Research Article

Tensorial Kernel Principal Component Analysis for Action Recognition

Cong Liu, Xu Wei-sheng, and Wu Qi-di

School of Electrical and Information Engineering, Tongji University, Shanghai 201804, China

Correspondence should be addressed to Cong Liu; 1210482congliu@tongji.edu.cn

Received 25 July 2013; Accepted 30 August 2013

Academic Editor: Gelan Yang

Copyright © 2013 Cong Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We propose the Tensorial Kernel Principal Component Analysis (TKPCA) for dimensionality reduction and feature extraction from tensor objects, which extends the conventional Principal Component Analysis (PCA) in two perspectives: working directly with multidimensional data (tensors) in their native state and generalizing an existing linear technique to its nonlinear version by applying the kernel trick. Our method aims to remedy the shortcomings of multilinear subspace learning (tensorial PCA) developed recently in modelling the nonlinear manifold of tensor objects and brings together the desirable properties of kernel methods and tensor decompositions for significant performance gain when the data are multidimensional and nonlinear dependencies do exist. Our approach begins by formulating TKPCA as an optimization problem. Then, we develop a kernel function based on Grassmann Manifold that can directly take tensorial representation as parameters instead of traditional vectorized representation. Furthermore, a TKPCA-based tensor object recognition is also proposed for application of the action recognition. Experiments with real action datasets show that the proposed method is insensitive to both noise and occlusion and performs well compared with state-of-the-art algorithms.

1. Introduction

Recent years have witnessed a dramatic increase in the quantity of multidimensional data which are so large and complex that it becomes difficult to process them using traditional data processing applications. Hence, there is a growing need for the development and application of feature extraction and dimensionality reduction to analyze multidimensional data.

Tensor provides a natural and efficient way to describe such multidimensional data. The entries of tensor are addressed by more than two indices. The number of indices defines the order of the tensor, and each index defines one of the so-called “modes.” In reality, there are a lot of data that can be formed by tensor. For example, two order tensors include gray level images in computer vision and pattern recognition [1–4], multichannel EEG signals in biomedical engineering [5–7]. Three-order tensors include diffusion tensor imaging (DTI) in brain research [8], hyperspectral cube in remote sensing [9], silhouette sequences in gait analysis [10], and gray video sequences in action recognition [11, 12]. There are also multidimensional signals that can be formed in more

than three-order tensor in the application of color video sequences surveillance, social network analysis [13], and so forth. Figure 1 shows two examples of 3-order tensor for a silhouette sequence and a reconstructed fiber tracts of human brain measured by DTI. The hypothesis behind DTI is that the bundles of fiber tracts make the water diffuse asymmetrically and DTI derives tract directional information from 3-order tensors describing this anisotropy.

Principal Component Analysis (PCA) [14] is one of the most important techniques in the class of unsupervised learning algorithms which linearly transforms a number of possibly correlated variables into uncorrelated features called *principal components* (PCs). The transformation is performed to find directions of maximal variation. Normally, only a few principal components can account for most of the variation in the original dataset. However, PCA is not suitable to discover nonlinear relationships among the original variables. To overcome this limitation, Schölkopf et al. [15] originally proposed Kernel Principal Component Analysis (KPCA) [15] which performs PCA in a Reproducing Kernel Hilbert Space (RKHS) rather than in the input space. In principle, the

FIGURE 1: Two examples of 3-order tensor: (a) a gait silhouette sequences with the column, row, and time mode; (b) a diffusion tensor imaging (DTI) scan of fiber tracts in human brain which derives tract directional information from 3-order tensors that describe anisotropic diffusion of water.

kernel methods nonlinearly map a set of training samples to a higher dimensional RKHS where conventional linear PCA is performed, with the resulting subspaces being nonlinear with regards to the original input space. In practice, the mapping is performed implicitly via the *kernel trick* [15], where an appropriately chosen kernel function is used to evaluate dot products of mapped input space vectors without having to explicitly carry out the mapping.

As a classical dimensionality reduction method, PCA and KPCA have been widely used in extracting feature from tensor objects. However, before feeding tensorial data to PCA or KPCA, the tensors have to be typically transformed into long vectors by concatenating the tensor entry-wise. This will present several problems. Firstly, the integral structure of tensor is disintegrated; therefore, the information correlated with surrounding entries could be lost. Secondly, the vectorized representation lies in a very high dimensional space, which will bring the *Curse of Dimensionality dilemma* [16]. Thirdly, only sparse data are available in many application areas such as web document classification, face recognition, and disease classification based on gene expression profiling; consequently, the *small sample size (SSS)* problem [17, 18] is inevitable there.

Due to the challenges above, recently, interests have grown in multilinear subspace learning (hereinafter referred to as *tensorial PCA*) that reduces dimensionality of multidimensional data directly from their tensorial representations. Initiated by the pioneer work of Yang et al. [3], a two dimensional PCA (2DPCA) algorithm is proposed. This algorithm solves for a linear transformation that projects an image to a low dimensional matrix while maximizing the variance measure. It works directly on image matrices, but there is only one linear transformation in the 2-mode. Thus, the image data are projected in the 2-mode (the row mode) only, while the projection in the 1-mode (the column mode) is ignored. A more general algorithm named the generalized PCA (GPCA) is introduced in [1], which takes into account the spatial correlation between the image pixels in neighborhood and applies double linear

transforms to both the left and right sides of input image matrices. However, it is formulated for matrices only. Later, the work of multilinear PCA (MPCA) [19] generalizes GPCA to work for tensors of any order, where the objective is to find a *core tensor* (see Section 2.2) that captures most of the original tensorial input variations. In [20], two robust MPCA (RMPCA) algorithms are proposed, where iterative algorithms are derived on the basis of Lagrange multipliers to deal with sample outliers and intrasample outliers. In [21], the nonnegative MPCA (NMPCA) extends MPCA to constrain the projection matrices to be nonnegative. NMPCA preserves the nonnegativity of the original tensor samples that is important when the underlying data have physical or psychological interpretation. RMPCA and NMPCA can be considered as an extension of MPCA. Although the above algorithms exploit tensorial structure for subspace learning, they are formulated on the multilinear projection of tensor to tensor only. Moreover, there exists a different projection scheme that projects tensor to vector. The tensor rank-one decomposition (TROD) algorithm introduced in [22] is one example. This algorithm looks for a second-order projection that projects an image to a low dimensional vector while minimizing a least square (reconstruction) error measure. Nonetheless, the input data are not centered before learning and the work is formulated only for matrices. Later, an uncorrelated MPCA (UMPCA) algorithm is proposed in [23] and adopted in [24], which extracts uncorrelated multilinear features through tensor to vector projection while capturing most of the variation in the original data input.

Although tensorial PCA methods achieve better performance than naive PCA, there remain several shortcomings in them. Firstly, the nonconvex optimality criterion and the suboptimal iterative solver, used by tensorial PCA, do not have any guarantee of global optimality of the solution found. Secondly, the objective of most tensorial PCA algorithms is to find the most expressive core tensor for each input tensor. Therefore, the disadvantage arising in this way is that more storage is required to represent the core tensor compared with the scalar representation used in PCA or KPCA. Last but not

TABLE 1: Comparison between several PCA Algorithms.

Method	Linear/nonlinear	Vector-based/tensor-based
PCA	Linear	Vector-based
KPCA	Nonlinear	Vector-based
Tensorial PCA	Linear	Tensor-based
TKPCA	Nonlinear	Tensor-based

important, none of the above methods takes into account the nonlinear relationships among the tensorial data. In other words, the tensorial PCA algorithms are multilinear method that neglects the higher-order statistics existed between neighborhood. However, it is well known that the object appearances lie on a nonlinear low-dimensional manifold in applications such as action recognition and face recognition where pose or illumination variations exist. Tensorial PCA methods cannot effectively model such nonlinearity, and this prevents them from higher recognition accuracy.

1.1. TKPCA. Motivated by the above drawbacks, in this paper, we propose a *Tensorial Kernel Principal Component Analysis* (TKPCA) to extend the conventional PCA to its kernelized tensor counterpart. TKPCA aims to overcome the drawbacks in traditional PCA and MPCA and brings together the desirable properties of kernel methods and tensor decompositions (see Section 2.1) for significant performance gain when the data are multidimensional and nonlinear dependencies do exist. Table 1 gives connections and differences with other PCA techniques. To the best of our knowledge, this is the first study that addresses the TKPCA problem.

Our approach begins by formulating TKPCA as an optimization problem. Unlike traditional PCA, the *Covariance Matrix* cannot be formed directly from tensorial data. We thus derive the TKPCA in a support vector machine (SVM) fashion, which leads to a convex optimization and fits into the *primal-dual framework* [16]. The primal problem can be solved by dual representation through kernel trick which is based on the Mercer theorem related to positive definite kernels [25]. Subsequently, a kernel function with tensorial inputs (*tensorial kernel*) can be plugged into the dual solution, which takes the nonlinear structure of tensorial representation into account. Furthermore, we design a novel tensorial kernel based on Grassmann Manifold and the positive definiteness which is also proved.

The benefits of the TKPCA can be summarized as follows.

- (i) Tensor representation of multidimensional data reduces the SSS problem and the Curse of Dimensionality phenomenon, facilitating a precise classification performance even for low number of training samples and complex data structure.
- (ii) Kernel method remedies the shortcoming of tensorial PCA in modelling the nonlinear manifold of tensor objects.
- (iii) TKPCA is equivalent to performing a standard KPCA except that the parameters of kernel function are in natural tensor representations, and in general, KPCA achieves higher compression rate (CR) than

tensorial PCA. Therefore, TKPCA offers better CR performance.

- (iv) TKPCA is insensitive to environmental variations and more robust to noise. This is because the Grassmann based kernel function compares similarity between subspaces that are low dimensional approximation of original data and the approximation can “fill in” the missing data. Moreover, the kernel is derived with the geodesic distance on the Grassmann manifold other than the Euclidean distance. Therefore, the TKPCA is expected to capture the topological structure underlying tensor dataset.
- (v) TKPCA is a convex optimization problem which means that any local minimum must also be global. Therefore, TKPCA do not suffer from the issue of local minima as tensorial PCA.

The main contributions of this paper include the following.

- (i) A new TKPCA is introduced for nonlinear dimensionality reduction and feature extraction from tensor object, by encoding the structured information embedded in the tensorial data into the kernels framework.
- (ii) A novel tensorial kernel function is proposed based on Grassmann kernel which can directly measure the similarity between tensorial inputs. Furthermore, the strict positive definiteness proof of proposed kernel function is given.
- (iii) A recognition system is developed for action recognition by selecting more discriminative features after TKPCA projection.

The rest of this paper is organized as follows. Section 2 introduces basic notations, kernel methods concepts, and the notion of multilinear projection for dimensionality reduction. In Section 3, the problem of TKPCA is formulated. Then, the detailed algorithm is summarized and discussed in detail. Moreover, a TKPCA-based tensor object recognition is proposed for application of action recognition. Section 4 lists experiments on action recognition and compares performance against state-of-the-art algorithm. We also assess the noise robustness and investigate sensitivity against occlusion and misalignment. Finally, Section 5 summarizes the major findings of this work.

2. Background and Notation

This section firstly reviews the notations and some basic multilinear operations that are necessary in defining our TKPCA. Then, a multilinear projection is introduced for dimensionality reduction and feature extraction from tensor object. We provide the conceptual foundations of kernel methods in the last part.

2.1. Notations and Basic Multilinear Algebra. Following the notation conventions in multilinear algebra, pattern recognition, and adaptive learning literature [26–29], vectors

TABLE 2: List of important notations.

Notation	Description
$\mathcal{X}_m$	The m th input tensor sample, $m = 1, \dots, M$
$\mathcal{Y}_m$	The projection of $\mathcal{X}_m$ on $\{U_{(n)}\}_{n=1}^N$
y	The projection on eigenvector
$\mathcal{A}_{(n)}$	The mode- n matricization of $\mathcal{A}$
$\mathcal{X}_{m(n)}$	The mode- n matricization of m th tensor sample
$\text{vec}(\mathcal{A})$	The vectorization of $\mathcal{A}$
$U^{(n)}$	The n th projection matrix
$U_{\mathcal{X}}^{(n)}$	The orthonormal base matrix of $\mathcal{X}_{(n)}$
$\mathcal{K}$	The Gram or Kernel Matrix
M	The number of training samples
N	The order of a tensor object, the number of indices/modes
P	The n -mode dimensionality in the projected space or the number of dominant eigenvectors
Q	The number of most discriminative components of y
$\phi(\cdot)$	The map from vector to RKHS
$\Phi(\cdot)$	The map from tensor to HSF

FIGURE 2: Visual illustration of the 1-mode unfolding of a third-order tensor.

are denoted by lowercase boldface letters, for example, $\mathbf{x}$; matrices by uppercase boldface, for example, $\mathbf{X}$; and tensors by calligraphic letters, for example, $\mathcal{X}$. Their elements are denoted with indices in parentheses. Indices are denoted by lowercase letters, spanning the range from 1 to the uppercase letter of the index, for example, $n = 1, 2, 3, \dots, N$. In addressing part of a vector/matrix/tensor, “:” denotes the full range of the respective index, and $n_1 : n_2$ denotes indices ranging from n_1 to n_2 . In this paper, only real-valued data are considered. Table 2 summarizes the important symbols used in this paper for quick reference.

An N th-order tensor is denoted as $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$, which is addressed by N indices $i_n, n = 1, \dots, N$, with each i_n addressing the n -mode of $\mathcal{A}$.

The n -mode vectors of $\mathcal{A}$ are defined as the I_n dimensional vectors obtained from $\mathcal{A}$ by varying its index i_n while keeping all the other indices fixed. The mode- n matricization of $\mathcal{A}$ is denoted as

$$\mathcal{A}_{(n)} \in \mathbb{R}^{I_n \times (I_1 \times \dots \times I_{n-1} \times I_{n+1} \times \dots \times I_N)}, \quad (1)$$

where the column vectors of $\mathcal{A}_{(n)}$ are the n -mode vectors of $\mathcal{A}$. Figure 2 illustrates the 1-mode (column mode) matricization of a third-order tensor.

FIGURE 3: The illustration of HOSVD scheme.

The n -mode product of a tensor $\mathcal{A}$ by a matrix $U \in \mathbb{R}^{j_n \times i_n}$, denoted by $\mathcal{A} \times_n U$, is a tensor defined with entries:

$$\begin{aligned} (\mathcal{A} \times_n U)(i_1, \dots, i_n - 1, j_n, i_n + 1, i_N) \\ = \sum_{i_n} \mathcal{A}(i_1, \dots, i_N) \cdot U(j_n, i_n). \end{aligned} \quad (2)$$

The two most commonly used tensor decompositions are Tucker and CANDECOMP/PARAFAC (CP). Both of which can be regarded as higher-order generalizations of the matrix Singular Value Decomposition (SVD). Let $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ denote an N th-order tensor; then, Tucker decomposition is defined as follows:

$$\mathcal{A} = \mathcal{S} \times_1 U^{(1)} \times_2 U^{(2)} \dots \times_N U^{(N)}, \quad (3)$$

where $\mathcal{S} \in \mathbb{R}^{P_1 \times P_2 \times \dots \times P_N}$, with $P_n < I_n$ denotes the core tensor and $U^{(n)} \in \mathbb{R}^{I_n \times P_n}$. When all $\{U^{(n)}\}_{n=1}^N$ are orthonormal and the core tensor is all orthogonal, this model is called High Order Singular Value Decomposition (HOSVD) [30]; see Figure 3. When all factor matrices have the same number of components and the core tensor is superdiagonal, Tucker model is simplified to CP model. In general, CP model is considered to be a multilinear low rank approximation, while Tucker model is regarded as a multilinear subspace approximation.

The distance between tensors $\mathcal{A}$ and $\mathcal{B}$ can be measured by the Frobenius norm [31], $\text{dist}(\mathcal{A}, \mathcal{B}) = \|\mathcal{A} - \mathcal{B}\|_F$. Although this is a tensor-based measure, it is equivalent to a distance measure of corresponding vector representations. Let $\text{vec}(\mathcal{A})$ be the vector representation (vectorization) of $\mathcal{A}$; then, $\text{dist}(\mathcal{A}, \mathcal{B}) = \|\text{vec}(\mathcal{A}) - \text{vec}(\mathcal{B})\|_2$. This implies that the distance between two tensors equals to the Euclidean distance between their vectorized representations.

2.2. Multilinear Principal Component Analysis. An N th-order tensor $\mathcal{X}$ resides in the tensor (multilinear) space $\mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2} \otimes \mathbb{R}^{I_N}$, where $\mathbb{R}^{I_1}, \dots, \mathbb{R}^{I_2}, \mathbb{R}^{I_N}$ are the N vector (linear) spaces. For typical image and video tensor objects, although the corresponding tensor space is of high dimensionality, tensor objects typically are embedded in a lower dimensional tensor subspace (or manifold), in analogy to the (vectorized) face image embedding problem where vector image inputs reside in a low-dimensional subspace of the original input space [15]. Thus, it is possible to find a tensor subspace that

captures most of the variation in the input tensor objects, and it can be used to extract features for recognition and classification applications. To achieve this objective, let us assume that a set of M tensor objects $\{\mathcal{X}\}_{m=1}^M$ is available for training. Each tensor object $\mathcal{X}_m \in \mathbb{R}_{I_1 \times I_2 \times \dots \times I_N}$ has values in a tensor space $\mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2} \otimes \dots \otimes \mathbb{R}^{I_N}$, where I_n is the n -mode dimension of the tensor. The objective of *Multilinear Principal Component Analysis of Tensors* (MPCA) [19] is to find a multilinear transformation $\{\mathbf{U}^{(n)} \in \mathbb{R}_{I_n \times P_n}\}_{n=1}^N$ that maps the original tensor space $\mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2} \otimes \dots \otimes \mathbb{R}^{I_N}$ into a tensor subspace $\mathbb{R}^{P_1} \otimes \mathbb{R}^{P_2} \otimes \dots \otimes \mathbb{R}^{P_N}$, with $P_n < I_n$, that is, $\mathcal{Y}_m = \mathcal{X}_m \times_1 \mathbf{U}^{(1)^T} \times_2 \mathbf{U}^{(2)^T} \dots \times_N \mathbf{U}^{(N)^T}$, such that $\{\mathcal{Y} \in \mathbb{R}^{P_1} \otimes \mathbb{R}^{P_2} \otimes \dots \otimes \mathbb{R}^{P_N}\}_{m=1}^M$ captures most of the variations observed in the original tensor objects, assuming that these variations are measured by the total tensor scatter: $\sum_{m=1}^M \|\mathcal{X}_m - \bar{\mathcal{X}}\|_F^2$, where $\bar{\mathcal{X}}$ is the empirical mean. In other words, the MPCA objective is the determination of the N projection matrices that maximize the total tensor scatter.

However, there is no known optimal solution which allows for the simultaneous optimization of the N projection matrices. Instead of global optimization, [19] propose a suboptimal iterative solution.

2.3. Kernel Methods. Kernel methods [25] have gained considerable popularity during the last few decades, providing attractive solutions to a variety of problems. The strategy adopted is to embed the data into a space where the patterns can be discovered as linear relations. This will be done in a modular fashion: the first module that performs a nonlinear mapping into *RKHS* or *feature space* implicitly through a kernel function and the second module that is a specific learning algorithm in a dual form designed to discover linear relations in the feature space. The basic assumption is that the obtained feature space reflects nonlinear structure of input data. Hence, the only information that is required is the similarity measure in the feature space, which leads us to avoid explicitly having to know the nonlinear mapping function. Instead, the similarity measure of two data points in the feature space, that is, an inner product, should be appropriately defined by a reproducing kernel formulated in the input space, which is called a *kernel trick*.

The main ingredients of kernel methods are elucidated through kernel PCA, given a set of centered observations $\{\mathbf{x}_m\}_{m=1}^M \in \mathbb{R}^I$ independent and identically distributed (i.i.d.) according to the generator $p(\mathbf{x})$. PCA optimally chooses a subspace that captures most of the variance of the data. The first principal component is defined as $y_m = \mathbf{w}^T \mathbf{x}_m$, where the weight $\mathbf{w}$ can be estimated as the leading eigenvector of sample covariance matrix $\mathbf{C} = (1/M) \sum_{m=1}^M \mathbf{x}_m \mathbf{x}_m^T$, satisfying $\lambda \mathbf{w} = \mathbf{C} \mathbf{w}$ which implies that $\mathbf{w}$ can be also expressed as a linear combination of the training samples, that is, $\mathbf{w} = \sum_{m=1}^M \alpha_m \mathbf{x}_m$. Thus, the *dual* representation of PCA is $M \lambda \boldsymbol{\alpha} = \mathcal{H} \boldsymbol{\alpha}$ with $\mathcal{H}(i, j) = \langle \mathbf{x}_i, \mathbf{x}_j \rangle$, referred to *Kernel Matrix* that consist of inner products between all pairs of training samples. After estimation of $\boldsymbol{\alpha}$ by diagonalizing $\mathcal{H}$, the first principal component of test sample $\mathbf{x}'$ is obtained

by $y' = \mathbf{w}^T \mathbf{x}' = \sum_{m=1}^M \alpha_m \langle \mathbf{x}_m, \mathbf{x}' \rangle$. Note that, for the dual representation of PCA, all information from training samples is given by the Kernel Matrix $\mathcal{H}$. This matrix acts as an information bottleneck, as all the information is available to a kernel algorithm.

let us consider an embedding or map $\phi : \mathbb{R}^I \rightarrow F$, where F refer to feature space which could have an arbitrarily large dimensionality. The pairwise inner products in feature space can be computed efficiently directly from the original data items using a kernel function $k(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$. Hence, the Kernel Matrix $\mathcal{H}$ can be computed without explicit knowledge of $\phi(\cdot)$. Finally, the first principle component of test sample embedded into feature space $\phi(\mathbf{x}')$ is computed by $\sum_{m=1}^M \alpha_m k(\mathbf{x}_m, \mathbf{x}') = \sum_{m=1}^M \alpha_m \langle \phi(\mathbf{x}_m), \phi(\mathbf{x}') \rangle$.

3. Kernel Principal Component Analysis of Tensor Objects

In this section, we propose a novel unsupervised learning method, called Tensorial Kernel Principal Component Analysis (TKPCA), for nonlinear dimensionality reduction and feature extraction from tensor objects. Unlike conventional PCA, there is no closed-form formula for Covariance Matrix of tensorial data. Therefore, our approach begins by formulating TKPCA as an optimization problem. Then, we develop a kernel function that can directly take tensorial data as parameters other than vectorial ones. Moreover, the detailed algorithm is summarized and discussed. A TKPCA-based tensor object recognition is also proposed for application of action recognition.

3.1. TKPCA as an Optimization Problem. As we have seen in Section 2.3, the KPCA is classically derived by constructing the Covariance Matrix explicitly. However, in statistics and probability theory, the Covariance Matrix is a matrix of covariance between elements of a random *vector*. This means that, before feeding tensors into Covariance Matrix, we have to transform them into vectors firstly, which conflicts with our purpose of this paper. To solve this difficulty, we derive TKPCA as an optimization problem. In this way, the explicit construction of Covariance Matrix is bypassed. Note that there is a number of other ways to derive the PCA [14], and a Generalized Covariance Matrix (GCM) [32] concept also provides an alternative solution from other perspective.

Given is a set of centered tensorial observations $\{\mathcal{X}_m\}_{m=1}^M \in \mathbb{R}_{I_1 \times I_2 \times \dots \times I_N}$, i.i.d., according to the generator $p(\mathcal{X})$ and a nonlinear mapping $\phi : \mathbb{R}_{I_1 \times I_2 \times \dots \times I_N} \rightarrow HSF$, where HSF refer to a space of multilinear functions corresponding the infinite dimensional tensors which could have an arbitrarily large dimensionality (see Section 3.2), to the objective is to optimally choose a subspace that captures most of the variance from tensorial samples. The starting point is to define projection onto weight w as

$$y_m = \mathbf{w}_m^T \phi(\mathcal{X}_m). \quad (4)$$

Recall that, while least squares support vector machine classifiers (LS-SVM) have a natural link with *kernel Fisher*

discriminant analysis (minimizing the within class scatter around targets +1 and -1), for TKPCA, we can take the interpretation of a one-class modeling problem with zero target value around which one maximizes the variance. Let us now reformulate the TKPCA problem as follows:

$$\max_{\mathbf{w}} \sum_{m=1}^M [0 - \mathbf{w}^T (\phi(\mathcal{X}_m))]^2, \quad (5)$$

where zero is considered as a single target value. For Kernel Fisher discriminant analysis one aims at minimizing the within scatter around the targets, while for TKPCA analysis one is interested in finding the direction(s) for which the variance is maximal. This interpretation leads to the following primal optimization problem

$$\max_{\mathbf{w}, y} J_p(\mathbf{w}, y) = \frac{\gamma}{2} \sum_{m=1}^M y_m^2 - \frac{1}{2} \mathbf{w}^T \mathbf{w}, \quad (6)$$

where $\gamma \in \mathbb{R}^+$. Equation (6) maximizes the empirical variance of y_m around value 0 while keeping the norm of the corresponding parameter $\mathbf{w}$ small by the regularization term $-(1/2)\mathbf{w}^T \mathbf{w}$. one can also include a bias term; see [33].

The Lagrangian corresponding to (6) is

$$\mathcal{L}(\mathbf{w}, y; \boldsymbol{\alpha}) = \frac{\gamma}{2} \sum_{m=1}^M y_m^2 - \sum_{m=1}^M \alpha_m (y_m - \mathbf{w}^T \phi(\mathbf{n})) - \frac{1}{2} \mathbf{w}^T \mathbf{w}, \quad (7)$$

with conditions for optimality given by

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \mathbf{w}} = 0 &\longrightarrow \mathbf{w} = \sum_{m=1}^M \alpha_m \phi(\mathcal{X}_m), \\ \frac{\partial \mathcal{L}}{\partial y_m} = 0 &\longrightarrow \alpha_m = \gamma y_m, \quad m = 1, \dots, M, \\ \frac{\partial \mathcal{L}}{\partial \mathbf{a}_m} = 0 &\longrightarrow y_m = \mathbf{w}^T \phi(\mathcal{X}_m), \quad m = 1, \dots, M. \end{aligned} \quad (8)$$

By eliminating the primal variables $\mathbf{w}$ and y , one obtains $(1/r)\boldsymbol{\alpha}(m) - \sum_{m=1}^M \boldsymbol{\alpha}(m)\phi(\mathcal{X}_m)^T \phi(\mathcal{X}_m) = 0$, for $m = 1, \dots, M$. This is an eigenvalue decomposition that can be present in matrix formulation as

$$\mathcal{H}\boldsymbol{\alpha} = \lambda \boldsymbol{\alpha}, \quad (9)$$

where $\lambda = 1/\gamma$, $\boldsymbol{\alpha} \equiv [\alpha_1, \dots, \alpha_M]^T$, and $\mathcal{H}$ is the centered Kernel Matrix defined entry-wise by

$$\begin{aligned} \mathcal{H}(i, j) &= \langle \phi(\mathcal{X}_i), \phi(\mathcal{X}_j) \rangle = k(\mathcal{X}_i, \mathcal{X}_j), \\ i, j &= 1, \dots, M, \end{aligned} \quad (10)$$

where $\mathcal{X}_i, \mathcal{X}_j \in \{\mathcal{X}_m\}_{m=1}^M$.

The optimal solution to the formulated problem is obtained by selecting the eigenvectors corresponding to the first P largest eigenvalues, where P is a slight abuse of the

notation which, however, simplifies the description. For test sample $\mathcal{X}'$, the first projection becomes

$$\begin{aligned} y' &= \mathbf{w}^T \phi(\mathcal{X}') = \sum_{m=1}^M \boldsymbol{\alpha}(m) \langle \phi(\mathcal{X}_m), \phi(\mathcal{X}') \rangle \\ &= \sum_{m=1}^M \boldsymbol{\alpha}(m) k(\mathcal{X}_m, \mathcal{X}'), \end{aligned} \quad (11)$$

where $\boldsymbol{\alpha} \equiv [\alpha_1, \dots, \alpha_M]^T$ is the first eigenvector of Kernel Matrix (9). For computing the kernel functions in (10) and (11), we present a *tensorial kernel* function in next section, which can directly take tensorial data as parameters other than vectorial ones.

3.2. RKHS Induced by Multilinear Functions. Kernel should be constructed from input space in a way that the high-dimensional feature space implicated by kernel function reflects the underlying structure of data in original input space. Although a number of kernels have been designed for tensorial objects, few approaches exploit the underlying structure of tensorial space. Recently, Signoretto et al. [34] generalized RKHS to adapt to multilinear functions, which allows a reproducing kernel to exploit algebraic geometry of tensorial space. In principle, the idea of Signoretto et al. is to propose a *tensorial kernel* that can directly take tensorial data as parameters other than vectorial ones. After that, the tensorial kernel is plugged into prime-dual framework to learn the structural information embodied in the tensors.

Let RKHS be $(\mathcal{H}, \langle \cdot, \cdot \rangle)$ equipped with some innerproduct.

A bounded (continuous) multilinear function on RKHSs denoted by $\psi : \mathcal{H}_1 \times \mathcal{H}_2 \times \dots \times \mathcal{H}_p \rightarrow \mathbb{R}$ is said to be *Hilbert-Schmidt* if it satisfies some constraints. The ensemble of such well behaved Hilbert-Schmidt functions equipped with the inner product $\langle \psi, \psi' \rangle$ forms a Hilbert Space denoted by HSF, which is a space of multilinear functions corresponding the infinite dimensional tensors. Using boldface ϕ denoting the map from tensor object to multilinear function space, we have $\phi : \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N} \rightarrow \text{HSF}$ and define $k : \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N} \times \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N} \rightarrow \mathbb{R}$ by

$$k(\mathcal{X}_i, \mathcal{X}_j) := \langle \phi(\mathcal{X}_i), \phi(\mathcal{X}_j) \rangle_{\text{HSF}}. \quad (12)$$

According to the theory of [34], for N-order tensors $\mathcal{X}_i, \mathcal{X}_j$, a kernel function, exploiting structural properties possessed by the given tensorial representations, can be stated as N product kernel:

$$k(\mathcal{X}_i, \mathcal{X}_j) = \prod_{n=1}^N k(\mathcal{X}_{i(n)}, \mathcal{X}_{j(n)}), \quad (13)$$

where $k(\mathcal{X}_i, \mathcal{X}_j)$ denotes *tensorial kernel*, $k(\mathcal{X}_{i(n)}, \mathcal{X}_{j(n)})$ is *nth factor kernel* of tensorial kernel, and $\mathcal{X}_{i(n)}$ and $\mathcal{X}_{j(n)}$ is the mode- n matricization of $\mathcal{X}_i$ and $\mathcal{X}_j$, respectively. Equation (13) implies that the similarity measure induced by the kernel function between two tensor objects can be represented as product of factor kernels which measure similarity between mode- n matricization of two tensors.

3.3. Factor Kernel on Grassmann Manifold. The factor kernel represents a similarity measure between two matrices obtained by mode- n matricization of two tensors. In [34], Signoretto et al. adopt Chordal distance as metric to measure such similarity that lead to an ad hoc approach to obtain tensorial kernel. This inconsistency can cause complications and weak guarantees. In our approach, the factor kernel is build from Grassmann kernel by a number of simple operations, resulting in a simpler and better-understood formulation. Note that our factor kernel differs from the result of Signoretto et al.

The fixed dimensional linear subspaces form a non-Euclidean and curved Riemannian manifold known as *Grassmann manifold*, allowing the subspaces to be represented as points on it. In TKPCA, such low dimensional subspaces is used to approximate the mode- n matricization of tensors. The benefits of using subspaces are two-fold: (a) comparing two subspaces is cheaper than comparing two n matricizations of tensor directly when they are very large, for example, too many frames per video, and (b) it is more robust to noise since the subspace can “fill in” the missing pictures.

Given a $I_n \times (I_1 \times \dots \times I_{n-1} \times I_{n+1} \times \dots \times I_N)$ mode- n matricizations $\mathcal{X}_{(n)}$ of rank r , we can represent it as a subspace (and hence as a point on a Grassmann manifold) through any orthogonalisation procedure like SVD. More specifically, let $\mathcal{X}_{(n)} = \mathbf{U}_{\mathcal{X}}^{(n)} \mathbf{D}_{\mathcal{X}}^{(n)} \mathbf{V}_{\mathcal{X}}^{(n)T}$, where the $I_n \times r$ orthonormal matrix $\mathbf{U}_{\mathcal{X}}^{(n)}$ represents an optimised subspace of order r (in the mean square sense) for $\mathcal{X}_{(n)}$ and can be seen as a point on Grassmann manifold $G(r, I_n)$, which is the set of r -dimensional linear subspaces of the $\mathbb{R}^{I_n}$. The Riemannian distance between two subspaces is the length of the shortest geodesic connecting the two points on the Grassmann manifold. Among many different distances, a few of them can be induced to form a positive definite kernel, and the *Projection metric* is the one.

The Projection metric can be understood by associating a point span $\text{span}(\mathbf{U}_{\mathcal{X}}^{(n)}) \in G(r, I_n)$ with its projection matrix $\mathbf{U}_{\mathcal{X}}^{(n)} \mathbf{U}_{\mathcal{X}}^{(n)T}$ by an embedding:

$$\phi : G(r, I_n) \longrightarrow \mathbb{R}^{I_n \times I_n}, \quad \text{span}(\mathbf{U}_{\mathcal{X}}^{(n)}) \longrightarrow \mathbf{U}_{\mathcal{X}}^{(n)} \mathbf{U}_{\mathcal{X}}^{(n)T}. \quad (14)$$

The image $\phi(G)$ is the set of rank r orthogonal projection matrices. This map is in fact an isometric embedding [35], and the projection metric is simply a Euclidean distance in $\mathbb{R}^{I_n \times I_n}$. The corresponding innerproduct of the space is $\text{tr}[(\mathbf{U}_{\mathcal{X}_i}^{(n)} \mathbf{U}_{\mathcal{X}_i}^{(n)T})(\mathbf{U}_{\mathcal{X}_j}^{(n)} \mathbf{U}_{\mathcal{X}_j}^{(n)T})] = \|\mathbf{U}_{\mathcal{X}_i}^{(n)} \mathbf{U}_{\mathcal{X}_j}^{(n)T}\|_F^2$, and therefore, the *projection kernel*

$$k(\mathbf{U}_{\mathcal{X}_i}^{(n)}, \mathbf{U}_{\mathcal{X}_j}^{(n)}) = \|\mathbf{U}_{\mathcal{X}_i}^{(n)T} \mathbf{U}_{\mathcal{X}_j}^{(n)}\|_F^2 \quad (15)$$

is a Grassmann kernel.

Motivated by classic Gaussian kernel $k(x, y) = \exp(-1/2\sigma^2\|x - y\|_2^2)$, we propose a novel factor kernel based on projection kernel (15), and by Theorem 1, it is provable positive definiteness as required by *Mercer's Theorem* [36].

Theorem 1. Let adjustable parameter $\sigma \in \mathbb{R}^+$; the function that exploit metric on Grassmann manifolds

$$k(\mathcal{X}_{i(n)}, \mathcal{X}_{j(n)}) = \exp\left(-\frac{\|\mathbf{U}_{\mathcal{X}_i}^{(n)} \mathbf{U}_{\mathcal{X}_i}^{(n)T} - \mathbf{U}_{\mathcal{X}_j}^{(n)} \mathbf{U}_{\mathcal{X}_j}^{(n)T}\|_F^2}{2\sigma^2}\right) \quad (16)$$

is positive definite kernel function.

Proof. We first verify that the Projection kernel (15) is positive definite kernel function.

The positive definiteness of Projection kernel follows from the properties of the Frobenius norm. For all $\mathbf{U}_{\mathcal{X}_i}^{(n)}, \mathbf{U}_{\mathcal{X}_j}^{(n)} \in \mathcal{G}$, and $c_i, c_j \in \mathbb{R}$, we have

$$\begin{aligned} \sum_{ij} c_i c_j \|\mathbf{U}_{\mathcal{X}_i}^{(n)} \mathbf{U}_{\mathcal{X}_j}^{(n)T}\|_F^2 &= \sum_{ij} c_i c_j \text{tr}(\mathbf{U}_{\mathcal{X}_i}^{(n)} \mathbf{U}_{\mathcal{X}_i}^{(n)T} \mathbf{U}_{\mathcal{X}_j}^{(n)} \mathbf{U}_{\mathcal{X}_j}^{(n)T}) \\ &= \text{tr}\left(\sum_i c_i \mathbf{U}_{\mathcal{X}_i}^{(n)} \mathbf{U}_{\mathcal{X}_i}^{(n)T}\right)^2 \\ &= \left\|\sum_i c_i \mathbf{U}_{\mathcal{X}_i}^{(n)} \mathbf{U}_{\mathcal{X}_i}^{(n)T}\right\|_F^2 \geq 0. \end{aligned} \quad (17)$$

Next, we use Projection kernel as a footstone to build the more complex factor kernel.

The exponential function can be arbitrarily closely approximated by polynomials with positive coefficients and hence is a limit of kernels. Since the positive definiteness property is closed under taking pointwise limits,

$$k(\mathcal{X}_{i(n)}, \mathcal{X}_{j(n)}) = \exp\left(\frac{k(\mathbf{U}_{\mathcal{X}_i}^{(n)}, \mathbf{U}_{\mathcal{X}_j}^{(n)})}{\sigma^2}\right) \quad (18)$$

is a positive definite kernel function for $\sigma \in \mathbb{R}^+$.

Assuming kernel (18) corresponds to a feature map $\phi(\cdot)$, normalising this kernel corresponds to the feature map

$$\mathbf{U}_{\mathcal{X}_i}^{(n)} \longrightarrow \phi(\mathbf{U}_{\mathcal{X}_i}^{(n)}) \longrightarrow \frac{\phi(\mathbf{U}_{\mathcal{X}_i}^{(n)})}{\|\phi(\mathbf{U}_{\mathcal{X}_i}^{(n)})\|}. \quad (19)$$

Hence, we can express the normalised kernel $\hat{k}$ in terms of $k(\mathcal{X}_{i(n)}, \mathcal{X}_{j(n)})$ as follows:

$$\begin{aligned} \hat{k}(\mathbf{U}_{\mathcal{X}_i}^{(n)}, \mathbf{U}_{\mathcal{X}_j}^{(n)}) &= \left\langle \frac{\phi(\mathbf{U}_{\mathcal{X}_i}^{(n)})}{\|\phi(\mathbf{U}_{\mathcal{X}_i}^{(n)})\|}, \frac{\phi(\mathbf{U}_{\mathcal{X}_j}^{(n)})}{\|\phi(\mathbf{U}_{\mathcal{X}_j}^{(n)})\|} \right\rangle \\ &= \frac{k(\mathbf{U}_{\mathcal{X}_i}^{(n)}, \mathbf{U}_{\mathcal{X}_j}^{(n)})}{\sqrt{k(\mathbf{U}_{\mathcal{X}_i}^{(n)}, \mathbf{U}_{\mathcal{X}_i}^{(n)}) k(\mathbf{U}_{\mathcal{X}_j}^{(n)}, \mathbf{U}_{\mathcal{X}_j}^{(n)})}}, \end{aligned} \quad (20)$$

where the $\hat{k}$ is a valid kernel because it was derived from the feature map (19).

Now, we can normalise the kernel (18) to obtain (16):

$$\begin{aligned}
& \frac{\exp\left(\left(k\left(\mathbf{U}_{\mathcal{X}_i}^{(n)}, \mathbf{U}_{\mathcal{X}_j}^{(n)}\right)\right)/\sigma^2\right)}{\sqrt{\exp\left(\left(k\left(\mathbf{U}_{\mathcal{X}_i}^{(n)}, \mathbf{U}_{\mathcal{X}_i}^{(n)}\right)\right)/\sigma^2\right) \exp\left(\left(k\left(\mathbf{U}_{\mathcal{X}_j}^{(n)}, \mathbf{U}_{\mathcal{X}_j}^{(n)}\right)\right)/\sigma^2\right)}} \\
&= \exp\left(\frac{\left\|\mathbf{U}_{\mathcal{X}_i}^{(n)T} \mathbf{U}_{\mathcal{X}_j}^{(n)T}\right\|_F^2}{\sigma^2} - \frac{\left\|\mathbf{U}_{\mathcal{X}_i}^{(n)T} \mathbf{U}_{\mathcal{X}_i}^{(n)}\right\|_F^2}{2\sigma^2} - \frac{\left\|\mathbf{U}_{\mathcal{X}_j}^{(n)T} \mathbf{U}_{\mathcal{X}_j}^{(n)}\right\|_F^2}{2\sigma^2}\right) \\
&= \exp\left(\frac{\text{tr}\left(2\mathbf{U}_{\mathcal{X}_i}^{(n)} \mathbf{U}_{\mathcal{X}_i}^{(n)T} \mathbf{U}_{\mathcal{X}_j}^{(n)} \mathbf{U}_{\mathcal{X}_j}^{(n)T} - \mathbf{U}_{\mathcal{X}_i}^{(n)} \mathbf{U}_{\mathcal{X}_i}^{(n)T} \mathbf{U}_{\mathcal{X}_i}^{(n)} \mathbf{U}_{\mathcal{X}_i}^{(n)T} - \mathbf{U}_{\mathcal{X}_j}^{(n)} \mathbf{U}_{\mathcal{X}_j}^{(n)T} \mathbf{U}_{\mathcal{X}_j}^{(n)} \mathbf{U}_{\mathcal{X}_j}^{(n)T}\right)}{2\sigma^2}\right) \\
&= \exp\left(\frac{\text{tr}\left(\left(\mathbf{U}_{\mathcal{X}_i}^{(n)} \mathbf{U}_{\mathcal{X}_i}^{(n)T}\right)^2 - 2\mathbf{U}_{\mathcal{X}_i}^{(n)} \mathbf{U}_{\mathcal{X}_i}^{(n)T} \mathbf{U}_{\mathcal{X}_j}^{(n)} \mathbf{U}_{\mathcal{X}_j}^{(n)T} + \left(\mathbf{U}_{\mathcal{X}_j}^{(n)} \mathbf{U}_{\mathcal{X}_j}^{(n)T}\right)^2\right)}{2\sigma^2}\right) \\
&= \exp\left(\frac{\text{tr}\left(\mathbf{U}_{\mathcal{X}_i}^{(n)} \mathbf{U}_{\mathcal{X}_i}^{(n)T} - \mathbf{U}_{\mathcal{X}_j}^{(n)} \mathbf{U}_{\mathcal{X}_j}^{(n)T}\right)^2}{2\sigma^2}\right) \\
&= \exp\left(\frac{\text{tr}\left(\left(\mathbf{U}_{\mathcal{X}_i}^{(n)} \mathbf{U}_{\mathcal{X}_i}^{(n)T} - \mathbf{U}_{\mathcal{X}_j}^{(n)} \mathbf{U}_{\mathcal{X}_j}^{(n)T}\right)^T \left(\mathbf{U}_{\mathcal{X}_i}^{(n)} \mathbf{U}_{\mathcal{X}_i}^{(n)T} - \mathbf{U}_{\mathcal{X}_j}^{(n)} \mathbf{U}_{\mathcal{X}_j}^{(n)T}\right)\right)}{2\sigma^2}\right) \\
&= \exp\left(\frac{\left\|\mathbf{U}_{\mathcal{X}_i}^{(n)} \mathbf{U}_{\mathcal{X}_i}^{(n)T} - \mathbf{U}_{\mathcal{X}_j}^{(n)} \mathbf{U}_{\mathcal{X}_j}^{(n)T}\right\|_F^2}{2\sigma^2}\right). \tag{21}
\end{aligned}$$

□

Finally, substituting (16) in (13), the tensorial kernel based on projection kernel can be represented as

$$\begin{aligned}
k(\mathcal{X}_i, \mathcal{X}_j) &= \prod_{n=1}^N k(\mathcal{X}_{i(n)}, \mathcal{X}_{j(n)}) \\
&= \prod_{n=1}^N \exp\left(-\frac{\left\|\mathbf{U}_{\mathcal{X}_i}^{(n)} \mathbf{U}_{\mathcal{X}_i}^{(n)T} - \mathbf{U}_{\mathcal{X}_j}^{(n)} \mathbf{U}_{\mathcal{X}_j}^{(n)T}\right\|_F^2}{2\sigma^2}\right). \tag{22}
\end{aligned}$$

Furthermore, the positive definiteness of (22) comes from the *closure properties of kernels*: a product of valid kernels is still a kernel [25].

We are now ready to summarize the basic steps for performing a TKPCA with the tensorial kernel, given a set of observation $\{\mathcal{X}_m\}_{m=1}^M \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$, i.i.d., according to the generator $p(\mathcal{X})$ and $\mathcal{X}_i, \mathcal{X}_j \in \{\mathcal{X}_m\}_{m=1}^M$. For test sample $\mathcal{X}'$, the pseudo code is summarized in Algorithm 1.

3.4. Properties of TKPCA. Before we proceed to the next section, the following observation is essential.

TKPCA can be seen to arise from a space of multilinear functionals which are, loosely speaking, infinite dimensional tensors. Therefore, TKPCA is equivalent to performing a standard KPCA in the RKHS of multilinear functions except that the parameters of kernel function are in natural tensor

representations. Furthermore, it can be shown that all the properties associated with the KPCA are still valid for the TKPCA. That is, (a) the dominant eigenvector directions optimally retain most of the variance, (b) the MSE (mean square error) in approximating a point in RKHS in terms of the P dominant eigenvectors is minimal, with respect to any other P directions, (c) projections onto the eigenvectors are uncorrelated, and (d) the entropy (under Gaussian assumption) is maximized.

TKPCA Algorithm has the same computational complexity as KPCA provided that the evaluation of kernel function is of complexity $O(1)$. The most time-consuming step is eigendecomposition of $\mathcal{K}$ with complexity $O(M^3)$. When taking into account the evaluation of kernel function, the complexity of computing $\mathbf{U}_{\mathcal{X}_m}^{(n)}$ by any standard SVD algorithm requires $O((I_1 \times \dots \times I_{n-1} \times I_{n+1} \times \dots \times I_N)^3)$ operations. Note that the computational complexity of TKPCA does not grow with the dimensionality of the feature space that we are implicitly working in.

TKPCA compresses each tensor sample of size $\mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ to P scalars, where P denotes the first P largest eigenvalues, while $\mathbb{R}^{P_1 \times P_2 \times \dots \times P_N}$ are needed to represent a tensor object in the tensorial PCA solutions. Thus, the compression ratio (CR) is a major advantage that TKPCA enjoys over other solutions, such as MPCA [19], 2DPCA [3], and GPCA [1].

There are several reasons behind the motivation of deriving the factor kernel with Gaussian kernel. Firstly, the Gaussian kernel is the most widely used nonlinear kernel and has been extensively studied in neighbouring fields. Secondly, there is no theoretical method for determining a kernel function. In the absence of expert knowledge, the Gaussian kernel makes a good default nonlinear kernel. Thirdly, the isotropic property of Gaussian kernel endows final tensorial kernel with intrinsic rotation-invariant, which is a desire feature that can be used in applications such as face recognition and action recognition.

Just like choosing kernel, there is no prior knowledge for setting the parameter σ which controls the flexibility of the kernel. Generally speaking, small values of σ allow classifiers to fit any labels, hence risking overfitting. In such cases, the Kernel Matrix becomes close to the identity matrix. On the other hand, large values of σ gradually reduce the kernel to a constant function, making it impossible to learn any nontrivial classifier. The feature space has infinite-dimension for every value of σ , but for large values, the weight decays very fast on the higher-order features. In other words, although the rank of the kernel matrix will be full, for all practical purposes, the data lie in a low-dimensional subspace of the feature space. For a full coverage of choosing kernel and parameters, please refer [37–39].

The PCA and Linear Discriminant Analysis (LDA) are two of the most commonly used subspace techniques. PCA produces an expressive subspace for object representation, while LDA produces a discriminating subspace. For the purpose of classification, LDA is generally believed to be superior to PCA when enough training samples per class are available [40]. However, when the number of available


```

(1) Tensorial Kernel Function  $k(\mathcal{X}_i, \mathcal{X}_j)$ :
(2) for  $n = 1$  to  $N$  do
(3)   Compute orthonormal bases  $\mathbf{U}_{\mathcal{X}_i}^{(n)} \mathbf{U}_{\mathcal{X}_j}^{(n)}$  from SVD:  $\mathcal{X}_{i(n)} = \mathbf{U}_{\mathcal{X}_i}^{(n)} \mathbf{D}_{\mathcal{X}_i}^{(n)} \mathbf{V}_{\mathcal{X}_i}^{(n)T}$ ,
       $\mathcal{X}_{j(n)} = \mathbf{U}_{\mathcal{X}_j}^{(n)} \mathbf{D}_{\mathcal{X}_j}^{(n)} \mathbf{V}_{\mathcal{X}_j}^{(n)T}$ .
(4) Compute  $k(\mathcal{X}_i, \mathcal{X}_j)$  by (22).
(5) TKPCA:
(6) Step 1. (Initialization):
(7) Center the input samples by subtracting empirical mean:  $(1/M) \sum_{m=1}^M \mathcal{X}_m$ .
(8) Step 2. (Training):
(9) for  $i = 1$  to  $M$  do
(10)   for  $j = 1$  to  $M$  do
(11)     if  $i \neq j$  then
(12)       Compute the Kernel Matrix (10):  $\mathcal{K}(i, j) = k(\mathcal{X}_i, \mathcal{X}_j)$ .
(13) Compute the  $P$  dominant eigenvectors  $\alpha_p$  by eign-decomposition  $\mathcal{K}$  (9).
(14) Step 3. (Projection):
(15) For test sample  $\mathcal{X}'$ , compute the  $P$  projections onto each one of the dominant eigenvectors (11),
       $y_p \equiv \langle \mathbf{w}, \phi(\mathcal{X}') \rangle = \sum_{m=1}^M \alpha_p(m) k(\mathcal{X}_m, \mathcal{X}'), p = 1, \dots, P$ .

```

ALGORITHM 1: TKPCA: Kernel Principal Component Analysis of tensor objects.

training samples per class is small, the situation considered in this paper, experimental analysis indicates that PCA outperforms LDA [41, 42].

3.5. TKPCA-Based Tensor Object Recognition. After a projection by TKPCA, a new *feature vector* is obtained for each tensor object. The classification tasks on tensor objects are reduced to classification tasks in vector spaces. More precisely, for any query tensor object $\mathcal{X}$, the projection $\mathbf{y} \in \mathbb{R}^P$ on P most dominant eigenvectors is obtained. Similarly, the gallery set containing data samples with labels is also represented by vectors. Then, any classification methods can be employed to label query.

However, as we have seen above, TKPCA maximizes not only the within-class variation but also the between-class variation. This is due to the fact that the TKPCA works as an unsupervised technique without considering the class label. To overcome this limitation, a feature selection strategy is proposed to select eigenvectors for a more discriminative subspace. The strategy works according to the criterion that is based on the maximization of the following ratio [43]:

$$J = \frac{\sum_{c=1}^C N_c (\bar{\mathbf{y}}_c - \bar{\mathbf{y}})^2}{\sum_{m=1}^M (\mathbf{y}_m - \bar{\mathbf{y}}_{c_m})^2}, \quad (23)$$

where C is the number of classes, M is the number of samples in the gallery set, N_c is the number of samples for class c , and c_m is the class label for the m th gallery sample $\mathcal{X}_m$. $\mathbf{y}$ is the feature vector of $\mathcal{X}_m$ in the projected nonlinear subspace. The mean feature vector $\bar{\mathbf{y}} = (1/M) \sum_m \mathbf{y}_m$, and class mean feature vector $\bar{\mathbf{y}}_c = (1/N_c) \sum_{m, c_m=c} \mathbf{y}_m$. For the eigenvector selection, only the first Q most discriminative components of $\mathbf{y}_q$ are kept for classification, with Q determined empirically or cross-validated.

Upon the extraction of the proper set of features, a classifier such as Nearest Neighbor Classifier, Bayesian Classifier,

Neural Network, and Support Vector Machine can be applied to recognize the objects. Here, we use a Nearest Neighbor Classifier for classification. The distance between two arbitrary feature vectors is defined by $d(\mathcal{X}_i, \mathcal{X}_j) = \sum_{p=1}^P \|\mathbf{y}_i^p - \mathbf{y}_j^p\|$, where the norm denotes the Euclidean distance between the two feature vectors. Such a simple classifier is selected to study the performance mainly contributed by the TKPCA-based feature extraction algorithm although better classifiers can be investigated.

4. Experiments

This section illustrates the efficacy of TKPCA in tensor object recognition, by applying it to the emerging application of Action Recognition [44] and comparing its performance against state-of-the-art algorithm. We also assess the noise robustness of the proposed approach and investigate sensitivity against occlusion and misalignment.

The action recognition is the process of labeling videos containing human motion with action labels. We will test our method on two action datasets: the KTH human motion dataset [45] and the Ballet dataset [46]. The proposed TKPCA-Based Tensor Object Recognition in Section 3.4 treats each action video as a 3rd-order tensor sample with the spatial row space, column space, and the time space accounting for the 3 modes. The whole dataset will be a 4th-order tensor, with the addition of the sample space.

4.1. KTH Dataset. The KTH human motion data set [45] contains six types of human actions walking, jogging, running, boxing, hand waving, and hand clapping performed several times by 25 subjects in four scenarios: outdoors, outdoors with scale variation, outdoors with different clothes, and indoors. See Figure 4 for sample frames. We first run an automatic pre processing step to track and stabilise the video sequences so that all of the figures appear in the center of the

TABLE 3: Recognition accuracy (in %) for the KTH action recognition dataset.

	STW	BoW-MKL	TCCA	DCCA	TKPCA
Walking	88	100	88	82	97
Jogging	90	80	75	53	98
Running	99	68	77	88	97
Boxing	98	97	92	98	100
Handclapping	100	99	100	86	96
Handwaving	97	99	88	93	100
Total	95	90	87	83	98

FIGURE 4: The KTH action dataset.

FIGURE 5: Visualization of samples of KTH action recognition dataset in the first two dimensional subspace: (a) represents TKPCA using a tensor kernel; (b) represents KPCA using a Gaussian kernel performed on the vectorization of tensors. Observe that the features obtained by TKPCA are more discriminative than KPCA features.

FIGURE 6: Confusion matrix (in %) for the TKPCA (a) and MPCA (b) methods on the KTH action recognition dataset using LOO protocol.

FIGURE 7: Some example frames from an instructional ballet DVD.

LR hand opening	81.50	2.50	2.50	0.00	3.00	5.00	0.00	5.50
RL hand opening	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00
Standing hand opening	0.00	0.00	87.88	4.55	0.00	6.06	0.00	1.52
Leg swinging	3.57	0.00	0.00	85.71	3.57	3.57	0.00	3.57
Jumping	4.55	4.55	9.09	13.14	50.54	0.00	13.60	4.55
Turing	0.00	0.00	5.82	0.00	0.00	94.18	0.00	0.00
Hopping	0.00	2.33	17.23	2.33	2.33	0.00	71.14	4.65
Standing still	6.00	0.00	27.50	11.00	0.00	10.00	0.00	45.50
	LR hand opening	RL hand opening	Standing hand opening	Leg swinging	Jumping	Turing	Hopping	Standing still

FIGURE 8: Confusion matrix (in %) for the TKPCA method on the Ballet dataset.

TABLE 4: Recognition accuracy (in %) along its standard deviation for the Ballet dataset.

Method	Recognition accuracy
KPCA	85.08 ± 1.8
TPM	78.05 ± 2.9
TCCA	77.63 ± 2.9

field of view. All videos were resized to $32 \times 32 \times 32$. In order to have a standard length of 32 frames per video, the middle 32 frames were used.

As this paper focuses on kernelizing Principal Component Analysis for tensor objects, the convolutional Kernel Principal Component Analysis using a Gaussian kernel performed on vectorization of tensor $\text{vec}(\mathcal{X})$ should be compared with our new approach TKPCA first. However, for each action video, the dimensionality of vectorized tensor is up to 32768 (32^3) which prevents KPCA from computing CPs efficiently. Thus, a reduced-order model [47] is adopted in KPCA. The kernel parameter σ is optimized by methods of cross-validation for both TKPCA and KPCA. The test samples are projected onto the feature subspace to obtain the discriminative features as shown in Figure 5. Observe that TKPCA outperforms KPCA with respect to the discriminative ability, and six classes are well separated even in two dimensional space.

In order to test the TKPCA's ability to capture nonlinear structure of input data, we conduct another experiment to compare TKPCA with the multilinear PCA (MPCA) [19]. Figure 6 illustrates the confusion matrices of TKPCA and MPCA. The confusion matrix is a specific table layout to see if the recognizer is confusing two classes in which rows correspond to the ground truth, and columns correspond to the classification results. It can be seen that TKPCA

FIGURE 9: Noise resilience analysis: (a) noisy (additive Gaussian) query and clean training set; (b) query and training are both noisy (additive Gaussian); (c) noisy (sparse spikes) query and clean training set; (d) query and training are both noisy (sparse spikes).

achieves average accuracy of 98%, while MPCA achieves 84%, and the confusion of TKPCA only appears among boxing, hand clapping, and hand waving, which is consistent with our intuition that these actions are easily confused. The superiority of TKPCA over MPCA indicates that nonlinear structures of video volumes captured by tensorial kernel learning significantly improve the discriminative performance.

Next, the proposed TKPCA algorithms are compared against the state-of-the-art action recognition algorithms. We compared TKPCA against spatial-temporal words (STW) [48] and bag of words model in conjunction with multiple kernel learning (BoW-MKL) [49, 50]. In STW, a video sequence is represented by a set of spatial-temporal words, extracted from space-time interest points. The algorithm

then utilises latent topic models such as the probabilistic latent semantic analysis [51] to learn the probability distributions of the spatial-temporal words. BoW-MKL exploits global spatial-temporal distribution of interest points by extracting holistic features from clouds of interest points accumulated over multiple temporal scales. Then, extracted features are fused using MKL. We also compared TKPCA against Tensor Canonical Correlation Analysis (TCCA) [11] and Discriminative Canonical Correlation Analysis (DCCA) [11]. TCCA is an extension of canonical correlation analysis (a principled tool to inspect linear relations between two sets of vectors) to tensor spaces and measures video-to-video tensors in a way similar to our method. DCCA implements a linear discriminant function that maximizes the canonical

FIGURE 10: Sensitivity analysis to occlusion and misalignment: (a) occlusion; (b) misalignment.

correlations of within-class sets and minimizes the canonical correlations of between-class sets. To facilitate comparison with prior work, we followed the leave-one-out (LOO) cross validation protocol used in STW [48] and TCCA [11].

Looking at the results in Table 3, the first thing to note is that no algorithm is universally the best. In terms of top classification rates, STW, BoW-MKL, and our method are, respectively, best for two of the six actions. However, when our method is better, it is typically by a larger amount, and this is reflected in the higher overall average classification rate of 98% versus 95% for STW and 90% for BoW-MKL.

4.2. Ballet Dataset. The Ballet dataset contains 44 real video sequences of 8 actions collected from an instructional ballet DVD. The dataset consists of 8 complex motion patterns performed by three subjects. The actions include “left-to-right hand opening”, “right-to-left hand opening”, “standing hand opening”, “leg swinging”, “jumping”, “turning”, “hopping”, and “standing still”. Figure 7 shows samples. This dataset has a uniform background and fair illumination and therefore minimises the effect of variations in illumination and background. Yet, at the same time, it is very challenging due to the significant within-class variations in terms of speed, spatial and temporal scale, clothing, and movement. Available samples of each action were randomly split into training and testing sets (the number of actions in both training and testing sets were fairly even). The process of random splitting was repeated ten times, and the average classification accuracy was recorded. All video sequences were uniformly resized to $50 \times 50 \times 50$. In order to have a standard length of 50 frames per video sequences, the middle 50 frames were used.

For the sake of comparison between tensor based methods, TKPCA algorithm is contrasted with the Tensor as a point on a Product Manifold (TPM) [12] and Tensor

Canonical Correlation Analysis (TCCA) [11]. TPM maps a video tensor to a point on a product manifold and the geodesic distance on a product manifold, is computed for tensor classification. Table 4 shows that the TKPCA algorithm obtains the highest accuracy and outperforms state-of-the-art tensor based methods of TPM and TCCA significantly. The confusion matrix of the proposed TKPCA method is shown in Figure 8. Our performance on this dataset is not as good as the previous ones, which might be because of the complexity of actions in this dataset and significant within-class variations.

4.3. Sensitivity Analysis

4.3.1. Sensitivity to Noise. In addition to the above, to assert robustness to noise we add two types of noise, to the clean Ballet dataset. We compare TKPCA, TPM and TCCA in case of additive Gaussian noise and sparse noise spikes. The noise process in the case of additive Gaussian noise is $N(0, p\sigma_i)$, added to i th frame with σ_i , the standard deviation of the frames and $p \in [0, 1]$. In case of spike noise, we randomly add values drawn from the normal process, $N(0, p\sigma_i)$, to randomly choose time points of each video sequences. The number of noisy time points is no more than 5% of the length of the time-series, spread uniformly over the full time-span. Figure 9 depicts TKPCA, TPM, and TCCA recognition accuracies in presence of different levels of noise. Figure 9 indicates that TKPCA outperforms others, the advantage of which is becoming obvious as the noise level grows. This could be due to our underlying kernel function built on the Grassmann kernel which is more robust to noise since the subspace points on Grassmann Manifold can fill in the missing pictures.

4.3.2. Sensitivity to Occlusion. An important aspect of the proposed approach relates to the sensitivity against occlusion.

We assess the performance at various levels of occlusion in Ballet dataset, from 1.56% up to 45%, by replacing a set of randomly located square blocks of size 4×4 in the query frames with a blank block. The location of occlusion is randomly chosen for each query frame and is unknown to the system. The training frames do not contain occlusions. Figure 10(a) shows the recognition rates of TKPCA, TPM, and TCCA. The proposed TKPCA method significantly outperforms the other two methods in almost all levels of occlusion. Up to 40 percent occlusion, the performance of TKPCA has dropped roughly by 20 percentage points. The proposed TKPCA method has better captured the nonlinear intrinsic geometry and is hence more robust to the missing parts.

4.3.3. Sensitivity to Misalignment. The temporal and spatial misalignment could deteriorate the performance of an action recogniser drastically. In this part, we only consider spatial misalignment and assess and contrast the sensitivity of TKPCA algorithm as compared to TPM and TCCA on Ballet dataset. To this end, we have introduced random displacements to the frames of query videos and measured the accuracy for various amounts of displacements. Figure 10(a) shows the result. The horizontal axis here demonstrates the degree of misalignment. Figure 10(a) reveals that all studied algorithms are sensitive to misalignment. The larger the displacement, the lower would be the recognition accuracy. This is mainly due to the tensorial representation of video highly depending on the relationships that are fragile to misalignment.

5. Conclusion

In this paper, we present a new TKPCA algorithm for dimensionality reduction and feature extraction from tensor objects, such as 2D/3D images and video sequences. TKPCA determines a subspace of lower dimensionality that captures most of the nonlinear variation present in the original tensorial representation. A novel tensorial kernel, which can directly measure the similarity between tensorial inputs, is also proposed based on Grassmann kernel to capture the topological structure underlying tensor dataset. Furthermore, the strict positive definiteness proof of proposed kernel function is given. Experimental results show that the TKPCA remedies the shortcoming of tensorial PCA in modelling the nonlinear manifold of tensor objects and reduces the SSS and Curse of Dimensionality problem. Furthermore, it achieves more compression rate and is robust to both noise and occlusion. To the best of our knowledge, the problem of TKPCA has not been considered in the existing literatures.

Finally, there are still some aspects of TKPCA that deserve further study. For example, TKPCA is essentially *batch* optimization problem, with all training tensor data being available in advance. Such assumption is unsuitable for large-scale data sets and thus unadapted for real-time applications [47]. Therefore, for applications like video surveillance [52] or social networks analysis [53], an online scheme of TKPCA is expected.

Conflict of Interests

The authors declare no conflict of interests.

References

- [1] J. Ye, R. Janardan, and Q. Li, "Gpca: an efficient dimension reduction scheme for image compression and retrieval," in *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 354–363, August 2004.
- [2] D. Zhang and Z.-H. Zhou, "(2D)2 PCA: Two-directional two-dimensional PCA for efficient face representation and recognition," *Neurocomputing*, vol. 69, no. 1–3, pp. 224–231, 2005.
- [3] J. Yang, D. Zhang, A. F. Frangi, and J.-Y. Yang, "Two-dimensional PCA: a new approach to appearance-based face representation and recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, pp. 131–137, 2004.
- [4] C. Ding and J. Ye, "2-dimensional singular value decomposition for 2d maps and images," in *Proceedings of the SIAM International Conference on Data Mining*, pp. 32–43, 2005.
- [5] F. Cong, I. Kalyakin, A.-H. Phan et al., "Benefits of multi-domain feature of mismatch negativity extracted by non-negative tensor factorization from eeg collected by low-density array," *International Journal of Neural Systems*, vol. 22, no. 6, 2012.
- [6] M. Kwon, J. S. Kang, and M. Lee, "Emotion classification in movie clips based on 3d fuzzy gist and eeg signal analysis," in *Proceedings of the International Winter Workshop on Brain-Computer Interface (BCI '13)*, pp. 67–68, 2013.
- [7] J. Ye, T. Li, T. Xiong, and R. Janardan, "Using uncorrelated discriminant analysis for tissue classification with gene expression data," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 1, no. 4, pp. 181–190, 2004.
- [8] B. Bosch, E. M. Arenaza-Urquijo, L. Rami et al., "Multiple DTI index analysis in normal aging, amnesic MCI and AD. Relationship with neuropsychological performance," *Neurobiology of Aging*, vol. 33, no. 1, pp. 61–74, 2012.
- [9] N. Renard and S. Bourennane, "Dimensionality reduction based on tensor modeling for classification methods," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 4, pp. 1123–1131, 2009.
- [10] M. W. Whittle, *Gait Analysis: An Introduction*, Butterworth-Heinemann, 2006.
- [11] T.-K. Kim and R. Cipolla, "Canonical correlation analysis of video volume tensors for action categorization and detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 8, pp. 1415–1428, 2009.
- [12] Y. M. Lui, J. R. Beveridge, and M. Kirby, "Action classification on product manifolds," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 833–839, June 2010.
- [13] J. Sun, D. Tao, S. Papadimitriou, P. S. Yu, and C. Faloutsos, "Incremental tensor analysis: theory and applications," *ACM Transactions on Knowledge Discovery from Data*, vol. 2, no. 3, article 11, 2008.
- [14] I. Jolliffe, *Principal Component Analysis*, Wiley Online Library, 2005.
- [15] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, 1998.

- [16] C. M. Bishop and N. M. Nasrabadi, *Pattern Recognition and Machine Learning*, vol. 1, Springer, New York, NY, USA, 2006.
- [17] X. He, D. Cai, and P. Niyogi, "Tensor subspace analysis," in *Advances in Neural Information Processing Systems*, pp. 499–506, 2005.
- [18] D. Tao, X. Li, W. Hu, S. Maybank, and X. Wu, "Supervised tensor learning," in *Proceedings of the 5th IEEE International Conference on Data Mining (ICDM '05)*, pp. 450–457, November 2005.
- [19] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "MPCA: Multilinear principal component analysis of tensor objects," *IEEE Transactions on Neural Networks*, vol. 19, no. 1, pp. 18–39, 2008.
- [20] K. Inoue, K. Hara, and K. Urahama, "Robust multilinear principal component analysis," in *Proceedings of the 12th International Conference on Computer Vision (ICCV '09)*, pp. 591–597, October 2009.
- [21] Y. Panagakis, C. Kotropoulos, and G. R. Arce, "Non-negative multilinear principal component analysis of auditory temporal modulations for music genre classification," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 18, no. 3, pp. 576–588, 2010.
- [22] A. Shashua and A. Levin, "Linear image coding for regression and classification using the tensor-rank principle," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '01)*, vol. 1, pp. 142–149, December 2001.
- [23] D. Li, K. Huang, H. Zhang, and L. Zhang, "Umpca based feature extraction for ecg," in *Advances in Neural Networks*, pp. 383–390, Springer, 2013.
- [24] H. Lu, K. N. K. Plataniotis, and A. N. Venetsanopoulos, "Uncorrelated multilinear principal component analysis for unsupervised multilinear subspace learning," *IEEE transactions on Neural Networks*, vol. 20, no. 11, pp. 1820–1836, 2009.
- [25] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*, Cambridge university press, 2004.
- [26] E. Kofidis and P. A. Regalia, "On the best rank-1 approximation of higher-order supersymmetric tensors," *SIAM Journal on Matrix Analysis and Applications*, vol. 23, no. 3, pp. 863–884, 2002.
- [27] L. de Lathauwer, B. de Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, 2000.
- [28] B. W. Bader and T. G. Kolda, "Algorithm 862: Matlab tensor classes for fast algorithm prototyping," *ACM Transactions on Mathematical Software*, vol. 32, no. 4, pp. 635–653, 2006.
- [29] M. Ding, Z. Tian, and H. Xu, "Adaptive kernel principal component analysis," *Signal Processing*, vol. 90, no. 5, pp. 1542–1553, 2010.
- [30] Q. Zhao, C. F. Caiafa, D. P. Mandic et al., "Multilinear subspace regression: an orthogonal tensor decomposition approach," in *Advances in Neural Information Processing Systems*, pp. 1269–1277, 2011.
- [31] S. Yan, D. Xu, Q. Yang, L. Zhang, X. Tang, and H.-J. Zhang, "Multilinear discriminant analysis for face recognition," *IEEE Transactions on Image Processing*, vol. 16, no. 1, pp. 212–220, 2007.
- [32] S. Shan, B. Cao, Y. Su, L. Qing, X. Chen, and W. Gao, "Unified principal component analysis with generalized covariance matrix for face recognition," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, pp. 1–7, June 2008.
- [33] J. A. K. Suykens, T. van Gestel, J. Vandewalle, and B. de Moor, "A support vector machine formulation to PCA analysis and its kernel version," *IEEE Transactions on Neural Networks*, vol. 14, no. 2, pp. 447–450, 2003.
- [34] M. Signoretto, L. de Lathauwer, and J. A. K. Suykens, "A kernel-based framework to tensorial data analysis," *Neural Networks*, vol. 24, no. 8, pp. 861–874, 2011.
- [35] Y. Chikuse, *Statistics on Special Manifolds*, vol. 174, Springer, 2003.
- [36] S. Theodoridis, K. Nos, and K. M. Bas, *Pattern Recognition*, Elsevier, 4th edition, 2009.
- [37] G. C. Cawley and N. L. C. Talbot, "Preventing over-fitting during model selection via bayesian regularisation of the hyperparameters," *The Journal of Machine Learning Research*, vol. 8, pp. 841–861, 2007.
- [38] G. C. Cawley and N. L. C. Talbot, "On over-fitting in model selection and subsequent selection bias in performance evaluation," *The Journal of Machine Learning Research*, vol. 99, pp. 2079–2107, 2010.
- [39] C. Lampert, *Kernel Methods in Computer Vision*, vol. 4, Now Publishers, 2009.
- [40] J. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "Face recognition using LDA-based algorithms," *IEEE Transactions on Neural Networks*, vol. 14, no. 1, pp. 195–200, 2003.
- [41] A. M. Martinez and A. C. Kak, "PCA versus LDA," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, pp. 228–233, 2001.
- [42] G. Shakhnarovich and B. Moghaddam, "Face recognition in subspaces," in *Handbook of Face Recognition*, pp. 19–49, Springer, 2011.
- [43] J. Wang, K. N. Plataniotis, and A. N. Venetsanopoulos, "Selecting discriminant eigenfaces for face recognition," *Pattern Recognition Letters*, vol. 26, no. 10, pp. 1470–1482, 2005.
- [44] R. Poppe, "A survey on vision-based human action recognition," *Image and Vision Computing*, vol. 28, no. 6, pp. 976–990, 2010.
- [45] C. Schödl, I. Laptev, and B. Caputo, "Recognizing human actions: a local SVM approach," in *Proceedings of the 17th International Conference on Pattern Recognition (ICPR '04)*, pp. 32–36, August 2004.
- [46] Y. Wang and G. Mori, "Human action recognition by semilattent topic models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 10, pp. 1762–1774, 2009.
- [47] P. Honeine, "Online kernel principal component analysis: a reduced-order model," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, pp. 1814–1826, 2012.
- [48] J. C. Niebles, H. Wang, and L. Fei-Fei, "Unsupervised learning of human action categories using spatial-temporal words," *International Journal of Computer Vision*, vol. 79, no. 3, pp. 299–318, 2008.
- [49] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan, "Multiple kernel learning, conic duality, and the SMO algorithm," in *Proceedings of the 21st International Conference on Machine Learning*, p. 6, July 2004.
- [50] M. Bregonzio, T. Xiang, and S. Gong, "Fusing appearance and distribution information of interest points for action recognition," *Pattern Recognition*, vol. 45, no. 3, pp. 1220–1234, 2012.
- [51] T. Hofmann, "Probabilistic latent semantic indexing," in *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 50–57, 1999.

- [52] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *International Journal of Computer Vision*, vol. 77, no. 1–3, pp. 125–141, 2008.
- [53] J. Wen, X. Gao, Y. Yuan, D. Tao, and J. Li, "Incremental tensor biased discriminant analysis: a new color-based visual tracking method," *Neurocomputing*, vol. 73, no. 4–6, pp. 827–839, 2010.