = \text{mod} (\text{abs}(\frac{q_2}{q_3} \times 2^{13}), 1) \times \pi,
\]
\[
\Delta \theta_k = \theta_k - \theta_1
\]
\( q_1, q_2, q_3 \) are pseudo-random sequences generated by the quantum logistic system. \( q_1 \) is involved in the quantum encoding generation of images.

Let the Bloch sphere coordinates of \( |c_k\rangle \) be \((x_k, y_k, z_k)\) to make \( |c_k\rangle \) rotate toward the Bloch sphere to the point \((0, 0, -1)\). The rotation axis should be:
\[
R_k = \frac{(x_k, y_k, z_k) \times (0,0,-1)}{|(x_k, y_k, z_k) \times (0,0,-1)|} = \frac{(-y_k, x_k, 0)}{\sqrt{x_k^2 + y_k^2}}.
\]

The rotation matrix is,
\[
M_k = \cos \frac{\Delta \theta_k}{2} I - i \sin \frac{\Delta \theta_k}{2} (R_k \times \sigma).
\]
\( I \) is the unit matrix, and \( i \) is the imaginary unit. \( \sigma \) represents the Bubbleley matrix \( \sigma = (\sigma_x, \sigma_y, \sigma_z) \):
\[
\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.
\]

Controlled revolving doors:
CM_k = \left( \sum_{j=0,j \neq k}^{2^n-1} |j\rangle \langle j| \otimes I + |k\rangle \langle k| \otimes M_k \right). \quad (29)

The rotation on the sphere can be defined as:

\[ |I'(\theta, \phi)\rangle = (\prod_{k=0}^{2^n-1} CM_k) |I(\theta, \phi)\rangle = \frac{1}{2^n} \sum_{k=0}^{2^n-1} |k\rangle |c'_k\rangle. \quad (30)\]

3.1.3. Bit-Level Encryption

Step 9: Quantum encoding image is scrambled in bit-level. Calculate the pixel value \(c'_1\) according to \(\Delta \theta_k\). Use Equation (25) to scramble pixel value \(c'_1\).

\[
\begin{align*}
\{ & X = \text{quantum\_Logistic}(M \times N \times 8, \text{parameters}) \\
& Q = \text{dec2hex}((X \times 2^{13}) \mod 256) \\
& C_1 = c'_1 \oplus Q_1 \oplus c'_{M \times N} \\
& C_i = c'_i \oplus Q_i \oplus c'_{i-1}, i \in (2, M \times N) 
\end{align*}
\]

\(X\) is the pseudo-random value generated by the iterative quantum logistic system. The \(\oplus\) is the bitwise-XOR operation.

Step 10: Semi-ciphertext \(C\) is scrambled by the josephTraverse function in bit level.

\[ [C, \text{index}] = \text{josephTraverse}(C, \text{start}, \text{space}) \quad (32)\]

At this point, the encryption process is all over. The encryption process flowchart is shown in Figure 1. Figures 2–4 show the encryption and decryption results of the proposed scheme.

Figure 1. Encryption flowchart.
3.2. Decryption Process

The decryption process is the reverse process of the encryption process. The details are as follows:

Step 1: Use the $ijosephTraverse$ function to inverse scrambling on the ciphertext $C$.

$$C' = ijosephTraverse(C, index). \quad (33)$$

The $index$ is the coordinate before $josephTraverse$. $C'$ is the decryption result.
Step 2: Bit-level inverse diffusion.

\[
\begin{align*}
X &= quantum\_\text{Logistic}(M \times N \times 8, \text{paramaters}) \\
Q &= \text{dec2hex}((X \times 2^{13}) \mod 256) \\
c_i &= C_i \oplus Q_i \oplus C_{i-1}, i \in (2, M \times N) \\
c_1 &= C_1 \oplus Q_1 \oplus C_{M \times N}.
\end{align*}
\] (34)

The \( X \) is the pseudo-random value generated by the iterative quantum logistic system. The \( \oplus \) is the bitwise-XOR operation.

Step 3: Decrypt the quantum coding image. The rotation matrices used in the encryption process are all unitary matrices and conform to the basic principles of quantum mechanics [21]. Therefore, we only need to change rotation matrices to conjugate transpose to decrypt the ciphertext in the decryption process.

\( M_k \) is rotation matrices. Let \( M_k^+ \) be the conjugate transpose matrix of \( M_k \). According to unity, we can obtain \( M_k^+ \):

\[
M_k M_k^+ = M_k^+ M_k = I. \tag{35}
\]

Calculate the decryption-controlled revolving door \( C_k^{\text{ent}} \) according to Equation (29). Use Equation (36) to decrypt quantum coding ciphertext image:

\[
|I(\theta, \varphi)\rangle = \left( \prod_{k=0}^{2^n-1} C_{k,1} \right) |I(\theta, \varphi)\rangle. \tag{36}
\]

Before decrypting at the bit level, the decrypted ciphertext image \( |I(\theta, \varphi)\rangle \) needs to be converted into a gray value by quantum measurement. In order to achieve quantum measurement, we define the measurement symbol \( K \),

\[
K = \sum_{k=0}^{2^n-1} (k_0^{(n)} p_0^{(n)} + k_1^{(n)} p_1^{(n)}).
\] (37)

\( p_0^{(n)} = |n\rangle|0\rangle \langle 0| \) and \( p_1^{(n)} = |n\rangle|1\rangle \langle 1| \) are a set of orthogonal projection matrices corresponding to the eigenvalues \( k_0^{(n)} \) and \( k_1^{(n)} \) of \( K \).

For the decrypted quantum image \( |I(\theta, \varphi)\rangle \), the probabilities of obtaining eigenvalues \( k_0^{(n)} \) and \( k_1^{(n)} \) with the measurement symbol \( K \) are:

\[
\begin{align*}
P(k_0^{(n)}) &= \langle I(\theta, \varphi)|k_0^{(n)}|I(\theta, \varphi)\rangle = \frac{1}{2^n} \cos^2 \frac{\theta_n}{2} \\
P(k_1^{(n)}) &= \langle I(\theta, \varphi)|k_1^{(n)}|I(\theta, \varphi)\rangle = \frac{1}{2^n} |e^{i\varphi_n}|^2 \sin^2 \frac{\theta_n}{2}.
\end{align*}
\] (38)

The gray value of the \( n \)-th pixel \( c_n \):

\[
\begin{align*}
\theta_n &= 2 \arcsin(\sqrt{2^n P(m_1^{(n)})}) \\
c_n &= \text{ent}\left(\frac{2^n - 1}{\pi} \theta_n\right).
\end{align*}
\] (39)

Step 4: Inversely scramble the image \( c_n \).

Step 5: Use OMP to reconstruct the image of the decrypted compressive sensing semi-plaintext image.

Step 6: Use inverse wavelet transform to restore the plaintext image \( P \).

The decryption process flowchart is shown in Figure 5.
4. Security Analysis

4.1. Keyspace Analysis

An excellent image encryption algorithm should have a large enough keyspace to resist the brute force attack analysis [36]. In theory, the keyspace needs to reach the power of $2^{100}$ to resist the brute force attack analysis of the attacker [37]. In our proposed algorithm, our algorithm key consists of two parts:

1. Quantum logistic parameters: $x, y, z, r, \beta$ and the number of max iterations; the number of wavelets transforms layers $N$, the start position and interval distance of the $josephTraverse$ algorithm; the quantum logistic parameter in the quantum encoding process and the rotation angle $\theta$.

2. The $q$ matrix and the $josephTraverse$ generated the index during the encryption process.

Assuming that the calculation accuracy of the computer is $10^{-14}$, our algorithm keyspace is:

$$\left(10^{14}\right)^{16} \times \left(10^{14}\right)^{2(MN)} = \left(10^{224}\right) \times \left(10^{14}\right)^{2(MN)} > 2^{300}.$$  

This result is much larger than the standard requirement of $2^{100}$, so the encryption algorithm we propose can resist brute force attack analysis. As shown in Table 1, compared with the research in references [4, 17, 33–40], our image encryption algorithm has a large enough keyspace to withstand all types of brute force attacks.

| Keyspace               | Our algorithm       | Reference [33] | Reference [4] | Reference [38] | Reference [39] | Reference [40] | Reference [17] |
|------------------------|---------------------|----------------|---------------|----------------|----------------|----------------|----------------|
|                        | >$2^{300}$          | 2$^{294}$       | 2$^{250}$      | >$10^8$        | 2$^{186}$       | 2$^{149}$       | 2$^{125}$       |
4.2. Key Sensitivity Analysis

In Figure 6, we use Lena256 as the encryption object to test the key sensitivity of our encryption system. During the test, we used the original key as: \[ \text{Key} = [x = 0.5, y = 0.0032, z = 0.12, r = 3.99, \beta = 7, q, \text{index}] \]. We add $10^{-10}$ to the above keys respectively and use the modified key to decrypt the ciphertext in the decryption process. The decryption result is shown in Figure 6. From Figure 6, we can see that even if the key changes $10^{-10}$, the wrong key cannot decrypt the ciphertext correctly, proving our encryption scheme is highly sensitive to the key.

![Figure 6](image)

**Figure 6.** Key sensitivity analysis. (a) Correct key. (b) $x + 10^{-10}$. (c) $y + 10^{-10}$. (d) $z + 10^{-10}$. (e) $r + 10^{-10}$. (f) $\beta + 10^{-10}$. (g) $\theta + 10^{-10}$. (h) $q + 10^{-10}$. (i) $\text{index} + 10^{-10}$.

4.3. Statistical Analysis

4.3.1. Histogram Analysis

Histogram analysis is one of the essential indicators to measure the security of an encryption algorithm. The histogram can measure the degree of pixel change of a ciphertext image. An excellent encryption algorithm can eliminate the correlation between adjacent pixels in the picture and the histogram feature of the original plaintext information. Section 4.3 uses Lena, the Boat, and 7.2.01.tiff as the plaintext image. The (a), (c) and (e) of Figure 7 show that the plaintext histogram has uneven peaks, reflecting the pixel values distribution of the original images. From (b), (d) and (f) of Figure 7, it can be found that
the histograms of the dense image are different from the histogram of the original images, and the uneven peaks become uniformly distributed. The attacker may not obtain the corresponding plaintext information by analyzing the histogram of the ciphertext image. Therefore, the proposed image compression and encryption algorithm can resist statistical analysis attacks.

Figure 7. Histogram. (a) Lena. (b) Ciphertext of Lena. (c) Boat. (d) Ciphertext of Boat. (e) 7.2.01.tiff. (f) Ciphertext of 7.2.01.tiff.

4.3.2. Adjacent Pixel Correlation

There is a strong correlation between adjacent pixels of a plaintext image, and an attacker can crack the algorithm by analyzing the correlation between adjacent pixels. Therefore, the correlation between pixels needs to be reduced to avoid statistical attacks in the design of the encryption algorithm. In this article, we use Equation (40) to calculate the correlation between adjacent pixels:

\[ r_{xy} = \frac{\text{cov}(x, y)}{\sqrt{D(x)} \sqrt{D(y)}} \]  

(40)
In Equation (40), \( \text{cov}(x, y) = \frac{1}{N} \sum_{i=1}^{N} (x_i - E(x))(y_i - E(y)), D(x) = \frac{1}{N} \sum_{i=1}^{N} (x_i - E(x))^2, \)

\( E(x) = \frac{1}{N} \sum_{i=1}^{N} x_i, \) \( x \) and \( y \) are adjacent pixel values. \( N \) is the total number of pixels selected from the image.

We select 17 images of different sizes as plaintext for encryption. Randomly select 10,000 pairs of adjacent pixel values, and calculate its correlation in three different directions: vertical, horizontal and diagonal. The specific calculation results are shown in Table 2, and the corresponding adjacent pixel correlation distribution is shown in Figure 8. Finally, we choose Lena’s ciphertext image to show the horizontal, vertical and diagonal pixel distribution in Figure 9.

### Table 2. Adjacent pixel correlation of ciphertext.

| Size     | Image           | Horizontal | Vertical | Diagonal |
|----------|-----------------|------------|----------|----------|
| 256 × 256| Lena.jpg        | 0.0011     | 0.0050   | −0.0032  |
|          | 4.1.01.tiff     | −0.0028    | −0.0126  | −0.0171  |
|          | 4.1.02.tiff     | 0.0052     | 0.0011   | 0.0057   |
|          | 4.1.03.tiff     | 8.8213 × 10^{-5} | −0.0081 | −0.0079  |
|          | 4.1.04.tiff     | 0.0034     | 0.0014   | 0.0011   |
|          | 4.1.05.tiff     | 0.0020     | 0.0023   | −0.0082  |
|          | 4.1.06.tiff     | 0.0044     | −0.0076  | 0.0020   |
|          | 4.1.07.tiff     | 0.0080     | 0.0019   | 0.0010   |
|          | 4.1.08.tiff     | −0.0094    | 0.0011   | 0.0033   |
| 512 × 512| 4.2.01.tiff     | 0.0011     | −0.0010  | −0.0015  |
|          | 4.2.03.tiff     | −0.0269    | 0.0019   | −0.0214  |
|          | 4.2.05.tiff     | 0.0055     | −0.0317  | 0.0017   |
|          | 4.2.06.tiff     | −0.0372    | −0.0155  | −0.0069  |
|          | 4.2.07.tiff     | 0.0093     | −0.0177  | 0.0078   |
| 1024 × 1024 | 7.2.01.tiff  | −0.0227    | −0.0015  | 0.0018   |
|          | 5.3.01.tiff     | 0.0022     | 0.0011   | 0.0022   |
|          | 5.3.02.tiff     | 0.0020     | 0.0075   | −0.0185  |

Figure 8. Adjacent pixel correlation of ciphertext.
Figure 9. Lena’s adjacent pixel distribution. (a,c,e) are plaintext Lena. (b,d,f) are ciphertext Lena.

We can see that all the pixel values in the plaintext Lena is concentrated near a certain line, (a), (c) and (e) of Figure 9. However, in the pixel distribution of the ciphertext, the pixel values are evenly distributed throughout the space in (b), (d) and (f) of Figure 9, proving that the proposed encryption algorithm can resist statistical attack analysis.

At the same time, we also compare our proposed algorithm with other similar encryption algorithms using an image of Lena256. Table 3 shows that our algorithm correlation coefficient of adjacent pixels is better than similar encryption algorithms.
Table 3. Compare with another algorithm.

|                      | Horizontal | Vertical | Diagonal |
|----------------------|------------|----------|----------|
| Our algorithm        | 0.0011     | 0.0050   | −0.0032  |
| Reference [4]        | 0.0062     | −0.1017  | 0.0052   |
| Reference [38]       | −0.0072    | 0.0086   | −0.0028  |
| Reference [39]       | 0.0010     | −0.0008  | 0.0008   |
| Reference [40]       | 0.0018     | 0.0014   | 0.0034   |
| Reference [17]       | 0.0009     | 0.0020   | −0.0016  |

4.3.3. Information Entropy

In 1948, Shannon proposed [41] the concept of information entropy, which is the degree of confusion in the information source. The concept of information entropy can describe the uncertainty of the ciphertext. In an ideal state, the closer the information entropy is to 8, the better the encryption effect.

\[
H(s) = -\sum_{i=0}^{2^L-1} p(s_i) \log_2 \frac{1}{p(s_i)}
\]  \hspace{1cm} (41)

We calculate the information entropy of 17 ciphertext images of different sizes, shown in Table 4. From Table 4, we can see that the information entropy of the 17 ciphertext images is close to the ideal value of 8. These results indicate that the ciphertext images are very messy and can resist the statistical attack analysis of the attacker.

Table 4. Entropy.

| Size          | Test Image      | Entropy             |
|---------------|-----------------|---------------------|
| 256 × 256     | Lena.jpg        | 7.999382215618442  |
|               | 4.1.01.tiff     | 7.999232866133528  |
|               | 4.1.02.tiff     | 7.999842999290084  |
|               | 4.1.03.tiff     | 7.999372347262450  |
|               | 4.1.04.tiff     | 7.999533419091807  |
|               | 4.1.05.tiff     | 7.998158601569759  |
|               | 4.1.06.tiff     | 7.997645074489912  |
|               | 4.1.07.tiff     | 7.998344070593309  |
|               | 4.1.08.tiff     | 7.999008694956089  |
| 512 × 512     | 4.2.01.tiff     | 7.999808123472354  |
|               | 4.2.03.tiff     | 7.998547327068983  |
|               | 4.2.05.tiff     | 7.998283067128929  |
|               | 4.2.06.tiff     | 7.998567490111828  |
|               | 4.2.07.tiff     | 7.998694994664158  |
| 1024 × 1024   | 7.2.01.tiff     | 7.999662345147449  |
|               | 5.3.01.tiff     | 7.999678856272011  |
|               | 5.3.02.tiff     | 7.999649730480633  |

Table 4 uses Lena256 as the information entropy test object to compare with other similar encryption algorithms. From the comparison results in Table 5, the information entropy of our proposed algorithm is better than similar algorithms.
Table 5. Compare with another algorithm.

|                | Entropy  |
|----------------|---------|
| Our algorithm  | 7.9993  |
| Reference [33] | -       |
| Reference [4]  | 7.9968  |
| Reference [38] | 7.9985  |
| Reference [39] | 7.9977  |
| Reference [40] | 7.9986  |
| Reference [17] | 7.9993  |

4.4. Robustness Analysis

Robustness is an important indicator that shows whether a cryptographic system can resist interference. In Section 4.4, we use noise attacks and blocking attacks to verify the robust performance of our proposed cryptographic system.

4.4.1. Noise Attack Analysis

Common noise types include Gaussian noise, Poisson noise, multiplicative noise and salt and pepper noise. Salt and pepper noise has the most significant impact on the ciphertext image among all the noises. We add salt and pepper noise to the ciphertext image in Section 4.4.1 and decrypt the ciphertext after adding noise. Figure 10 shows the decrypted image with different noise intensities. From Figure 10, we can see that when the salt and pepper noise intensity reaches 0.1, the decryption result can also be distinguished with the naked eye, proving that our algorithm can resist noise attacks.

![Figure 10. Decrypted images under different intensities of noise. (a) Decrypted 0.0001. (b) Decrypted 0.001. (c) Decrypted 0.01. (d) Decrypted 0.1.](image-url)
4.4.2. Crop Attack Analysis

We crop the $8 \times 8$, $16 \times 16$ and $32 \times 32$ information of the ciphertext image for testing. The quality of the decrypted image will decrease as the block size increases. In Figures 11–13, we show the ciphertext image cropped with different sizes of block and the corresponding decrypted image. From the experimental results, it can be seen that when the crop size is $32 \times 32$, the decryption result has the greatest impact, but even when the decryption result is the worst, we can still understand the information carried by the decrypted image, which proves that our algorithm, to a certain extent, can withstand tailoring attacks.

![Figure 11](image1.png)

**Figure 11.** Loss data of $8 \times 8$. (a) Top left. (b) Center. (c) Bottom right. (d) Decrypted image (top left). (e) Decrypted image (center). (f) Decrypted image (bottom right).

![Figure 12](image2.png)

**Figure 12.** Loss data of $16 \times 16$. (a) Top left. (b) Center. (c) Bottom right. (d) Decrypted image (top left). (e) Decrypted image (center). (f) Decrypted image (bottom right).
4.5. Resistance to Difference Analysis

Differential attack refers to a cryptanalysis technique that obtains the encryption key by analyzing the difference between two ciphertexts after making small changes on the plaintext. The number of pixels changes rate (NPCR) and unified average changing intensity (UACI) are generally used as evaluation indicators when measuring whether a cryptographic system can resist differential attacks. The ideal values of the two indicators are NPCR $\geq 99.6093\%$ and UACI $\geq 33.4635\%$.

\[
NPCR = \frac{\sum_{i,j} D(i,j)}{W \times H} \times 100
\]  
\[
UACI = \frac{1}{W \times H} \left[ \frac{\sum_{i,j} |c_1(i,j) - c_2(i,j)|}{255} \right] \times 100
\]
\[
D(i,j) = \begin{cases} 
1 & \text{if } c_1(i,j) \neq c_2(i,j) \\
0 & \text{if } c_1(i,j) = c_2(i,j)
\end{cases}
\]

$W$ and $H$ represent the image’s width and height, respectively. $c_1$ and $c_2$ are two ciphertext pixels of images. $i$ and $j$ are the same position index of the $c_1$ and $c_2$.

We calculated the NPCR and UACI of 15 ciphertext images with 1bit data changed in Table 6. We can see that our NPCR and UACI are close to their respective ideal values, proving that our algorithm can resist differential attacks. Table 7 uses Lena256 as the NPCR and UACI test object to compare with other similar encryption algorithms. From the comparison results, the NPCR and UACI of our proposed algorithm are better than similar algorithms.
Table 6. NPCR and UACI.

| Size       | Image         | NPCR (%)  | UACI (%)     |
|------------|---------------|-----------|--------------|
| 256 × 256  | Lena.jpg      | 0.996376037597656 | 0.336031279919194 |
|            | 4.1.01.tif    | 0.996520996093750  | 0.33512089269301  |
|            | 4.1.02.tif    | 0.989776611328125  | 0.33527267156863  |
|            | 4.1.03.tif    | 0.9970703125000000 | 0.342420630361520  |
|            | 4.1.04.tif    | 0.996459960937500  | 0.33628314568015  |
|            | 4.1.05.tif    | 0.998565673828125  | 0.344157978132659  |
|            | 4.1.06.tif    | 0.979400634765625  | 0.343144435508578  |
|            | 4.1.07.tif    | 0.996490478515625  | 0.336412018420650  |
|            | 4.1.08.tif    | 0.987884521484375  | 0.336220894607843  |
| 512 × 512  | 4.2.01.tif    | 0.996910095214844  | 0.33531793325674  |
|            | 4.2.03.tif    | 0.996177673339844  | 0.335232107125077  |
|            | 4.2.05.tif    | 0.997024536132813  | 0.331339602002911  |
|            | 4.2.06.tif    | 0.995491027832031  | 0.336372472462471  |
|            | 4.2.07.tif    | 0.996109008789063  | 0.337656806497013  |
| 1024 × 1024| 7.2.01.tif    | 0.994831085205078  | 0.333878127453374  |
|            | 5.3.01.tif    | 0.997009277343750  | 0.352036449955959  |
|            | 5.3.02.tif    | 0.996718551635742  | 0.336304112752279  |

Table 7. Compare with another algorithm.

|                | NPCR (%) | UACI (%) |
|----------------|----------|----------|
| Our algorithm  | 99.637   | 33.60    |
| Reference [33] | -        | -        |
| Reference [4]  | -        | -        |
| Reference [38] | -        | -        |
| Reference [39] | 99.621   | 33.51    |
| Reference [40] | -        | -        |
| Reference [17] | 99.607   | 33.51    |

4.6. Time Complexity Analysis

Time complexity analysis is a critical indicator to measure an encryption algorithm [42]. The primary encryption time consumption concentrates on wavelet transform (Step 1), sparse (Step 2), iterative quantum logistic system (Steps 2–5), normalization (Step 6), joseph-Traversal (Steps 3, 5), image quantum coding (Step 7) and quantum scrambling (Step 8) in the encryption process. In the compressive sensing process (wavelet transform (Step 1) and sparse (Step 2)), the time complexity of constructing the measurement matrix for compressed sensing is $O(n \log n)$. In the quantum encoding stage, we use the same number of quantum gates in the encoding method as FRQI, so the computational complexity of the two is the same, that is, $O(\log^2 n)$, and the time complexity during the axis rotation phase is easy to calculate as $O(n^2)$. The total time complexity of bit-level scrambling and diffusion does not exceed $O(n^2)$ in Steps 2–5, Step 6 and Steps 3 and 5. The proposed encryption algorithm has better performance than classical algorithms in computational complexity.

5. Conclusions

This paper uses the FRQI quantum coding method on the classical computer. Firstly, we use compressive sensing algorithms to compress plaintext images to reduce data dimensions for shortening the time of quantum encoding. Secondly, FRQI is used to encode the compressed image and the scrambling operation by rotating the qubit around the axis. Finally, the bit-level diffusion and scrambling in the classical computer is performed. The computational complexity is not high in the whole encryption process. However, there is no way to achieve quantum parallelism in the simulation of classical computers, resulting in low computational efficiency. Although we compress the image data in the encryption
process, the encoding time is still unsatisfactory. Because the existing quantum image coding methods and image storage models are currently in the theoretical research stage, the coding efficiency is not satisfactory. In addition, numerical simulation and theoretical analysis show that our proposed algorithm has better performance when compared with similar algorithms. Therefore, our scheme can guarantee the security of the image during transmission in the future quantum era.

**Author Contributions:** Y.W.: conceptualization, methodology, software, writing—original draft; L.C.: resources, supervision, project administration, funding acquisition; K.Y.: writing—review and editing, visualization; Y.G.: writing—review and editing, investigation; Y.M.: writing—review and editing, investigation. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research is supported by the National Natural Science Foundation of China (62002058) and the National Key Research and Development program of China (2020YFE0200600).

**Conflicts of Interest:** All the authors of this research work declare that the work presented is in its original form and contains no conflict of interest.

**References**

1. Hua, Z.; Yi, S.; Zhou, Y. Medical image encryption using high-speed scrambling and pixel adaptive diffusion. *Signal Process. 2018*, 144, 134–144. [CrossRef]
2. Ye, G. Image scrambling encryption algorithm of pixel bit based on chaos map. *Pattern Recognit. Lett. 2010*, 31, 347–354. [CrossRef]
3. Hu, G.; Hou, W.; Dong, J.; Peng, J. A novel image encryption algorithm based on cellular neural networks hyper chaotic system. In *Proceedings of the 2018 IEEE 4th International Conference on Computer and Communications*, Chengdu, China, 7–10 December 2018; pp. 1878–1882. [CrossRef]
4. Wang, H.; Xiao, D.; Li, M.; Xiang, Y.; Li, X. A visually secure image encryption scheme based on parallel compressive sensing. *Signal Process. 2019*, 155, 218–232. [CrossRef]
5. Zhang, L.; Tian, X.; Xia, S. Scrambling algorithm of image encryption based on Rubik’s cube rotation and logistic sequence. In *Proceedings of the 2011 International Conference on Multimedia and Signal Processing*, Guilin, China, 14–15 May 2011; pp. 312–315. [CrossRef]
6. Li, C.; Lin, D.; Lu, J.; Hao, F. Cryptanalyzing an image encryption algorithm based on autoboiling and electrocardiography. *IEEE Multimed. 2018*, 25, 46–56. [CrossRef]
7. Yang, T.; Ma, J.; Wang, Q.; Miao, Y.; Wang, X.; Meng, Q. Image feature extraction in encrypted domain with privacy-preserving hahn moments. *IEEE Access 2018*, 6, 47521–47534. [CrossRef]
8. Li, C.; Lin, D.; Lu, J. Cryptanalyzing an Image-Scrambling Encryption Algorithm of Pixel Bits. *IEEE Multimed. 2017*, 24, 64–71. [CrossRef]
9. Ali, D.S.; Alwan, N.A.; Al-Saidi, N.M. Image encryption based on highly sensitive chaotic system. *AIP Conf. Proc. 2019*, 2183, 080007. [CrossRef]
10. Gopalakrishnan, T.; Ramakrishnan, S. Image Encryption Using Hyper-chaotic Map for Permutation and Diffusion by Multiple Hyper-chaotic Maps. *Wired. Pers. Commun. 2019*, 109, 437–454. [CrossRef]
11. Chen, H.; Liu, Z.; Tanougast, C.; Liu, F.; Blondel, W. A novel chaos based optical cryptosystem for multiple images using DNA-blend and gyrator transform. *Opt. Lasers Eng. 2021*, 138, 106448. [CrossRef]
12. Choi, U.S.; Cho, S.J.; Kim, J.G.; Kang, S.W.; Kim, H.D. Color image encryption based on programmable complemented maximum length cellular automata and generalized 3-D chaotic cat map. *Multimed. Tools Appl. 2020*, 79, 22825–22842. [CrossRef]
13. Matthews, R. On the derivation of a “chaotic” encryption algorithm. *Cryptologia 1989*, 13, 29–42. [CrossRef]
14. Liu, Q.; Li, P.-y.; Zhang, M.-c; Sui, Y.-x.; Yang, H.-j. A novel image encryption algorithm based on chaos maps with Markov properties. *Commun. Nonlinear Sci. Numer. Simul. 2015*, 20, 506–515. [CrossRef]
15. Goggin, M.E.; Sundaram, B.; Milonni, P.W. Quantum logistic map. *Phys. Rev. A 1990*, 41, 5705–5708. [CrossRef] [PubMed]
16. Zhou, N.; Hu, Y.; Gong, L.; Li, G. Quantum image encryption scheme with iterative generalized Arnold transforms and quantum image cycle shift operations. *Quantum Inf. Process. 2017*, 16, 164. [CrossRef]
17. Butt, K.K.; Li, G.; Masood, F.; Khan, S. A digital image confidentiality scheme based on pseudo-quantum chaos and Lucas sequence. *Entropy 2020*, 22, 1276. [CrossRef] [PubMed]
18. Lanzagorta, M.; Uhlmann, J. Quantum algorithmic methods for computational geometry. *Math. Struct. Comput. Sci. 2010*, 20, 1117–1125. [CrossRef]
19. Venegas-Andraca, S.E.; Ball, J.L. Processing images in entangled quantum systems. *Quantum Inf. Process. 2010*, 9, 1–11. [CrossRef]
20. Zhang, W.W.; Gao, F.; Liu, B.; Wen, Q.Y.; Chen, H. A watermark strategy for quantum images based on quantum fourier transform. *Quantum Inf. Process. 2013*, 12, 793–803. [CrossRef]
21. Le, P.Q.; Dong, F.; Hirota, K. A flexible representation of quantum images for polynomial preparation, image compression, and processing operations. *Quantum Inf. Process. 2011*, 10, 63–84. [CrossRef]
22. Li, P.C.; Cao, Z.Q. Quantum description method of color image and its application. *Kongzhi Yu Juece/Control Decis.* 2017, 32, 443–450. [CrossRef]
23. Yao, X.W.; Wang, H.; Liao, Z.; Chen, M.C.; Pan, J.; Li, J.; Zhang, K.; Lin, X.; Wang, Z.; Luo, Z.; et al. Quantum image processing and its application to edge detection: Theory and experiment. *Phys. Rev. X* 2017, 7, 031041. [CrossRef]
24. Eleuch, H.; Rotter, I. Clustering of exceptional points and dynamical phase transitions. *Phys. Rev. A* 2016, 93, 042116. [CrossRef]
25. Venegas-Andraca, S.E.; Bose, S. Storing, processing, and retrieving an image using quantum mechanics. *Quantum Inf. Comput.* 2003, 3, 105, 137. [CrossRef]
26. Zhang, Y.; Lu, K.; Gao, Y.; Wang, M. NEQR: A novel enhanced quantum representation of digital images. *Quantum Inf. Process.* 2013, 12, 2833–2860. [CrossRef]
27. Ma, S.Y.; Khalil, A.; Hajjdiab, H.; Eleuch, H. Quantum dilation and erosion. *Appl. Sci.* 2020, 10, 4040. [CrossRef]
28. Zidan, M.; Abdel-Aty, A.H.; El-shafei, M.; Feraig, M.; Al-Sbou, Y.; Eleuch, H.; Abdel-Aty, M. Quantum classification algorithm based on competitive learning neural network and entanglement measure. *Appl. Sci.* 2019, 9, 1277. [CrossRef]
29. Chai, X.; Wu, H.; Gan, Z.; Zhang, Y.; Chen, Y. Hiding cipher-images generated by 2-D compressive sensing with a multi-embedding strategy. *Signal Process.* 2020, 171, 107525. [CrossRef]
30. Candès, E.J.; Romberg, J.; Tao, T. Robust Uncertainty Principles: Exact Signal Reconstruction From Highly Incomplete Frequency Information. *IEEE Trans. Inf. Theory* 2006, 52, 489–509. [CrossRef]
31. Duarte, M.F.; Davenport, M.A.; Takbar, D.; Laska, J.N.; Sun, T.; Kelly, K.F.; Baraniuk, R.G. Single-pixel imaging via compressive sampling: Building simpler, smaller, and less-expensive digital cameras. *IEEE Signal Process. Mag.* 2008, 25, 83–91. [CrossRef]
32. Donoho, D.L. Compressed sensing. *IEEE Trans. Inf. Theory* 2006, 52, 1289–1306. [CrossRef]
33. Chai, X.; Wu, H.; Gan, Z.; Han, D.; Zhang, Y.; Chen, Y. An efficient approach for encrypting double color images into a visually meaningful cipher image using 2D compressive sensing. *Inf. Sci.* 2021, 556, 305–340. [CrossRef]
34. Xu, Q.; Sun, K.; He, S.; Zhu, C. An effective image encryption algorithm based on compressive sensing and 2D-SLIM. *Opt. Lasers Eng.* 2020, 134, 106178. [CrossRef]
35. Zhang, H.; Xiao, S.; Zhou, P. A matching pursuit algorithm for backtracking regularization based on energy sorting. *Symmetry* 2020, 12, 231. [CrossRef]
36. Xian, Y.; Wang, X. Fractal sorting matrix and its application on chaotic image encryption. *Inf. Sci.* 2021, 547, 1154–1169. [CrossRef]
37. Alvarez, G.; Li, S. Some basic cryptographic requirements for chaos-based cryptosystems. *Int. J. Bifurc. Chaos* 2006, 16, 2129–2151. [CrossRef]
38. Liang, H.R.; Tao, X.Y.; Zhou, N.R. Quantum image encryption based on generalized affine transform and logistic map. *Quantum Inf. Process.* 2016, 15, 2701–2724. [CrossRef]
39. Zhu, C. A novel image encryption scheme based on improved hyperchaotic sequences. *Opt. Commun.* 2012, 285, 29–37. [CrossRef]
40. Chen, J.; Zhang, Y.; Qi, L.; Fu, C.; Xu, L. Exploiting chaos-based compressed sensing and cryptographic algorithm for image encryption and compression. *Opt. Laser Technol.* 2018, 99, 238–248. [CrossRef]
41. Shannon, C.E. Communication Theory of Secrecy Systems. *Bell Syst. Tech. J.* 1949, 28, 656–715. [CrossRef]
42. Wang, X.; Wang, Y.; Wang, S.; Zhang, Y.; Wu, X. A novel pseudo-random coupled LP spatiotemporal chaos and its application in image encryption. *Chin. Phys. B* 2018, 27, 110502. [CrossRef]