Hyperspectral Image Spectral-Spatial Feature Extraction via Tensor Principal Component Analysis

Yuemei Ren, Liang Liao, *Member, IEEE*, Stephen John Maybank, *Fellow, IEEE*, Yanning Zhang, *Senior Member, IEEE*, and Xin Liu

***Abstract*— We consider the tensor-based spectral-spatial feature extraction problem for hyperspectral image classification. First, a tensor framework based on circular convolution is proposed. Based on this framework, we extend the traditional principal component analysis (PCA) to its tensorial version tensor PCA (TPCA), which is applied to the spectral-spatial features of hyperspectral image data. The experiments show that the classification accuracy obtained using TPCA features is significantly higher than the accuracies obtained by its rivals.**

***Index Terms*— Feature extraction, hyperspectral image classification, principal component analysis (PCA), tensor model.**

I. Introduction

HYPERSPECTRAL imaging sensors collect hyperspectral images in the form of 3-D arrays, with two spatial dimensions representing the image width and height, and a spectral dimension describing the spectral bands, whose number is usually more than one hundred. Due to the redundancy of the raw representation, it is advantageous to design effective feature extractors to exploit the spectral information of hyperspectral images [1], [2]. For example, via a linear projection, after an eigenanalysis, principal component analysis (PCA) reduces a high-dimensional vector to a lower dimensional feature vector. By choosing the so-called principal components, the obtained feature vectors can retain most of the available information. But PCA, like many other vector-based counterparts, lacks, in its model, a prior mechanism to capture the spatial information in the relative positions of the pixels. This deficiency of vectorial models can be overcome using a tensorial representation of hyperspectral imagery.

Manuscript received October 31, 2016; revised February 3, 2017; accepted March 17, 2017. Date of publication July 26, 2017; date of current version August 25, 2017. This work was supported in part by the National Natural Science Foundation of China under Grant U1404607, in part by the Open Foundation Program of the Shaanxi Provincial Key Laboratory of Speech and Image Information Processing under Grant SJ2013001, in part by the High-end Foreign Experts Recruitment Program under Grant GDW20134100119, and in part by the Key Science and Technology Program of Henan under Grant 142102210557. Yuemei Ren, Liang Liao, and Stephen John Maybank contributed equally to this work. *(Corresponding author: Liang Liao.)*

Y. Ren is with the Department of Electronic Information Engineering, Henan Polytechnic Institute, Nanyang 473000, China, and also with the School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China.

L. Liao is with the School of Electronics and Information, Zhongyuan University of Technology, Zhengzhou 450007, China, and also with the School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China (e-mail: liaoliangis@126.com).

S. J. Maybank is with the Birkbeck College, University of London, London WC1E 7HX, U.K. (e-mail: sjmaybank@dcs.bbk.ac.uk).

Y. Zhang is with the School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China.

X. Liu is with the Information Center of Yellow River Conservancy Commission, Zhengzhou 450000, China.

Digital Object Identifier 10.1109/LGRS.2017.2686878

There exist many different tensor models derived from different perspectives. Tensor models are essentially the extensions of traditional vector models. In the recent years, tensor-based approaches have been successfully applied in many different areas, including image analysis, video processing [3], [4], and remote sensing imagery analysis [5], [6]. For example, Zhang *et al.* [6] reported a tensor discriminative locality alignment method, called tensor discriminative locality alignment (TDLA), to extract features from hyperspectral images. Zhong *et al.* [7] proposed a tensorial discriminant extractor called local tensor discriminant analysis (LTDA) to obtain spectral-spatial features from hyperspectral images for image classification.

The aforementioned works have shown the superior representation capability of a tensor-based multilinear algebra, compared with that of the traditional matrix algebra. For example, Kilmer and Martin [8] introduced the t-product model and defined a generalization of matrix multiplication for tensors in the form of 3-D arrays (tensors of order three). The generalized matrix multiplication is based on the circular convolution operation and can be implemented more efficiently via the Fourier transform. The t-product model is further developed using the concepts of tubal scalar (also known as tubal fiber), frontal slice, array folding and unfolding, and so on, to establish its connection to traditional linear algebra [9], [10]. This development makes it possible to generalize all the classical algorithms formulated in linear algebra.

Inspired by the recently reported "t-product" tensor model, we propose a tensor-based spectral-spatial feature extractor for classifying the pixels of a hyperspectral image. First, a novel straightforward tensor algebraic framework is proposed. In the framework, the "t-product" tensors are confined to the same size and, therefore, form an algebraic tensor ring. The algebraic framework combines the "multiway" merits of high-order arrays and the "multiway" intuitions of traditional matrices, since the "t-product" tensors serve as the entries of our proposed tensor-vectors or tensor-matrices. This algebraic framework is backward compatible with the traditional linear algebra based on nontensorial vectors and matrices. With the help of the proposed tensor algebra, we extend PCA to its tensorial counterpart tensor PCA (TPCA), which has a prior mechanism to exploit the spatial information of images. We demonstrate, in our experiments, on some publicly available images that the TPCA outperforms PCA and some other vector or tensorial feature extractors in terms of classification accuracy.

This letter is organized as follows. In Section II, we discuss the tensor algebraic framework. In Section III, we propose a tensorial feature extractor called TPCA and its fast version via the Fourier transform. In Section IV, we present the experimental results and analysis. Finally, we conclude this letter in Section V.

II. Tensor Algebra

In this section, we extend the "t-product" model [8]–[10] to a novel tensor algebraic framework, which is backward compatible to the traditional matrix algebra. Without losing generality, the tensors presented in this letter are second order. It is easy to extend the method to higher order tensors.

Before extending the "t-product" model, we first confine the "t-product" tensors to the same size. For the second-order tensors, the tensor size is defined by $m \times n$. In the following discussion, given the second-order array X (such as tensor, matrix, and so on), $[X]_{i,j}$ denotes its (i, j)th entry. If X is the first order, $[X]_i$ denotes its ith entry. Then, some denotations and definitions on the "t-product tensor" are summarized as follows [8]–[10].

Tensor Addition: Given tensors x_t and y_t of the same size, the sum $c_t = x_t + y_t$ is a tensor of the same size such that $[c_t]_{i,j} = [x_t]_{i,j} + [y_t]_{i,j}, \forall i, j$.

Tensor Multiplication: Given tensors x_t and y_t of size $m \times n$, the product $d_t = x_t \circ y_t$ is defined by the result of the circular convolution of x_t and y_t, such that

$$[d]_{i,j} = \sum_{k_1=1}^{m} \sum_{k_2=1}^{n} [x_t]_{k_1,k_2} [y_t]_{\mathrm{mod}((i-k_1),m)+1,\mathrm{mod}((j-k_2),n)+1}.$$

Given tensors x_t and y_t of the same size, their product can be computed efficiently via the fast Fourier transform and the inverse fast Fourier transform because of the following Fourier transform theorem.

1) *Fourier Transform:* Given tensors x_t, y_t, $d_t = x_t \circ y_t$ and their Fourier transforms $F(x_t)$, $F(y_t)$, and $F(d_t)$, the following equation holds $[F(d_t)]_{i,j} = [F(x_t)]_{i,j} \cdot [F(y_t)]_{i,j} \quad \forall i, j$.
   By the virtue of the Fourier transform, the $(mn)^2$ scalar multiplications of the circular convolution are reduced to the mn-independent scalar multiplications in the Fourier domain. Thus, the Fourier transform can be employed to speed up a convolution-based algorithm.
2) *Zero Tensor:* The zero tensor z_t is a tensor whose entries are all 0, namely $[z_t]_{i,j} \equiv 0 \quad \forall i, j$.
3) *Identity Tensor:* The identity tensor e_t is a tensor satisfying $[e_t]_{1,1} = 1$ and $[e_t]_{i,j} = 0$ if $(i, j) \neq (1, 1)$.

It is not difficult to prove that the tensors of size $m \times n$ defined previously form an algebraic ring R. The algebraic operations of addition and multiplication in R are backward compatible with the analogous operations in the field $\mathbb{R}$ of real numbers. Based on the above-mentioned "t-product" definitions, we extend the "t-product" model to a novel straightforward algebraic framework by the following definitions.

Definition 1 (Scalar Multiplication): Given tensor x_t and scalar α, the product $d_t = \alpha x_t$ is a tensor of the same size as x_t such that $[d_t]_{i,j} = \alpha[x_t]_{i,j} \quad \forall i, j$.

Definition 2 (Vectors and Matrices of Tensors): A vector of tensors is a list of elements of R. A matrix of tensors is an array of elements of R.

In our tensor algebraic framework, the mathematical operations between the elements of the vectors of tensors or of the matrices of tensors comply with the operations defined in R. Some definitions of the mathematical manipulations of the vectors and matrices of tensors are given as follows.

Definition 3 (Tensor Matrix Multiplication): Given $X_{\mathrm{tm}} \in R^{D_1 \times D_2}$ and $Y_{\mathrm{tm}} \in R^{D_2 \times D_3}$, their product $C_{\mathrm{tm}} = X_{\mathrm{tm}} \circ Y_{\mathrm{tm}} \in R^{D_1 \times D_3}$ is a new tensor matrix, such that $[C_{\mathrm{tm}}]_{i,j} = \sum_{k=1}^{D_2} [X_{\mathrm{tm}}]_{i,k} \circ [Y_{\mathrm{tm}}]_{k,j}, \forall i, j$.

Definition 4 (Identity Matrix of Tensors): The identity matrix of tensors E_{tm} is a matrix such that $[E_{\mathrm{tm}}]_{i,j} = e_t$ if $i = j$, otherwise $[E_{\mathrm{tm}}]_{i,j} = z_t$.

Definition 5 (Tensor Transposition): Given tensor x_t, its transpose $x_t^\top$ is a tensor of the same size, satisfying $[x_t^\top]_{i,j} = [x_t]_{\mathrm{mod}(1-i,m)+1,\mathrm{mod}(1-j,n)+1}, \forall i, j$.

Definition 6 (Tensor Matrix Transposition): Given $X_{\mathrm{tm}} \in R^{D_1 \times D_2}$, $X_{\mathrm{tm}}^\top \in R^{D_2 \times D_1}$ is a new tensor matrix satisfying $[X_{\mathrm{tm}}^\top]_{i,j} = [X_{\mathrm{tm}}]_{j,i}^\top$, $\forall i, j$, where $[X_{\mathrm{tm}}]_{j,i}^\top$ is the transpose of tensor $[X_{\mathrm{tm}}]_{j,i}$, as defined by Definition 5.

Definition 7 (Orthonormal Matrix of Tensors): If the matrix X_{tm} of tensors satisfies $X_{\mathrm{tm}}^\top \circ X_{\mathrm{tm}} = E_{\mathrm{tm}}$, we call X_{tm} an orthonormal matrix of tensors.

Definition 8 (Singular Value Decomposition of a Square Tensor Matrix): Each matrix $G_{\mathrm{tm}} \in R^{D \times D}$ has a singular value decomposition $G_{\mathrm{tm}} = U_{\mathrm{tm}} \circ S_{\mathrm{tm}} \circ V_{\mathrm{tm}}^\top$ such that $U_{\mathrm{tm}} \in R^{D \times D}$ and $V_{\mathrm{tm}} \in R^{D \times D}$ are both orthonormal tensor matrices and $S_{\mathrm{tm}} \in R^{D \times D}$ is a diagonal tensor matrix and $[S_{\mathrm{tm}}]_{i,i}^\top = [S_{\mathrm{tm}}]_{i,i}, \forall i$.

To speed up the computations with the vectors and matrices of tensors, we extend the Fourier transform of a tensor to the Fourier transform of a vector or matrix of tensors.

Definition 9 (Fourier Transform of a Tensor Matrix): Given a tensor matrix X_{tm}, let its Fourier transform be $F(X_{\mathrm{tm}})$, such that $[F(X_{\mathrm{tm}})]_{i,j} = F([X_{\mathrm{tm}}]_{i,j}), \forall i, j$.

The Fourier transform of a tensor matrix is defined by the Fourier transform on its matrix tensorial entities. This definition gives a mechanism to decompose a tensor matrix in the Fourier domain to a range of separate traditional matrix.

To have the mechanism, we define the following slice of a tensor matrix by the index of its tensorial entity (ω_1, ω_2), for all $\omega_1 = 1, \ldots, m$ and $\omega_2 = 1, \ldots, n$.

Definition 10 (Slice of a Tensor Matrix): Given $X_{\mathrm{tm}} \in R^{D_1 \times D_2}$, let its slice by index (ω_1, ω_1) be $X_{\mathrm{tm}}(\omega_1, \omega_2)$ such that $X_{\mathrm{tm}}(\omega_1, \omega_2) \in \mathbb{R}^{D_1 \times D_2}$ and

$$[X_{\mathrm{tm}}(\omega_1, \omega_2)]_{i,j} = [[X_{\mathrm{tm}}]_{i,j}]_{\omega_1,\omega_2} \quad \forall i, j, \omega_1, \omega_2.$$

Let the slice of $X_{ftm} \doteq F(X_{\mathrm{tm}})$ by index (ω_1, ω_2) be $X_{ftm}(\omega_1, \omega_2)$, such that $X_{ftm}(\omega_1, \omega_2) \in \mathbb{C}^{D_1 \times D_2}$ and

$$[X_{ftm}(\omega_1, \omega_2)]_{i,j} = [[X_{ftm}]_{i,j}]_{\omega_1,\omega_2} \quad \forall i, j, \omega_1, \omega_2.$$

Tensor vector is a special case of tensor matrix, and thus, the slicing of a tensor vector complies with the definition of the slicing of a tensor matrix.

By the virtue of Definitions 9 and 10, given the tensor size of $m \times n$, a mathematical tensor matrix operation can be decomposed to and efficiently computed by $m \times n$ separate traditional matrix operations in the Fourier domain.

III. Tensor Principal Component Analysis

A. PCA

Traditional PCA can be briefly described as follows—given $x_1, \ldots, x_N \in \mathbb{R}^D$ and $\bar{x} = (1/N) \sum_{k=1}^{N} x_k$, the covariance matrix G is given by $G = (1/N - 1) \sum_{k=1}^{N} (x_k - \bar{x})(x_k - \bar{x})^\top$.

The singular value decomposition of G is computed such that $G = U \cdot S \cdot V^\top$, where both U and V are $D \times D$ orthonormal matrices, in that $U^\top \cdot U = V^\top \cdot V = I_{D\times D}$ and $S = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_D)$, where $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \lambda_D \geqslant 0$.

Then, given $y \in \mathbb{R}^D$, its PCA features are given by $\hat{y} = U^\top \cdot (y - \bar{x})$. To reduce the dimension of $\hat{y}$ from D to d ($d < D$), the last $(D - d)$ entries of $\hat{y}$ are simply discarded.

B. TPCA

The TPCA is a straightforward tensorial extension of the traditional PCA—given $X_{\mathrm{tv},1}, X_{\mathrm{tv},2}, \ldots, X_{\mathrm{tv},N} \in R^D$ and $\bar{X}_{\mathrm{tv}} = (1/N)\sum_{k=1}^{N} X_{\mathrm{tv},k}$, tensor matrix $G_{\mathrm{tm}} \in R^{D\times D}$ is defined by

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

Its singular value decomposition is computed as in Definition 8, namely

$$G_{\mathrm{tm}} = U_{\mathrm{tm}} \circ S_{\mathrm{tm}} \circ V_{\mathrm{tm}}^\top. \quad (2)$$

Then, given a queryaq tensor vector $Y_{\mathrm{tv}} \in R^D$, its tensor feature vector $\hat{Y}_{\mathrm{tv}}$ is given by

$$\hat{Y}_{\mathrm{tv}} = U_{\mathrm{tm}}^\top \circ (Y_{\mathrm{tv}} - \bar{X}_{\mathrm{tv}}). \quad (3)$$

We call $\hat{Y}_{\mathrm{tv}}$ the TPCA vector of Y_{tv}. To accommodate the traditional algorithms, which only deal with traditional vectors, with the help of the tensor slicing operation, we propose a mapping $\delta(\cdot)$ to transform $\hat{Y}_{\mathrm{tv}} \in R^D$ to $\delta(\hat{Y}_{\mathrm{tv}}) \in \mathbb{R}^D$—let the tensor size be $m \times n$, $\delta(\hat{Y}_{\mathrm{tv}})$ is given by

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

We call $\hat{y} = \delta(U_{\mathrm{tm}}^\top \circ (Y_{\mathrm{tv}} - \bar{X}_{\mathrm{tv}}))$ the traditional TPCA feature vector, which can be conveniently employed by traditional nontensorial algorithms. To reduce the dimension of $\hat{y}$ from D to d, the last $(D - d)$ entries are simply discarded.

TPCA is organized in Algorithm 1.

Algorithm 1 Tensorial Principal Component Analysis

Input: Query tensor vector $Y_{tv} \in R^D$ and N training tensor vectors $X_{tv,1}, \ldots, X_{tv,N} \in R^D$.
Output: non-tensorial TPCA feature vector $y \in \mathbb{R}^D$
Compute G_{tm} and $\bar{X}_{tv}$ as in equation (1).
Compute U_{tm} as in equation (2).
Compute $\hat{Y}_{tv}$ as in equation (3).
return $y \leftarrow \delta(\hat{Y}_{tv})$.

C. Fast TPCA via the Fourier Transform

Note that, with the help of the slicing operation, (2) can be computed much more efficiently via a series of traditional SVDs computed in the Fourier domain. More specifically, given the tensor matrix G_{tm}, and $\Omega \doteq \{(\omega_1, \omega_2) : \omega_1 = 1, \ldots, m,\ \omega_2 = 1, \ldots, n\}$, (2) can be computed as in the following pseudocode.

Algorithm 2 Fast Tensorial SVD via the Fourier Transform

$G_{ftm} \leftarrow F(G_{tm})$.
for all $(\omega_1, \omega_2) \in \Omega$ **do**
  Compute the traditional SVD of slice $G_{ftm}(\omega_1, \omega_2)$ such that $G_{ftm}(\omega_1, \omega_2) = U \cdot S \cdot V^H$, where V^H is the Hermitian transpose of the matrix V.
  $U_{ftm}(\omega_1, \omega_2) \leftarrow U$, $S_{ftm}(\omega_1, \omega_2) \leftarrow S$, $V_{ftm}(\omega_1, \omega_2) \leftarrow V$.
end for
return $U_{tm} \leftarrow F^{-1}(U_{ftm})$, $S_{tm} \leftarrow F^{-1}(S_{ftm})$, $V_{tm} \leftarrow F^{-1}(V_{ftm})$.

Furthermore, the Fourier transform and the slicing operation are also applicable to (1)–(3). Thus, the whole TPCA procedure can be implemented by a series of traditional PCAs in the Fourier domain. More specifically, with the tensors of size $m \times n$, the fast implementation of the whole TPCA can be carried as follows.

Algorithm 3 Fast TPCA via the Fourier Transform

$\bar{X}_{tv} \leftarrow (1/N)\sum_{i=1}^{N} X_{tv,i}$.
$Y_{ftv} \leftarrow F(Y_{tv} - \bar{X}_{tv})$, $X_{ftv,i} \leftarrow F(X_{tv,i} - \bar{X}_{tv})$, $\forall i$.
for all $(\omega_1, \omega_2) \in \Omega$ **do**
  $G \leftarrow \frac{1}{N-1}\sum_{i=1}^{N} X_{ftv,i}(\omega_1, \omega_2)\left(X_{ftv,i}(\omega_1, \omega_2)\right)^H$.
  Compute SVD of G, such that $G = U \cdot S \cdot V^H$.
  $\hat{Y}_{ftv}(\omega_1, \omega_2) \leftarrow U^H \cdot Y_{ftv}(\omega_1, \omega_2)$.
end for
return $y \leftarrow \delta(F^{-1}(\hat{Y}_{ftv}))$.

D. Computational Complexity

By above-mentioned speeding-up scheme, a tensorial operation (such as TPCA or the tensor SVD) is decomposed to $|\Omega|$ separate corresponding traditional nontensorial operations computed in the Fourier domain. If the traditional operation in the Fourier involves one flop, the corresponding tensorial operation needs $|\Omega|$ flops. If the cost of Fourier transform is ignored, the computational complexity of TPCA or tensor SVD is proportional to $|\Omega|$. A complex number is a tensor with $|\Omega| = 1$, and therefore, the computational complexity of PCA (or SVD) is $O(1)$. The computational complexity of TPCA (or tensor SVD) is $O(|\Omega)|$.

IV. Experiments

Two publicly available image sets are employed in our experiments. One is the Indian Pines scene. It consists of 145×145 pixels and 200 spectral bands. The groundtruth of the Indian Pines scene is based on 16 classes. Another image set is the Pavia University scene. It consists of 610×340 pixels with 103 spectral bands. Its groundtruth is based on nine classes.

A. Tensorization

We employ a straightforward approach to tensorize the given images—given a hyperspectral image $Z \in \mathbb{R}^{D_1\times D_2\times D_3}$, let $Z_{i,j} \in \mathbb{R}^{D_3}$ denote the traditional vector representation of the (i, j)th pixel such that $[Z_{i,j}]_k = [Z]_{i,j,k}, \forall i, j, k$. Then, we

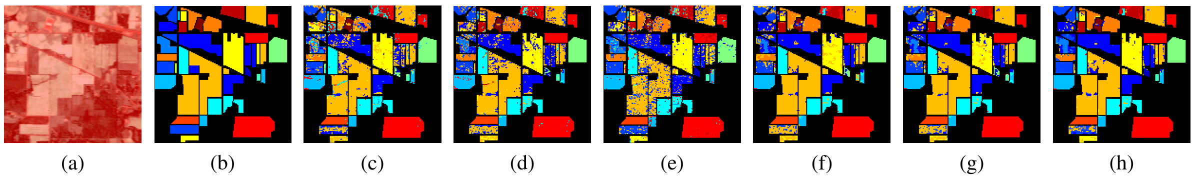

Fig. 1. Classification maps obtained using RF with different types of features on the Indian Pines images. (a) Indian Pines scene. (b) Groundtruth. (c) Original. (d) PCA. (e) LDA. (f) TDLA. (g) LTDA. (h) TPCA.

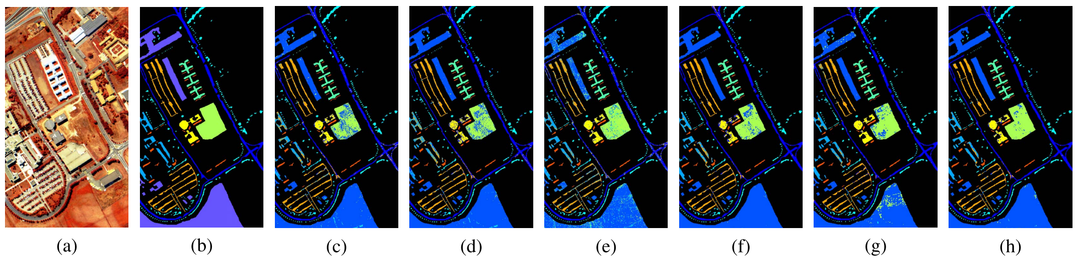

Fig. 2. Classification maps obtained using RF with different types of features on the Pavia University images. (a) Pavia University scene. (b) Groundtruth. (c) Original. (d) PCA. (e) LDA. (f) TDLA. (g) LTDA. (h) TPCA.

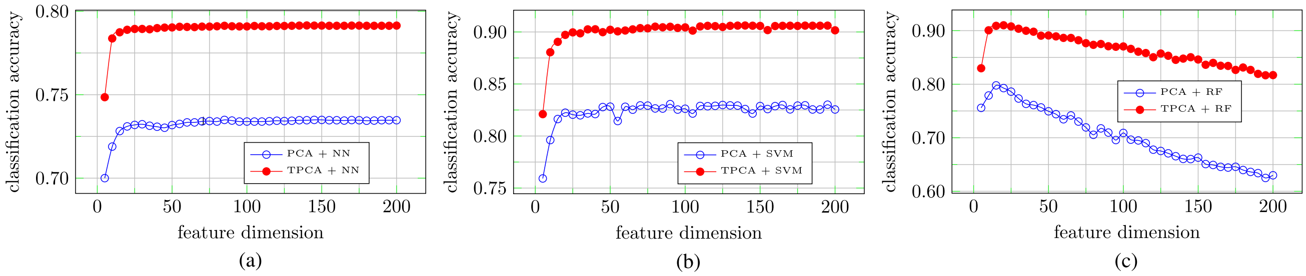

Fig. 3. Classification accuracy curves obtained using NN, SVM, and RF with PCA/TPCA on the Indian Pines image. (a) NN. (b) SVM. (c) RF.

tensorize each pixel sample to a tensor vector whose entries are 3×3 tensors, transforming traditional vector $Z_{i,j} \in \mathbb{R}^{D_3}$ to tensor vector $Z_{\mathrm{tv},(i,j)} \in R^{D_3}$ $\forall i, j$. More specifically, $Z_{\mathrm{tv},i,j}$ is defined via its slice $Z_{\mathrm{tv},(i,j)}(\omega_1, \omega_2) \in \mathbb{R}^{D_3}$ as

$$Z_{\mathrm{tv},(i,j)}(\omega_1, \omega_2) = Z_{i',j'} \quad \forall i, j, \omega_1, \omega_2$$

where $i' \doteq \mathrm{mod}(i + \omega_1 - 3, D_1) + 1$ and $j' \doteq \mathrm{mod}\,(j + \omega_2 - 3, D_2) + 1$.

We call $\{(i', j') : \omega_1 = 1, 2, 3 \text{ and } \omega_2 = 1, 2, 3\}$ the 3×3 circular-shift neighborhood of (i, j). Given a hyperspectral image with N training pixels and D-bands ($D = D_3$), we denote the acquired tensor vectors in order by $X_{\mathrm{tv},1}, \ldots, X_{\mathrm{tv},N} \in R^D$, which are inputs in Algorithm 1.

B. Experimental Results

We report the overall accuracies (OAs) and the κ coefficient of the classification results obtained by five feature extractors and three classifiers. Higher values of OA and κ indicate a better classification result [11]. The employed classifiers are the nearest neighbor (NN), support vector machine (SVM), and random forest (RF). The Gaussian radial basis function kernel is employed for SVM and, in the experiments, the Gaussian parameter $\sigma \in \{2^i\}_{i=-15}^{10}$ and the regularization parameter $C \in \{2^i\}_{i=-5}^{15}$ are trained from the fivefold cross validation.

The extractors include two classical vector-based algorithms—PCA and linear discriminant analysis (LDA), and three state-of-the-art tensor-based algorithms—TDLA [6], LTDA [7], and TPCA (ours). We also give the quantitative results obtained by the original raw vector representation (denoted as "original") as a comparison baseline. For PCA, TPCA, and LDA, we use a range of values for the feature dimension D, namely $D = 5, 10, \cdots, D_{\max}$ with $D_{\max} = 200$ for the Indian Pines image and $D_{\max} = 100$ for the Pavia University image. The highest classification accuracy for the range of values of D is reported. In the experiments, 10% of the pixels of interests are uniformly randomly chosen as the training samples, the rest of the pixels are chosen as query samples. The classification experiment is repeated independently ten times and the average OA and κ is recorded.

The quantitative comparison of the classification results is given in Table I. The third and the fourth column of Table I show the classification result of OA and κ obtained on the Indian Pines image. It is clear that the results obtained by the tensorial features (TDLA, LTDA, and TPCA) are better than those obtained by the vectorial features (PCA and LDA). Furthermore, among all the extractors, TPCA always yields the highest OA and the largest κ, outperforming its tensorial rivals TDLA and LTDA. For a comparison with its vectorial counterpart, the OA obtained by TPCA is about 6%–11% higher than that obtained by PCA—with RF, the TPCA accuracy is 91.01% compared with the baseline 76.78% and the PCA accuracy 79.78%. The Indian Pines image and the University of Pavia image, the groundtruth, and the classification maps with the highest classification accuracies for the different experimental settings are shown, respectively, in Figs. 1 and 2. It is clear that TPCA yields the best classification.

Since PCA and TPCA have a similar structure, to compare their performances, the OA curves of PCA and TPCA over varying feature dimension, obtained by classifiers NN, SVM,

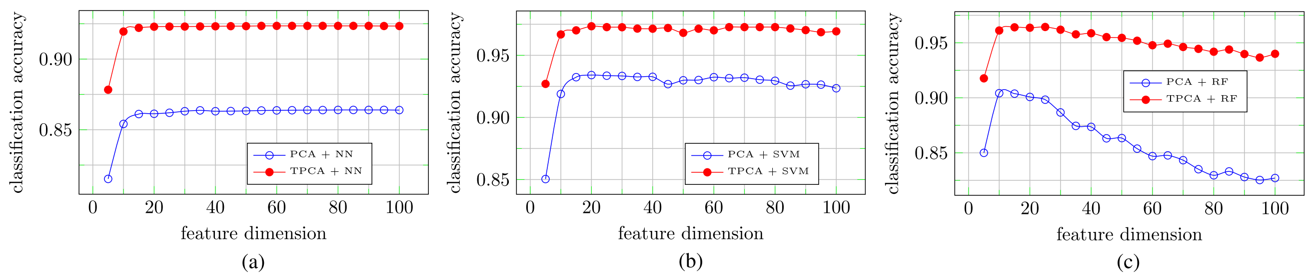

Fig. 4. Classification accuracy curves obtained using NN, SVM, and RF with PCA/TPCA on the Pavia University image. (a) NN. (b) SVM. (c) RF.

TABLE I
CLASSIFICATION ACCURACY COMPARISON OBTAINED ON THE INDIAN PINES IMAGE AND THE PAVIA UNIVERSITY IMAGE

Classifier	Extractor	Indian Pines		Pavia	
		OA	κ	OA	κ
NN	original	73.43	0.6972	86.35	0.8170
	PCA	73.49	0.6977	86.40	0.8178
	LDA	71.54	0.6642	85.04	0.8037
	TDLA[6]	74.21	0.7209	89.27	0.8568
	LTDA[7]	75.14	0.7216	90.48	0.8718
	TPCA[ours]	**79.15**	**0.7624**	**92.35**	**0.8979**
SVM	original	82.95	0.8053	93.60	0.9143
	PCA	83.06	0.8065	93.41	0.9123
	LDA	79.53	0.7673	89.76	0.8606
	TDLA[6]	83.51	0.8168	96.14	0.9498
	LTDA[7]	85.68	0.8372	94.91	0.9324
	TPCA[ours]	**90.62**	**0.8930**	**97.34**	**0.9648**
RF	original	76.78	0.7315	89.79	0.8624
	PCA	79.78	0.7667	90.42	0.8712
	LDA	76.59	0.7353	87.74	0.8395
	TDLA[6]	84.96	0.8237	94.32	0.9245
	LTDA[7]	86.57	0.8475	93.03	0.9178
	TPCA[ours]	**91.01**	**0.8969**	**96.44**	**0.9526**

and RF on the Indian Pines image, are given in Fig. 3. Fig. 3 shows that, no matter which classifier and feature dimension are chosen, TPCA always outperforms PCA.

The classification results obtained using a range of classifiers and feature extractors on the Pavia University image are given in the last two columns of Table I. The visual classification maps obtained by RF with different extractors are shown in Fig. 2. The maps with the highest classification accuracies for the different experimental settings are shown. To compare the performances of PCA and TPCA, the OA curves of PCA and TPCA for changing feature dimension, obtained by classifiers NN, SVM, and RF on the Pavia University image, are given in Fig. 4. From Table I and Figs. 2 and 4, a similar observation that TPCA outperforms its rivals can be drawn, supporting our claims for TPCA.

V. CONCLUSION

A novel tensor-based feature extractor called TPCA is proposed for hyperspectral image classification. First, we propose a new tensor matrix algebraic framework, which combines the merits of the recently emerged t-product model, which is based on the circular convolution, and the traditional matrix algebra. With the help of the proposed algebraic framework, we extend the traditional PCA algorithm to its tensorial variant TPCA. To speed up the tensor-based computing of TPCA, we also propose a fast TPCA for which the calculations are conducted in the Fourier domain. With a tensorization scheme via a neighborhood of each pixel, each sample is defined by a tensorial vector whose entries are all the second-order tensors and TPCA can effectively extract the spectralspatial information in a given hyperspectral image. To make TPCA applicable to traditional vector-based classifiers, we design a straightforward but effective approach to transform TPCA's output tensor vector to a traditional vector. Experiments to classify the pixels of two publicly available benchmark hyperspectral images show that TPCA outperforms its rivals, including PCA, LDA, TDLA, and LDLA in term of classification accuracy.

ACKNOWLEDGMENT

In letter, L. Liao and S. Maybank are primarily responsible for the algebraic framework of tensorial matrices and tensor principal component analysis. Y. Ren is primarily responsible for the experiments on the hyperspectral data. L. Liao also contributed significantly to the experiments.

REFERENCES

[1] J. M. Bioucas-Dias, A. Plaza, G. Camps-Valls, P. Scheunders, N. M. Nasrabadi, and J. Chanussot, "Hyperspectral remote sensing data analysis and future challenges," *IEEE Geosci. Remote Sens. Mag.*, vol. 1, no. 2, pp. 6–36, Jun. 2013.
[2] M. He and S. Mei, "Dimension reduction by random projection for endmember extraction," in *Proc. 5th IEEE Conf. Ind. Electron. Appl. (ICIEA)*, Jun. 2010, pp. 2323–2327.
[3] T. Gkamas, F. Renard, C. Heinrich, and S. Kremer, "A fourth order tensor statistical model for diffusion weighted MRI," in *Proc. Int. Conf. Pattern Recognit. Appl. Methods*, vol. 2. 2015, pp. 277–282.
[4] D. Ai, G. Duan, X. Han, and Y. W. Chen, "Generalized N-dimensional independent component analysis and its application to multiple feature selection and fusion for image classification," *Neurocomputing*, vol. 103, pp. 186–197, Mar. 2013.
[5] M. Tao, F. Zhou, Y. Liu, and Z. Zhang, "Tensorial independent component analysis-based feature extraction for polarimetric SAR data classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 5, pp. 2481–2495, May 2015.
[6] L. Zhang, L. Zhang, D. Tao, and X. Huang, "Tensor discriminative locality alignment for hyperspectral image spectral–spatial feature extraction," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 1, pp. 242–256, Jan. 2013.
[7] Z. Zhong *et al.*, "Discriminant tensor spectral–spatial feature extraction for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 5, pp. 1028–1032, May 2015.
[8] M. E. Kilmer and C. D. Martin, "Factorization strategies for third-order tensors," *Linear Algebra Appl.*, vol. 435, no. 3, pp. 641–658, 2011.
[9] K. Braman, "Third-order tensors as linear operators on a space of matrices," *Linear Algebra Appl.*, vol. 433, no. 7, pp. 1241–1253, 2010.
[10] M. E. Kilmer, K. Braman, N. Hao, and R. C. Hoover, "Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging," *SIAM J. Matrix Anal. Appl.*, vol. 34, no. 1, pp. 148–172, 2013.
[11] R. W. Fitzgerald and B. G. Lees, "Assessing the classification accuracy of multisource remote sensing data," *Remote Sens. Environ.*, vol. 47, no. 3, pp. 362–368, 1994.