Sparse Alignment for Robust Tensor Learning

Zhihui Lai, Wai Keung Wong, Yong Xu, *Member, IEEE*, Cairong Zhao, and Mingming Sun

***Abstract*— Multilinear/tensor extensions of manifold learning based algorithms have been widely used in computer vision and pattern recognition. This paper first provides a systematic analysis of the multilinear extensions for the most popular methods by using alignment techniques, thereby obtaining a general tensor alignment framework. From this framework, it is easy to show that the manifold learning based tensor learning methods are intrinsically different from the alignment techniques. Based on the alignment framework, a robust tensor learning method called sparse tensor alignment (STA) is then proposed for unsupervised tensor feature extraction. Different from the existing tensor learning methods, L_1- and L_2-norms are introduced to enhance the robustness in the alignment step of the STA. The advantage of the proposed technique is that the difficulty in selecting the size of the local neighborhood can be avoided in the manifold learning based tensor feature extraction algorithms. Although STA is an unsupervised learning method, the sparsity encodes the discriminative information in the alignment step and provides the robustness of STA. Extensive experiments on the well-known image databases as well as action and hand gesture databases by encoding object images as tensors demonstrate that the proposed STA algorithm gives the most competitive performance when compared with the tensor-based unsupervised learning methods.**

***Index Terms*— Feature extraction, local alignment, manifold learning, sparse representation, tensor learning.**

Manuscript received December 15, 2012; revised October 10, 2013; accepted December 8, 2013. Date of publication January 13, 2014; date of current version September 15, 2014. This work was supported in part by the Natural Science Foundation of China under Grant 61203376, Grant 61375012, Grant 61203247, Grant 61005005, Grant 61071179, Grant 61125305, Grant 61170077, Grant 61362031, Grant 61332011, Grant 61370163, and Grant 61263032, in part by the General Research Fund of Research Grants Council of Hong Kong under Project 531708, in part by the China Postdoctoral Science Foundation under Project 2012M510958 and Project 2013T60370, in part by the Guangdong Natural Science Foundation under Project S2012040007289, and in part by the Shenzhen Municipal Science and Technology Innovation Council under Grant JC201005260122A, Grant JCYJ20120613153352732, Grant JCYJ20120613134843060, and Grant JCYJ20130329152024199.

Z. Lai is with the Bio-Computing Research Center, Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen 518055, China, and also with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518055, China (e-mail: lai_zhi_hui@163.com).

W. K. Wong is with the Institute of Textiles and Clothing, The Hong Kong Polytechnic University, Hong Kong, and also with Shenzhen Research Institute, Shenzhen 518055, China (e-mail: calvin.wong@polyu.edu.hk).

Y. Xu is with the Bio-Computing Research Center, Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen 518055, China (e-mail: yongxu@ymail.com).

C. Zhao is with the Department of Computer Science and Technology, Tongji University, Shanghai 201804, China (e-mail: zhaocairong@126.com).

M. Sun is with the Department of Computer Science and Technology, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: sunmingming@gmail.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2013.2295717

I. Introduction

In recent years, high-order tensor learning methods have been widely used in the fields of computer vision, pattern recognition, and machine learning to deal with the curse of dimensionality problem. The classical dimensionality reduction method principal component analysis (PCA) [1] was first extended to second-order cases (i.e., 2-D PCA (2-DPCA) [2] and generalized low-rank approximations of matrices (GLRAM) [3]), and then to the high-order case, (i.e., multilinear PCA (MPCA) [4] and its uncorrelated variation [5]). Similarly, linear discriminant analysis (LDA) [6] was also extended to 2-D LDA (2-DLDA) [7], [8] and multilinear discriminant analysis (MDA) [9]. By using the differential scatter discriminant criterion (DSDC) [10], Tao *et al.* [11] proposed the general tensor discriminant analysis (GTDA) for gait recognition. With the maximum margin criterion (MMC) [12], Laplacian bidirectional MMC (LBMMC) [13] and tensor MMC (TMMC) [14] were proposed for object recognition. The high-order tensor-based methods performed better than the classical ones in feature extraction and classification.

However, the above methods only use the global structure information of the dataset. Results from manifold learning methods developed in the past decade show that the local geometric structure is more important than the global structure since the high-dimensional data lies on the low-dimensional manifold. The representative manifold learning methods include locally linear embedding (LLE) [15], ISOMAP [16], Laplacian eigenmaps (LE) [17] and local tangent space alignment (LTSA) [18], and so on. All of these nonlinear manifold learning methods suffer from the out-of-sample problem [19], and one of the simplest but frequently used technique is to learn the explicit linear mappings of the corresponding nonlinear manifold learning methods. Therefore, locality preserving projections (LPP) [20], the linearization of LE, neighborhood-preserving embedding (NPE) [21] and orthogonal neighborhood preserving projections (ONPP) [22], the linearization of LEE, the linear LTSA (LLTSA) [23] and its supervised variations [24], [25], the linearization of LTSA were proposed for dimensionality reduction. Recently, Rozza *et al.* [26] proposed the truncated isotropic PCA classifier (T-IPCAC) for feature extraction and classifier design.

Since these linear dimensionality reduction methods cannot deal with the high-order tensor data, some of these methods were further extended to be multilinear cases, and many tensor-based and manifold learning based methods were proposed by using higher order tensor decomposition [27]–[29].

Within the past 10 years, there has been great interest in high-order tensor feature extraction, and the tensor-based methods have been popular in computer vision and pattern recognition [30]–[33]. For example, He *et al*. [34] proposed tensor subspace analysis (TSA) for second-order learning. Dai and Yeung [35] proposed tensor LPP (TLPP), tensor NPE (TNPE), and tensor LDE (TLDE). Yan *et al*. [36] proposed the marginal Fisher analysis (MFA) and graph embedding framework for dimensionality reduction from the viewpoint of graph construction, in which some classical methods can be included. Recently, Liu and Ruan [37] proposed orthogonal tensor NPE (OTNPE) for facial expression recognition. By integrating the manifold learning and the MDA methods, discriminant locally linear embedding (DLLE) [38] and some variations such as those in [39]–[42] were proposed for face recognition, gait recognition, action recognition, etc. (For more details, see the latest survey of multilinear subspace learning for tensor data [43].) In addition, the tensor voting methods [44], [45] also used the tensor representation to perform the dimensionality estimation, manifold learning, and function approximation.

However, until now, a systematic analysis on the intrinsic relationship among these tensor learning methods and designing a robust method for tensor learning have not been done. Therefore, this paper proposes to use the alignment techniques to unify the tensor learning methods and design a robust tensor alignment method which integrates L_1- and L_2-norms for sparse alignment. The contributions of this paper are as follows. First, this paper proposes a general framework for tensor learning and a concrete method called sparse tensor alignment (STA) for feature extraction. Second, it provides a comprehensive analysis and comparison on some of the most representative tensor learning methods and puts them into a unified framework by using the tensor alignment techniques. Therefore, this framework leads us to understand the common properties and intrinsic differences in existing tensor learning algorithms. Based on the unified framework summed up in this paper, a novel tensor learning method using the L_1- and L_2-norms penalty is proposed for robust tensor learning. Thus, it is natural for the proposed STA to avoid the difficulty in selecting the neighborhood size in the manifold learning based tensor learning methods.

The rest of this paper is organized as follows. In Section II, a systematic analysis on the tensor learning methods is provided. In Section III, a robust tensor alignment method is presented and used for tensor learning. Experiments are carried out to evaluate the proposed tensor learning method in Section IV, and conclusions are given in Section V.

II. Tensor Alignment Techniques

In this section, some basic multilinear notations, definitions and operations similar to those in [9] and [38] are briefly reviewed at first and then the tensor alignment representation of the most representative methods is presented. Thus a unified tensor learning framework is obtained.

A. Multilinear Algebras

In this paper, lowercase and uppercase italic letters (i.e. i, j, N, etc.) denote scalars, bold lowercase letter (i.e., $\mathbf{e}, \mathbf{h}, \mathbf{x}$ etc.) denote vectors, bold uppercase letters (i.e., $\mathbf{A}, \mathbf{B}, \mathbf{C}$, etc.) denote matrices, and the Lucida calligraphy Italic letters, (i.e. $\mathcal{X}, \mathcal{Y}$) denote the tensors.

It is assumed that the training samples are represented as the nth order tensor $\{\mathcal{X}_i \in R^{m_1 \times m_2 \times \cdots \times m_n}, i = 1, 2, \ldots, N\}$, where N denotes the total number of training samples.

Definition 1: The mode-k flattening of the nth-order tensor $\mathcal{X} \in R^{m_1 \times m_2 \times \cdots \times m_n}$ $(i = 1, 2, \ldots, N)$ into matrix $\mathbf{X}^{(k)} \in R^{m_k \times \prod_{i \neq k} m_i}$, i.e. $\mathbf{X}^{(k)} \Leftarrow_k \mathcal{X}$, is defined as $\mathbf{X}^{(k)}_{i_k, j} = \mathcal{X}_{i_i, i_2, \ldots, i_n}$, $j = 1 + \sum_{l=1, l \neq k}^{n} (i_l - 1) \prod_{o=l+1, o \neq k}^{n} m_o$.

Definition 2: The mode-k product of tensor $\mathcal{X}$ with matrix $\mathbf{U} \in R^{m'_k \times m_k}$ is defined as $\mathcal{Y} = \mathcal{X} \times_k \mathbf{U}$, where $\mathcal{Y}_{i_1, \ldots, i_{k-1}, i, i_{k+1}, \ldots, i_n} = \sum_{j=1}^{m_k} \mathcal{X}_{i_1, \ldots, i_{k-1}, j, i_{k+1}, \ldots, i_n} \mathbf{U}_{i,j}$ $(j = 1, \ldots, m'_k)$.

The common properties of the tensor learning methods aim to obtain a set of projection matrices $\{\mathbf{U}_i \in R^{m_i \times d_i}, d_i \leq m_i, i = 1, 2, \ldots, n\}$ and map the original high-order tensor data into a low-order tensor space, as

$$\mathcal{Y}_i = \mathcal{X}_i \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \cdots \times_n \mathbf{U}_n. \tag{1}$$

Different tensor learning methods use different strategies to learn the projection matrices for feature extraction. With the above preparations, some popular tensor learning methods can be unified into a general framework which provides the comprehensive understanding on different tensor feature extraction methods.

B. Proposed Tensor Alignment Technique and its Models

Concerning the tensor learning methods, since the tensor $\mathcal{X}_i$ is unfolded into a large size matrix $\mathbf{X}_i^{(k)}$ for computing, we only need to give the alignment method about the unfolded matrix.

Let $\hat{\mathbf{X}}_i^{(k)} = [\mathbf{X}_i^{(k)}, \mathbf{X}_{i_1}^{(k)}, \mathbf{X}_{i_2}^{(k)}, \ldots, \mathbf{X}_{i_K}^{(k)}]$ be the matrix containing $\mathbf{X}_i^{(k)}$ and its K unfolding nearest neighbors tensors. The projection matrix $\mathbf{U}_k$ maps the unfolding tensor into a low-dimensional subspace: $\mathbf{U}_k : \mathbf{X}_i^{(k)} \rightarrow \mathbf{Y}_i^{(k)}$. Let $\mathbf{L}_i$ be the local alignment matrix of size $(K + 1) \times (K + 1)$ designed for different tensor learning algorithms, and then the local alignment optimization problem is formed

$$\min \operatorname{tr}\left(\hat{\mathbf{Y}}_i^{(k)} (\mathbf{L}_i \otimes \mathbf{I}_k) \hat{\mathbf{Y}}_i^{(k)T}\right) \tag{2}$$

where $\otimes$ denotes the Kronecker product of matrices and $\hat{\mathbf{Y}}_i^{(k)} = [\mathbf{Y}_i^{(k)}, \mathbf{Y}_{i_1}^{(k)}, \ldots, \mathbf{Y}_{i_K}^{(k)}]$ be the local coordinate. The selection matrix $\mathbf{S}_i$ with the size of $N \times (K+1)$ is defined as

$$(\mathbf{S}_i)_{\mathrm{pq}} = \begin{cases} 1, & \text{if } p = \mathbf{f}_i\{q\} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where $\mathbf{f}_i = \{i, i_1, i_2, \ldots, i_K\}$ denotes the set of indices for the ith alignment matrix formed by $\mathbf{X}_i^{(k)}$ (or tensor $\mathcal{X}_i$) and its K unfolding nearest neighbors tensors. Let $\mathbf{Y}^{(k)} = [\mathbf{Y}_1^{(k)}, \mathbf{Y}_2^{(k)}, \ldots, \mathbf{Y}_N^{(k)}]$ be the global coordinates; then we have

$$\hat{\mathbf{Y}}_i^{(k)} = \mathbf{Y}^{(k)} (\mathbf{S}_i \otimes \mathbf{I}_k). \tag{4}$$

Then (2) can be rewritten as

$$\begin{aligned}&\min \operatorname{tr}(\hat{\mathbf{Y}}_i^{(k)}(\mathbf{L}_i \otimes \mathbf{I}_k)\hat{\mathbf{Y}}_i^{(k)T}) \\ &\quad = \min \operatorname{tr}(\mathbf{Y}^{(k)}(\mathbf{S}_i \otimes \mathbf{I}_k)(\mathbf{L}_i \otimes \mathbf{I}_k)(\mathbf{S}_i^T \otimes \mathbf{I}_k)\mathbf{Y}^{(k)T}) \\ &\quad = \min \operatorname{tr}(\mathbf{Y}_i^k(\mathbf{S}_i\mathbf{L}_i\mathbf{S}_i^T \otimes \mathbf{I}_k)\mathbf{Y}_i^{(k)T}). \end{aligned} \tag{5}$$

By summing up all the alignments together, the whole alignment can be obtained as

$$\begin{aligned}&\min \sum\nolimits_i \operatorname{tr}(\hat{\mathbf{Y}}_i^{(k)}(\mathbf{S}_i\mathbf{L}_i\mathbf{S}_i^T \otimes \mathbf{I}_k)\hat{\mathbf{Y}}_i^{(k)T}) \\ &\quad = \min \operatorname{tr} \sum\nolimits_i \hat{\mathbf{Y}}_i^{(k)}(\mathbf{S}_i\mathbf{L}_i\mathbf{S}_i^T \otimes \mathbf{I}_k)\hat{\mathbf{Y}}_i^{(k)T} \\ &\quad = \min \operatorname{tr}(\mathbf{Y}^{(k)}(\mathbf{L} \otimes \mathbf{I}_k)\mathbf{Y}^{(k)T}) \end{aligned} \tag{6}$$

where $\mathbf{L} = \sum_i \mathbf{S}_i\mathbf{L}_i\mathbf{S}_i^T$ is the alignment matrix [18], which can be obtained by the iterative procedure as

$$\mathbf{L}(\mathbf{f}_i, \mathbf{f}_i) \leftarrow \mathbf{L}(\mathbf{f}_i, \mathbf{f}_i) + \mathbf{L}_i \tag{7}$$

with the initialization $\mathbf{L} = \mathbf{0}$.

Let $\mathbf{L}_a$ and $\mathbf{L}_b$ be some kinds of alignment matrices by different methods. If the linear transformation $\mathbf{Y}_i^{(k)} = \mathbf{U}_k^T\mathbf{X}_i^{(k)}$ is considered, then the following optimization model is obtained:

$$(i)\begin{cases}\min \operatorname{tr}(\mathbf{Y}^{(k)}(\mathbf{L}\otimes\mathbf{I}_k)\mathbf{Y}^{(k)T}) = \min \operatorname{tr}(\mathbf{U}_k^T\mathbf{X}^{(k)}(\mathbf{L}_a\otimes\mathbf{I}_k)\mathbf{X}^{(k)T}\mathbf{U}_k) \\ \text{s.t. } \mathbf{U}_k^T\mathbf{X}^{(k)}(\mathbf{L}_b \otimes \mathbf{I}_k)\mathbf{X}^{(k)T}\mathbf{U}_k = \mathbf{I}_{d_k}. \end{cases} \tag{8}$$

Or, if only one alignment matrix $\mathbf{L}_a$ is used and $\mathbf{Y}^{(k)}$ is uniquely determined, the constraint $\mathbf{Y}^{(k)}\mathbf{Y}^{(k)T} = \mathbf{I}_{d_k}$ can be imposed and then the optimization model is obtained as

$$\begin{cases}\min \operatorname{tr}(\mathbf{U}_k^T\mathbf{X}^{(k)}(\mathbf{L}_a \otimes \mathbf{I}_k)\mathbf{X}^{(k)T}\mathbf{U}_k) \\ \quad \text{s.t. } \mathbf{U}_k^T\mathbf{X}^{(k)}\mathbf{X}^{(k)T}\mathbf{U}_k = \mathbf{I}_{d_k}. \end{cases} \tag{9}$$

Specifically, (9) is the special case of model (i) with $\mathbf{L}_b = \mathbf{I}_N$.

In addition, one can alternatively impose the following orthogonal constraint and obtain another model

$$(ii)\begin{cases}\min \operatorname{tr}(\mathbf{U}_k^T\mathbf{X}^{(k)}(\mathbf{L}_a \otimes \mathbf{I}_k)\mathbf{X}^{(k)T}\mathbf{U}_k) \\ \text{s.t. } \mathbf{U}_k^T\mathbf{U}_k = \mathbf{I}_{d_k}. \end{cases} \tag{10}$$

These two models can be solved by using the Lagrangian multiplier method and their solutions can be obtained by using generalized or standard eigenvalue decomposition, respectively. Since there are not closed-form solutions for tensor subspace learning methods, the iterative strategy is usually used for computing the local optimal solutions. As can be seen from the following sections, these two models are the basic forms of the tensor subspace learning methods.

C. Alignment for MPCA and T-IPCAC

MPCA maximizes the trace of the total scatter matrix of the unfolded tensors in the projected subspace. The basic model of MPCA is

$$\begin{aligned}&\min \operatorname{tr}(\mathbf{U}_i^T\mathbf{S}_t^{(k)}\mathbf{U}_i) \\ &= \min \operatorname{tr}(\mathbf{U}_k^T \sum\nolimits_i (\mathbf{X}_i^{(k)} - \bar{\mathbf{X}}^{(k)})(\mathbf{X}_i^{(k)} - \bar{\mathbf{X}}^{(k)})^T\mathbf{U}_k) \\ &= \min \sum_{i=1}^{N} \operatorname{tr}\left(\frac{1}{N^2}\sum_{j=1}^{N-1}(\mathbf{Y}_i^{(k)} - \mathbf{Y}_{i_j}^{(k)})(\mathbf{Y}_i^{(k)} - \mathbf{Y}_{i_j}^{(k)})^T\right) \\ &= \min \sum_{i=1}^{N} \operatorname{tr}\left(\frac{1}{N^2}\left(\hat{\mathbf{Y}}_i^{(k)}\left(\begin{pmatrix}N-1\\ -\mathbf{e}_{N-1}\end{pmatrix}\otimes\mathbf{I}_k\right)\right)\right. \\ &\qquad \left.\times\left(\hat{\mathbf{Y}}_i^{(k)}\left(\begin{pmatrix}N-1\\ -\mathbf{e}_{N-1}\end{pmatrix}\otimes\mathbf{I}_k\right)\right)^T\right) \\ &= \min \sum_{i=1}^{N} \operatorname{tr}\left(\frac{1}{N^2}\hat{\mathbf{Y}}_i^{(k)}\left(\begin{pmatrix}N-1\\ -\mathbf{e}_{N-1}\end{pmatrix}\begin{pmatrix}N-1\\ -\mathbf{e}_{N-1}\end{pmatrix}^T\otimes\mathbf{I}_k\right)\hat{\mathbf{Y}}_i^{(k)T}\right) \\ &= \min \sum_{i=1}^{N} \operatorname{tr}(\hat{\mathbf{Y}}_i^{(k)}(\mathbf{L}_i^{\text{MPCA}} \otimes \mathbf{I}_k)\hat{\mathbf{Y}}_i^{(k)T}) \end{aligned} \tag{11}$$

where $\mathbf{Y}_{i_j}^{(k)}(j = 1, 2, \ldots, N-1)$ are the rest unfolded tensors of $\mathbf{Y}_i^{(k)}$, $\bar{\mathbf{X}}^{(k)}$ is the unfolded mean tensor, and $\mathbf{e}_{N-1} = [1, 1, \ldots, 1]^T$ with $N-1$ elements, $\hat{\mathbf{Y}}_i^{(k)} = [\mathbf{Y}_i^{(k)}, \mathbf{Y}_{i_1}^{(k)}, \ldots, \mathbf{Y}_{i_K}^{(k)}]$ and $K = N-1$

$$\mathbf{L}_i^{\text{MPCA}} = \begin{pmatrix}N-1\\ -\mathbf{e}_{N-1}\end{pmatrix}\begin{pmatrix}N-1\\ -\mathbf{e}_{N-1}\end{pmatrix}^T \otimes \mathbf{I}_k$$

and

$$\mathbf{L}^{\text{MPCA}} = \sum\nolimits_i \mathbf{L}_i^{\text{MPCA}}.$$

Therefore, MPCA can be viewed as a global tensor alignment method since $\hat{\mathbf{Y}}_i^{(k)}$ contains all the unfolded tensors. And model (ii) represents the optimization model of MPCA with $\mathbf{L} = \sum_i \mathbf{L}_i^{\text{MPCA}}$.

The key points of T-IPCAC [26] are the recovery of the residuals and the Fisher subspace estimation. Since T-IPCAS uses labeled data to estimate the Fisher subspace, it can also be included in the MDA set. One of the key steps in T-IPCAC is to compute the d eigenvectors corresponding to the d largest eigenvalues of the sample covariance matrix $\mathbf{S}_t$ on the vector space instead of higher order tensor space, where the definition of $\mathbf{S}_t$ and its alignment are as follows:

$$\mathbf{S}_t = \frac{1}{N}\widehat{\mathbf{X}}\widehat{\mathbf{X}}^T = \sum_{i=1}^{N}\widehat{\mathbf{X}}\left(\frac{1}{N^2}\mathbf{I}\right)\widehat{\mathbf{X}} = \sum_{i=1}^{N}\widehat{\mathbf{X}}\mathbf{L}_i^{\text{T-IPCAC}}\widehat{\mathbf{X}} \tag{12}$$

where $\widehat{\mathbf{X}} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]$ denotes the vector-based sample matrix, and $\mathbf{L}_i^{\text{T-IPCAC}} = (1/N^2)\mathbf{I}$, and $\mathbf{I}$ is the $N \times N$ identity matrix. This indicates that the whitening step in T-IPCAC is a global alignment with the identity matrix.

D. Alignment for TLPP

TLPP preserves the local neighborhood relationship of the tensors. Similar to LPP, TLPP first constructs the local neighborhood matrix $\mathbf{W}_{ij} = \exp(-\|X_i - X_j\|^2/t)$ if X_j is one of the K nearest neighbors of X_i; otherwise 0, and t

is the tuning parameter. The objective function of TLPP is defined as

$$\begin{aligned}
&\min \sum\nolimits_i \sum\nolimits_j \left\| (\mathbf{Y}_i^{(k)} - \mathbf{Y}_j^{(k)}) \right\|^2 \mathbf{W}_{ij} \\
&= \min \sum_{i=1}^{N} \sum_{l=1}^{K} (\mathbf{Y}_i^{(k)} - \mathbf{Y}_l^{(k)})(\mathbf{Y}_i^{(k)} - \mathbf{Y}_l^{(k)})^T \bar{\mathbf{W}}_{il} \\
&= \min \sum_{i=1}^{N} \mathrm{tr} \left(\begin{bmatrix} (\mathbf{Y}_i^{(k)} - \mathbf{Y}_{i_1}^{(k)})^T \\ \cdots \\ (\mathbf{Y}_i^{(k)} - \mathbf{Y}_{i_l}^{(k)})^T \end{bmatrix} \right. \\
&\qquad \left. \times [\mathbf{Y}_i^{(k)} - \mathbf{Y}_{i_1}^{(k)}, \ldots, \mathbf{Y}_i^{(k)} - \mathbf{Y}_{i_l}^{(k)}] \mathrm{diag}(\bar{\mathbf{W}}_{i,:}) \right) \\
&= \min \sum_{i=1}^{N} \mathrm{tr} \left(\hat{\mathbf{Y}}_i^{(k)} \left(\begin{bmatrix} -\mathbf{e}_K^T \\ \mathbf{I}_K \end{bmatrix} \mathrm{diag}(\bar{\mathbf{W}}_{i,:}) \begin{bmatrix} -\mathbf{e}_K^T & \mathbf{I}_K \end{bmatrix} \otimes \mathbf{I}_k \right) \hat{\mathbf{Y}}_i^{(k)T} \right) \\
&= \min \sum_{i=1}^{N} \mathrm{tr} (\hat{\mathbf{Y}}_i^{(k)} (\mathbf{L}_i^{\mathrm{TLPP}} \otimes \mathbf{I}_k) \hat{\mathbf{Y}}_i^{(k)T}) \qquad (13)
\end{aligned}$$

where $\mathbf{L}_i^{\mathrm{TLPP}} = \begin{bmatrix} -\mathbf{e}_K^T \\ \mathbf{I}_K \end{bmatrix} \mathrm{diag}(\bar{\mathbf{W}}_{i,:}) \begin{bmatrix} -\mathbf{e}_K^T & \mathbf{I}_K \end{bmatrix} \otimes \mathbf{I}_k$, and $\mathbf{I}_K$ is the $K \times K$ identity matrix, $\hat{\mathbf{Y}}_i^k = [\mathbf{Y}_i^k, \mathbf{Y}_{i_1}^k, \ldots, \mathbf{Y}_{i_K}^k]$, $\mathbf{e}_K = [1, 1, \ldots, 1]^T$ with K elements, and $\bar{\mathbf{W}}_{il} = \exp(-\left\| \chi_i - \chi_l \right\|^2 / t)$ (i.e., matrix $\bar{\mathbf{W}}$ only contains nonzero elements in $\bar{\mathbf{W}}$).

In addition, TLPP has the following constraint which can also be represented by using the alignment technique:

$$\mathrm{tr} \left(\sum\nolimits_i \mathbf{Y}_i^{(k)} \mathbf{Y}_i^{(k)T} \mathbf{D}_{ij} \right) = \sum_{i=1}^{N} \mathrm{tr} \left(\mathbf{Y}_i^k \left(\mathbf{D} \otimes \mathbf{I}_k \right) \mathbf{Y}_i^{(k)T} \right) = 1 \quad (14)$$

where the diagonal elements $\mathbf{D}_{ii}$ of matrix $\mathbf{D}$ is defined as $\mathbf{D}_{ii} = \sum_j \mathbf{W}_{ij}$. Equation (14) can be viewed as the single alignment with weight $\mathbf{D}_{ii}$ since matrix $\hat{\mathbf{Y}}_i^{(k)} = \mathbf{Y}_i^{(k)}$ only contains one element (i.e., the unfolded tensor of $\mathbf{U}_k^T \mathbf{X}_i^{(k)}$). Therefore, both parts of TLPP can be represented by using the alignment technique.

It should be noted that the tensor version of the graph embedding framework proposed in [36] can also be represented and concluded in the tensor alignment framework with the same way as TLPP. To avoid repetition, it is omitted in this paper.

E. Alignment for TNPE

TNPE preserves the local linear reconstruction coefficients of tensors in the low-dimensional subspace. Suppose the coefficient matrix $\mathbf{M}$ (of size $N \times K$) is obtained in the same way as in LLE, and $\mathbf{M}$ only contains the reconstruction coefficients (zero elements are not included). The cost function of TNPE is defined as

$$\begin{aligned}
&\min \sum_{i=1}^{N} \left\| \mathbf{Y}_i^{(k)} - \sum_{j=1}^{K} \mathbf{M}_{i,j} \mathbf{Y}_{i_j}^{(k)} \right\|^2 \\
&= \min \sum_{i=1}^{N} \mathrm{tr} \left(\hat{\mathbf{Y}}_i^{(k)} \left(\begin{bmatrix} -1 \\ \mathbf{M}_{i,:} \end{bmatrix} \begin{bmatrix} -1 & \mathbf{M}_{i,:}^T \end{bmatrix} \otimes \mathbf{I}_k \right) \hat{\mathbf{Y}}_i^{(k)T} \right) \\
&= \min \sum_{i=1}^{N} \mathrm{tr} (\hat{\mathbf{Y}}_i^{(k)} (\mathbf{L}_i^{\mathrm{TNPE}} \otimes \mathbf{I}_k) \hat{\mathbf{Y}}_i^{(k)T}) \qquad (15)
\end{aligned}$$

where $\hat{\mathbf{Y}}_i^{(k)} = [\mathbf{Y}_i^{(k)}, \mathbf{Y}_{i_1}^{(k)}, \ldots, \mathbf{Y}_{i_K}^{(k)}]$, i.e., $\hat{\mathbf{Y}}_i^{(k)}$ only contains the $\mathbf{Y}_i^{(k)}$ and its K nearest neighbor tensor unfolded matrices.

It can be found that TNPE is different from MPCA. The essential difference is in the alignment matrices. TLPP uses the local alignment method to construct $\mathbf{L}_i^{\mathrm{TNPE}}$.

F. Alignment for MDA

MDA aims to find the multilinear subspaces that can minimize the trace of the within-class unfolded tensor scatter matrix $\mathbf{S}_w^{(k)}$ and maximize the trace of the between-class unfolded tensor scatter $\mathbf{S}_b^{(k)}$.

For the mode-k within-class tensor scatter matrix $\mathbf{S}_w^{(k)}$, we have

$$\begin{aligned}
&\min \mathrm{tr}(\mathbf{S}_w^{(k)}) \\
&= \sum_{i=1}^{C} \sum_{j=1}^{N_i} (\mathbf{Y}_i^{j(k)} - \bar{\mathbf{Y}}_i^{(k)})(\mathbf{Y}_i^{j(k)} - \bar{\mathbf{Y}}_i^{(k)})^T \\
&= \min \mathrm{tr} \sum_{i=1}^{N} \frac{1}{N^2} \left(\sum_{j=1}^{N_i - 1} (\mathbf{Y}_i^{(k)} - \mathbf{Y}_{i_j}^{(k)}) \right) \left(\sum_{j=1}^{N_i - 1} (\mathbf{Y}_i^{(k)} - \mathbf{Y}_{i_j}^{(k)}) \right)^T \\
&= \min \sum_{i=1}^{N} \mathrm{tr} \left(\frac{1}{N^2} \hat{\mathbf{Y}}_i^{(k)} \left(\begin{bmatrix} N_i - 1 \\ -\mathbf{e}_{N_i - 1} \end{bmatrix} \begin{bmatrix} N_i - 1 \\ -\mathbf{e}_{N-1} \end{bmatrix}^T \otimes \mathbf{I}_k \right) \hat{\mathbf{Y}}_i^{(k)T} \right) \\
&= \min \sum_{i=1}^{N} \mathrm{tr} \left(\frac{1}{N^2} \hat{\mathbf{Y}}_i^{(k)} \left(\mathbf{L}_i^w \otimes \mathbf{I}_k \right) \hat{\mathbf{Y}}_i^{(k)T} \right) \qquad (16)
\end{aligned}$$

where $\bar{\mathbf{Y}}_i^{(k)}$ denotes the mean value of the mode-k flattening of the tensor samples in the ith class, C is the number of classes, N_i is the number of the tensors in the ith class.$\mathbf{Y}_i^{j(k)}$ is the jth tensor in the ith class, $\mathbf{e}_{N_i - 1} = [1, 1, \ldots, 1]^T$ with $N_i - 1$ elements and $\hat{\mathbf{Y}}_i^{(k)} = [\mathbf{Y}_i^{(k)}, \mathbf{Y}_{i_1}^{(k)}, \ldots, \mathbf{Y}_{i_{N_i - 1}}^{(k)}]$

$$\mathbf{L}_i^w = \begin{bmatrix} N_i - 1 \\ -\mathbf{e}_{N_i - 1} \end{bmatrix} \begin{bmatrix} N_i - 1 \\ -\mathbf{e}_{N-1} \end{bmatrix}^T.$$

For the mode-k between-class tensor scatter matrix $\mathbf{S}_b^{(k)}$, we have

$$\begin{aligned}
&\max \mathrm{tr}(\mathbf{S}_b^{(k)}) = \sum_{i=1}^{C} N_i (\bar{\mathbf{Y}}_i^{(k)} - \bar{\mathbf{Y}}^{(k)})(\bar{\mathbf{Y}}_i^{(k)} - \bar{\mathbf{Y}}^{(k)})^T \\
&= \max \mathrm{tr} \left(\sum_{i=1}^{C} N_i \frac{1}{C^2} \sum_{j=1}^{C-1} (\bar{\mathbf{Y}}_i^{(k)} - \bar{\mathbf{Y}}_{i_j}^{(k)}) \sum_{j=1}^{C-1} (\bar{\mathbf{Y}}_i^{(k)} - \bar{\mathbf{Y}}_{i_j}^{(k)})^T \right)
\end{aligned}$$

$$= \max \sum_{i=1}^{C} \mathrm{tr}\left(\frac{N_i}{C^2}\hat{\mathbf{Y}}_i^{(k)}\left(\begin{bmatrix} C-1 \\ -\mathbf{e}_{C-1}\end{bmatrix}\begin{bmatrix} C-1 \\ -\mathbf{e}_{C-1}\end{bmatrix}^T \otimes \mathbf{I}_k\right)\hat{\mathbf{Y}}_i^{(k)T}\right)$$

$$= \max \sum_{i=1}^{C} \mathrm{tr}\left(\frac{N_i}{C^2}\hat{\mathbf{Y}}_i^{(k)}\left(\mathbf{L}_i^b \otimes \mathbf{I}_k\right)\hat{\mathbf{Y}}_i^{(k)T}\right) \quad (17)$$

where $\bar{\mathbf{Y}}^{(k)}$ denotes the mean value of the mode-k flattening of the tensor samples of all the training samples. $\bar{\mathbf{Y}}_{i_j}^{(k)}(j = 1, \ldots, C-1)$ is the mean unfolded tensor of the different classes form $\bar{\mathbf{Y}}_i^{(k)}$, and

$$\mathbf{L}_i^b = \begin{bmatrix} C-1 \\ -\mathbf{e}_{C-1}\end{bmatrix}\begin{bmatrix} C-1 \\ -\mathbf{e}_{C-1}\end{bmatrix}^T$$

$\mathbf{e}_{C-1} = [1, 1, \ldots, 1]^T$ with $C-1$ elements and $\hat{\mathbf{Y}}_i^{(k)} = [\bar{\mathbf{Y}}_i^{(k)}, \bar{\mathbf{Y}}_{i_1}^{(k)}, \ldots, \bar{\mathbf{Y}}_{i_{C-1}}^{(k)}]$.

As can be seen from (16) and (17), $\mathbf{S}_w^{(k)}$ is aligned by the samples within each class, and $\mathbf{S}_b^{(k)}$ is aligned by the unfolded matrices of the sample mean tensor of different classes. The objective function of MDA can be constructed by using the model (i).

G. Alignment for Tensor Voting

Tensor voting (TV) was proposed in [45] for dimensionality estimation, manifold learning, and function approximation. The key operation in the voting process is to compute the voting accumulator through the local neighboring points. This step can be viewed as the local alignment with the eigenvectors in a special form. For $\mathbf{x}_i$, the voting accumulator of its neighborhood point $\mathbf{x}_j$ can be expressed as

$$\mathbf{L}_j^{\mathrm{TV}} \leftarrow \mathbf{L}_j^{\mathrm{TV}} + (\lambda_1 - \lambda_2)S_{\mathrm{vote}}(\mathbf{x}_i, \mathbf{x}_j, \hat{\mathbf{e}}_1) + \lambda_N B_{\mathrm{vote}}(\mathbf{x}_i, \mathbf{x}_j) + \sum_{d=2}^{N-1}(\lambda_d - \lambda_{d-1})V_{\mathrm{vote}}(\mathbf{x}_i, \mathbf{x}_j, T_{e,d}^i) \quad (18)$$

where λ_ks are the eigenvalues corresponding to the voter; $S_{\mathrm{vote}}(\mathbf{x}_i, \mathbf{x}_j, \hat{\mathbf{e}}_1)$ denote the stick vote from $\mathbf{x}_i$ to $\mathbf{x}_j$ with $\hat{\mathbf{e}}_1$ being the normal at $\mathbf{x}_i$; $B_{\mathrm{vote}}(\mathbf{x}_i, \mathbf{x}_j)$ denotes the ball vote from $\mathbf{x}_i$ to $\mathbf{x}_j$; $V_{\mathrm{vote}}(\mathbf{x}_i, \mathbf{x}_j, T_{e,d}^i)$ denotes the vote from the generic tensor, and $T_{e,d}^i$ denotes the elementary voting tensor with d equal nonzero eigenvalues. The readers are referred to [45] for more details.

Comparing (18) with (7), we can find that the tensor voting method in [45] has a procedure similar to the tensor alignment framework and thus it can be included in the proposed framework.

H. Discussions on the Tensor Learning Models

It can be seen from above sections that the tensor learning methods are different in terms of constructing the alignment matrices $\mathbf{L}_i^*$ and $\hat{\mathbf{Y}}_i^k$. MPCA and MDA use the global alignment techniques, but TLPP and TNPE use local alignment techniques. One of the limitations of MPCA and MDA is that the alignment matrices cannot reflect the local geometric structure of a tensor dataset. However, TNPE and TLPP use the local geometric structure information in constructing the alignment matrices and preserve it in the low-dimensional subspace, therefore the manifold properties can be maintained.

For the supervised tensor methods based on LPP or NPE, the only difference from TLPP or TNPE lies in the construction of the local neighborhood graphs $\mathbf{W}$ or $\mathbf{M}$ using the label information. Thus, the corresponding supervised tensor learning methods, such as those in [38]–[42], have the same alignment methods as in TLPP or TNPE. Therefore, this paper does not discuss it in detail except for the representative example MDA. Table I summarizes the details of the tensor learning methods using the proposed tensor alignment framework.

Although the manifold learning based tensor learning methods usually outperform the globality-based methods such as MPCA and MLDA, the neighborhood size K in the manifold learning based tensor subspace learning methods is difficult to decide in application. Moreover, since the tensor data usually contains large quantities of information redundancy and noise, designing a robust method for alignment becomes crucial but has yet to be explored. In this paper, we introduce the recently proposed sparse representation for robust alignment in the next section.

III. Sparse Tensor Alignment

In this section, a new multilinear dimensionality reduction technique called STA is developed as a special application of the above tensor alignment framework.

A. Motivation of Sparse Tensor Alignment

Sparse representation has been widely used in signal processing, image processing, feature extraction, and pattern recognition. Wright *et al.* [46] proposed the use of sparse representation for robust face recognition, Qiao *et al.* [47] proposed sparsity-preserving projections (SPPs) for feature extraction, and Cheng *et al.* [48] used the L_1-graph for image data clustering and subspace learning. As demonstrated in [47] and [48], the graphs constructed by the L_1-norm have the advantages of greater robustness to data noise, automatic sparsity and adaptive neighborhood for individual datum. Another important advantage is that sparse representation has the potential discriminative ability since most nonzero elements are located on the samples in the same class as the represented sample [46]–[48]. Thus, these advantages can be naturally fused to tensor learning if the L_1-norm is used in the tensor representation in constructing the alignment matrices. However, only using the L_1-norm penalty such as in LASSO [49] has its limitation as indicated in [50]: if there is a group of variables among which the pairwise correlations are very high, LASSO tends to select any one variable from the group and does not consider which one is selected. Fortunately, it is known that combining the L_1- and L_2-norm penalty can result in grouping effectiveness in regression and thus enhance the prediction accuracy by using the elastic net [50] which overcomes the limitation of only using the L_1-norm penalty.

In short, it is expected that the elastic net is used to group a set of sparse coefficients to construct the sparse alignment matrices, in which the sparse representation information or the potential discriminative information is encoded to enhance the

TABLE I
SUMMARY OF THE ALGORITHMS

Method	Data matrix	Alignment matrix	Alignment form	Goal	Related methods
MPCA	$\mathcal{X}_i$ and its remainders	$\begin{pmatrix} N-1 \\ -\mathbf{e}_{N-1} \end{pmatrix}\begin{pmatrix} N-1 \\ -\mathbf{e}_{N-1} \end{pmatrix}^T \otimes \mathbf{I}_k$	Unsupervised, global	Maximizing scatter matrix	2DPCA, PCA, T-IPCAC
TLPP	$\mathcal{X}_i$ and its neighbors	$\begin{bmatrix} -\mathbf{e}_K^T \\ I_K \end{bmatrix} diag(\overline{\mathbf{W}}_{i,:})\begin{bmatrix} -\mathbf{e}_K^T & \mathbf{I}_K \end{bmatrix} \otimes \mathbf{I}_k$	Unsupervised, local neighborhood	Preserving local nearest neighbor relationship	TLDE, MFA, LE
TNPE	$\mathcal{X}_i$ and its neighbors	$\begin{bmatrix} -1 \\ \mathbf{M}_{i,:} \end{bmatrix}\begin{bmatrix} -1 & \mathbf{M}_{i,:}^T \end{bmatrix} \otimes I_k$	Unsupervised, local neighborhood	Preserving local reconstruction relationship	OTNPE, ONPP,LLE
MDA	$\mathcal{X}_i$ and its withinclass remainders, Centroid and the different class points	$\begin{bmatrix} N_i-1 \\ -\mathbf{e}_{N_i-1} \end{bmatrix}\begin{bmatrix} N_i-1 \\ -\mathbf{e}_{N-1} \end{bmatrix}^T \otimes \mathbf{I}_k$ $\begin{bmatrix} C-1 \\ -\mathbf{e}_{C-1} \end{bmatrix}\begin{bmatrix} C-1 \\ -\mathbf{e}_{C-1} \end{bmatrix}^T \otimes \mathbf{I}_k$	Supervised, global	Maximizing the Fisher criterion	TMMC, GTDA, 2DLDA, LDA, T-IPCAC
TV	$\mathbf{x}_i$ and its neighbors	$(\lambda_1-\lambda_2)S_{vote}(\mathbf{x}_i,\mathbf{x}_j,\hat{\mathbf{e}}_1)+\lambda_N B_{vote}(\mathbf{x}_i,\mathbf{x}_j)$ $+\sum_{d=2}^{N-1}(\lambda_d-\lambda_{d-1})V_{vote}(\mathbf{x}_i,\mathbf{x}_j,T_{e,d}^i)$	Unsupervised, local neighborhood	Manifold learning and dimensionality estimate	------
STA	$\mathcal{X}_i$ and its remainders	$(\frac{1}{N}\mathbf{e}_i-\mathbf{H}_{i,:})(\frac{1}{N}\mathbf{e}_i-\mathbf{H}_{i,:})^T \otimes \mathbf{I}_k$	Unsupervised, global	Preserving the sparsity reconstruction relationship	SPP

discriminative ability in an unsupervised manner. Therefore, it is reasonable to integrate these advantages to design a more robust and effective tensor learning method. As a result, we first introduce sparse representation and the SPP algorithm in the next section, and then STA is proposed.

B. Sparse Representation and SPP

The goal of sparse representation is to represent the high-dimensional vector $\mathbf{x}$ as few entries of $\widehat{\mathbf{X}} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$ as possible. This can be formally expressed as follows:

$$\min_{\mathbf{h}} \|\mathbf{h}\|_0 \quad \text{s.t.} \quad \mathbf{x} = \widehat{\mathbf{X}}\mathbf{h} \tag{19}$$

where $\mathbf{h} \in R^N$ is the coefficient vector and $\|\mathbf{h}\|_0$ is the L_0-norm which is equal to the number of nonzero components in $\mathbf{h}$. Unfortunately, this problem is not convex, and finding the sparse solution is NP-hard. It has been shown that, if the solution of (19) is sparse enough, this difficulty can be bypassed by convexizing the problem [51] and solving

$$\min_{\mathbf{h}} \|\mathbf{h}\|_1 \quad \text{s.t.} \quad \mathbf{x} = \widehat{\mathbf{X}}\mathbf{h}. \tag{20}$$

SPP takes the advantage of L_1-norm sparse representation and preserves such reconstructive weights for dimensionality reduction. For each $\mathbf{x}_i$, SPP first solves the following L_1-norm minimization problem:

$$\min \|\mathbf{h}_i\|_1 \quad \text{s.t.} \quad \mathbf{x}_i = \widehat{\mathbf{X}}\mathbf{h}_i, \quad 1 = \mathbf{e}^T\mathbf{h}_i \tag{21}$$

where $\mathbf{h}_i = [h_{i,1}, \ldots, h_{i,i-1}, 0, h_{i,i+1}, \ldots, h_{i,N}]^T$ is an N-dimensional vector in which the ith element is equal to zero (implying that the $\mathbf{x}_i$ is removed from $\widehat{\mathbf{X}}$), and the elements $h_{i,j}(j \neq i)$ denote the contribution of each $\mathbf{x}_j$ to reconstruct $\mathbf{x}_i$; $\mathbf{e}$ is an N-dimensional vector of all 1s. Then the optimal solution, denoted as $\tilde{\mathbf{h}}_i$, is used to construct the following objective function which aims to preserve the optimal weight vector $\tilde{\mathbf{h}}_i$

$$\sum_{i=1}^{N}\left\|\mathbf{U}^T\mathbf{x}_\mathbf{i} - \mathbf{U}^T\widehat{\mathbf{X}}\tilde{\mathbf{h}}_i\right\|^2 = tr(\mathbf{U}^T\widehat{\mathbf{X}}(\mathbf{I}-\tilde{\mathbf{H}})(\mathbf{I}-\tilde{\mathbf{H}})^T\widehat{\mathbf{X}}^T\mathbf{U}) \tag{22}$$

where $\tilde{\mathbf{H}} = [\tilde{\mathbf{H}}_1, \tilde{\mathbf{h}}_2, \ldots, \tilde{\mathbf{H}}_N]$. The optimal projections of SPP are the eigenvectors corresponding to the smaller eigenvalues of the following generalized eigenvalue problem:

$$\widehat{\mathbf{X}}(\mathbf{I}-\tilde{\mathbf{H}})(\mathbf{I}-\tilde{\mathbf{H}})^T\widehat{\mathbf{X}}^T\mathbf{U} = \widehat{\mathbf{X}}\widehat{\mathbf{X}}^T\mathbf{U}\Lambda. \tag{23}$$

C. Efficient Method for Computing the Sparse Coefficients

Since SPP only focuses on the vector-based sparse representation problem using the L_1-norm, in this section, the sparse representation for tensor data combining the L_1-and L_2-norm penalty is introduced. First, in order to obtain the optimal sparse representation coefficients, the tensor representation of the following L_1- and L_2-norm penalty optimization problem should be solved

$$\mathbf{H}^* = \arg\min_{\mathbf{H}}\left(\left\|X_i - \sum_{j,j\neq i}\mathbf{H}_{ij}X_j\right\|^2 + \alpha\left\|\mathbf{H}_{i,:}\right\|^2 + \beta\left|\mathbf{H}_{i,:}\right|\right)(\forall i) \tag{24}$$

where the $N \times N$ matrix $\mathbf{H}$ is the representation coefficient matrix satisfying $\text{diag}(\mathbf{H}) = \mathbf{0}$ (this is similar to the $\tilde{\mathbf{H}}$ in SPP), and $\mathbf{H}_{i,:}$ denotes the ith row vector, $|\cdot|$ denotes the L_1-norm of vector $\mathbf{H}_{i,:}$, the coefficient $\alpha \geq 0$ is a parameter to control the amounts of shrinkage, and β is the L_1-norm term coefficient. Because of the nature of the L_1-norm penalty, some coefficients are shrunk exactly to zeros if β is large enough. The difference for learning the reconstruction matrix between the STA and SPP is that STA uses the L_1- and L_2-norm penalty which can result in sparsity and improving the grouping effectiveness in regression [50] in unsupervised manner. However, it is impossible to directly solve the above optimization problem with tensor representation. Fortunately, it is easy to obtain the following proposition from Definition 1.

Proposition 1: The optimization problem of (24) is equivalent to the following optimization problem:

$$\mathbf{H}^* = \arg\min_{\mathbf{H}} \left(\left\| \mathbf{x}_i - \sum_{j,j\neq i} \mathbf{H}_{ij}\mathbf{x}_j \right\|_2^2 + \alpha \left\| \mathbf{H}_{i,:} \right\|^2 + \beta \left| \mathbf{H}_{i,:} \right| \right) (\forall i) \tag{25}$$

where $\mathbf{x}_i$ denotes the high-dimensional vector concatenated by the columns of matrix $\mathbf{X}_i^{(k)}$ (or tensor X_i) for any mode k.

Therefore, one can solve the N optimization problem (25) to obtain sparse matrix $\mathbf{H}$ by using the elastic net algorithm [50]. However, since $\mathbf{x}_i$ is a very high-dimensional vector, directly solving (25) is also time consuming. Fortunately, the following theorem can guarantee the equivalence of the sparse representation coefficients, which can be computed efficiently.

Theorem 1: Suppose $\mathbf{x}_i$s are the independent random vectors, for any unitary matrix $\Phi = [\mathbf{A}\ \mathbf{A}^c]$ where $\text{span}(\mathbf{A}) = \text{span}(\mathbf{x}_1, \ldots, \mathbf{x}_N)$ and $\mathbf{A}^c$ is the complement of $\mathbf{A}$, the following optimization problem (26) has the same solution as (25)

$$\mathbf{H}^* = \arg\min_{\mathbf{H}} \left(\left\| \mathbf{A}^T\mathbf{x}_i - \sum_{j,j\neq i} \mathbf{H}_{ij}\mathbf{A}^T\mathbf{x}_j \right\|_2^2 + \alpha \left\| \mathbf{H}_{i,:} \right\|^2 + \beta \left| \mathbf{H}_{i,:} \right| \right). \tag{26}$$

Proof: Since $\text{span}(\mathbf{A}) = \text{span}(\mathbf{x}_1, \ldots, \mathbf{x}_N)$ and $\mathbf{A}^c$ is the orthogonal complement of $\mathbf{A}$, then $\Phi^T\Phi = \Phi\Phi^T = \mathbf{I}$ and $\mathbf{A}^{cT}\mathbf{x}_i = 0$. We have

$$\begin{aligned}
&\left\| \mathbf{A}^T\mathbf{x}_i - \sum_{j,j\neq i} \mathbf{H}_{ij}\mathbf{A}^T\mathbf{x}_j \right\|_2^2 + \alpha \left\| \mathbf{H}_{i,:} \right\|^2 + \beta \left| \mathbf{H}_{i,:} \right| \\
&= \left\| [\mathbf{A}\ \mathbf{A}^c]^T\mathbf{x}_i - \sum_{j,j\neq i} \mathbf{H}_{ij}[\mathbf{A}\ \mathbf{A}^c]^T\mathbf{x}_j \right\|_2^2 + \alpha \left\| \mathbf{H}_{i,:} \right\|^2 + \beta \left| \mathbf{H}_{i,:} \right| \\
&= \left\| \Phi^T\mathbf{x}_i - \sum_{j,j\neq i} \mathbf{H}_{ij}\Phi^T\mathbf{x}_j \right\|_2^2 + \alpha \left\| \mathbf{H}_{i,:} \right\|^2 + \beta \left| \mathbf{H}_{i,:} \right| \\
&= \text{tr}\left[\Phi^T \left(\mathbf{x}_i - \sum_{j,j\neq i} \mathbf{H}_{ij}\mathbf{x}_j \right) \left(\mathbf{x}_i - \sum_{j,j\neq i} \mathbf{H}_{ij}\mathbf{x}_j \right)^T \Phi \right] \\
&\quad + \alpha \left\| \mathbf{H}_{i,:} \right\|^2 + \beta \left| \mathbf{H}_{i,:} \right| \\
&= \text{tr}\left[\left(\mathbf{x}_i - \sum_{j,j\neq i} \mathbf{H}_{ij}\mathbf{x}_j \right) \left(\mathbf{x}_i - \sum_{j,j\neq i} \mathbf{H}_{ij}\mathbf{x}_j \right)^T \Phi\Phi^T \right] \\
&\quad + \alpha \left\| \mathbf{H}_{i,:} \right\|^2 + \beta \left| \mathbf{H}_{i,:} \right| \\
&= \text{tr}\left[\left(\mathbf{x}_i - \sum_{j,j\neq i} \mathbf{H}_{ij}\mathbf{x}_j \right) \left(\mathbf{x}_i - \sum_{j,j\neq i} \mathbf{H}_{ij}\mathbf{x}_j \right)^T \mathbf{I} \right] \\
&\quad + \alpha \left\| \mathbf{H}_{i,:} \right\|^2 + \beta \left| \mathbf{H}_{i,:} \right| \\
&= \text{tr}\left[\left(\mathbf{x}_i - \sum_{j,j\neq i} \mathbf{H}_{ij}\mathbf{x}_j \right) \left(\mathbf{x}_i - \sum_{j,j\neq i} \mathbf{H}_{ij}\mathbf{x}_j \right)^T \right] \\
&\quad + \alpha \left\| \mathbf{H}_{i,:} \right\|^2 + \beta \left| \mathbf{H}_{i,:} \right| \\
&= \left\| \mathbf{x}_i - \sum_{j,j\neq i} \mathbf{H}_{ij}\mathbf{x}_j \right\|_2^2 + \alpha \left\| \mathbf{H}_{i,:} \right\|^2 + \beta \left| \mathbf{H}_{i,:} \right|.
\end{aligned}$$

Therefore, the optimal sparse reconstruction coefficients are invariant when $\mathbf{x}_i$s are projected to the low-dimensional subspace $\mathbf{A}$.

The theorem indicates that the sparse representation coefficients can be efficiently computed in a low-dimensional subspace spanned by the $\mathbf{x}_i$s instead of being in the original high-dimensional space. Therefore, the computational complexity can be greatly reduced in solving the sets of optimization problem (26).

D. Sparse Tensor Alignment Algorithm

Once the optimal sparse coefficient matrix $\mathbf{H}$ is obtained, it can be incorporated into the tensor alignment framework, in which the sparse representation coefficients are preserved. Thus a novel unsupervised tensor dimensionality reduction method called STA is obtained.

The objective function of STA is defined as

$$\begin{aligned}
&\min \sum_{i=1}^{N} \left\| \mathbf{Y}_i^{(k)} - \sum_{j\neq i, j=1}^{N} \mathbf{H}_{i,j}\mathbf{Y}_j^{(k)} \right\|^2 \\
&= \min \sum_{i=1}^{N} \left\| \sum_{j=1}^{N} \left(\frac{1}{N}\mathbf{Y}_i^{(k)} - \mathbf{H}_{i,j}\mathbf{Y}_j^{(k)} \right) \right\|^2 \\
&= \min \sum_{i=1}^{N} \left\| \sum_{j=1}^{N} \left(\frac{1}{N}\mathbf{e}_i - \mathbf{H}_{i,j} \right) \mathbf{Y}_j^{(k)} \right\|^2 \\
&= \min \sum_{i=1}^{N} \text{tr}\left(\hat{\mathbf{Y}}^k \left(\left(\frac{1}{N}\mathbf{e}_i - \mathbf{H}_{i,:} \right) \left(\frac{1}{N}\mathbf{e}_i - \mathbf{H}_{i,:} \right)^T \otimes \mathbf{I}_k \right) \hat{\mathbf{Y}}^{(k)T} \right) \\
&= \min \sum_{i=1}^{N} \text{tr}(\hat{\mathbf{Y}}^{(k)}(\mathbf{L}_i^{\text{STA}} \otimes \mathbf{I}_k)\hat{\mathbf{Y}}^{(k)T}) \\
&= \min \text{tr}\left(\hat{\mathbf{Y}}^{(k)} \left((\mathbf{I} - \mathbf{H})(\mathbf{I} - \mathbf{H})^T \otimes \mathbf{I}_k \right) \hat{\mathbf{Y}}^{(k)T} \right) \\
&= \min \text{tr}(\mathbf{U}_k^T\mathbf{X}^{(k)}(\mathbf{L}^{\text{STA}} \otimes \mathbf{I}_k)\mathbf{X}^{(k)T}\mathbf{U}_k)
\end{aligned}$$

TABLE II
STA ALGORITHM PROCEDURES

Input: Tensor samples $\{\mathcal{X}_i \in R^{m_1 \times m_2 \times \cdots \times m_n}, i = 1, 2, ..., N\}$, the numbers of iterations $T_{\max}$ and the dimensions $d_i (\leq m_i), i = 1, 2, ..., n$

Output: Low-dimensional features $\mathcal{Y}_i = \mathcal{X}_i \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \cdots \times_n \mathbf{U}_n \ (i = 1, 2, ..., N)$

Step 1: Solve the optimization problem (17) to get the sparse matrix H.

Step 2: Initialize $\mathbf{U}_k^0 |_{k=1}^n$ as arbitrary column-wise orthogonal matrices.

Step 3: For $t = 1 : T_{\max}$ do

For $k = 1 : n$ do

*Compute $\mathbf{X}_i^k$: $\mathbf{X}_i^k = \mathcal{X}_i \times_1 \mathbf{U}_1^{(t-1)} \cdots \times_{k-1} \mathbf{U}_{k-1}^{(t-1)} \times_{k+1} \mathbf{U}_{k+1}^{(t-1)} \cdots \times_n \mathbf{U}_n^{(t-1)}$

*Perform the mode-k flattening of the n th-order tensors $\mathcal{X}_i$ to matrices: $\mathbf{X}_i^{(k)} \Leftarrow_k \mathcal{X}_i$

*Compute the scatter matrices $\mathbf{X}^{(k)}(\mathbf{L}^{STA} \otimes \mathbf{I}_k)\mathbf{X}^{(k)T}$ and $\mathbf{X}^{(k)}\mathbf{X}^{(k)T}$.

*Solve the eigen problem $\mathbf{X}^{(k)}(\mathbf{L}^{STA} \otimes \mathbf{I}_k)\mathbf{X}^{(k)T}\mathbf{U}_k^{(t)} = \mathbf{X}^{(k)}\mathbf{X}^{(k)T}\mathbf{U}_k^{(t)}\mathbf{\Lambda}$ to get $\mathbf{U}_k^{(t)}$

End

End

Step 4: Project the tensor samples to the low-dimensional tensor subspace $\mathcal{Y}_i = \mathcal{X}_i \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \cdots \times_n \mathbf{U}_n \ (i = 1, 2, ..., N)$

where $\hat{\mathbf{Y}}^{(k)} = [\mathbf{Y}_1^{(k)}, \mathbf{Y}_2^{(k)}, \ldots, \mathbf{Y}_N^{(k)}]$, $\mathbf{L}_i^{\text{STA}} = (1/N\mathbf{e}_i - \mathbf{H}_{i,:})(1/N\mathbf{e}_i - \mathbf{H}_{i,:})^T$ and $\mathbf{L}^{\text{STA}} = (\mathbf{I} - \mathbf{H})(\mathbf{I} - \mathbf{H})^T$, and the N-dimensional vector $\mathbf{e}_i = [0, 0, \ldots, 0, 1, 0, ...0]$, i.e., only ith element is 1.

By using the model (ii), the whole optimization model of STA is obtained as follows:

$$\begin{cases} \min \operatorname{tr}(\mathbf{U}_k^T \mathbf{X}^{(k)}(\mathbf{L}^{\text{STA}} \otimes \mathbf{I}_k)\mathbf{X}^{(k)T}\mathbf{U}_k) \\ \text{s.t. } \mathbf{U}_k^T \mathbf{X}^{(k)}\mathbf{X}^{(k)T}\mathbf{U}_k = \mathbf{I}_{d_k}. \end{cases} \tag{27}$$

For each mode k, the optimal projection matrix of STA can be obtained by solving the following eigen equation:

$$\mathbf{X}^{(k)}(\mathbf{L}^{\text{STA}} \otimes \mathbf{I}_k)\mathbf{X}^{(k)T}\mathbf{U}_k = \mathbf{X}^{(k)}\mathbf{X}^{(k)T}\mathbf{U}_k\Lambda. \tag{28}$$

Similar to other tensor learning methods, the optimal projection matrices of STA have no closed-form solutions. However, the suboptimal solutions can be obtained by iteratively optimizing different projection matrices while fixing the other projection matrices. The details of the STA algorithm are shown in Table II.

E. Computational Complexity Analysis

For simplicity, we assume that $m_1 = m_2 = \cdots = m_n = m$ and the total number of training samples N is comparable in magnitude to the feature dimension m^n. Usually, the computational complexity of multilinear extension methods is less than that of vector-based methods on higher tensor data. For example, the complexity of PCA is $O(m^{3n})$. The total complexity of MPCA is $O(t((n+1)Nnm^{n+1} + nm^3))$, where t denotes the number of iterations in the outside loop.

In STA, solving the elastic net to get the coefficient matrix needs $O(m^n N^2 J)$, where J denotes the number of nonzero elements and usually is a very small number. Thus it can be rewritten as $O(m^n N^2)$. Computing the scatter matrices in (27) needs $O(nNm^{n+1})$ (upper bounded). Solving (27) needs $O(m^3)$ and the tensor projection needs $O(Nm^{n+1})$. Therefore, the total complexity of STA is $O(m^n N^2 + t(nNm^{n+1} + Nm^{n+1} + m^3))$.

From the above analysis and the literature [3]–[5], [9], [11], [14], [30]–[33], and [38]–[43], we can conclude that when the data is a second- or high-order tensor, the tensor-based learning methods can improve computational efficiency and avoid small sample size problem, thereby obtaining better performances than the vector-based learning methods in this case. However, since the tensor-based learning models are the nonconvex optimization problems, they cannot obtain the global optimal solution.

IV. EXPERIMENT

In this section, a set of experiments are presented to evaluate the proposed STA, the baseline method (nearest classifier on the original data), and other unsupervised algorithms, i.e., MPCA, TLPP, TNPE and ONPP, for recognition tasks, including second-order tensor (image matrix) in face/objective recognition and high-order tensor (3-D matrix data) in action recognition. The Yale face database was used to explore the robustness of STA with the variations in expressions, illumination, block subtraction, and noise. The COIL-20 database was used to test the robustness of STA for pose variations in noise and block subtraction. The FERET face database was used to test the robustness of STA with variations in face expression and lighting conditions. The Weizmann database was used to test the performance of STA in high-order learning. The nearest neighbor classifier with Euclidean distance was used in all the experiments. Section IV-F summarizes the experimental results.

A. Robustness Test on Yale Face Database

The Yale face database [52] (http://www.cvc.yale.edu/projects/yalefaces/yalefaces.html) contains 165 images of 15 individuals (each person providing 11 different images) with various facial expressions and lighting conditions. In our experiments, each image was manually cropped and resized to 50 × 40 pixels. In order to test the robustness of the algorithms, some areas of images were first replaced by a randomly

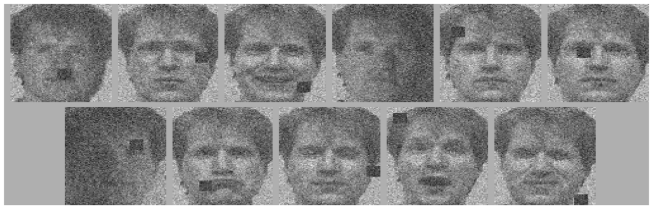

Fig. 1. Processed sample images of one person from the Yale face database.

TABLE III
AVERAGE RECOGNITION RATES (PERCENT), STANDARD ERROR, AND THE BEST PARAMETER VALUE OF FIVE METHODS ON THE YALE FACE DATABASE

Block size	Method	Recognition rate (std)	Best parameter value
5×5	Baseline	83.12±6.77	----------
	MPCA	84.20±6.11	28×28
	TLPP	85.20±3.32	34×34, $K=5$
	TNPE	83.60±6.58	22×22, $K=5$
	OTNPE	86.33±3.41	39×39, $K=4$
	STA	90.67±3.46	29×29, $\alpha=100$
10×10	Baseline	75.33±8.41	----------
	MPCA	76.67±8.11	24×24
	TLPP	80.33±4.93	24×24, $K=8$
	TNPE	78.13±7.68	37×37, $K=4$
	OTNPE	81.73±4.87	38×38, $K=4$
	STA	84.13±4.38	33×33, $\alpha=10$

located square block of size 5 × 5 or 10 × 10. Then Gaussian noise was added to the two groups of occluded images by using the MATLAB code "a = `awgn (a, 1, 35)`," where "a" denotes the image matrix. Fig. 1 shows the occluded sample images (block size 10 × 10) of one person used in the experiments.

In the experiments, six images of each individual were randomly selected and used as training set, and half of the remaining images as test and validation set, respectively. The experiments were independently performed 10 times and the average recognition results of the test set were calculated. For each run, the validation set was used for parameter selection (i.e., the local neighbor size K and the optimal subspace dimensions) in MPCA, TLPP, TNPE, and ONPP. When using the elastic net, the optimal parameter α is selected from $\{0.001, 0.01, \ldots, 10\,000\}$. The parameter β can be automatically determined since the elastic net algorithm could provide the optimal solution path of β [50]. The average recognition rates of each method and the corresponding best parameter values (in average) are shown in Table III. The recognition rate versus the number of the dimensions is shown in Fig. 2(a), and the variation of the recognition rate versus the parameter α of a single run is shown in Fig. 2(b), which indicates that the STA is robust to this parameter when it is large enough. Fig. 2(b) also shows that, when $\alpha = 0$ (i.e., without the L_2 norm penalty), the STA usually is less effective than when using a suitable L_2-norm penalty coefficient. Thus, the grouping effectiveness in regression can enhance the performance of the algorithm.

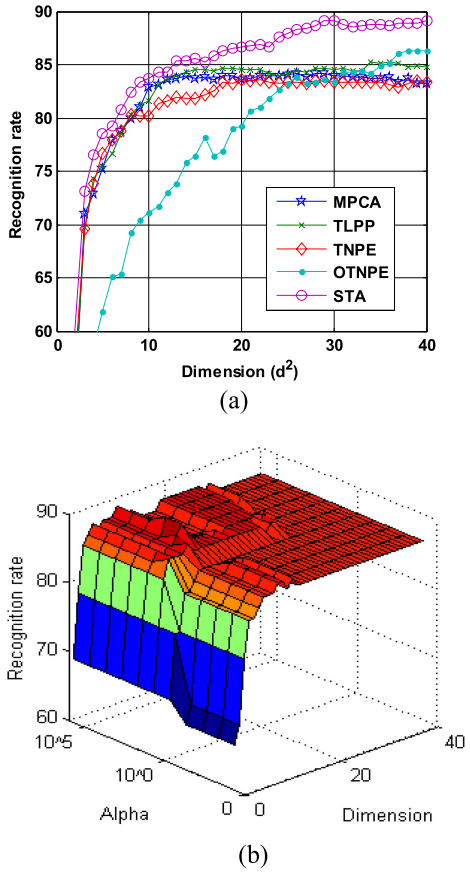

Fig. 2. (a) Average recognition rates (%) versus the number of dimensions on the Yale face database. (b) Recognition rates (%) versus the variation of α and the number of dimension on STA algorithm.

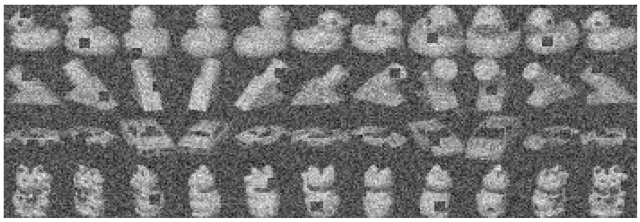

Fig. 3. Processed sample images from the COIL-20 image database.

As can be seen form Table III and Fig. 2, STA obtains the best recognition rates in the two groups of experiments, which shows the robustness for block subtraction and noise when there are variations in expressions and illumination.

B. Robust Objective Recognition on COIL-20 Image Database

The COIL-20 database [53] (http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php) consists of 20 × 72 = 1440 images of 20 objects where the images of each object were taken at pose intervals of 5° (i.e., 72 poses per object). The original images were normalized to 128 × 128 pixels. Each image was converted to a gray-scale image of 32 × 32 pixels for computational efficiency in the experiments. The images were also preprocessed as in Section IV-A by using the 5 × 5 square block for occlusion. Some sample images of four objects are shown in Fig. 3.

TABLE IV
AVERAGE RECOGNITION RATES (PERCENT), STANDARD ERROR, AND THE BEST PARAMETER VALUE OF FIVE METHODS ON THE COIL-20 DATABASE

Method	Recognition rate (std)	Best parameter value
Baseline	75. 78±4.22	----------
MPCA	80.97±2.06	11×11
TLPP	83.01±1.62	10×10, $K=5$
TNPE	76.90±3.27	11×11, $K=5$
OTNPE	83.06±1.81	31×31, $K=5$
STA	85.02±1.70	15×15, $\alpha=100$

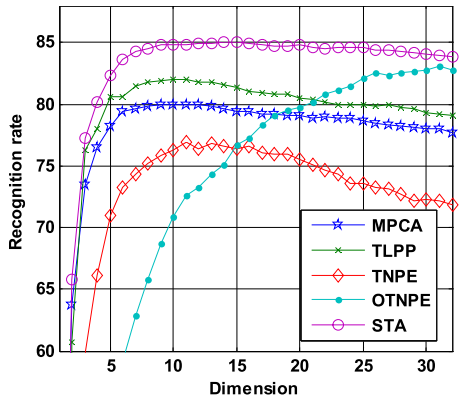

Fig. 4. Average recognition rates (%) versus the number of dimensions on the COIL 20 database.

Fig. 5. Sample images of one person on FERET face database.

In this database, the experiments were performed in the same way as in Section IV-A. The recognition rates of each method are shown in Table IV. The recognition rates versus the variations of the dimension are shown in Fig. 4. In Table IV and Fig. 4, STA obtains the best recognition rate, which shows the robustness for block subtraction and noise when there are variations in rotations of the objectives.

C. Experiments on FERET Face Database

The FERET face database is a result of the FERET program, which was sponsored by the U.S. Department of Defense [54]. It has become a standard database for testing and evaluating state-of-the-art face recognition algorithms. The proposed method was tested on a subset of the FERET database. This subset includes 1400 images of 200 individuals (each individual has seven images) and involves variations in facial expression, illumination, and pose. In the experiment, the facial portion of each original image was automatically cropped based on the location of the eyes, and the cropped images were resized to 40 × 40 pixels. The sample images of one person are shown in Fig. 5.

In the experiments, four images of each individual were randomly selected and used for training, and the remaining images were used for testing. The experiments were performed

TABLE V
AVERAGE RECOGNITION RATES (PERCENT), STANDARD ERROR, AND THE BEST PARAMETER VALUE OF FIVE METHODS ON THE FERET DATABASE

Method	Recognition rate (std)	Best parameter value
Baseline	50.34±8.13	----------
MPCA	56.82±5.47	15×15
TLPP	52.12±6.30	17×17, $K=16$
TNPE	57.80±3.27	15×15, $K=4$
OTNPE	58.22±3.45	39×39, $K=12$
STA	60.75±3.32	28×28, $\alpha=100$

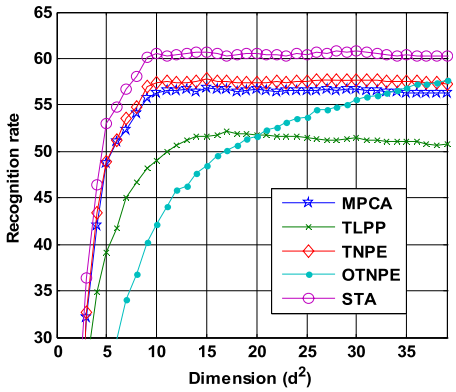

Fig. 6. Average recognition rates (%) versus the number of dimensions on the FERET database.

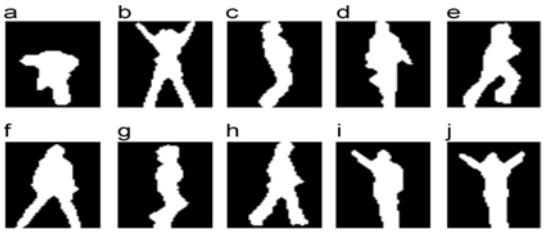

Fig. 7. Key silhouettes of 10 actions from the Weizmann database. (a) Bend. (b) Jack. (c) Jump. (d) Pjump. (e) Run. (f) Side. (g) Skip. (h) Walk. (i) Wave1. (j) Wave2.

as the same way in Section IV-A. Table V lists the recognition rates of each method and Fig. 6 shows the variations of the recognition rates versus the dimensions. Again, STA performs better than the other methods.

D. Experiments on Weizmann Action Database

The experiment was performed on the Weizmann database [55], which is a commonly used database for human action recognition. The 90 videos coming from 10 categories of actions included bending (bend), jacking (jack), jumping (jump), jumping in places (pjump), running (run), galloping sideways (side), skipping (skip), walking (walk), single-hand waving (wave1), and both-hands waving (wave2), which were performed by nine subjects. The centered key silhouettes of each action are shown in Fig. 7.

In order to represent the spatiotemporal feature of the samples, 10 successive frames of each action were used to

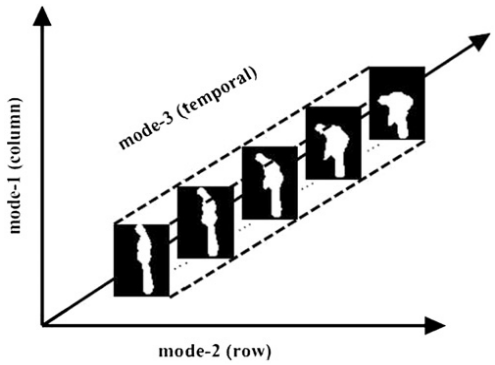

Fig. 8. Example of the bending action in the spatiotemporal domain from Weizmann database.

TABLE VI
AVERAGE RECOGNITION RATES (PERCENT), STANDARD ERROR, AND THE BEST PARAMETER VALUE OF FIVE METHODS ON THE WEIZMANN ACTION DATABASE

Method	Recognition rate (std)	Best parameter value
Baseline	70.03±3.07	----------
MPCA	70.14±2.83	10^3
TLPP	76.33±3.62	9^3, $K=5$
TNPE	75.62±3.77	10^3, $K=5$
OTNPE	77.56±3.08	10^3, $K=4$
STA	79.33±3.54	8^3, $\alpha=10$

extract the temporal feature. Fig. 8 shows a tensor sample of the bending action. Each centered frame was normalized to the size of 32 × 24 pixels. Thus the tensor sample was represented in the size of 32 × 24 × 10 pixels. It should be noted that there are no overlapped frames in any two tensors and the starting frames of the tensors are not normalized to the beginning frames of each action. Thus, the recognition tasks are difficult. Therefore, if one wants to get high recognition accuracy, the methods used for feature extraction should be robust to the starting frames and the actions' variations.

In the experiments, six action tensors of each category were randomly selected and used for training and the remaining tensors were used for testing. The experimental procedures were the same as in Section IV-A. The recognition rates of each method are listed in Table VI. The average recognition rates (%) versus the number of dimensions are shown in Fig. 9(a). The variations of the average recognition rate versus the number of training sample of different methods are shown in Fig. 9(b). It can be found that STA also outperforms the other algorithms in action tensor feature extraction. Since many experimental details are quite different from each other in different articles, all the results obtained by different algorithms in this paper are based on the same database and the same experimental background, and thus it is fair to compare them. When the number of training samples is slightly increased, the recognition rates of STA and the compared methods are above 90%. However, the superiority of the proposed STA over previous tensor-based subspace learning methods is still maintained. As indicated in [56], a possible improvement direction on the action recognition is to extend previous works on spatiotemporal alignments by incorporating manifold learning.

E. Experiments on Cambridge Hand Gesture Database

The Cambridge hand gesture database [57] consists of 900 image sequences of nine gesture classes, which are defined by three primitive hand shapes and three primitive motions. The objective of using this dataset is to classify different shapes as well as different motions at a time. Each class contains 100 image sequences (5 different illuminations × 10 arbitrary motions × 2 subjects). Each sequence was recorded in front of a fixed camera having roughly isolated gestures in space and time. Thus, fairly large intraclass variations in spatial and temporal alignment are reflected in the dataset. Some sample images of the nine different gesture classes are shown in Fig. 10. The experimental procedures are the same as in the Weizmann action database. The recognition rates of each method are listed in Table VII and the average recognition rates (%) versus the number of dimensions are shown in Fig. 9(c). It can be found that STA also outperforms the other algorithms in hand gesture tensor feature extraction.

F. Discussion

Based on the experimental results shown in the above sections, the following observations can be made.

1) Although the label information was not used in all methods, STA obtained the best recognition rates. STA performed better than the TNPE and OTNPE, which indicates that combining the L_1- and L_2-norms for sparse alignment provides more discriminative information than local linear reconstruction.

2) STA was more robust than the other compared methods. OTNPE outperformed MPCA, TLPP, and TNPE in higher dimensional subspace, but it usually obtained low accuracies in the lower dimensional subspace. That is, with increasing number of dimensions, OTNPE is more effective than MPCA, TLPP, and TNPE in different cases.

3) STA performed better than TLPP and TNPE, which indicates that sparsity is more important than locality. In addition, TLPP and TNPE had almost the same performance in action recognition, which indicates that only using the L_2-norm as a metric to measure the local geometric structure cannot always improve performance.

4) In the experiments presented in Sections IV-A and IV-B, it was found that the local neighborhood graphs could not explore the latent discriminant information for discrimination when noise was added to the data, which gave rise to the lower recognition rates of TLPP, TNPE, and OTNPE. However, STA did not introduce the local neighborhood parameter K and thus there was essential difference. In STA, the L_1- and L_2-norms are combined together for grouping the reconstruction coefficients with sparse properties; thus the advantages of robustness to data noise and the potential discriminative ability proven in [46]–[48] are encoded in the representation coefficients, which are preserved in the low-dimensional subspace. These are the essential reasons for STA to achieve good performance.

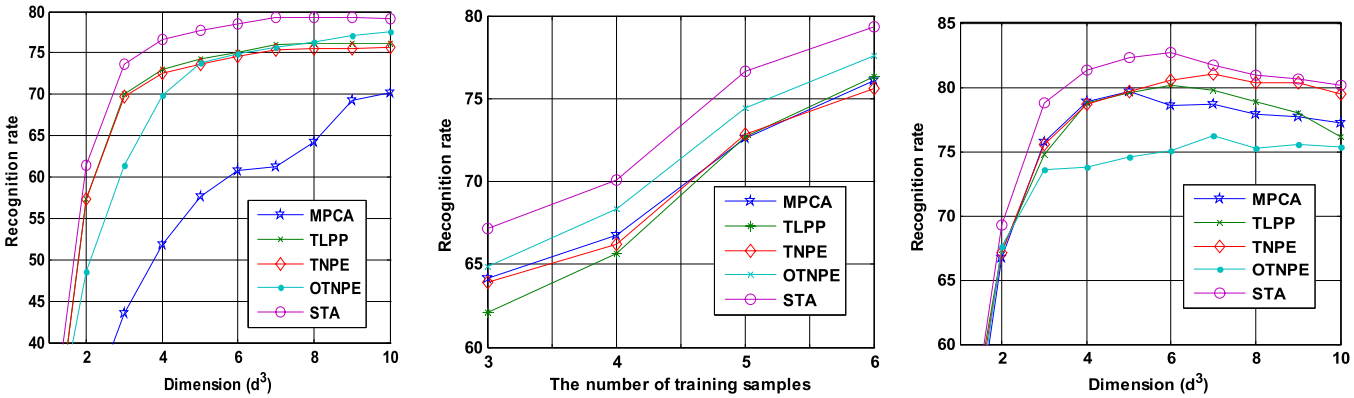

Fig. 9. (a) Variations of the average recognition rates (%) versus the number of dimensions on the Weizmann action database. (b) Average recognition rate versus the number of training sample of different methods on the Weizmann action database. (c) Variations of the average recognition rates (%) versus the number of dimensions on Cambridge hand gesture database.

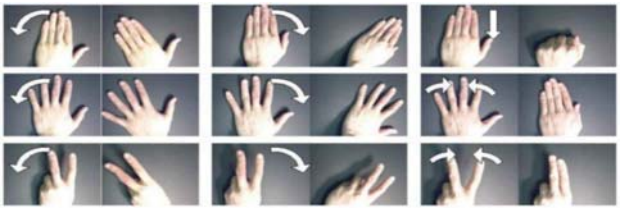

Fig. 10. Some sample images on Cambridge hand gesture database.

TABLE VII
AVERAGE RECOGNITION RATES (PERCENT), STANDARD ERROR, AND THE BEST PARAMETER VALUE OF FIVE METHODS ON THE CAMBRIDGE HAND GESTURE DATABASE

Method	Recognition rate (std)	Best parameter value
Baseline	75.11 ± 1.74	----------
MPCA	79.61 ± 1.42	5^3
TLPP	80.18 ± 1.18	6^3, $K = 4$
TNPE	81.07 ± 1.87	7^3, $K = 5$
OTNPE	76.24 ± 1.55	7^3, $K = 5$
STA	82.74 ± 1.56	6^3, $\alpha = 100$

5) From the experiments, we also found the limitation of the STA algorithm. The properties of STA were very similar to those of the compared algorithms. However, since the coefficient matrix was obtained from elastic net which selected the group of the most correlated samples, if the group of coefficients corresponding to the correlated samples was distributed in different classes (i.e., elastic net cannot explore more discriminant information than the local geometric structure in TLPP, TNPE), STA might not obtain higher recognition rate than TLPP or TNPE. However, this special case seldom happened.

V. CONCLUSION

In this paper, we chose a set of tensor learning algorithms and unified them by using the alignment technique. As a result, a general tensor learning framework was obtained. By using this framework as a platform, STA was proposed to explore the latent discriminative information by using the L_1- and L_2-norm penalty. STA preserved the sparse tensor representation coefficients, which encoded the discriminative information and robustness in the low-dimensional subspace. Experimental results on five well-known databases showed the excellent performance of STA against the state-of-the-art tensor learning methods in face recognition, objective recognition, and action recognition. It was shown that STA is robust to noise, block subtraction, rotation of the object, and starting frames of different actions. For future research, we plan to enforce the sparsity on the projection matrix/vector and investigate the sparse projection learning methods for tensor recognition.

REFERENCES

[1] M. Turk, "Eigenfaces for recognition," *J. Cognit. Neurosci.*, vol. 3, no. 1, pp. 71–86, Jan. 1991.
[2] J. Yang, D. Zhang, A. F. Frangi, and J. Yang, "Two-dimensional PCA: A new approach to appearance-based face representation and recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 1, pp. 131–137, Jan. 2004.
[3] J. Ye, "Generalized low rank approximations of matrices," *Mach. Learn.*, vol. 61, nos. 1–3, pp. 167–191, 2005.
[4] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "MPCA: Multilinear principal component analysis of tensor objects," *IEEE Trans. Neural Netw.*, vol. 19, no. 1, pp. 18–39, Jan. 2008.
[5] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "Uncorrelated multilinear principal component analysis for unsupervised multilinear subspace learning," *IEEE Trans. Neural Netw.*, vol. 20, no. 11, pp. 1820–1836, Nov. 2009.
[6] C. Fraley and A. E. Raftery, "Model-based clustering, discriminant analysis and density estimation," *J. Amer. Statist. Assoc.*, vol. 97, no. 1, pp. 611–631, 2002.
[7] J. Yang, D. Zhang, X. Yong, and J. Yang, "Two-dimensional discriminant transform for face recognition," *Pattern Recognit.*, vol. 38, no. 7, pp. 1125–1129, 2005.
[8] M. Li and B. Yuan, "2D-LDA: A statistical linear discriminant analysis for image matrix," *Pattern Recognit. Lett.*, vol. 26, no. 5, pp. 527–532, 2005.
[9] S. Yan, D. Xu, Q. Yang, L. Zhang, X. Tang, and H.-J. Zhang, "Multilinear discriminant analysis for face recognition," *IEEE Trans. Image Process.*, vol. 16, no. 1, pp. 212–220, Jan. 2007.
[10] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed. San Diego, CA, USA: Academic Press, 1990.
[11] D. Tao, X. Li, X. Wu, and S. J. Maybank, "General tensor discriminant analysis and Gabor features for gait recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 10, pp. 1700–1715, Oct. 2007.
[12] H. Li, T. Jiang, and K. Zhang, "Efficient and robust feature extraction by maximum margin criterion," *IEEE Trans. Neural Netw.*, vol. 17, no. 1, pp. 157–165, Jan. 2006.

[13] W. Yang, J. Wang, M. Ren, and J. Yang, "Feature extraction based on Laplacian bidirectional maximum margin criterion," *Pattern Recognit.*, vol. 42, no. 11, pp. 2327–2344, 2009.
[14] R.-X. Hu, W. Jia, D.-S. Huang, and Y.-K. Lei, "Maximum margin criterion with tensor representation," *Neurocomputing*, vol. 73, nos. 10–12, pp. 1541–1549, Jun. 2010.
[15] S. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 22, pp. 2323–2326, 2000.
[16] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 22, pp. 2319–2323, 2000.
[17] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Comput.*, vol. 15, no. 6, pp. 1373–1396, Jun. 2003.
[18] Z. Zhang and H. Zha, "Principal manifolds and nonlinear dimensionality reduction via tangent space alignment," *SIAM J. Sci. Comput.*, vol. 26, no. 1, pp. 313–338, 2004.
[19] Y. Bengio, J. F. Paiement, P. Vincent, O. Delalleau, N. Roux, and M. Ouimet, "Out-of-sample extensions for LLE, Isomap, MDS, Eigenmaps, and spectral clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2004, pp. 1–8.
[20] X. He and P. Niyogi, "Locality preserving projections," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 16. 2004, pp. 153–160.
[21] X. He, D. Cai, S. Yan, and H.-J. Zhang, "Neighborhood preserving embedding," in *Proc. 10th IEEE Int. Conf. Comput. Vis.*, vol. 2, Oct. 2005, pp. 1208–1213.
[22] E. Kokiopoulou and Y. Saad, "Orthogonal neighborhood preserving projections: A projection-based dimensionality reduction technique," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 12, pp. 2143–2156, Dec. 2007.
[23] T. Zhang, J. Yang, D. Zhao, and X. Ge, "Linear local tangent space alignment and application to face recognition," *Neurocomputing*, vol. 70, nos. 7–9, pp. 1547–1553, Mar. 2007.
[24] Y. Li, D. Luo, and S. Liu, "Orthogonal discriminant linear local tangent space alignment for face recognition," *Neurocomputing*, vol. 72, nos. 4–6, pp. 1319–1323, Jan. 2009.
[25] T. Zhang, D. Tao, X. Li, and J. Yang, "Patch alignment for dimensionality reduction," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1229–1313, Sep. 2009.
[26] A. Rozza, G. Lombardi, E. Casiraghi, and P. Campadelli, "Novel Fisher discriminant classifiers," *Pattern Recognit.*, vol. 45, no. 1, pp. 3725–3737, 2012.
[27] T. G. Kolda, "Orthogonal tensor decompositions," *SIAM J. Matrix Anal. Appl.*, vol. 23, no. 1, pp. 243–255, 2001.
[28] L. Lathauwer, B. De Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM J. Matrix Anal. Appl.*, vol. 21, no. 4, pp. 1253–1278, 2000.
[29] L. Lathauwer, B. De Moor, and J. Vandewalle, "On the best rank-1 and randk-(R1, R2, ...RN) approximation of high-order tensors," *SIAM J. Matrix Anal. Appl.*, vol. 21, no. 4, pp. 1324–1342, 2000.
[30] M. A. O. Vasilescu and D. Terzopoulos, "Multilinear analysis of image ensembles: Tensiofaces," in *Proc. Eur. Conf. Comput. Vis.*, 2002, pp. 447–460.
[31] M. A. O. Vasilescu and D. Terzopoulos, "Multilinear image analysis for facial recognition," in *Proc. 16th Int. Conf. Pattern Recognit.*, vol. 2. 2002, pp. 511–514.
[32] M. A. O. Vasilescu and D. Terzopoulos, "Multilinear subspace analysis for image ensembles," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2003, pp. 93–99.
[33] S. Rana, W. Liu, M. Lazarescu, and S. Venkatesh, "A unified tensor framework for face recognition," *Pattern Recognit.*, vol. 42, no. 11, pp. 2850–2862, Nov. 2009.
[34] X. He, D. Cai, and P. Niyogi, "Tensor subspace analysis," in *Proc. Adv. Neural Inf. Process. Syst.*, 2005, pp. 499–506.
[35] G. Dai and D. Yeung, "Tensor embedding methods," in *Proc. 21st AAAI Conf. Artif. Intell.*, vol. 1. 2005, pp. 330–335.
[36] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin, "Graph embedding and extensions: A general framework for dimensionality reduction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 40–51, Jan. 2007.
[37] S. Liu and Q. Ruan, "Orthogonal tensor neighborhood preserving embedding for facial expression recognition," *Pattern Recognit.*, vol. 44, no. 7, pp. 1497–1513, Jul. 2011.
[38] X. Li, S. Lin, S. Yan, and D. Xu, "Discriminant locally linear embedding with high-order tensor data," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 2, pp. 342–352, Apr. 2008.
[39] W. Zhang, Z. Lin, and X. Tang, "Tensor linear Laplacian discrimination (TLLD) for feature extraction," *Pattern Recognit.*, vol. 42, no. 9, pp. 1941–1948, Sep. 2009.
[40] F. Wang and X. Wang, "Neighborhood discriminant tensor mapping," *Neurocomputing*, vol. 72, nos. 7–9, pp. 2035–2039, Mar. 2009.
[41] Y. Liu, Y. Liu, and K. C. C. Chan, "Tensor distance based multilinear locality-preserved maximum information embedding," *IEEE Trans. Neural Netw.*, vol. 21, no. 11, pp. 1848–1854, Nov. 2010.
[42] D. Xu, S. Yan, D. Tao, S. Lin, and H.-J. Zhang, "Marginal Fisher analysis and its variants for human gait recognition and content-based image retrieval," *IEEE Trans. Image Process.*, vol. 16, no. 11, pp. 2811–2821, Nov. 2007.
[43] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "A survey of multilinear subspace learning for tensor data," *Pattern Recognit.*, vol. 44, no. 7, pp. 1540–1551, Jul. 2011.
[44] G. Guy and G. Medioni, "Inferring global perceptual contours from local features," *Int. J. Comput. Vis.*, vol. 20, nos. 1–2, pp. 113–133, 1996.
[45] P. Mordohai and G. Medioni, "Dimensionality estimation manifold learning and function approximation using tensor voting," *J. Mach. Learn. Res.*, vol. 11, no. 1, pp. 411–450, 2010.
[46] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.
[47] L. Qiao, S. Chen, and X. Tan, "Sparsity preserving projections with applications to face recognition," *Pattern Recognit.*, vol. 43, no. 1, pp. 331–341, Jan. 2010.
[48] B. Cheng, J. Yang, S. Yan, Y. Fu, and T. S. Huang, "Learning with L1 graph for image analysis," *IEEE Trans. Image Process.*, vol. 19, no. 4, pp. 858–866, Apr. 2010.
[49] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. R. Statist. Soc.*, vol. 58, no. 1, pp. 267–288, 1996.
[50] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *J. R. Statist. Soc. Series B (Statist. Methodol.)*, vol. 67, no. 2, pp. 301–320, 2005.
[51] R. Baraniuk, "A lecture on compressive sensing," *IEEE Signal Process. Mag.*, vol. 24, no. 4, pp. 118–121, Jul. 2007.
[52] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriengman, "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 711–720, Jul. 1997.
[53] S. A. Nene, S. K. Nayar, and H. Murase, "Columbia object image library (COIL-20)," Dept. Comput. Sci., Columbia Univ., New York, NY, USA, Tech. Rep. CUCS-005-96, 1996.
[54] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss, "The FERET evaluation methodology for face recognition algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 10, pp. 1090–1104, Oct. 2000.
[55] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 12, pp. 2247–2253, Dec. 2007.
[56] D. Gong and G. Medioni, "Dynamic manifold warping for view invariant action recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2011, pp. 571–578.
[57] T.-K. Kin, S.-B. Wong, and R. Cipolla, "Tensor canonical correlation analysis for action classification," in *Proc. IEEE CVPR*, Jun. 2007, pp. 1–8.

Zhihui Lai received the B.S. degree in mathematics from South China Normal University, Guangzhou, China, the M.S. degree from Jinan University, Guangzhou, China, and the Ph.D. degree in pattern recognition and intelligence system from the Nanjing University of Science and Technology, Nanjing, China, in 2002, 2007, and 2011, respectively.

He has been a Research Associate and Post-Doctoral Fellow with The Hong Kong Polytechnic University, Hong Kong, from 2010 to 2013. His current research interests include face recognition, image processing and content-based image retrieval, pattern recognition, compressive sense, human vision modelization, and applications in the fields of intelligent robot research.

Wai Keung Wong received the Ph.D. degree from The Hong Kong Polytechnic University, Hong Kong.

He is an Associate Professor with The Hong Kong Polytechnic University. He has published more than 50 scientific articles in refereed journals, including the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, *Pattern Recognition*, the *International Journal of Production Economics*, the *European Journal of Operational Research*, the *International Journal of Production Research, Computers in Industry*, and the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS. His current research interests include artificial intelligence, pattern recognition, and optimization of manufacturing scheduling, planning, and control.

Yong Xu (M'06) received the B.S. and M.S. degrees from the Air Force Institute of Meteorology, Nanjing, China, in 1994 and 1997, respectively, and the Ph.D. degree in pattern recognition and intelligence system from the Nanjing University of Science and Technology, Nanjing, in 2005.

He was with the Shenzhen Graduate School, Harbin Institute of Technology (HIT), Harbin, China, from 2005 to 2007, as a Post-Doctoral Research Fellow. He is currently a Professor with the Shenzhen Graduate School, HIT. He was a Research Assistant Researcher with The Hong Kong Polytechnic University, Hong Kong, from 2007 to 2008. He has published more than 40 scientific papers. His current research interests include pattern recognition, biometrics, and machine learning.

Cairong Zhao received the B.S. degree from Jilin University, Jilin, China, the M.S. degree from the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun, China, and the Ph.D. degree from the Nanjing University of Science and Technology, Nanjing, China, in 2011, 2006, and 2003, respectively.

He is currently an Assistant Professor with Tongji University, Shanghai, China. He is the author of more than 15 scientific papers in pattern recognition and computer vision. His current research interests include face recognition, feature extraction, and computer vision.

Mingming Sun received the B.S. degree in mathematics from Xinjiang University, Urumqi, China, in 2002, and the Ph.D. degree in pattern recognition and intelligence systems from the Department of Computer Science, Nanjing University of Science and Technology (NUST), Nanjing, China, in 2007.

He is currently a Lecturer with the School of Computer Science and Technology, NUST. His current research interests include pattern recognition, machine learning, and image processing.