HYPERSPECTRAL TARGET DETECTION: A PREPROCESSING METHOD BASED ON TENSOR PRINCIPAL COMPONENT ANALYSIS

Zehao Chen[1, 2, 3], Bin Yang[1, 2, 3], and Bin Wang[1, 2, 3]*

1. Key Laboratory for Information Science of Electromagnetic Waves (MoE), Fudan University, Shanghai, China;
2. State Key Laboratory of Earth Surface Processes and Resource Ecology, Beijing Normal University, Beijing, China;
3. Research Center of Smart Networks and Systems, School of Information Science and Technology, Fudan University, Shanghai, China

ABSTRACT

Traditional target detection (TD) methods for hyperspectral imagery (HSI) suffer from background interference. In this paper, we propose a novel preprocessing method based on tensor principal component analysis (TPCA) to separate the background and target apart. In our approach, HSI is decomposed into the sum of the principal component (PC) part and the residual part, and TD is performed on the latter. TPCA takes spatial and spectral information into account jointly, and treats spatial and spectral information differently, which is in line with HSI physical meanings. Experiments on both synthetic and real data indicate that our TPCA-based method outperforms other feature extraction preprocessing methods in terms of TD results.

Index Terms—Hyperspectral imagery, target detection, tensor principal component analysis (TPCA)

1. INTRODUCTION

Hyperspectral remote sensors capture images in hundreds of narrow spectral bands, which span the visible to infrared spectrum. Each pixel of HSI is presented as a continuous spectral curve. Abundant spatial and spectral information exist in HSI, jointly. This property enables discrimination of materials. Hyperspectral target detection (TD) is essentially a dichotomous problem where pixels are labeled as target or background based on their spectral characteristics. Hyperspectral TD is widely used in many fields.

In decades, many TD algorithms have been proposed. Probability and statistics based model is one of the most classic TD models, and several traditional TD algorithms have been proposed based on it. The detection operator of constrained energy minimization (CEM) [1] algorithm minimizes the output energy of unknown background pixel while fixing the output energy of a priori target spectrum in order to accomplish the TD task. The adaptive cosine / coherence estimation (ACE) [2] and the adaptive matching filter (AMF) [3] algorithms are based on Kelly's generalized likelihood ratio detection operator, assuming a mixed background model subject to multivariate Gaussian distribution.

Nevertheless, the aforementioned methods suffer from background interference. Therefore, it is important to separate the background and target apart using preprocessing method before TD. Feature extraction algorithms such as principal component analysis (PCA) and Tucker decomposition [5] decompose HSI into the PC part and the residual part. According to the model of hyperspectral TD problem, most of HSI pixels belong to background while target only resent in few pixels. Therefore, the PC part and residual part represent background and target, respectively. However, PCA fails to consider spatial information of HSI, and Tucker decomposition based on tensor model shows no respect to the difference of spatial and spectral dimensions of HSI. Since neither of these two feature extraction methods accord with HSI physical meanings, we need a new feature extraction approach which coincides HSI property better for hyperspectral TD preprocessing.

Recently, an innovative feature extraction algorithm has been proposed for HSI classification via tensor principal component analysis (TPCA) [4]. TPCA takes both spatial and spectral information into account, and treats spatial and spectral dimensions differently. Thus it accords with the HSI property. In this paper, we propose a TD preprocessing algorithm based on TPCA. By using our TPCA-based preprocessing algorithm, the background and the target are well separated, and better TD results are obtained compared with other feature extraction methods.

2. RELATED WORKS

2.1. Feature extraction

The traditional PCA algorithm transforms the samples in the original space to vectors in feature space that can retain most of the available information through linear transformation. For samples $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N \in R^D$, singular value decomposetion (SVD) is performed on its covariance matrix

This work was supported in part by the National Natural Science Foundation of China under Grant 61572133 and by the Research Fund for the State Key Laboratory of Earth Surface Processes and Resource Ecology under Grant 2017-KF-19.

$\mathbf{G} = (1/(N-1))\sum_{k=1}^{N}(\mathbf{x_k} - \bar{\mathbf{x}})(\mathbf{x_k} - \bar{\mathbf{x}})^T$, where $\bar{\mathbf{x}}$ is the sample mean, and $\mathbf{G} = \mathbf{U} \cdot \mathbf{S} \cdot \mathbf{V}^T$ is obtained. Using the orthogonal transformation matrix $\mathbf{U}$, the PCA result $\hat{\mathbf{y}} = \mathbf{U}^T(\mathbf{y} - \bar{\mathbf{x}})$ of the test sample can be obtained. The elements are arranged according to the orders of the PCs. The elements ranked ahead are the PCs of the test samples in the feature space.

Tucker decomposition is a feature extraction algorithm based on tensor models. For a third-order tensor, the form of Tucker decomposition is $\mathcal{X} = G \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C}$, where $\times_n$ represents the *n*-mode product. G is called the core tensor. $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ are the transformation matrices in three dimensions, respectively.

2.2. Tensor algebra

We briefly give some definitions about tensor algebra [6].

Tensor multiplication: Given tensors $x_t, y_t \in R^{m \times n}$, their multiplication $z_t = x_t \circ y_t \in R^{m \times n}$ is defined as $[z_t]_{i,j} = \sum_{k_1=1}^{m}\sum_{k_2=1}^{n}[x_t]_{k_1,k_2}[y_t]_{\mathrm{mod}(i-k_1,m)+1,\mathrm{mod}(j-k_2,n)+1}$.

Tensor space: Each element of a tensor space is a tensor with a certain size. Tensor space is noted as $\mathcal{R}_t$.

Tensor vector/tensor matrix: Vectors or matrices defined in tensor spaces are called tensor vectors or tensor matrices.

SVD of tensor matrix: Given square tensor matrix $\mathbf{X_{tm}} \in \mathcal{R}_t^{D \times D}$, we have the singular value decomposition $\mathbf{X_{tm}} = \mathbf{U_{tm}} \circ \mathbf{S_{tm}} \circ \mathbf{V_{tm}}^T$.

Slice of tensor matrix: Given tensor matrix $\mathbf{X_{tm}} \in \mathcal{R}_t^{D_1 \times D_2}$, define its (ω_1, ω_2) slice as $[\mathbf{X_{tm}}(\omega_1, \omega_2)]_{i,j} = [[\mathbf{X_{tm}}]_{i,j}]_{\omega_1,\omega_2}$.

Fourier transform of tensor matrix: The Fourier transform of a tensor matrix X_{tm} is calculated as $[F(\mathbf{X_{tm}})]_{i,j} = F([\mathbf{X_{tm}}]_{i,j})$.

2.3. TPCA

TPCA is the tensorial extension of PCA, performing feature extraction in a tensor space. Given samples $\mathbf{X_{tv,1}}, \mathbf{X_{tv,2}}, ..., \mathbf{X_{tv,N}} \in \mathcal{R}_t^D$, the sample mean is $\bar{X}_{tv}$, and the sample covariance tensor matrix is

$$\mathbf{G_{tm}} = \frac{1}{N-1}\sum_{k=1}^{N}(\mathbf{X_{tv,k}} - \bar{\mathbf{X}}_{\mathbf{tv}}) \circ (\mathbf{X_{tv,k}} - \bar{\mathbf{X}}_{\mathbf{tv}})^T . \quad (1)$$

We operate singular value decomposition on the covariance tensor matrix as defined in the last section:

$$\mathbf{X_{tm}} = \mathbf{U_{tm}} \circ \mathbf{S_{tm}} \circ \mathbf{V_{tm}}^T . \quad (2)$$

In this way, for any sample $Y_{tv} \in \mathcal{R}_t^D$, the feature tensor vector transformed into the feature tensor space is

$$\hat{\mathbf{Y}}_{\mathbf{tv}} = \mathbf{U}_{\mathbf{tm}}^T \circ (\mathbf{Y_{tv}} - \bar{\mathbf{X}}_{\mathbf{tv}}) . \quad (3)$$

After obtaining the sample transformed feature tensor vector, TPCA algorithm uses the tensor slice to restore it to the vector in real number space:

$$\hat{\mathbf{y}} = \delta(\hat{\mathbf{Y}}_{\mathbf{tv}}) = \frac{1}{mn}\sum_{\omega_1=1}^{m}\sum_{\omega_2=2}^{n}\hat{\mathbf{Y}}_{\mathbf{tv}}(\omega_1, \omega_2) . \quad (4)$$

Since the tensor multiplication is defined in the form of a circular convolution, it can be implemented more efficiently via the Fourier transform. The fast implementation of TPCA via Fourier transform can be carried out as follows:

Algorithm 1: Fast TPCA computation via the Fourier transform

Input: $\mathbf{X_{tv,1}}, \mathbf{X_{tv,2}}, ..., \mathbf{X_{tv,N}} \in \mathcal{R}_t^D$: Training tensor vector samples

$\mathbf{Y_{tv}} \in \mathcal{R}_t^D$: Test tensor vector sample

Output: $\hat{\mathbf{y}} \in R^D$: Real number space TPCA result

Steps:

1. Compute sample mean as $\bar{\mathbf{X}}_{\mathbf{tv}} = (1/N)\sum_{k=1}^{N}\mathbf{X_{tv,k}}$
2. Compute the Fourier transform of samples:
   $\mathbf{X_{ftv,k}} = F(\mathbf{X_{tv,k}} - \bar{\mathbf{X}}_{\mathbf{tv}})$, $\mathbf{Y_{ftv}} = F(\mathbf{Y_{tv}} - \bar{\mathbf{X}}_{\mathbf{tv}})$
3. **For all slices** (ω_1, ω_2) **do**
   (1) Compute $\mathbf{G} = (1/(N-1))\sum_{k=1}^{N}(\mathbf{X_{ftv,k}}(\omega_1, \omega_2))(\mathbf{X_{ftv,k}}(\omega_1, \omega_2))^H$
   (2) SVD of $\mathbf{G}$ that $\mathbf{G} = \mathbf{U} \cdot \mathbf{S} \cdot \mathbf{V}^H$
   (3) $\hat{\mathbf{Y}}_{\mathbf{ftv}}(\omega_1, \omega_2) = \mathbf{U}^H \cdot \mathbf{Y_{ftv}}(\omega_1, \omega_2)$

end for

4. Inverse Fourier transform $\hat{\mathbf{Y}}_{\mathbf{tv}} = F^{-1}(\hat{\mathbf{Y}}_{\mathbf{ftv}})$
5. Compute real number space TPCA result as in equation (4)

3. THE PROPOSED METHOD

3.1. Tensorization

In this work, we introduce the TPCA method into TD. TPCA requires the input sample to be a tensor vector, but each pixel of HSI is a vector in real number space. Therefore, it is necessary to tensorize the HSI pixel vectors. Let HSI be expressed as $\mathcal{X} \in R^{D_1 \times D_2 \times D_3}$, where each pixel is $\mathbf{x} \in R^{D_3}$, and becomes $\mathbf{X_{tv}} \in \mathcal{R}_t^{D_3}$ after tensorization. The way of tensorization is to add spatial neighborhood information. We use the $n \times n$ circular-shift neighborhood of pixel x to form the tensor vector defined in the $n \times n$ second-order tensor space. More specifically, the formula of tensorization is:

$$\mathbf{X}_{\mathbf{tv},(i,j)}(\omega_1, \omega_2) = \mathbf{X}_{(i',j')} , \quad \omega_1, \omega_2 = 1, 2, ..., n \quad (5)$$

where $[\mathbf{X}_{(i,j)}]_k = \mathcal{X}_{i,j,k}$ and $i' = \mathrm{mod}(i - n + \omega_1, D_1) + 1$, $j' = \mathrm{mod}(j - n + \omega_2, D_2) + 1$.

3.2. Target detection based on TPCA

In Section 2, we introduced the steps of TPCA feature extraction. As the transformation tensor matrix $\mathbf{U_{tm}}$ obtained in equation (2) is orthogonal, we can use $\mathbf{U_{tm}}$ to transform the sample tensor vectors that have been transformed into the feature tensor space back into the original tensor space. Since the columns of $\mathbf{U_{tm}}$ are arranged in the order of PCs, we can use the first few columns of $\mathbf{U_{tm}}$ as the PC part and the remaining columns as the residual part. Obviously, these two tensor matrices are also orthogonal. By using these two parts as transformation matrices, respectively, we perform inverse transformation on the pixels of HSI that have been transformed into the feature space by TPCA, and we obtain the PC part $\mathcal{X}'$ and residual

part $\mathcal{E}$: $\mathcal{X}=\mathcal{X}'+\mathcal{E}$. It is assumed that the PC part $\mathcal{X}'$ contains the background of the original imagery and the target will appear in the residual part $\mathcal{E}$ because the target only takes up few pixels. In this way, we can perform target detection in the residual part $\mathcal{E}$ using target spectrum with its PC part removed to get rid of the background interference.

We first tensorize the original HSI and the priori target spectrum. Since the target spectrum contains no spatial information, the target spectrum is chosen as its own neighborhood spectrum when tensorized, that is

$$\mathbf{T}_{\mathbf{tv},(i,j)}(\omega_1,\omega_2)=\mathbf{t} \quad \omega_1,\omega_2=1,2,...,n. \tag{6}$$

Here $\mathbf{t}\in R^{D_3}$ is the priori target spectrum. Pixels $\mathbf{x_1},\mathbf{x_2},...,\mathbf{x_N}\in R^{D_3}$ are randomly selected as samples. Then, we tensorize the samples and the target spectrum, and perform TPCA to obtain a transformation tensor matrix $\mathbf{U_{tm}}$. Finally, its n_{PC} columns are removed to obtain a residual transformation tensor matrix $\mathbf{U}_{\mathcal{E},tm}$: $[\mathbf{U}_{\mathcal{E},tm}]_{i,j}=[\mathbf{U}_{tm}]_{i,j+n_{PC}}$.

For each pixel of HSI, the feature tensor vector Y_{tv} and the target spectral feature tensor vector $\hat{\mathbf{T}}_{\mathbf{tv}}=\mathbf{U}_{\mathbf{tm}}^{T}\circ(\mathbf{T}_{\mathbf{tv}}-\bar{\mathbf{X}}_{\mathbf{tv}})$ are also with the first n_{PC} rows removed. Then they are transformed into the original tensor space using the residual transformation tensor matrix $\mathbf{U}_{\mathcal{E},tm}$. Finally, they are restored to the real number space to obtain the residual part $\mathcal{E}$ and the residual target spectrum $\mathbf{t}_{\mathcal{E}}$. A variety of target detection algorithms are performed in the residual part $\mathcal{E}$ using the residual target spectrum $\mathbf{t}_{\mathcal{E}}$ to obtain the TD result. Similarly, the operation can be sped up by Fourier transform. Target detection algorithm based on TPCA using Fourier transform is presented as follows:

Algorithm 2: TPCA-based TD via the Fourier transform

Input: $\mathcal{X}\in R^{D_1\times D_2\times D_3}$: HSI, $\mathbf{t}\in R^{D_3}$: Priori target spectrum

Output: $\mathbf{Z}\in R^{D_1\times D_2}$: Target detection result

Steps:

1. Tensorize each pixel of $\mathcal{X}$ and $\mathbf{t}$ as in equation (5) and (6)
2. Randomly select part of pixels as samples
3. Compute the Fourier transform: $\mathbf{X}_{\mathbf{ftv}}=F(\mathbf{X}_{\mathbf{tv}}-\bar{\mathbf{X}}_{\mathbf{tv}})$, $\mathbf{T}_{\mathbf{ftv}}=F(\mathbf{T}_{\mathbf{tv}}-\bar{\mathbf{X}}_{\mathbf{tv}})$
4. For all slices (ω_1,ω_2) do
   (1) Compute $\mathbf{G}=(1/(N-1))\sum_{k=1}^{N}(\mathbf{X}_{\mathbf{ftv},\mathbf{k}}(\omega_1,\omega_2))(\mathbf{X}_{\mathbf{ftv},\mathbf{k}}(\omega_1,\omega_2))^H$
   (2) SVD of $\mathbf{G}$ that $\mathbf{G}=\mathbf{U}\cdot\mathbf{S}\cdot\mathbf{V}^H$
   (3) Compute $\mathbf{U}_{\mathcal{E}}$ as $[\mathbf{U}_{\mathcal{E}}]_{i,j}=[\mathbf{U}]_{i,j+n_{PC}}$
   (4) **For all pixels do**
      a) $\hat{\mathbf{X}}_{\mathbf{ftv}}(\omega_1,\omega_2)=\mathbf{U}^H\cdot\mathbf{X}_{\mathbf{ftv}}(\omega_1,\omega_2)$
      b) Compute $\hat{\mathbf{Y}}_{\mathbf{ftv}}(\omega_1,\omega_2)=\mathbf{U}_{\mathcal{E}}\cdot\hat{\mathbf{X}}_{\mathbf{ftv}}(\omega_1,\omega_2)$
   end for, thus we obtain (ω_1,ω_2) slice of residual part $E_{fti}(\omega_1,\omega_2)$
   (5) Compute $\hat{\mathbf{T}}_{\mathcal{E},ftv}(\omega_1,\omega_2)$ as $[\hat{\mathbf{T}}_{\mathcal{E},ftv}(\omega_1,\omega_2)]_i=[\mathbf{T}_{ftv}(\omega_1,\omega_2)]_{i+n_{PC}}$
   (6) Compute $\mathbf{T}_{\mathcal{E},ftv}(\omega_1,\omega_2)=\mathbf{U}_{\mathcal{E}}\cdot\hat{\mathbf{T}}_{\mathcal{E},ftv}(\omega_1,\omega_2)$
5. Inverse Fourier transform $\mathbf{T}_{\mathcal{E},tv}=F^{-1}(\mathbf{T}_{\mathcal{E},ftv})$, $E_{\mathcal{E},ti}=F^{-1}(E_{\mathcal{E},fti})$
6. Compute real number space TPCA result $t_{\mathcal{E}}$ and $\mathcal{E}$ as in equation (4)
7. Perform target detection with $\mathcal{E}$ and $\mathbf{t}_{\mathcal{E}}$ to obtain $\mathbf{Z}$

4. EXPERIMENTAL RESULTS

4.1. Synthetic data experiments

The background of the synthetic data comes from the Cuprite dataset collected by AVIRIS, containing a total of 224 bands, and 188 bands remained after removal of low SNR and water absorption bands. A region with size of 250×191 is selected as the background. The embedded target spectrum is Muscovite_GD111 from the USGS spectral set, with same bands removed as the Cuprite data as background. 5 targets of size 1×1 and 5 targets of size 2×2 are randomly embedded. Based on the linear mixing model, a synthetic subpixel target with spectrum $\mathbf{z}$ and a random abundance fraction f between 0 and 1 is generated by fractionally implanting a desired target with spectrum $\mathbf{t}$ in a given pixel of the background with spectrum $\mathbf{b}$ as follows:

$$\mathbf{z}=f\cdot\mathbf{t}+(1-f)\cdot\mathbf{b}.$$

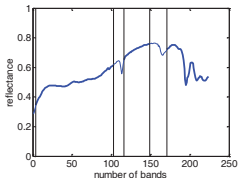
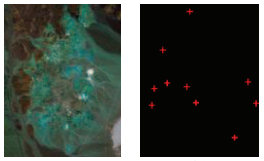


Fig. 1. Embedded target spectrum



(a) (b)

Fig. 2. Synthetic data (a)pseudo-color image (b) groundtruth

First we discuss the parameter choices in synthetic data experiments. With regard to the number of PCs, we consider the energy of the residual part after extracting PCs. The number of PCs to be chosen is the smallest n that satisfies the following equation:

$$\Delta_{\mathcal{E}}=\left(\left\|\mathcal{E}(n_{PC}=n)\right\|_2-\left\|\mathcal{E}(n_{PC}=n+1)\right\|_2\right)\Big/\left\|\mathcal{X}\right\|_2<0.5\% \tag{7}$$

where $\mathcal{E}(n_{PC}=n)$ is the residual part after removing n_{PC} PCs. The curve of $\left\|\mathcal{E}(n_{PC})\right\|_2$ varies with n_{PC} is shown in Fig. 3. According to equation (7), we find out that the number of PCs should be 4. It can also be qualitatively observed from Fig. 3 that the energy of the residual part changes little after the 4th PC. We use 3, 4, 5 PCs and perform TD in the TPCA residual part. Experiment results are shown in Fig. 4, proving not only our criterion is valid, but also our TPCA-based method is rather robust to the number of PCs.

Then we consider the impact of the sample rate on the TD results. We take 10% up to 90% of the dataset as training samples. The AUCs obtained for the ROC curves are given in Fig. 5. It indicates that the rate of training samples has no obvious effect on the detection results. Here we select 40% of the pixels in the image as training samples.

The size of the tensorization neighborhood will also affect the TD results. We select different neighborhoods sizes, set the number of PCs as 4 and sample rate as 40%, and perform TD in the residual part. Experimental results are displayed in Table 1. We can see that when the size of neighborhood is 3×3, the best detection result and the least running time are obtained.

The feature extraction algorithms used for comparative experiments are PCA and Tucker decomposition. The original imagery is also used for comparison. PC numbers of PCA and Tucker decomposition spectral dimension are set the same as TPCA. PC numbers of Tucker decomposition spatial dimensions are set to best results, 5 and 5. Sample rate of PCA is set the same as TPCA. Then TD is performed with CEM, ACE and AMF. The ROC curves are shown in Fig. 6. We repeat the experiments for 20 times and acquire AUCs in Table 2. Experimental results show that our TPCA-based algorithm has the best and the most stable detection results.

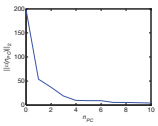
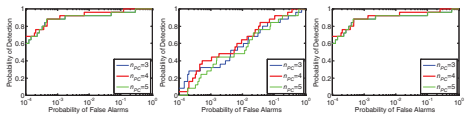


Fig. 3. The curve of $\left\|\mathcal{E}(n_{PC})\right\|_2$ varies with n_{PC}



(a) (b) (c)

Fig. 4. ROC curves with different number of PCs. (a) CEM, (b) ACE, (c) AMF.

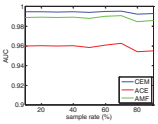
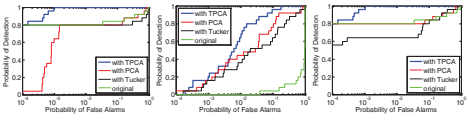


Fig. 5. AUCs with different sample rates



(a) (b) (c)

Fig. 6. ROC curves with different feature extraction methods. (a) CEM, (b) ACE, (c) AMF.

Table 1. AUCs and running time with different size of neighborhood

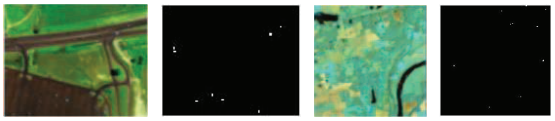
TD algorithm/ Running time	Size of neighborhood			
	3 × 3	5 × 5	7 × 7	9 × 9
CEM	**0.9945**	0.9891	0.9904	0.9843
ACE	**0.9602**	0.9376	0.9358	0.9148
AMF	**0.9891**	0.9779	0.9804	0.9674
Running Time (s)	**227.4857**	499.2109	1093.2675	2055.3608

Table 2. AUCs of ROC curves for TD on synthetic data repeated 20 times

TD algorithm	Feature extraction algorithm			
	origin	PCA	Tucker	TPCA
CEM	0.8951 ± 0.0301	0.9225 ± 0.0221	0.9005 ± 0.0187	**0.9995 ± 0.0011**
ACE	0.2416 ± 0.0938	0.8987 ± 0.0227	0.8879 ± 0.0395	**0.9845 ± 0.0073**
AMF	0.8935 ± 0.0304	0.9195 ± 0.0239	0.9031 ± 0.0259	**0.9991 ± 0.0022**

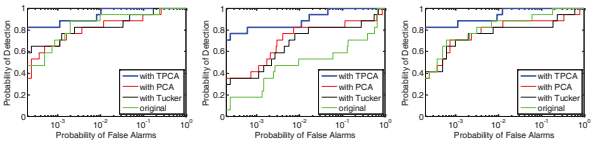
4.2. Real data experiments

In real data experiments, we use datasets collected by HYDICE and Hyperion hyperspectral sensors. Each contains 210 and 242 bands, respectively. After removal of low SNR and water absorption bands, there are 188 bands remaining in HYDICE data, and 155 bands remaining in Hyperion data. A region of 80× 100 pixels is used for HYDICE data, and Hyperion is with size of 150 × 150. Their pseudocolor images and groundtruth are shown in Fig. 7. The number of PCs is selected according to equation (7). PCA and Tucker decomposition are used as comparative experiments. And the original imagery is also used for comparison. We operate TD in the residual part with CEM, ACE, and AMF. Experiment results are presented in Fig. 8 and Table 3.



(a) (b) (c) (d)

Fig. 7. HYDICE data (a) pseudocolor image (b)groundtruth. Hyperion data (c) pseudocolor image (d) groundtruth



(a) (b) (c)

Fig. 8. ROC curves on HYDICE data. (a) CEM, (b) ACE, (c) AMF.

Table 3. AUCs of ROC curves for TD on real datasets

Dataset	TD algorithm	Feature extraction algorithm			
		origin	PCA	Tucker	TPCA
HYDICE	CEM	0.9847	0.9671	0.9916	**0.9988**
	ACE	0.8079	0.9041	0.9371	**0.9956**
	AMF	0.9847	0.9325	0.9579	**0.9986**
Hyperion	CEM	0.9709	0.9918	0.9281	**0.9988**
	ACE	0.7053	0.8524	0.9079	**0.9965**
	AMF	0.9643	0.9492	0.9165	**0.9824**

5. CONCLUSION

A novel preprocessing method for hyperspectral TD based on TPCA was proposed in this paper, which decomposed HSI into the sum of PC part and residual part. TD in residual part eliminated the background interference. Compared with other feature extraction methods, our method performed better in revealing the HSI physical property. Experimental results showed that various TD methods obtained better results in both synthetic and real datasets using our preprocessing procedure.

6. REFERENCES

[1] W. H. Farrand and J. C. Harsanyi, "Mapping the distribution of mine tailings in the Coeur d'Alene River Valley, Idaho, through the use of a constrained energy minimization technique," *Remote Sensing of Environment,* vol. 59, pp. 64-76, 1997.

[2] S. Kraut, L. Scharf and L. McWhorter, "Adaptive sub-space detectors," *IEEE Trans. Signal Process.,* vol. 49, no. 1, pp. 1–16, Jan. 2001.

[3] F. C. Robey, D. R. Fuhrmann, E. J. Kelly, and R. Nitzberg, "A CFAR adaptive matched filter detector," *IEEE Trans. Aerosp. Electron. Syst.,* vol. 28, no. 1, pp. 208–216, Jan. 1992.

[4] Y. Ren, L. Liao, S. J. Maybank, Y. Zhang and X. Liu, "Hyperspectral Image Spectral-Spatial Feature Extraction via Tensor Principal Component Analysis," *IEEE Geosci. Remote Sens. Lett.,* vol. 14, no. 9, pp. 1431-1435, Sept. 2017.

[5] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.,* vol. 51, no. 3, pp. 455–500, Mar. 2009.

[6] M. E. Kilmer and C. D. Martin, "Factorization strategies for third-order tensors," *Linear Algebra Appl.,* vol. 435, no. 3, pp. 641–658, 2011.