doi:10.3772/j. issn. 1006-6748. 2022. 02. 003

A multispectral image compression and encryption algorithm based on tensor decomposition and chaos[①]

XU Dongdong(徐冬冬)[②], DU Limin
(Electronic Information Engineering College, Changchun University, Changchun 130022, P. R. China)

Abstract

A multispectral image compression and encryption algorithm that combines Karhunen-Loeve (KL) transform, tensor decomposition and chaos is proposed for solving the security problem of multi-spectral image compression and transmission. Firstly, in order to eliminate residual spatial redundancy and most of the spectral redundancy, the image is performed by KL transform. Secondly, to further eliminate spatial redundancy and reduce block effects in the compression process, two-dimensional discrete 9/7 wavelet transform is performed, and then Arnold transform and encryption processing on the transformed coefficients are performed. Subsequently, the tensor is decomposed to keep its intrinsic structure intact and eliminate residual space redundancy. Finally, differential pulse filters are used to encode the coefficients, and Tent mapping is used to implement confusion diffusion encryption on the code stream. The experimental results show that the method has high signal-to-noise ratio, fast calculation speed, and large key space, and it is sensitive to keys and plaintexts with a positive effect in spectrum assurance at the same time.

Key words: Karhunen-Loeve (KL) transform, tensor decomposition, differential pulse filter, Tent map

0 Introduction

Remote sensing is a comprehensive detection technology based on aerial photography and interpretation[1]. Due to its wide coverage, strong detection capabilities, and comprehensive data acquisition, it has been used for decades on reconnaissance, meteorological observation, and resource census. Multispectral imaging systems as an important part of satellites are developing towards high resolution and large field of view. It gets more detailed and accurate information to improve the recognition ability of ground and ocean targets with high resolution; and it also makes the coverage of the space camera larger and effectively improves the work efficiency. With the increase in resolution and field of view, the amount of data output by the space camera becomes larger and larger, which puts forward higher requirements on the compression and decompression algorithms of related images. In addition, it is necessary to encrypt multispectral images to ensure data security.

For two-dimensional images and multispectral images, many compression encryption algorithms have been proposed. Ref. [2] proposed a 3D chaotic encryption scheme for compressed image. First, the set partitioning in hierarchical trees (SPIHT) encoding algorithm is used to compress the image, and then the images are mapped to a three-dimensional bit matrix. Next Lorenz is used to generate a chaotic sequence, and a series of processing is performed on the generated three-dimensional bit matrix. Finally the compressed and encrypted image is got. The method has a positive encryption effect, but the amount of data is huge. Ref. [3] proposed a hyper-spectral image compression algorithm based on adaptive spectrum recombination. It has a small amount of encrypted data but lacks security. Ref. [4] proposed an algorithm of joint hyperspectral image compression and encryption based on optimal spectrum prediction of inter-band, SPIHT and chaos mapping. It has a better compression efficiency and encryption effect, but the compression efficiency and key space need further improvement. Ref. [5] proposed a joint image compression-encryption scheme using entropy coding and compressive sensing, which has a better compression and encryption performance.

The compression encryption algorithm mentioned above are aimed at two-dimensional images, with little

[①] Supported by the National Natural Science Foundation of China (No. 61801455).
[②] To whom correspondence should be addressed. E-mail: 1069292478@qq. com.
Received on Apr. 30, 2021

research on multispectral images. Different from ordinary two-dimensional images, multispectral images contain hundreds of continuous spectrum imaging information with spatial and spectral dimensions. The amount of data is huge, which makes the subsequent processing of multispectral images more complicated. Considering the characteristics of multispectral images and referring to the latest Karhunen-Loeve (KL) transform technology[6-7], this paper proposes a multispectral image compression encryption algorithm based on chaos and fast wavelet transform, which closely combines the compression process with the encryption process together. While improving the efficiency of image storage and transmission, the security of the image is guaranteed, and higher compression efficiency and better encryption effect are achieved.

1 KL transformation

KL transformation (KLT) can be used in principal component analysis (PCA). It is a linear and reversible transform with decorrelation performance. The calculation process of the original algorithm is as follows.

First, the multispectral image matrix containing N spectral bands is integrated into a two-dimensional matrix $\boldsymbol{Y}$ using the row stacking method, and the average value of the vector $\boldsymbol{Y}$ is calculated. The process of calculation is shown in Eq. (1).

$$m = E\{\boldsymbol{Y}\} \approx \frac{1}{N}\sum_{i=1}^{N} Y_i \tag{1}$$

where m is the mean of $\boldsymbol{Y}$, $E\{\}$ represents the mean of the vector.

Secondly, the covariance matrix $\boldsymbol{C}$ of the vector $\boldsymbol{Y}$ is calculated, and the calculation process is shown in Eq. (2).

$$\begin{aligned}\boldsymbol{C} &= E\{(\boldsymbol{Y} - m)(\boldsymbol{Y} - m)^{\mathrm{T}}\}\\ &= \frac{1}{N}\sum_{i=1}^{N} Y_i Y_i^{\mathrm{T}} - mm^{\mathrm{T}}\end{aligned} \tag{2}$$

Finally, the eigenvalues and eigenvectors of the covariance matrix $\boldsymbol{C}$ are calculated. The KL transformation expression can be obtained as shown in Eq. (3).

$$\boldsymbol{Y} = \boldsymbol{A}^{\mathrm{T}}(\boldsymbol{Y} - m) \tag{3}$$

where $\boldsymbol{A}$ is the eigenvector of $\boldsymbol{C}$. The algorithm requires a huge amount of calculations, among which the calculation of the $M \times N$ size spectrum requires $(M \times N - 1)$ times additions and one division, and the calculation of normalization of the data requires $(M \times N)$ times subtractions. Other operations require higher computational complexity. Therefore, the following improvements will be made to solve this problem.

First of all, when the covariance is calculated, a subset of the spectral vectors is randomly selected instead of using all the spectral vectors, and the size of the subset is appropriately selected to minimize the computational complexity while ensuring image quality. The effect of sampled size on compression performance is shown in Fig. 1. From Fig. 1 it can be seen when the sample size is 1/1000 of the traditional method, the compression performance starts to decline, and the calculation complexity is also low. The value is selected as the sampling ratio to estimate the covariance considering the compression performance and calculation time.

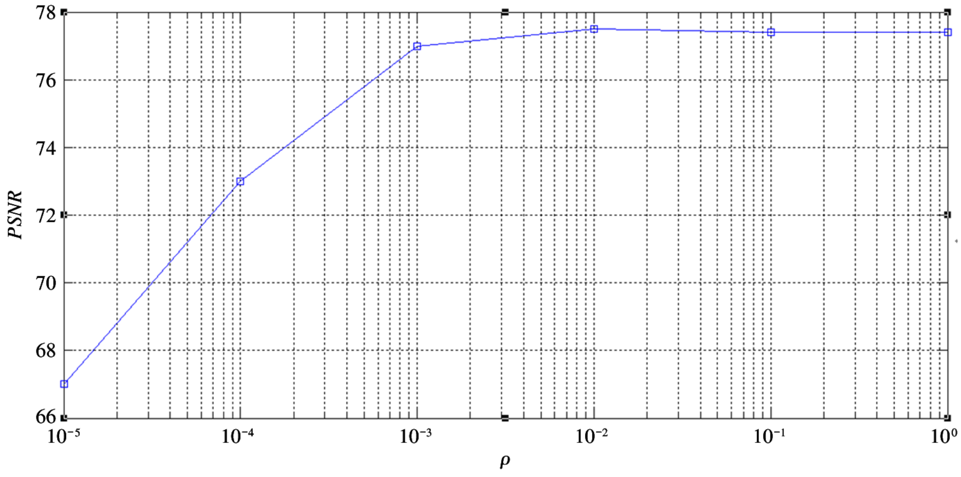

Fig. 1 The effect of sampling ratio on compression performance

Table 1 shows the comparison of the KLT execution time of the 512 × 512 × 4 image using the downsampling method. It can be seen that under the premise of ensuring certain compression performance, the calculation time of this method is greatly reduced.

Table 1 Calculation time comparison

Methods	Computing time/s	Total time/s
Sample	0.65	3.28
Without Sample	1.73	4.56
Enhance	62.43%	28.07%

The Jacobi algorithm is often used to find the eigenvalues and eigenvectors of the symmetric matrix. However, the algorithm requires not only the main element, but also the row and column rotation transformation to obtain the eigenvalues at the same time, which makes the whole calculation process very complex and difficult to implement in parallel.

A series of transformations will be used to transform the matrix $\boldsymbol{M}$ into a square matrix $\boldsymbol{T}$ with each column pairwise orthogonal to calculate the eigenvalues of the symmetric matrix $\boldsymbol{M}$, namely $\boldsymbol{MV}_1 \cdots \boldsymbol{V}_k = \boldsymbol{T}$, so $\boldsymbol{T}^{\mathrm{T}}\boldsymbol{T} = \boldsymbol{V}_k^{\mathrm{T}} \cdots \boldsymbol{V}_1^{\mathrm{T}}\boldsymbol{M}^{\mathrm{T}}\boldsymbol{MV}_1 \cdots \boldsymbol{V}_k$. According to the relationship between $\boldsymbol{T}$ and $\boldsymbol{T}^{\mathrm{T}}\boldsymbol{T}$, it is known that the spectral norm of each column of the square matrix $\boldsymbol{T}$ is the absolute value of the eigenvalue of the symmetric matrix, and its sign can be determined by the relationship between the eigenvalue and the eigenvector from the equation $\boldsymbol{Mb}_i = \lambda_i \boldsymbol{b}_i$. λ_i is positive if the signs of $\boldsymbol{Mb}$ and $\boldsymbol{b}_i$ are the same. Otherwise it is negative.

Finally, the eigenvector matrix is calculated using a lifting scheme instead of matrix multiplication (mean × eigenvector) in classic KLT. The eigenvector matrix is decomposed into $\boldsymbol{A}^{\mathrm{T}} = \boldsymbol{PLUS}$, where $\boldsymbol{P}$ is the permutation matrix, $\boldsymbol{L}$ is the unit lower triangular matrix, $\boldsymbol{U}$ is the unit upper triangular matrix, and $\boldsymbol{S}$ is the diagonal matrix. The output elements are rounded after each stage to maintain no floating point output. The improved KL expression is shown in Eq. (4).

$$Y = round(round(round(\boldsymbol{X} \times \boldsymbol{S}) \times \boldsymbol{U}) \times \boldsymbol{L}) \times \boldsymbol{P} \tag{4}$$

In the formula, *round* means rounding. Multiplying by $\boldsymbol{P}$ is the element swap, where the multiplication is only performed by 1 and 0. The permutation is not computationally intensive, because it only needs to loop through the vector to exchange certain elements. The $\boldsymbol{S}$ matrix is a sparse lower triangular matrix, and therefore fewer element multiplication operations can be required by applying the zero check technique to the multiplication operation.

The transformed effect diagram is shown in Fig. 2. KLT is performed on the upper three spectral bands (a, b, c) of a multi-spectral image (including 4 spectral bands) respectively to obtain three transformed images (d, e, f), which eliminate most of the inter-spectral redundancy, and the energy is mainly concentrated in the first two spectral bands.

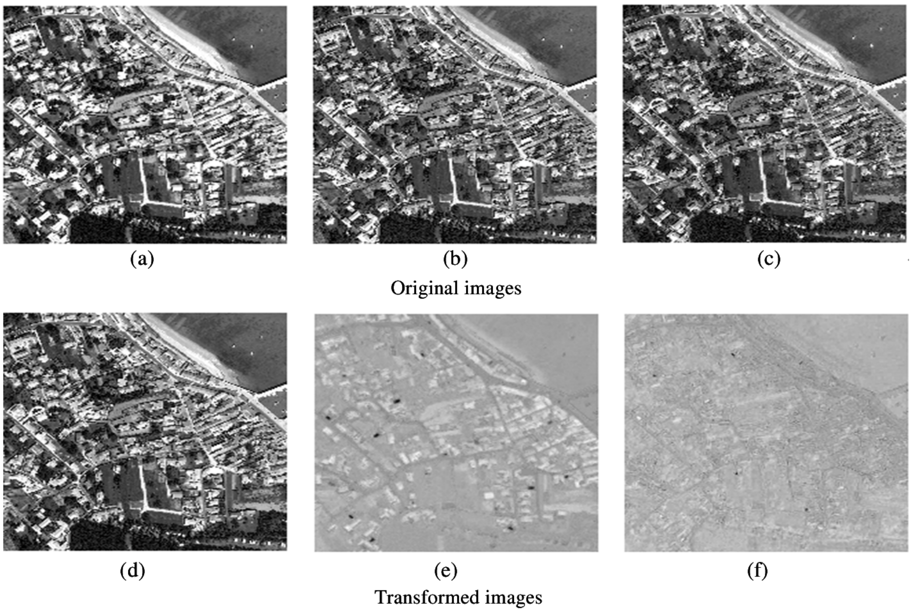

Fig. 2 KL transformation effect diagram of each spectrum

2 Wavelet encryption algorithm

The encryption algorithm is composed of three parts: subkey generation, wavelet coefficient scrambling and data stream encryption.

2.1 Subkey generation

The chaotic system is improved Logistic mapping and Tent mapping, the improved mapping equation is

$$x_{n+1} = [\mu_0 + (4 - \mu_0) \times \cos((10^{-6} + x_n) \times \pi/2)] \times x_n \times (1 - x_n/n) \quad (5)$$

$$x_{n+1} = \begin{cases} \dfrac{x_n}{p} & 0 < x_n < p \\ \dfrac{1 - x_n}{1 - p} & \text{otherwise} \end{cases} \quad (6)$$

where $p \in (0,1)$, $x_n \in [0,1]$, μ_0 is 3.569945673, and $1/n \in (0,1)$ is the amplification factor. When $\mu \in (3.569945673, 4]$, the sequence generated by Logistic mapping is in a chaotic state. When its value is 4, the system is in the best chaotic state, but the encryption effect is poor at this time. In order to solve the problem of poor encryption effect when value is 4, an infinite approach expression of 4 is adopted to replace 4 to achieve the expected chaotic characteristics and safety.

The output hash values are divided into 5 groups, denoted as f_1, f_2, f_3, f_4, and f_5. The sub-key is generated by Eq. (7) as the initial value of Logistic mapping.

$$\boldsymbol{x}_0^m = \mathrm{mod}\left(\sum_{i=1}^{5} f_i/2^{32},\ 1\right) \quad (7)$$

where, mod stands for modular operation. Given the initial keys x_0, x_1, and x_2, the subkeys are generated by perturbing the initial keys by Eq. (8), and they are respectively used as the initial values, control parameters and initial ciphertext blocks of the Tent mapping.

$$\begin{cases} \boldsymbol{x}'_0 = \mathrm{mod}(x_0 + (f_1 \oplus f_2)/2^{32}, 1) \\ \boldsymbol{x}'_1 = \mathrm{mod}(x_1 + (f_3 \oplus f_4)/2^{32}, 1) \\ \boldsymbol{x}'_2 = x_2 \oplus f_5 \end{cases} \quad (8)$$

where, $\oplus$ is the XOR operation.

2.2 Scrambling of wavelet coefficients

After the general pixel is constructed, it is necessary to perform wavelet transformation on the pixel to reduce the blocking effect in the compression process and protect the real information of the image. Wavelet transform is a new transform method proposed on the basis of Fourier transform. While continuing the localized advantage of short-time Fourier transform, it overcomes a series of shortcomings of Fourier transform. The two-dimensional discrete wavelet transform can be obtained by Eq. (9).

$$f(m,n) = \langle f(m,n), \tilde{\psi}_{j,\, i_1,\, i_2}(m,n) \rangle \psi_{j,\, i_1,\, i_2}(m,n) \quad (9)$$

After performing a wavelet transform on the image, four subbands, namely low frequency subband (LL), horizontal high frequency subband (LH), vertical high frequency subband (HL), and diagonal high frequency subband (HH) will be generated. The secondary transformation is to repeat a similar division on the basis of the LL. Better research results will be achieved by processing differently for different subband coefficients.

In this paper, Arnold transform is used to scramble the coefficients after wavelet transform. According to the Arnold transform periodic table, the appropriate number of scrambling times are selected to transform and scramble the target pixel to obtain a transform scrambling map.

2.3 Tensor decompositon

The multispectral image is a kind of third-order tensor, its tensor representation form is $\boldsymbol{Y} \in \mathrm{R}_+^{I_1 \times I_2 \times I_3}$, which can be decomposed into three non-negative mode matrices, namely $\boldsymbol{A}^{(n)} = [a_1^{(n)}, a_2^{(n)}, \cdots, a_{R_n}^{(n)}] \in \mathrm{R}_+^{I_n \times R_n}$ $(n = 1,2,3)$, and a low-dimensional non-negative core tensor, namely $\boldsymbol{G} \in \mathrm{R}^{R_1 \times R_2 \times R_3}$. The decomposed approximate tensor can be used to replace the original tensor to achieve the purpose of compression.

Most of the non-negative tensor Tucker decomposition is to minimize the cost function of the following formula.

$$\boldsymbol{D} = \| \boldsymbol{Y} - \boldsymbol{G} \times \{\boldsymbol{A}\} \|_F^2 \quad (10)$$

In order to optimize the above formula, many methods have been proposed mathematically. Among them, the dGN algorithm is derived from the Newton method. However, due to the relatively large amount of calculation and difficulty in implementation, it has not been used in multi-spectral image processing. The algorithm has low complexity and fast convergence speed, making it more suitable for multi-spectral image compression through appropriate improvements.

Through the following formula, the dGN iterate can be got.

$$\begin{aligned} &\boldsymbol{\beta} \leftarrow \boldsymbol{\beta} + (\boldsymbol{H} + \mu \boldsymbol{I})^{-1} \boldsymbol{g} \\ &\boldsymbol{H} = \boldsymbol{J}^{\mathrm{T}} \boldsymbol{J},\ \boldsymbol{g} = \boldsymbol{J}^{\mathrm{T}} (y - \hat{y}) \end{aligned} \quad (11)$$

Among them, $\boldsymbol{\beta} = [\mathrm{vec}(\boldsymbol{A}^{(1)})^{\mathrm{T}}, \mathrm{vec}(\boldsymbol{A}^{(2)})^{\mathrm{T}}, \mathrm{vec}(\boldsymbol{A}^{(3)})^{\mathrm{T}}]$, $\boldsymbol{I}$ is the identity matrix, $\boldsymbol{H}$ is the Hessian matrix, $\boldsymbol{\mu}$ is the damping parameter, $\boldsymbol{g}$ is the gradient, $\hat{y} = \mathrm{vec}(\hat{Y})$, $y = \mathrm{vec}(Y)$, and $\boldsymbol{J}$ is the Jacobian matrix.

The huge amount of calculation required for the iterative process mainly comes from the calculation of $\boldsymbol{H}$ and $\boldsymbol{g}$. In order to speed up the calculation, this paper improves the calculation of the Hessian matrix $\boldsymbol{H}$ and the gradient $\boldsymbol{g}$ respectively. Suppose the symbols of Kronecker product, Khatri-Rao product, and Hadamard product are $\otimes$, $\odot$ and Θ, respectively, Eq. (12) is defined as

$$\underset{n=1}{\overset{N}{\Theta}} A^{(n)} = A^{(N)} \Theta \cdots \Theta A^{(n)} \Theta \cdots \Theta A^{(1)},\ I_n = I,\ \forall n,$$

$$\underset{k\neq n}{\Theta} A^{(k)} = A^{(N)} \Theta \cdots \Theta A^{(n+1)} \Theta A^{(n-1)} \Theta \cdots \Theta A^{(1)},$$
$$I_n = I, \ \forall n,$$
$$\underset{k\neq n}{\odot} A^{(k)} = A^{(N)} \odot \cdots \odot A^{(n+1)} \odot A^{(n-1)} \odot \cdots A^{(1)} \tag{12}$$

The biggest problem in optimizing the dGN algorithm is how to reduce the computational complexity of the Hessian matrix $\boldsymbol{H}$ and its inverse matrix. To solve this problem, the following improvements is put forward.

First of all, the matrix $\boldsymbol{\Gamma}(\boldsymbol{n}, \boldsymbol{m})$ of size $R \times R$ and the block matrix $\boldsymbol{K}$ of size $NR^2 \times NR^2$ composed of matrix $\boldsymbol{K}(\boldsymbol{m}, \boldsymbol{n})$ are as follows:

$$\Gamma^{(n,m)} = [\Gamma^{(n,m)}]^{\mathrm{T}} = [\Gamma^{(m,n)}]^{\mathrm{T}} = \underset{k\neq n,\ m}{\Theta} C^{(k)} \tag{13}$$

$$\mathrm{K}^{(n,m)} = (1 - \delta_{n,m}) P_R \operatorname{diag}(\operatorname{vec}(\Gamma^{n,m})) \in \mathrm{R}^{R^2 \times R^2} \tag{14}$$

Among them, $C^{(n)} = A^{(n)\mathrm{T}} A^{(n)} \in \mathrm{R}^{\mathrm{R}\times\mathrm{R}}$, $\delta_{n,m}$ is Kronecker δ, $\boldsymbol{P}_{\boldsymbol{I},\boldsymbol{J}}$ is the permutation matrix of matrix $\boldsymbol{X}$ of size $\boldsymbol{I} \times \boldsymbol{J}$, and $P_{I,J}\operatorname{vec}(X^{\mathrm{T}}) = \operatorname{vec}(X)$, $P_{\mathrm{R}} \equiv P_{\mathrm{R,R}}$, $\Gamma^{(n)} \equiv \Gamma^{(n,n)}$, $n = 1, 2, \cdots, N$, $m = 1, 2, \cdots, N$. When $NR << T$, the fast dGN algorithm can be expressed as Eq. (15).

$$A^{(n)} \leftarrow A_{\mu}^{(n)} + A^{(n)}[I_R - (F_n + \Gamma^{(n)})\Gamma_{\tilde{\mu}}^{(n)}],$$
$$n = 1, 2, \cdots, N \tag{15}$$

Among them, $\boldsymbol{A}_{\mu}^{(n)}$ is a modification of the ALS update rule with suppression factor $\mu (\mu > 0)$, the matrix $\boldsymbol{F}_n$ of size $R \times R$ is the first few pieces of matrix $\boldsymbol{\Phi}$, $\operatorname{vex}(\boldsymbol{\Phi}) = B_{\mu}\omega_{\mu}$, the approximate Hessian matrix $\boldsymbol{H}$ can be obtained, namely:

$$\boldsymbol{H} = G + ZKZ^{\mathrm{T}} \tag{16}$$

$G = \operatorname{blkdiag}(\Gamma^{(n)} \otimes I_{In})_{n=1}^{N} \in \mathrm{R}^{RT\times RT}$, $Z = \operatorname{blkdiag}(I_R \otimes A^{(n)})_{n=1}^{N} \in \mathrm{R}^{RT\times NR^2}$.

The inverse matrix of the approximate damped Hessian matrix $\boldsymbol{H}_{\mu}(\boldsymbol{H}_{\mu} = \boldsymbol{H} + \mu I_{RT})$ can be obtained by Eq. (17).

$$H_{\mu}^{-1} = \tilde{G}_{\mu} - L_{\mu} B_{\mu} L_{\mu}^{\mathrm{T}} \tag{17}$$

$\tilde{G}_{\mu} = \operatorname{blkdiag}(\tilde{\Gamma}_{\mu}^{(n)} \otimes I_{In})_{n=1}^{N}$, $L_{\mu} = \operatorname{blkdiag}(\tilde{\Gamma}_{\mu}^{(n)} \otimes A^{(n)})_{n=1}^{N}$. Via calculation, Eq. (18) and Eq. (19) can be got.

$$\beta \leftarrow \beta + \tilde{G}_{\mu} g - \boldsymbol{L}_{\mu} B_{\mu} \boldsymbol{L}_{\mu}^{\mathrm{T}} g \tag{18}$$

$$\begin{aligned} \omega_{\mu} &= \boldsymbol{L}_{\mu}^{\mathrm{T}} \boldsymbol{J}^{\mathrm{T}} \operatorname{vec}(\varepsilon) \\ &= \{[\operatorname{vec}(A^{(n)\mathrm{T}}(A_{\mu}^{(n)} - A^{(n)}\Gamma^{(n)}\ \tilde{\Gamma}_{\mu}^{(n)}))^{\mathrm{T}}]_{n=1}^{N}\}^{\mathrm{T}} \\ &= \operatorname{vec}\{[A^{(n)\mathrm{T}} A_{\mu}^{(n)} - \Gamma\, \tilde{\Gamma}_{\mu}^{(n)}]_{n=1}^{N}\} \end{aligned} \tag{19}$$

Eq. (19) still requires a large amount of calculation. The main calculation amount is concentrated on the calculation of the Jacobian matrix $\boldsymbol{J}$ and the creation of the matrix $\boldsymbol{L}_{\mu}$. This article will further simplify the three main parts of the iterative Eq. (19), the simplification process is shown in Eq. (20) and Eq. (21) respectively.

$$\begin{aligned} \tilde{G}_{\mu} g &= (\tilde{G}_{\mu} \operatorname{vec}(\varepsilon))^{\mathrm{T}} \\ &= [\operatorname{vec}(\varepsilon)^{\mathrm{T}} Q_n(((\underset{k\neq n}{\odot} A^{(k)})\tilde{\Gamma}_{\mu}^{(n)}) \otimes I_{In})]_{n=1}^{N} \\ &= [\operatorname{vec}(E_{(n)}(\underset{k\neq n}{\odot} A^{(k)})\tilde{\Gamma}_{\mu}^{(n)})^{\mathrm{T}})]_{n=1}^{N} \\ &= [\operatorname{vec}(Y_{(n)}(\underset{k\neq n}{\odot} A^{(k)})\tilde{\Gamma}_{\mu}^{(n)} \\ &\quad - A^{(n)}(\underset{k\neq n}{\odot} A^{(k)})^{\mathrm{T}}(\underset{k\neq n}{\odot} A^{(k)})\ \tilde{\Gamma}_{\mu}^{(n)})^{\mathrm{T}}]_{n=1}^{N} \\ &= [\operatorname{vec}(A_{\mu}^{(n)} - A^{(n)}\Gamma^{(n)}\tilde{\Gamma}_{\mu}^{(n)})^{\mathrm{T}}]_{n=1}^{N} \end{aligned} \tag{20}$$

$$L_{\mu} B_{\mu} \omega_{\mu} = \begin{bmatrix} \operatorname{vec}(A^{(1)} F_1 \tilde{\Gamma}_{\mu}^{(1)}) \\ \vdots \\ \operatorname{vec}(A^{(n)} F_n \tilde{\Gamma}_{\mu}^{(n)}) \\ \vdots \\ \operatorname{vec}(A^{(N)} F_N \tilde{\Gamma}_{\mu}^{(N)}) \end{bmatrix} \tag{21}$$

Among them $Q_n = I_{I_{N+1:N}} \otimes P_{I_{1:n-1},\, I_n}$. From the above formulas, the core tensor $\boldsymbol{G}$ can be obtained and expressed as follows.

$$\begin{aligned} &\boldsymbol{G} = \operatorname{blkdiag}(\Gamma^{(n)} \otimes I_{I_n})_{n=1}^{N} \in \mathrm{R}^{RT\times RT}, \\ &\Gamma^{(n)} = \Theta_{k\neq n} C^{(k)}, C^{(n)} = A^{(n)\mathrm{T}} A^{(n)}, \\ &a_{j_n}^{(n)} \leftarrow a_{j_n}^{(n)} / \| a_{j_n}^{(n)} \|_2 \end{aligned} \tag{22}$$

The norm is a reinforced concept of distance, and l_P norm is not a norm, but a set of norms. l_1 norm can lead to a sparse solution. And it can produce a more sparse model than l_2 norm, and l_1 norm can be used to feature selection.

The above formula uses the l_2 norm, which is not suitable for image compression, so we change it to the l_1 norm , and the new iteration rule is as in Eq. (22).

Fig. 3 is a performance comparison chart between the algorithm and other three algorithms. It can be seen from Fig. 3 that compared with the hierarchical alternating least squares (HALS) and the least squares (LS), the algorithm proposed in this paper has the advantage of fast convergence; compared with the original

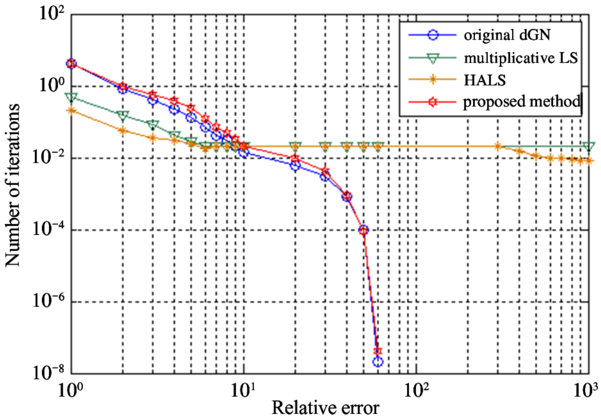

Fig. 3 Convergence speed comparison of decomposed three-dimensional tensor

dGN algorithm, although the convergence speed is slightly slower, the calculation amount of this system is much smaller than the latter, and the improved algorithm is more suitable for multi-spectral image compression.

2.4 Data stream encryption

For further compression, the original signal needs to be converted into a new integer stream. Then the integer stream is converted into a binary coded stream, and Huffman coding is adopted. However, if these integers are directly converted into a binary bit stream through Huffman coding, too much storage space will be required. Therefore, the differential pulse code modulation (DPCM) filter is used to convert the original signal into a new integer array.

Firstly, the non-zero coefficients are uniformly quantized with a quantizer. The original signal is converted into a new integer array to save storage space and further compress the non-zero coefficients. Finally, the integer stream is converted into a binary coded stream to achieve efficient image coding.

Secondly, Tent mapping is used to perform confusion diffusion encryption on the compressed code stream. Assuming that the compressed data stream of the plaintext is R, the initial value of the Tent mapping and the control parameters are $x_0^{'}$, $x_1^{'}$, and the initial ciphertext block is $x_2^{'}$. The data stream encryption process is as follows.

R is divided into n sub-blocks, and the length of each sub-block is 32.

Then, $x_0^{'}$ and $x_1^{'}$ are used as the initial values and control parameters, and they are iterated n times. The iteration process is shown in Eq. (23).

$$y_q = \mathrm{mod}(round(x_q^{'} \times 10^{16}), 2^{32}) \tag{23}$$

Finally, the encrypted sequence is obtained by Eq. (24) with $x_2^{'}$ as the initial ciphertext block.

$$\begin{aligned} x_i = SR[&(\mathrm{mod}((r_i \oplus m_q + x_{i-1}), 2^{32}), \\ &L_5(m_{(1+\mathrm{mod}(x_{i-1}^{'}, n))})] \end{aligned} \tag{24}$$

where $SR[e, f]$ means shifting e to the right by f bits, and L_5 means taking the lower five bits of the sequence.

3 Experimental results

3.1 Encryption performance analysis

The algorithm has 5 initial keys in total, and the precision space generated by each key is close to 10^{16}. During the encryption process, the key changes continuously as the plaintext changes. Therefore, it has a larger key space and known plaintext attacks and brute force attacks can be effectively resisted.

In order to verify the sensitivity of the algorithm to the key, the original image is encrypted with a key of a slight difference, and the rate of change of the code stream is compared. It can be obtained through experiments that the output bit stream change rate remains between 47.62% -47.72%, which has good key sensitivity.

With the key unchanged and the value of a certain pixel in the image being randomly changed, the sensitivity of this algorithm to plaintext is evaluated through the encrypted bit stream change. 100 simulation experiments on QuickBird with different characteristics and specific compression ratios are performed. The experiment shows that the ciphertext bit stream change rate remains at 47.45% -47.53%. It can be seen that this algorithm is very sensitive to plaintext images and the differential attacks can be effectively resisted.

The number of pixels change rate (NPCR) and the unified average changing intensity (UACI) are important indicators for effective analysis of resistance to differential attacks. Among them, NPCR represents the ratio of different gray values of different ciphertext images at the same position, and UACI represents the average change density between different ciphertext images. Table 2 lists the NPCR and UACI test results of this algorithm and several other algorithms.

Table 2 Test results of NPCR and UACI

Algorithm	Minor plaintext changes (pixel value)	NPCR/%	UACI/%
Ref. [8]	1	99.60	33.50
Ref. [9]	1	99.62	31.59
This work	1	99.79	34.21

The higher the values of NPCR and UACI, the better the encryption effect. It can be seen from Table 2 that the NPCR and UACI of this algorithm are closer to ideal values. Therefore, compared with similar algorithms, this algorithm can better resist differential attacks.

3.2 Compression performance analysis

A multispectral image with three spectral bands is selected as the test image to verify the feasibility of the algorithm. The whole algorithm is simulated on the computer. The pixel depth is 8 bit/pixel(b/p), and the compression ratio is 16∶1. The experimental results are shown in Fig. 4.

It can be seen from Fig. 4 that when the bit rate is relatively high, the PSNR is particularly high, and the reconstructed image is not much different from the original image. After DWT and KLT has been performed on images, most of the pixel values are located in 1 bit, the spectral redundancy is eliminated.

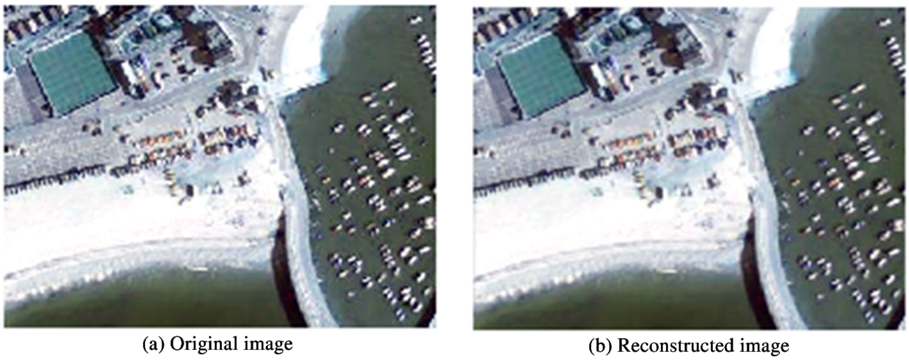
(a) Original image
(b) Reconstructed image

Fig. 4 Comparison of original image and compressed reconstructed image

To verify the quality of the compressed image further, 4 groups of QuickBird multispectral images with different characteristics were selected for testing. And the recently proposed multispectral image compression algorithm is used for comparison. The comparison results are shown in Table 3.

Table 3 Compression system test results

Methods	4 : 1	8 : 1	16 : 1	32 : 1
IKLT[10]	50.21	48.35	45.23	41.15
DWT-Tucker[11]	53.11	50.22	46.85	41.78
ATS[12]	50.17	45.87	43.36	41.23
JPEG2000[13]	49.45	48.11	45.07	40.31
This work	52.83	48.74	46.12	41.53

It can be seen from Table 3, within the compression ratio range of 4 : 1 – 32 : 1, the proposed algorithm achieves a high peak signal-to-noise ratio, which is superior to many existing compression algorithms. Also, this algorithm is feasible and particularly suitable for the occasions of higher compression ratio.

Table 4 shows the data processing speed of this algorithm compared with some existing algorithms. It can be seen from Table 4 that the data throughput rate of the compression algorithm proposed in this paper is lower than the existing compression algorithm, and the processing speed is equivalent to JPEG 2000, but the compression ratio is much higher than JPEG 2000 and other compression algorithms. Compared with other compression algorithms, one reason for the slightly lower throughput rate of this algorithm is that the encryption algorithm is added to the algorithm, which increases the complexity of the algorithm. As the hardware improves, the processing speed of the algorithm will continue to increase, and the advantage of the algorithm, namely high peak signal-to-noise ratio, will become more obvious at the same time.

Table 4 Comparison results of data processing speed

Methods	Throughput rate (M pixels/s)	Frequency /MHz
JPEG2000[13]	5.52	88
KLT[14]	9.77	88
DWT-Tucker[11]	11.26	88
3DSPIHT[15]	16.04	88
This work	5.06	88

The spectral distortion between the original pixel and the reconstructed pixel is often used to measure the fidelity, and the spectral angular distance (SAD) is one of the most commonly used criteria for evaluating multispectral images. Smaller SAD means that compression has lost less information, and the reconstructed multispectral image is more reliable for subsequent applications. For convenience, the mean SAD is used as standard deviation for all pixels to reveal the average spectral distortion of the multispectral image. The results of the mean SAD for different methods are shown in Fig. 5. It can be seen from the figure that the method achieves a smaller mean SAD at the maximum bit rates, which indicates that the average spectral distortion of this method is less than other methods. Therefore, the proposed method has good spectral fidelity.

4 Conclusion

This paper proposes a multi-spectral image compression encryption algorithm that combines chaos, wavelet transform and KL transform to solve the security problem of multispectral image compression and transmission. The experimental results show that the method has high signal - to - noise ratio , short operation

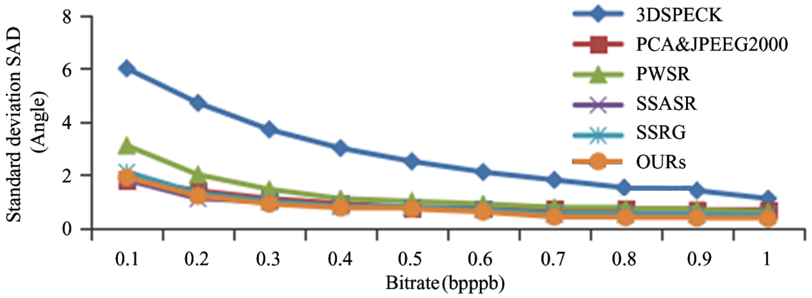

Fig. 5 Mean SADs

time, large key space, and it is sensitive to keys and plaintexts with a fine effect in spectrum assurance at the same time.

References

[1] MA Y, ZHANG J, ZHANG J. Analysis of unmanned aerial vehicle (UAV) hyperspectral remote sensing monitoring key technology in coastal wetland[C] // Selected Proceedings of the Chinese Society for Optical Engineering Conferences, Suzhou, China, 2015:97962S-1-97962S-9

[2] LI J, FENG Y, YANG X Q. 3D chaotic encryption scheme for compressed image[J]. *ACTA OPTICA SINCA*, 2010, 30(2): 399-404

[3] ZHOU Z. Hyperspectral Image Compression Algorithm Based on Adaptive Spectrum Reorganization[D]. Wuhan: Department of Electronic and Information Engineering, Huazhong University of Science and Technology, 2007 (In Chinese)

[4] CHEN T, ZHANG S W. Algorithm of joint hyperspectrum image compression and encryption based on optimal spectrum prediction of inter-band, SPIHT and chaos mapping [J]. *Journal of Jiangnan University (Natural Science Edition)*, 2014, 13(3): 258-263 (In Chinese)

[5] SONG Y J, ZHU Z L, ZHANG W, et al. Joint image compression-encryption scheme using entropy coding and compressive sensing[J]. *Nonlinear Dynamics*, 2019, 95 (3): 2235-2261

[6] LARA M, MULGREW B. Performance of the distributed KLT and its approximate implementation[C] // IEEE European Signal Processing Conference, Bucharest, Romania, 2012: 724-728

[7] EGHO C, VLADIMIROVA T. Adaptive hyperspectral image compression using the KLT and integer KLT algorithms [C] // 2014 IEEE NASA/ESA Conference on Adaptive Hardware and Systems (AHS), Leicester, UK, 2014: 112-119

[8] GUO Y, SHAO L P, YANG L. Bit-level image encryption algorithm based on Josephus and Henon Chaotic map [J]. *Application Research of Computers*, 2015, 32 (4): 1131-1137

[9] ZHAO X L, LI B, JIA P, et al. Image encryption algorithm based on improved Joseph traversal and piecewise logistic mapping[J]. *Chinese Journal of Electron Devices*, 2021, 44(1):125-130

[10] XU D D, FU T J, ZHANG Y, et al. A compression algorithm of multispectral images based on improved Karhunen-Loeve transform[J]. *Journal of Optoelectronics Laser*, 2015, 26(6): 1200-1205

[11] LI J, JIN L X, LI G N. Hyper-spectral remote sensing image compression based on nonnegative tensor factorizations in discrete wavelet domain[J]. *Journal of Electronics and Information Technology*, 2013, 35(2): 489-493

[12] BAYAZIT U. Adaptive spectral transform for wavelet-based color image compression [J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2011, 21 (7): 983-992

[13] GONZALEZ-CONEJERO J, BARTRINA-RAPESTA J, SERRA-SAGRISTA J. JPEG2000 encoding of remote sensing multispectral images with no-data regions [J]. *IEEE Geoscience and Remote Sensing Letters*, 2010, 7 (2): 251-255

[14] BLANE I, BARTRINA-RAPESTA J. Cost and scalability improvements to the Karhunen-Loeve transform for remote sensing image coding [J]. *IEEE Tranctions on Geoscience and Remote Sensing*, 2010, 48(7): 2854-2863

[15] KHELIFI F, BOURIDANE A, KURUGOLLU F. Joined spectral trees for scalable SPIHT-based multispectral image compression[J]. *IEEE Transactions on Multimedia*, 2008, 10(3):316-329

XU Dongdong, born in 1987. He received the B. E. degree in electronic information engineering from Northwestern Polytechnical University in 2010, and the M. S. and Ph. D degrees in optical engineering from Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences in 2013 and 2016. He is currently a lecturer at the Changchun University. His current research interests include image compression and electro-optical imaging.