A Low-Rank Tensor Decomposition Based Hyperspectral Image Compression Algorithm

Mengfei Zhang[1], Bo Du[1], Lefei Zhang[1](✉), and Xuelong Li[2]

[1] School of Computer, Wuhan University, Wuhan, China
{mzhang, remoteking, zhanglefei}@whu.edu.cn
[2] Center for OPTIMAL, State Key Laboratory of Transient Optics and Photonics, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an, China
xuelong_li@opt.ac.cn

Abstract. Hyperspectral image (HSI), which is widely known that contains much richer information in spectral domain, has attracted increasing attention in various fields. In practice, however, since a hyperspectral image itself contains large amount of redundant information in both spatial domain and spectral domain, the accuracy and efficiency of data analysis is often decreased. Various attempts have been made to solve this problem by image compression method. Many conventional compression methods can effectively remove the spatial redundancy but ignore the great amount of redundancy exist in spectral domain. In this paper, we propose a novel compression algorithm via patch-based low-rank tensor decomposition (PLTD). In this framework, the HSI is divided into local third-order tensor patches. Then, similar tensor patches are grouped together and to construct a fourth-order tensor. And each cluster can be decomposed into smaller coefficient tensor and dictionary matrices by low-rank decomposition. In this way, the redundancy in both the spatial and spectral domains can be effectively removed. Extensive experimental results on various public HSI datasets demonstrate that the proposed method outperforms the traditional image compression approaches.

Keywords: HSI · Compression · Reconstruction · Tensor representation · Low-rank decomposition

1 Introduction

Hyperspectral imaging systems record per-pixel reflectance spectroscopy in a number of narrow wavelength bands over a wide range of the electromagnetic spectrum, thereby providing a much higher spectral resolution than the typical gray-level and RGB images, and are thus able to better capture material-specific information. More recently, thanks to the development and decreasing cost of the latest hyperspectral cameras, they have become affordable and popular in many computer vision and multimedia applications [1] such as face recognition, iris recognition, and medical diagnosis [2–4], etc.

However, the provided numerous spectral bands have led to some problems [5]. Compared to the conventional gray-level and RGB images, despite possessing much

E. Chen et al. (Eds.): PCM 2016, Part I, LNCS 9916, pp. 141–149, 2016.
DOI: 10.1007/978-3-319-48890-5_14

more information, HSIs are trapped in increased computational and time cost owning to the large number of spectral bands. Thus it's quite important to find a way to efficiently process and analyze these HSIs which have hundreds of thousands of contiguous spectral bands. As an ideal solution, image compression algorithms played a great role in solving this problem.

There are many conventional and classical image compression methods that can solve aforementioned problems. Pixel coding-based methods such as JPEG [6], SPIHT [7] and EZW [8] are most widely applied compression methods. Du [9] proposed to deploy PCA [10, 11] in JPEG 2000 for gray-level image processing. 3D coding based algorithms such as the 3D reversible integer lapped transform [12] and the DWT coupled with tucker decomposition [13] are proved to be effective for both spectral and spatial information. However, these methods do not consider the internal characteristics of the HSI, which might result in a loss of the important information in the subsequent image analysis (e.g., classification and recognition). To address this issue, tensor [14] was applied in image compression method to preserve both neighborhood relationship across the spatial dimension as well as global correlation among the spectral dimension. Typical examples are multilinear principal component analysis [15] (MPCA) generalized by PCA, concurrent subspace analysis [16], tensor canonical correlation analysis [17], and some other tensor based HSI compression methods [18, 19]. Although, at first glance, these methods may appear suitable for HSI compression, they were actually designed for a group of tensor objects, and they may not be effective for single tensor data based image compression.

Conventional compression methods usually process the HSI by spectral channel or by pixel, while it is quite different in our framework. To fully exploit the advantages of HSI, we propose a novel patch-based low-rank tensor decomposition (PLTD) method, combining patch-based tensor representation and the tensor extension of dictionary learning [20]. Firstly, HSI is blocked into local tensor patches. Then, by exploring the nonlocal similarity over the spatial domain [21, 22], similar tensor patches are grouped together by clustering method, and a fourth-order tensor are constituted for each cluster. Since the tensor data is assumed to be redundant, each big tensor can be finally low-rank decomposed into low-dimensional coefficient tensor and dictionary matrices. In this way, both spatial and spectral redundancy are optimally removed by our proposed framework.

The remainder of this paper is organized as follows. Section 2 presents the detailed PLTD algorithm for HSI compression and reconstruction. Experiments and analysis are provided in Sect. 4. The last section is a conclusion.

2 Patch-Based Low-Rank Tensor Decomposition (PLTD) for HSI Compression

The compression process of the PLTD framework can be summarized as blocking, clustering, tensor dictionary learning and low-rank decomposition. It can realize simultaneous compression of both the spatial and spectral modes. The reconstructed HSI can be simply obtained by combining the recovered patches which are the tensor product of coefficient tensor and dictionaries per cluster. The overall process is illustrated in the

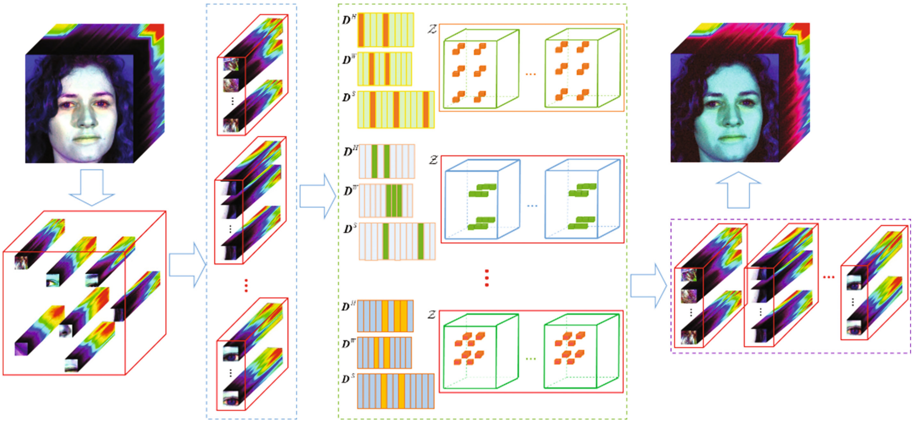


Fig. 1. Flow chart of the PLTD algorithm.

flow chart in Fig. 1. The detailed theory of the framework are given in the following sections, in which notations and multilinear algebras can refer to paper [23].

2.1 Patch-Based Tensor Representation

Given a third-order tensor HSI $\mathcal{I} \in \mathrm{R}^{L^W \times L^H \times L^S}$, we divide it by block size $l^W \times l^H$ ($l^W < L^W$, $l^H < L^H$) and consider there is no overlap between patches. We can then construct a group of 3D patches $\{\mathcal{P}_{i,j}\}_{1 \le i \le n^W, 1 \le j \le n^H} \subset R^{l^W \times l^H \times l^S}$, where $n^W = L^W/l^W$ and $n^H = L^H/l^H$, and the patch number n is equal to $n = n^W n^H$. Each patch is a cube preserving all the spectral information of the original HSI.

To make full use of the local information in the spatial and spectral domain and to simplify the deduction of the algorithm, we divide the patches into several clusters according to nonlocal similarity. After that, each cluster is reformed as a fourth-order tensor and can be resolved individually.

2.2 Tensor Dictionary Learning

Dictionary learning was originally used for image restoration. The objective function of the dictionary learning model can be given as:

$$\min_{\boldsymbol{D},\boldsymbol{Z}} \| \boldsymbol{A} - \boldsymbol{D}\boldsymbol{Z} \|, s.t. \mathcal{O}(z_i) \le s \tag{1}$$

where $\boldsymbol{D} = [\boldsymbol{d}_1, \ldots, \boldsymbol{d}_m] \in \mathrm{R}^{d \times m}, m > d$ is a redundant dictionary; and $\boldsymbol{A} = [\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n] \in \mathrm{R}^{d \times n}$ is a collection of n sample images, each image is originally 2D and is vectorized to a long vector $\boldsymbol{a}_i$. $\boldsymbol{Z} = [z_1, \ldots, z_n] \in \mathrm{R}^{m \times n}$ is the coefficient matrix. $\mathcal{O}(\cdot)$ denotes a sparsity controlling operator such as the l_0 or l_1 norm.

The tensor form of dictionary learning can be written as:

$$\min_{\boldsymbol{D}^W, \boldsymbol{D}^H, \boldsymbol{D}^S, \mathcal{Z}_i} \sum\nolimits_{k=1}^{K} \parallel \mathcal{P}^{(k)} - \mathcal{Z}_i \times_1 \boldsymbol{D}^W \times_2 \boldsymbol{D}^H \times_3 \boldsymbol{D}^S \parallel, s.t. \mathcal{O}(\mathcal{Z}_i) \leq \boldsymbol{s} \tag{2}$$

In this function, $\boldsymbol{D}^W \in \mathrm{R}^{l^W \times d^W}$, $\boldsymbol{D}^H \in \mathrm{R}^{l^H \times d^H}$, $\boldsymbol{D}^S \in \mathrm{R}^{L^S \times d^S}$, and $(l^W < d^W, l^H < d^H, L^S < d^S)$. When applied to HSI compression problem, $\mathcal{P}_i$ is the original data and $\mathcal{Z}_i$, $\boldsymbol{D}^W$, $\boldsymbol{D}^H$ and $\boldsymbol{D}^S$ are the compressed data.

Describing clustering by algebraic representation, the clusters after dividing are denoted as $\left\{\mathcal{P}_{k,j}\right\}_{j=1}^{n_k} (k = 1, \ldots, K)$, where K is the cluster number and n_k is the number of patches of cluster k. We reform cluster k to be a fourth-order tensor $\mathcal{P}^{(k)} \in R^{l^W \times l^H \times L^S \times n_k}$. Similarly, we define the corresponding coefficient tensors of cluster k as $\left\{\mathcal{Z}_{k,j}\right\}_{j=1}^{n_k} (k = 1, \ldots, K)$ and its fourth-order tensor as $\mathcal{Z}^{(k)} \in R^{d^W \times d^H \times d^S \times n_k}$.

In addition, it is verified that for an nth-order tensor $\mathcal{T}$ there is a smallest subset $I^1, \ldots, I^n (I^i|_{i=1,\ldots,n} \in R^{m^i})$ satisfying that the tensor value $t(i_1, \ldots, i_n) = 0$ for all $(i_1, \ldots, i_3) \notin [I^1, I^2, \ldots, I^{n-1}]$. On this condition, the tensor $\mathcal{T}$ is sparse [24, 25]. We can denote $idt(\mathcal{T})$ as an intrinsic dense tensor of $\mathcal{T}$ composing of all the nonzero entries from $\mathcal{T}$. According to this analysis, the objective function can then be reformed as follows:

$$\begin{aligned} &\min_{\boldsymbol{D}^W, \boldsymbol{D}^H, \boldsymbol{D}^S, \mathcal{Z}^{(k)}} \sum\nolimits_{k=1}^{K} \parallel \mathcal{P}^{(k)} - \mathcal{Z}^{(k)} \times_1 \boldsymbol{D}^W \times_2 \boldsymbol{D}^H \times_3 \boldsymbol{D}^S \parallel \\ &s.t. \mathcal{O}_t(\mathcal{Z}_{k,j}|_{j=1,\ldots,n_k, k=1,\ldots,K}) \prec (m_k^W, m_k^H, m_k^S) \end{aligned} \tag{3}$$

The constraint means that the number of nonzero elements of first three modes are less than m_k^W, m_k^H, and m_k^S, respectively, while the fourth-order is unchanged. Thus constraining the tensor $\mathcal{Z}^{(k)}$ to be sparse in the spatial and spectral domains.

2.3 Low-Rank Tensor Decomposition

Each patch is linearly combined by a rather small number of dictionary atoms, leading to high redundancy of the three dictionaries. Considering this fact, we split the dictionaries so that each sub-dictionary consists of only those atoms that are utilized by a single cluster. For example, the dictionaries of cluster k can be reformulated as: $\boldsymbol{D}_k^W \in \mathrm{R}^{l^W \times m_k^W}, \boldsymbol{D}_k^H \in \mathrm{R}^{l^H \times m_k^H}, \boldsymbol{D}_k^S \in \mathrm{R}^{L^S \times m_k^S}$, where m_k^w, m_k^H and m_k^S represent the number of atoms that are utilized in cluster k along the width, height, and spectral modes, respectively. In the same time, we can replace the coefficient tensor with its intrinsic dense tensor. As a result, the objective function can be reformulated like:

$$\min_{\boldsymbol{D}_k^W, \boldsymbol{D}_k^H, \boldsymbol{D}_k^S, idt(\mathcal{Z}^{(k)})} \sum\nolimits_{k=1}^{K} \parallel \mathcal{P}^{(k)} - idt(\mathcal{Z}^{(k)}) \times_1 \boldsymbol{D}_k^W \times_2 \boldsymbol{D}_k^H \times_3 \boldsymbol{D}_k^S \parallel \tag{4}$$

The size of coefficient tensor and dictionaries in each cluster are much smaller than the original ones, and the coefficient tensor is dense now so there is no longer any need to impose the sparsity constraint on the values. After replacing $idt(\mathcal{Z}^{(k)})$ with $\mathcal{S}^{(k)}$ for simplification, the objective function is equivalent to the following:

$$\min_{\boldsymbol{D}_k^W, \boldsymbol{D}_k^H, \boldsymbol{D}_k^S, \mathcal{S}^{(k)}} \sum\nolimits_{k=1}^{K} \| \mathcal{P}^{(k)} - \mathcal{S}^{(k)} \times_1 \boldsymbol{D}_k^W \times_2 \boldsymbol{D}_k^H \times_3 \boldsymbol{D}_k^S \| \tag{5}$$

This problem is much easier since it has no constraints. Considering that there are only three dictionaries for the fourth-order core tensor, this function can be approximated as a low-rank tensor decomposition problem by adding a matrix:

$$\min_{\boldsymbol{U}_k^{(1)}, \boldsymbol{U}_k^{(2)}, \boldsymbol{U}_k^{(3)}, \boldsymbol{U}_k^{(4)}, \mathcal{G}^{(k)}} \sum\nolimits_{k=1}^{K} \| \mathcal{X}^{(k)} - \mathcal{G}^{(k)} \times_1 \boldsymbol{U}_k^{(1)} \times_2 \boldsymbol{U}_k^{(2)} \times_3 \boldsymbol{U}_k^{(3)} \times_4 \boldsymbol{U}_k^{(4)} \| \tag{6}$$

Matrices $\boldsymbol{U}_k^{(1)}, \boldsymbol{U}_k^{(2)}$ and $\boldsymbol{U}_k^{(3)}$ are three dictionaries, and the result of $\mathcal{G}^{(k)} \times_4 \boldsymbol{U}_k^4$ is equal to coefficient tensor $\mathcal{S}^{(k)}$. Note that $\boldsymbol{U}_k^3 \in \mathrm{R}^{L^S \times m_k^S}$ and $L^S > m_k^S$, so the spectral bands are compressed. Therefore, the redundant spectral information is removed. The detailed steps are exhibited in the following Algorithm 1.

Algorithm 1: The PLTD algorithm for HSI reconstruction

```
Input: HSI $\mathcal{I} \in \mathrm{R}^{L^W \times L^H \times L^S}$
Output: dictionaries $\boldsymbol{D}^W = [\boldsymbol{D}_1^W, \cdots, \boldsymbol{D}_K^W]$, $\boldsymbol{D}^H = [\boldsymbol{D}_1^H, \cdots, \boldsymbol{D}_K^H]$, $\boldsymbol{D}^S = [\boldsymbol{D}_1^S, \cdots, \boldsymbol{D}_K^S]$, coefficient tensors $\mathcal{Z}^{(k)}$, and the reconstructed HSI $\mathcal{R} \in \mathrm{R}^{L^W \times L^H \times L^S}$.
Split the HSI into patches $\{\mathcal{P}_{i,j}\}_{1 \le i \le L^W/l^W, 1 \le j \le L^H/l^H}$.
Group all the patches into $K$ clusters $\{\mathcal{P}_{k,j}\}_{j=1}^{n_k} (k = 1, \cdots, K)$.
For $k$=1:$K$
    Given $m_k^W, m_k^H, m_k^S$ of cluster $k$.
    According to the Tucker decomposition, compute (6) to tain $\mathcal{G}^{(k)}, \boldsymbol{U}_k^{(1)}, \boldsymbol{U}_k^{(2)}, \boldsymbol{U}_k^{(3)}$, and $\boldsymbol{U}_k^{(4)}$ of cluster $k$.
    Calculate the three dictionaries and the coefficient tensor of cluster $k$ by $\boldsymbol{D}_k^W = \boldsymbol{U}_k^{(1)}$, $\boldsymbol{D}_k^H = \boldsymbol{U}_k^{(2)}$, $\boldsymbol{D}_k^S = \boldsymbol{U}_k^{(3)}$, and $idt(\mathcal{Z}^{(k)}) = \mathcal{G}^{(k)} \times_4 \boldsymbol{U}_k^{(4)}$.
End for
Compute the reconstructed fourth-order tensor patch group by $\mathcal{R}^{(k)} = \mathcal{G}^{(k)} \times_1 \boldsymbol{U}_k^{(1)} \times_2 \boldsymbol{U}_k^{(2)} \times_3 \boldsymbol{U}_k^{(3)} \times_4 \boldsymbol{U}_k^{(4)}$.
Combine the reconstructed patches $\{\mathcal{R}^{(k)}\}$.
Output $\boldsymbol{D}^W, \boldsymbol{D}^H, \boldsymbol{D}^S, \mathcal{Z}^{(k)}$, and $\mathcal{R}$.
```

3 Experiments and Analysis

In this section, we present the experiments undertaken on a natural scene hyperspectral dataset and provide the experimental analysis.

3.1 Experimental Setup

We employed five image compression methods for comparison: MPCA [15], PCA [10], JPEG [6], PCA+JPEG, and KSVD [26]. Among which, JPEG and KSVD are originally proposed for gray-level image processing. When dealing with HSIs, JPEG treats them as a group of gray-level images and compresses one by one. PCA treats all the spectral values of one pixel as a sample, *i.e.*, PCA compresses the band information only. MPCA takes the gray-level image of each band as a sample and compress the spatial redundancy only. For the PCA+JPEG method, this method involves successively compressing an image by PCA and JPEG.

The HSI has to be clipped so that the spatial dimension is divisible by the patch size. For JPEG, the spatial dimension size has to be divisible by 8×8 since the DCT translation splits the original gray-level image into 8×8 data blocks. Considering these limitations, we adopt the clipping step in all the comparison methods. Furthermore, to alleviate the influence caused by data diversity, we normalized the hyperspectral data to keep all the values in the interval [0, 255].

We employed five indices for the image compression performance evaluation: the peak signal-to-noise ratio (PSNR), the structural similarity (SSIM) index, the feature similarity (FSIM) index, the erreur relative globale adimensionnelle de synthese (ERGAS) [27], and the spectral angle mapper (SAM) [28]. For PSNR, SSIM, and FSIM, the larger the values, the better the performance of the algorithm. The opposite goes for ERGAS and SAM.

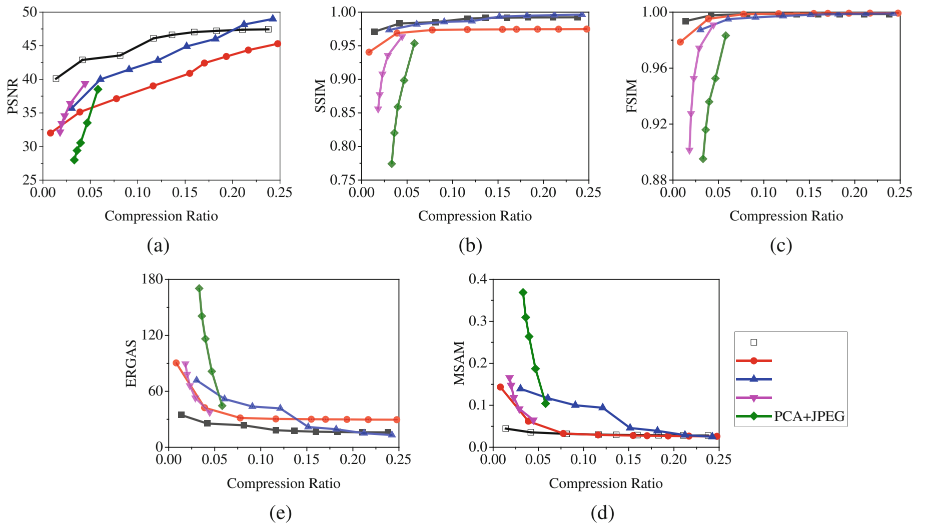
3.2 Performance on Natural Scene HSI

To begin with, we choose an image from a natural scene HSI database for compression, the HSI is named scene 8 in this paper. The original dimension of scene 8 is $1018 \times 1340 \times 33$. We clip it into the size of $512 \times 512 \times 33$. Then, we normalize it to keep its value be within the scope 0–255. The results and comparison are listed in Table 1. The compression ratio is the ratio between the size of the original data and the sum of the compressed data size. It is reasonable that the compression ratio of the six methods could not be completely the same; however, we tried as far as possible to keep them approximately equal.

Note that the above comparisons were carried out at only one compression ratio. So what would the situation be with other compression ratios? Would the other methods outperform the PLTD algorithm? To answer these questions, we varied the compression ratio and tested the compression results. We changed the ratio from 0 to 0.25, which was adequate to test these methods. The comparisons are exhibited in Fig. 2.

Table 1. Comparision of compression and reconstruction results on hyperspectral scene 8.

	Proposed	MPCA	PCA	JPEG	PCA+JPEG
Rate	0.0441	0.0467	0.0445	0.0465	0.0467
PSNR	**43.1343**	39.1176	39.3904	37.5290	33.5215
SSIM	**0.9836**	0.9666	0.9637	0.9511	0.8984
FSIM	**0.9980**	0.9960	0.9909	0.9764	0.9526
ERGAS	**24.7816**	42.6079	37.5006	46.2918	81.4623
MSAM	**0.0351**	0.0498	0.0648	0.0465	0.1871
Time	10.56	14.58	128.72	1.06	235.90



(a) (b) (c)

(e) (d)

Fig. 2. Comparison of the compression and reconstruction results when the compression ratio varies from 0 to 0.25.

According to Fig. 2, despite the certain interval within which the MPCA algorithm or the PCA algorithm perform better than the PLTD algorithm, the PLTD algorithm obtains the best performance in most cases. Exceptions occur because the PCA algorithm only compresses the spectral information, and MPCA only compresses the spatial information, while PLTD compresses both the spatial and spectral information. Therefore, most of the time, the tensor-based PLTD algorithm performs better than the other methods.

To sum up the experiments, we implemented enough experiments on the hyperspectral scene 8 image to confirm that the proposed method outperforms all the other methods.

4 Conclusion

In this paper, a new compression method called patched-based low-rank tensor decomposition (PLTD) have been proposed. The new method employs tensor patch representation, nonlocal similarity, tensor dictionary learning together to solve the problem of effectively compressing HSIs. According to the paper, by clustering, a big data problem is decomposed into several small data problems which can be quickly solved by low-rank decomposition algorithm. Since the decomposed data discard both the spectral and spatial redundancy, and only retain the most discriminant information, the data after compression can better improve the efficiency and accuracy for the subsequent analysis. As far as we know, this is the first time that a patch-based tensor structure has been proposed for HSI compression. And it is confirmed in the experiments that the proposed method outperforms others and has a wide applicability in various fields.

Acknowledgment. This work was supported by the National Natural Science Foundation of China under Grants 61401317, 61471274, 91338111, and U1536204.

References

1. Jia, X., Kuo, B.-C., Crawford, M.M.: Feature mining for hyperspectral image classification. Proc. IEEE **101**(3), 676–697 (2013)
2. Lee, S.H., Choi, J.Y., Ro, Y.M., Plataniotis, K.: Local color vector binary patterns from multichannel face images for face recognition. IEEE TIP **21**(4), 2347–2353 (2012)
3. Chen, R., Ding, X.L.T.: Liveness detection for iris recognition using multispectral images. PR Lett. **33**(12), 123–134 (2012)
4. Zhou, Y., Chang, H., Barner, K., Spellman, P., Parvin, B.: Classification of histology sections via multispectral convolutional sparse coding. In: Proceedings of CVPR, pp. 3081–3088 (2014)
5. Holloway, J., Priya, T., Veeraraghavan, A., Prasad, S.: Image classification in natural scenes: are a few selective spectral channels sufficient? In: Proceedings of ICIP, pp. 655–659 (2014)
6. Wallace, G.: The jpeg still picture compression standard. IEEE TCE **38**(1), 18–34 (1992)
7. Said, A., Pearlman, W.A.: A new fast and efficient image codec based on set partitioning in hierarchical trees. IEEE TCSVT **6**(6), 243–250 (1996)
8. Shapiro, J.M.: Embedded image coding using zero-trees of wavelet coefficients. IEEE TSP **41**(12), 3445–3462 (1993)
9. Du, Q., Fowler, J.E.: Hyperspectral image compression using jpeg2000 and principal component analysis. Geosci. Remote Sens. Lett. **4**(2), 201–205 (2007)
10. Jolliffe, I.T.: Principal Component Analysis. Springer, Heidelberg (2002)
11. Guo, Y., Lin, X., Teng, Z., Xue, X., Fan, J.: A covariance-free iterative algorithm for distributed principal component analysis on vertically partitioned data. PR **45**(3), 1211–1219 (2011)
12. Wang, L., Jiao, L., Bai, J., Wu, J.: Hyperspectral image compression based on 3D reversible integer lapped transform, electron. Electron. Lett. **46**(24), 1601–1602 (2010)

13. Karami, A., Yazdi, M., Mercier, G.: Compression of hyperspectral images using discrete wavelet transform and tucker decomposition. IEEE JSTARS **5**(2), 444–450 (2012)
14. Lathauwer, L.D.: Signal Processing Based on Multilinear Algebra. Katholieke Universiteit Leuven (1997)
15. Lu, H., Plataniotis, K.N., Venetsanopoulos, A.N.: Mpca: multilinear principal component analysis of tensor objects. IEEE SPM **19**(1), 18–39 (2008)
16. Xu, D., Yan, S., Zhang, L., Lin, S.: Reconstruction and recognition of tensor-based objects with concurrent subspaces analysis. IEEE TCSVT **18**(1), 36–47 (2008)
17. Luo, Y., Tao, D., Wen, Y., Ramamohanarao, K., Xu, C., Wen, Y.: Tensor canonical correlation analysis for multi-view dimension reduction. IEEE TKDE **27**(11), 3111–3124 (2015)
18. Zhang, L., Zhang, L., Tao, D., Huang, X., Du, B.: Compression of hyperspectral remote sensing images by tensor approach. Neurocomputing **147**, 358–363 (2015)
19. Zhang, L., Zhang, L., Tao, D., Huang, X.: Tensor discriminative locality alignment for hyperspectral image spectral-spatial feature extraction. IEEE TGRS **51**(1), 242–256 (2013)
20. Yang, C., Shen, J., Peng, J., Fan, J.: Image collection summarization via dictionary learning for sparse representation. PR **46**, 948–961 (2013)
21. Buades, A., Coll, B., Morel, J.-M.: A non-local algorithm for image denoising. In: Proceedings of CVPR, pp. 60–65 (2005)
22. Zhou, C., Güney, F., Wang, Y., Geiger, A.: Exploiting object similarity in 3D reconstruction. In: Proceedings of ICCV (2015)
23. Lu, H., Plataniotis, K.N., Venetsanopoulos, A.N.: A survey of multilinear subspace learning for tensor data. PR **44**(7), 1540–1551 (2011)
24. Peng, Y., Meng, D., Xu, Z., Gao, C., Yang, Y., Zhang, B.: Decomposable nonlocal tensor dictionary learning for multispectral image denoising. In: Proceedings of CVPR, pp. 4321–4328 (2014)
25. Zhang, Z., Xu, Y., Yang, J., Li, X., Zhang, D.: A survey of sparse representation: algorithms and applications. CoRR abs, **3**, 490–530 (2015)
26. Aharon, M., Elad, M., Bruckstein, A., Katz, Y.: K-SVD: an algorithm for designing over complete dictionaries for sparse representation. IEEE TSP **54**(11), 4311–4322 (2006)
27. Wald, L.: Data Fusion: Definitions and Architectures: Fusion of Images of Different Spatial Resolutions. Les Presses Ecole des Mines, Paris (2002)
28. Yuhas, R.H., Boardman, J.W., Goetz, A.F.H.: Determination of semi-arid landscape endmembers and seasonal trends using convex geometry spectral unmixing techniques. Ratio **4**(22) (1990)