

A Compression Algorithm for Hyperspectral Image

Lin Xiao-jun*, Xu Dong-Dong and Jin Long-xu

*Changchun institute of optics, fine mechanics and physics,
Chinese Academy of Sciences, China
1098453552@qq.com, 1069292478@qq.com, jinlx@ciomp.ac.cn*

Abstract

An effective compression algorithm of hyperspectral image based on discrete wavelet transform (DWT) and improved Karhunen-Loeve transform(KLT) is proposed in this paper. It could achieve better performance with effective combination of the two methods which convert the energy of images to a small number of coefficients. In order to reduce the most redundancy of space, it deals with each spectrum with fast 5/3 2-D DWT firstly. Then the spectral 1-D KLT is applied on the 2-D DWT coefficients to reduce spectral redundancy and the other spatial redundancy. Finally, entropy coding is executed to obtain the compressed code stream. The experimental results show that the average peak signal to noise ratio (PSNR) of the proposed compression algorithm is 3.01 bit per pixel (bpp), which improves greatly through comparing with the other approaches, and reduces the operating time and improves the performance of hyperspectral image compression algorithm. It proposes the hardware implementation strategy at the same time, verifies the correctness and feasibility of our method. The work in this paper has a good reference value and significance for the similar case.

Keywords: *hyperspectral; image compression; Karhunen-Loeve transform; hardware implementation strategy*

1. Introduction

With the developments in the fields of aerospace, geological, medical, *etc.* the traditional two-dimensional images can not satisfy the needs of the users, so arise the hyperspectral technologies which can carry more information at the historic moment. However, the amount of data transmission increasingly more and more fast when the spectrum compositions of hyperspectral image are more and more complex. Existing transfer and storage systems cannot adapt to a large amount data of hyperspectral images, in order to transmit and process these data faster, and reduce the storage space at the same time, the hyperspectral image must be compressed^[1-6].

At present, the basic method for compressing hyperspectral image is using discrete cosine transform (DCT) to remove spatial correlation or using KLT to remove spectral correlation^[7, 8]. The correlation among DCT coefficients diminishes, and most of the energy concentrates in just a few coefficients after two-dimensional discrete cosine transform (2D-DCT), then it could achieve the purpose of compression through quantization for correspond coefficients and entropy coding. DCT is the closest method with KLT in terms of the removing pixel correlation, and it achieves low implementation complexity by using its fast algorithm. However, the compression algorithm based on DCT transform is divided into several non-overlapping small pieces, then DCT transformation is applied with each sub-block, which makes the reconstruction images appear obvious “block effect” when in low bit rate. It will increase the complexity of the

Received (December 12, 2016), Review Result (November 7, 2017), Accepted (November 9, 2017)

algorithm in order to remove the “block-effect”; So DCT is not used directly on the image in consequence.

KLT has many good qualities, such as strongly remove correlation, highly concentrate after transform, and it could carry out the optimal sparse representation, *etc.* It rarely used in the practical engineering due to its large amount of calculation without the fast algorithm, so it is necessary to do the further improvement for KLT.

Combining the spectral characteristics of hyperspectral and referring to the latest technology of the KLT [9-11], it puts forward a kind of efficient compression algorithm based on DWT and improvement KLT. It transforms the energy into a series of coefficients in spatial direction using 2D-DWT firstly, and then transforms the coefficients with KLT which bases on different basis. The algorithm overcomes the limitation of compression performance with the single KLT basis, effectively protects the hyperspectral information, and implements the high compression performance and compression efficiency.

2. Design Thought of Compression Algorithm

Hyperspectral image is one kind of 3D images which generally composed by two-dimensional spatial geometry information and one dimensional spectral information^[12]. The image has not only space correlation, but also spectral correlation. It puts forward the following structure (take 3 bands from a hyperspectral image as an example) to remove the spatial correlation and spectral correlation effectively.

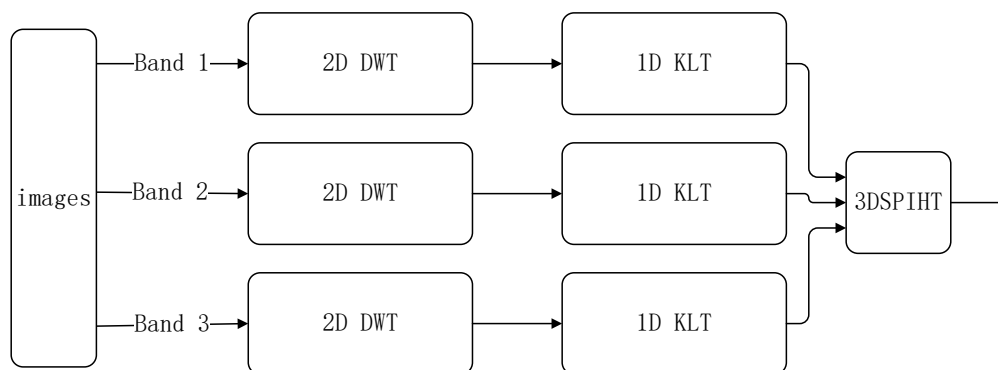


Figure 1. Structure of the Compression System

Traditional algorithm often takes KLT first, and then carries on the DCT and DWT. Because of the inconsistencies within the same spectrum, it is hard to reach the ideal compression effect using the same KLT basis. Referencing the related technology [13-17] and combining with the characteristics of hyperspectral image, it puts forward a near lossless compression algorithm which is applicable to the hyperspectral image. Firstly, two dimensional discrete wavelet transform is used for the image, and then comes one dimensional KLT with different KLT basis for different areas of the same spectrum, and optimizes the algorithm of every part later. It greatly improves the distortion problem of adjacent blocks, and realizes easier at the same time.

3. Key Technology

3.1. Discrete Wavelet Transform (DWT)

Discrete wavelet transform is a kind of transformation which discretizes the scale factor and shift factor of continuous wavelet. It has smaller redundancy and computational

complexity comparing with continuous wavelet, so it is more suitable for hardware implementation.

It will produce four sub zones, low frequency sub band (LL), the horizontal high frequency sub band (LH), vertical high frequency sub band (HL), and diagonal direction high frequency sub band (HH) respectively for one wavelet transform. The second transformation is similar with the first whose division is on the basis of the low frequency sub band (LL).

Lifting scheme is the efficient method to realize discrete wavelet transform (DWT). The lifting wavelet transformation can be thought as the decomposition process for the multiple matrix of target wavelet filter, which is decomposed into a constant diagonal matrix, M order triangular matrix, and lower triangular matrix in the form of product. The decomposition formula for lifting 5/3 wavelet transform matrix is expressed as follows:

$$d(2n+1) = x(2n+1) - \frac{x(2n) + x(2n+2)}{2}$$

$$d(2n) = x(2n) + \frac{d(2n-1) + d(2n+1) + 2}{4}$$

The lifting wavelet can be achieved by a forecast and update operations. Implementation process is shown in Figure 2.

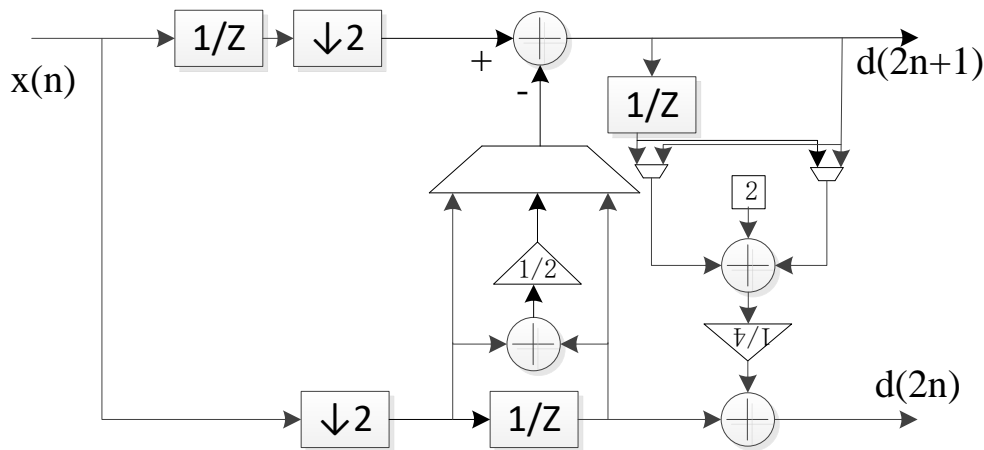


Figure 2. Achievement for The Lifting Wavelet

3.2. KLT

KLT is a kind of linear invertible transform to remove the correlation, which can be used in the principal component analysis (PCA). Compared with other orthogonal transformation, the energy is the most concentrated, and the error is minimum. The larger variances are all concentrate on a few coefficients after Transform, so that it could achieve good image compression effect through the appropriate coding. It is difficult for hardware implementation due to its large amount of calculation, so the method is rarely used in practical engineering. Aiming at this disadvantage, we make the appropriate improvement on the basis of the original algorithm to make it more suitable for hardware implementation. The calculation process of original algorithm is as follows:

Firstly, we get the average of a spectrum matrix for the target image, and then the matrix will minus the average for getting matrix H . Secondly, we can work out the covariance of H , namely $COV(H)$, and the eigenvalues R of the covariance and characteristic vector A . Finally we can get the KLT expression $Y=A \times H$. It has huge computational complexity, needs $M \times N - 1$ times addition and 1 time division when calculating the band size of $M \times N$, needs $M \times N$ times subtraction for calculating H , and

need more for other steps, so we will do the following improvement in view of the above problems.

On the premise of good quality and less computational complexity for the image, we randomly select a subset of the spectral vector and properly choose the subset size when calculating the covariance rather than using them all. The Figure 3 shows that the compression performance just begin to decline and computational complexity is very low when the sampling size is thousandth of the traditional method, this value is chosen as the sample to estimate covariance considering the compression performance and calculation time.

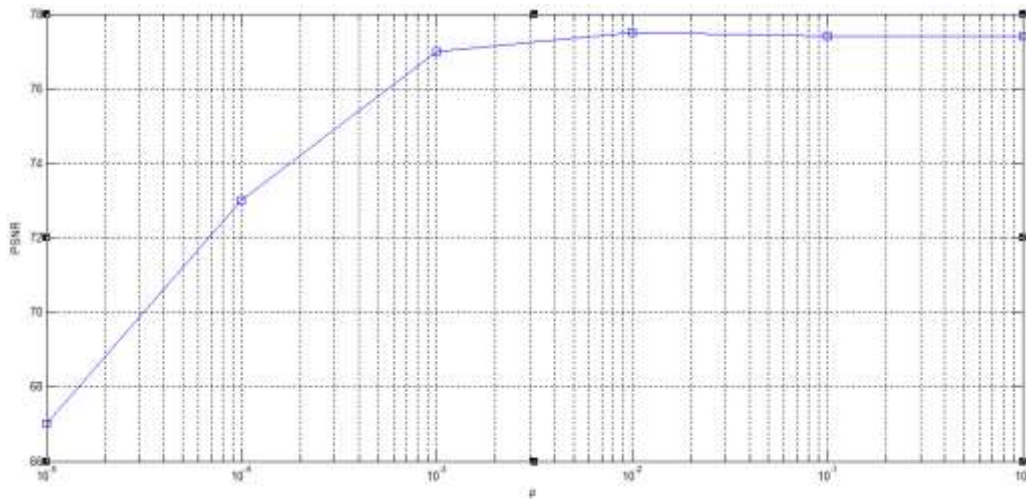


Figure. 3 Sampling Ratio on Compression Performance

Table 1 is the execution time of KLT for the use of sampling method to calculate the images with the size of $512 \times 512 \times 4$.

Table.1 Computation 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 Figure 3 and Table 1 shows that on the premise of guaranteeing the compression performance, computing time is greatly reduced using the method. We will adopt the following methods for calculating the eigenvalue and eigenvector.

$$V(i, j, \theta) = \begin{pmatrix} 1 & \dots & 0 & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \dots & \vdots & \dots & \vdots \\ 0 & \dots & c & \dots & -s & \dots & 0 \\ \vdots & \dots & \vdots & \dots & \vdots & \dots & \vdots \\ 0 & \dots & s & \dots & c & \dots & 0 \\ \vdots & \dots & \vdots & \dots & \vdots & \dots & \vdots \\ 0 & \dots & 0 & \dots & 0 & \dots & 1 \end{pmatrix} \begin{matrix} i \\ j \end{matrix} \quad (1)$$

Covariance matrix is a real symmetric matrix, which can be calculated through Jacobi algorithm^[18] normally. Jacobi algorithm adopts a series of Jacobi rotation planes to transform the symmetry matrix M into a diagonal matrix T , namely

$V_k \dots V_1 M^T \dots V_k^T = T$, Order $V_k \dots V_1 = V$, then $MM^T = T$, $M^T = V^T T$, T is an

orthogonal array, whose first I column is the eigenvalues for the feature vector λ_i of the matrix M . The rotation transform structure for Jacobi algorithm is shown as type (1).

(i, j) presents the positions of the eliminated elements in the matrix. $c = \cos \theta$, $s = \sin \theta$, and θ is the rotation angle. Iterative process of characteristic value is shown as type (2).

$$M^{k+1} = (V^{(k)})^T M^{(k)} V^{(k)} \quad (2)$$

Take A as the characteristic vector, so the iterative process of characteristic vector is shown as type (3).

$$A^{k+1} = A^{(k)} \times V^{(k)} \quad (3)$$

It is complex to calculate the correlation, and difficult to parallel implementation with Jacobi algorithm for not only selecting characteristic value of the main elements, but also rotating the rows and columns. To solve the above problems, we will use the following improved algorithm.

For calculating the eigenvalue of symmetric matrix M , we adopt a series of transformation taking M into square matrix T whose every two columns are all orthogonal, namely $M_1 \dots V_k = T$, so $T^T T = V_k^T \dots V_1^T M M_1 \dots V_k$. We can get the spectrum norm of every column for the square matrix T according to the relationship between matrix T and $T^T T$, namely, the absolute value of eigenvalue for symmetrical matrix. The positive and

negative can be judged through the relation $M b_i = \lambda_i b_i$ and b_i is same, and negative otherwise.

The transform structure of the algorithm is shown as type (4).

$$V(i, j, \theta) = \begin{pmatrix} 1 & \dots & 0 & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \dots & \vdots & \dots & \vdots \\ 0 & \dots & c & \dots & s & \dots & 0 \\ \vdots & \dots & \vdots & \dots & \vdots & \dots & \vdots \\ 0 & \dots & -s & \dots & c & \dots & 0 \\ \vdots & \dots & \vdots & \dots & \vdots & \dots & \vdots \\ 0 & \dots & 0 & \dots & 0 & \dots & 1 \end{pmatrix} \quad (4)$$

M^{k+1} is the same as M^k unless the column i and j , namely

$$a_i^{k+1} = a_i^k \cos \theta - a_j^k \sin \theta \quad (5)$$

$$a_j^{k+1} = a_i^k \sin \theta + a_j^k \cos \theta \quad (6)$$

The rotation angle θ could be gotten through the following formula.

$$\begin{cases} \tan 2\theta = \frac{2t}{1-t^2}, & x \neq 0 \\ \theta = 0, & x = 0 \end{cases} \quad (7)$$

$$t = \frac{\text{sign}(d)}{|d| + \sqrt{1+d^2}}, \quad d = \frac{z - y}{2x}$$

Among them, x is the vector inner product between column i and column j , y is the vector inner product between column i and column i , z is the vector inner product between column j and column j .

It only affects the elements related to i and j when transformation, the other elements are not be affected. So we can use 2×2 submatrix $S_{i,j} \in R^{2 \times 2}$ to rotate the transformation process.

$$S^{(k)} = \begin{bmatrix} S_{i,i}^{(k)} & S_{i,j}^{(k)} \\ S_{j,i}^{(k)} & S_{j,j}^{(k)} \end{bmatrix}, i, j = 1, 2 \quad (8)$$

The Iterative process is as type (9) -(11):

$$S^{(k+1)} = S^{(k)} \cdot R(a_{j,j}^{(k)}) \quad (9)$$

$$R(\theta^{(k)}) = \begin{pmatrix} \cos(\theta^{(k)}) & -\sin(\theta^{(k)}) \\ \sin(\theta^{(k)}) & \cos(\theta^{(k)}) \end{pmatrix} \quad (10)$$

$$\tan(2\theta_{i,i}^{(k)}) = \tan(2\theta_{j,j}^{(k)}) = \frac{2s_{i,j}^{(k)}}{s_{j,j}^{(k)} - s_{i,i}^{(k)}} \quad (11)$$

Among them, $S^{(0)} = S \cdot R(\theta^{(k)})$ is a rotate matrix, $\theta_{i,i}^{(k)}$ and $\theta_{j,j}^{(k)}$ are rotate angles. After a series of iterative, we get the first diagonal matrix $S^{(h)} \in R^{2 \times 2}$. The feature vector for matrix S is defined as $V \in R^{2 \times 2}$, where $V^{(0)} = I \in R^{2 \times 2}$, each iteration process is shown as (12).

$$V_{i,j}^{(k+1)} = V_{i,j}^{(k)} R(a_{j,j}^{(k)}) \quad (12)$$

In order to speed the algorithm and make it suitable for hardware implementation, we propose a pipelining parallel structure. The iterative process is as follows:

1: Store the matrix S^T . we know that $S \in R^{2 \times 2}$ is a symmetric matrix, so we can use its upper triangular matrix namely $S^T \in R^{2 \times 2}$ to get its eigenvalue and eigenvector.

2: The eigenvector matrix will be Initialized, and stored with the same memory size as S^T .

3: Formula (10) will be calculated through CORDIC rotation algorithm, and the rotation Angle $a_{i,j}^{(k)}$ will be gotten through the formula, namely $S_{i,j}^T \in R^{2 \times 2}$. Then the rotation angle is stored in order to calculate the eigenvalue and eigenvector.

4: Calculate the eigenvalue and eigenvector. We get each $S_{i,j}^T \in R^{2 \times 2}$ through formula (9), which include two multiplication operations of 2×2 child matrix. Because the multiplication contains vector rotation for $a_{i,j}^{(k)}$, therefore, the rotation pattern for CORDIC algorithm [19] will be used.

5: Store $S_{i,j}^T$ and $V_{i,j}^{(k)}$ with the same storage size as the initialized one.

6: Performing the previous steps to complete the first iteration, and then repeat Step 3 to Step5 for h times.

Repeat steps 3 to 5 to complete the parallel operation of the proposed algorithm after the first iteration.

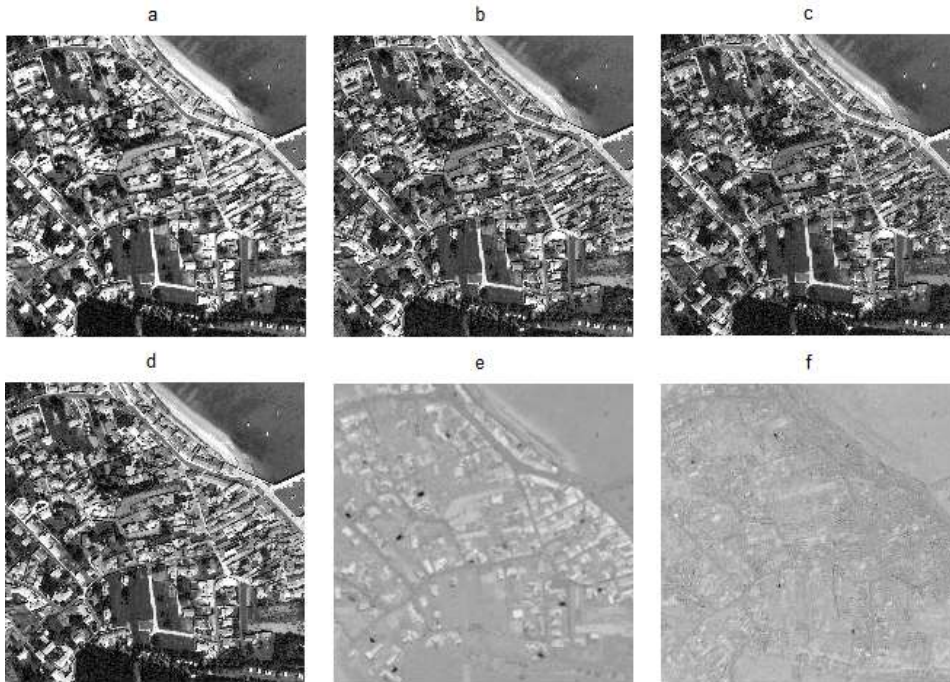


Figure 4. KL Transform for Each Spectrum

From Figure 4 we can see that KLT is used for the first three spectrums (**a**, **b**, **c**) of hyperspectral image (including four spectrums). It eliminates the redundancy between most of the spectrums, and energy spectrums mainly concentrate in the first two segments after transformation.

4. Experiment Results

In order to verify the feasibility of the algorithm, the hyperspectral image AVIRIS taken by JPL is chosen for testing. We use lifting 5/3 wavelet transform as the DWT filter first, then the improved KLT is used for removing spectral redundancy, finally, we use 3D-SPIHT for encoding. It can be seen that **a** is the original image, and **b** is the reconstruction image in Figure 5.

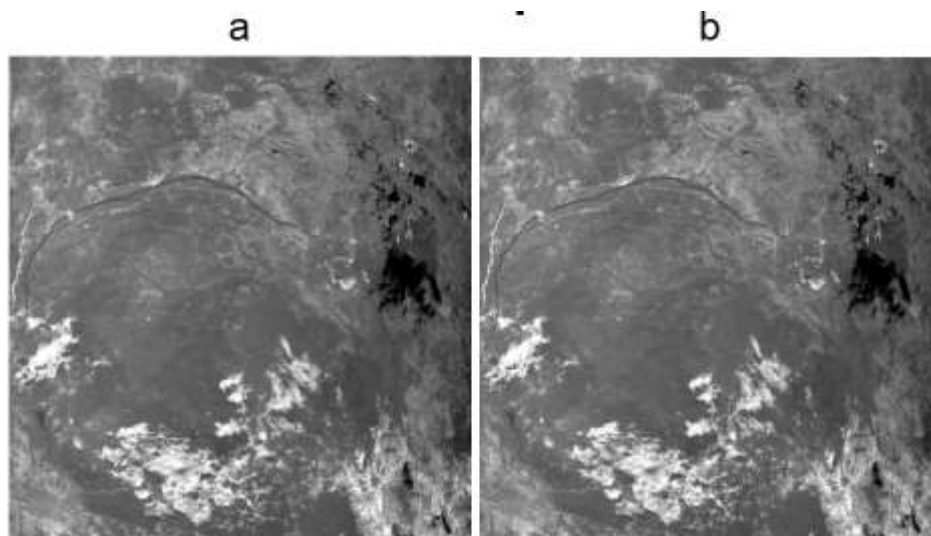


Figure 5. The Original Image and The Restored One

Because lossless compression is used, there is nearly no difference between the original image and reconstruction image theoretically. We use mean square error (MSE) formula to compare the original image and the reconstructed image of Figure 5, the MSE formula is as follows:

$$MSE = \sqrt{\frac{\sum_{x=1}^M \sum_{y=1}^N (f(x, y) - g(x, y))^2}{MN}} \quad (13)$$

Among them, $f(x, y)$ and $g(x, y)$ are the original image and the reconstructed image respectively, and M and N are the size of image. The MSE of image is approximate to 0 through calculation, so it is feasible. In addition, most of the space redundancy is eliminated after 2D 5/3 wavelet transform, depth of pixels for HL ranges from 8 bit to 0 ~ 6 bit, and most of the pixel values are 5 bit. It eliminates most of the spectral redundancy after KLT, and most of the depth for pixels is 1 bit. We use image entropy formula to see the values, image entropy formula for predicting residual is:

$$E = \sum_{i \in I} p_i \times \log(p_i) \quad (14)$$

Among them, p_i is the probability of the pixel i . Through calculation, we can get the values of image entropy for the three hyperspectrums were 4.39, 3.91, 3.25 respectively, so it eliminates most of the spectral redundancy, and greatly improves the compression ratio of the system.

Table 2. Compression Results

Images	Average compression ratio
Jasper Ridge	3.05
Lake Monona	2.88
Moffett Field	3.20
Low Altitude	2.91
Average	3.01

For testing the compression performance for our algorithm, a mass of AVIRIS and Hyperion hyperspectral images are used. The results are shown in Table 2.

In order to further verify the quality of compressed image, the average compression ratio is compared with the traditional hyperspectral image compression algorithm, and the results are shown in Table 3.

The Table 3 shows that the average compression ratio of the algorithm increased 0.04 ~ 2.23 than existing methods, and the average compression ratio is less than 4 bpp, so it has a good compression performance and suitable for hyperspectral image compression.

Table 3. The Versus for This Algorithm and Other Algorithms

algorithms	compression ratio
1.s-DSC-inf[20]	3.22
2.bidirectional prediction[21]	5.24
3.improved LUT[22]	3.05
4.OUR METHORD	3.01

5. Conclusion

Traditional compression algorithm cannot achieve good compression performance for using the single KLT basis. Aiming at this problem, it puts forward a hyperspectral image compression algorithm based on DWT and multiple KLT basis, both make up for the deficiency of the traditional algorithms, and make a further improvement in hardware

implementation, greatly reduces the computing time. Experimental results show that the algorithm guarantees the signal-to-noise ratio, effectively protects the texture and edge information of hyperspectral image, and establishes a certain foundation for later study of hyperspectral image compression.

References

- [1] Y.-J. Gao, "Research on efficient compression technology of satellite multispectral image", Xian: Xidian University, (2013), pp. 2-5.(in Chinese)
- [2] N. Sun, B.-L. Hu and S. Wang, "Reconstruction of compressive spectral imaging system of a FISTA algorithm-based coded aperture", *Infrared and Laser Engineering*, no. 01, (2014), pp. 963-966.
- [3] Z.-C. Gao and X.-L. Zhang, "Lossless compression of hyperspectral images using improved locally averaged interband scaling lookup tables", *IEEE international conference*, (2011), pp. 91-96.
- [4] Y.-J. Nian, Q. Xin and Y. Yang, "Distributed lossless compression of hyperspectral images based on multi-band prediction", *Optics and Precision Engineering*, vol. 20, no. 4, (2012), pp. 906-912. (in Chinese)
- [5] J. Song, Y.-S. Li and C.-K. Wu, "Compression for distributed interferential multispectral image based on set partitioning in hierarchical trees", *Acta Optica Sinica*, vol. 30, no. 2, (2010), pp. 358-363. (in Chinese)
- [6] L. Sun, J.-S. Luo and D.-F. Gu, "Hyperspectral image compression algorithm based on prediction between bands and rate pre-allocation", *Optics and Precision Engineering*, vol. 16, no. 4, (2008), pp. 752-757. (in Chinese)
- [7] Y.-B. Sun, "Sparse representations of images and its application to inverse problems in image processing", Nan Jing: Nan Jing University of science and technology, (2010), pp. 5~14. (in Chinese)
- [8] A. Moinuddin, E. Khan and M. Ghanbari, "An efficient wavelet based embedded color image coding technique using block-tree approach", *IEEE Int. Conf. Image Process*, (2006), pp. 1889-1892.
- [9] M. U. Torun and A. N. Akansu, "An Efficient Method to Derive Explicit KLT Kernel for First-order Autoregressive Discrete Process", *IEEE Transactions on signal processing*, vol. 61, no. 15, (2013), pp. 3944-3953.
- [10] H. Ju and S. Lee, "Complexity Reduction in Karhunen-Loeve Transform based Speech Coder for Voice Transmission", *IEEE Transactions on Consumer Electronics*, vol. 60, no. 1, (2014), pp. 130-136.
- [11] U. Bayazit, "Adaptive Spectral Transform for Wavelet-Based Color Image Compression", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 7, (2011), pp. 983-992.
- [12] C.-L. Zhang and X. Yang, "Correction of image uniformity for image spectrometer", *Chinese Optics*, vol. 6, no. 4, (2013), pp. 584-590. (in Chinese)
- [13] J.-W. Song, Z.-W. Zhang and X.-M. Chen, "Hyperspectral imagery compression via linear prediction and lookup tables", *Optics and Precision Engineering*, vol. 21, no. 8, (2013), pp. 2201-2208. (in Chinese)
- [14] J. W. Lee, "Optimized Quad tree for Karhunen-Loeve Transform in Multispectral Image Coding", *IEEE Transactions on Image Processing*, vol. 8, no. 4, (1999), pp. 453-461.
- [15] Y.-J. Nian, Q. Xin and Y. Tang, "Hyperspectral image compression based on 3D SPIHT algorithm", *Optics and Precision Engineering*, vol. 16, no. 6, (2008), pp. 1146-1151. (in Chinese)
- [16] G.-J. Li, "Improvement of realtime properties of MPEG-4 encoder by DSP underlying structure", *Chinese Optics*, vol. 4, no. 5, (2011), pp. 461-467. (in Chinese)
- [17] L.-B. Du, X.-M. Xiao and W.-S. Luo, "Quantification removing for satellite on-board remote image JPEG2000 compression algorithm", *Optics and Precision Engineering*, vol. 17, no. 3, (2009), pp. 690-694. (in Chinese)
- [18] R. P. Brent, F. T. Luk and C. Van Loan, "Computation of the singular value decomposition using mesh-connected processors", *J. VLSI Comput. Syst.*, vol. 1, no. 3, (1983), pp. 242-270.
- [19] J. S. Walther, "A unified algorithm for elementary functions", in *Proceedings of the Spring Joint Computer Conference*, (1971), pp. 379-385.
- [20] J. Song, Y.-S. Li and C.-K. Wu, "Lossless and near-lossless compression of hyperspectral images based on search for L_∞ minimum and coset coding", *Acta Electronic Sinica*, vol. 20, no. 7, (2011), pp. 1551-1555. (in Chinese)
- [21] J. Chang, W.-S. Tao and Y.-J. Nian, "Lossless compression for hyperspectral images based on bi-direction", *Electronics Optics & Control*, vol. 17, no. 10, (2010), pp. 65-68.
- [22] J. Li, L.-X. Jin and G.-N. Li, "Lossless compression of hyperspectral images for space-borne application", *Spectroscopy and Spectral Analysis*, vol. 32, no. 8, (2012), pp. 2264-2269.

Authors



Lin Xiao-Jun, he was born in 1992, male, master of Changchun institute of optics, fine mechanics and physics, Chinese academy of sciences, major: infrared imaging and image reconstruction.