MPCA: Multilinear Principal Component Analysis of Tensor Objects

Haiping Lu, *Student Member, IEEE*, Konstantinos N. (Kostas) Plataniotis, *Senior Member, IEEE*, and Anastasios N. Venetsanopoulos, *Fellow, IEEE*

Abstract—This paper introduces a multilinear principal component analysis (MPCA) framework for tensor object feature extraction. Objects of interest in many computer vision and pattern recognition applications, such as 2-D/3-D images and video sequences are naturally described as tensors or multilinear arrays. The proposed framework performs feature extraction by determining a multilinear projection that captures most of the original tensorial input variation. The solution is iterative in nature and it proceeds by decomposing the original problem to a series of multiple projection subproblems. As part of this work, methods for subspace dimensionality determination are proposed and analyzed. It is shown that the MPCA framework discussed in this work supplants existing heterogeneous solutions such as the classical principal component analysis (PCA) and its 2-D variant (2-D PCA). Finally, a tensor object recognition system is proposed with the introduction of a discriminative tensor feature selection mechanism and a novel classification strategy, and applied to the problem of gait recognition. Results presented here indicate MPCA's utility as a feature extraction tool. It is shown that even without a fully optimized design, an MPCA-based gait recognition module achieves highly competitive performance and compares favorably to the state-of-the-art gait recognizers.*

Index Terms—Dimensionality reduction, feature extraction, gait recognition, multilinear principal component analysis (MPCA), tensor objects.

I. INTRODUCTION

THE term tensor object is used here to denote a multidimensional object, the elements of which are to be addressed by more than two indices [1]. The number of indices used in the description defines the order of the tensor object and each index defines one of the so-called "modes." Many image and video data are naturally tensor objects. For example, color images are 3-D (third-order tensor) objects with column, row, and color modes [2]. Gait silhouette sequences, the input to most if not all gait recognition algorithms [3]–[7], as well as other grayscale video sequences can be viewed as third-order tensors

with column, row, and time modes. Naturally, color video sequences are fourth-order tensors with the addition of a color mode. In the most active area of biometrics research, namely, that of face recognition, 3-D face detection and recognition using 3-D information with column, row, and depth modes, in other words, a third-order tensor, has emerged as an important research direction [8]–[10]. Moreover, the research problem of matching still probe images to surveillance video sequences can be viewed as a pattern recognition problem in a third-order tensorial setting [11]. Beyond biometrics signal analysis, many other computer vision and pattern recognition tasks can be also viewed as problems in a multilinear domain. Such tasks include 3-D object recognition tasks [12] in machine vision, medical image analysis, and content-based retrieval, space-time analysis of video sequences for gesture recognition [13] and activity recognition [14] in human-computer interaction (HCI), and space-time super resolution [15] for digital cameras with limited spatial and temporal resolution. The wide range of applications explains the authors' belief that a comprehensive study of a specialized feature extraction problem, such as multilinear feature extraction, is worthwhile.

A typical tensor object in pattern recognition or machine vision applications is commonly specified in a high-dimensional tensor space. Recognition methods operating directly on this space suffer from the so-called curse of dimensionality [16]: Handling high-dimensional samples is computationally expensive and many classifiers perform poorly in high-dimensional spaces given a small number of training samples. However, since the entries of a tensor object are often highly correlated with surrounding entries, it is reasonable to assume that the tensor objects encountered in most applications of interest are highly constrained and thus the tensors are confined to a subspace, a manifold of intrinsic low dimension [16], [17]. Feature extraction or dimensionality reduction is thus an attempt to transform a high-dimensional data set into a low-dimensional equivalent representation while retaining most of the information regarding the underlying structure or the actual physical phenomenon [18].

Principal component analysis (PCA) is a well-known unsupervised linear technique for dimensionality reduction. The central idea behind PCA is to reduce the dimensionality of a data set consisting of a larger number of interrelated variables, while retaining as much as possible the variation present in the original data set [19]. This is achieved by transforming to a new set of variables, the so-called principal components (PCs), which are uncorrelated, and ordered so that the first few retain most of the original data variation. Naive application of PCA to tensor objects requires their reshaping into vectors with

high dimensionality (vectorization), which obviously results in high processing cost in terms of increased computational and memory demands. For example, vectorizing a typical gait silhouette sequence of size $(120 \times 80 \times 20)$ results in a vector with dimensionality $(192\,000 \times 1)$, the singular value decomposition (SVD) or eigendecomposition processing of which may be beyond the computing processing capabilities of many computing devices. Beyond implementation issues, it is well understood that reshaping breaks the natural structure and correlation in the original data, removing redundancies and/or higher order dependencies present in the original data set and losing potentially more compact or useful representations that can be obtained in the original form [20]. Vectorization as PCA preprocessing ignores the fact that tensor objects are naturally multidimensional objects, e.g., gait sequences are 3-D objects, instead of 1-D objects. Therefore, a dimensionality reduction algorithm operating directly on a tensor object rather than its vectorized version is desirable.

Recently, dimensionality reduction solutions representing images as matrices (second-order tensors) rather than vectors (first-order tensors) have been introduced. A 2-D PCA algorithm is proposed in [21], where the image covariance matrix is constructed using image matrices as inputs. However, a linear transformation is applied only to the right-hand side of the input image matrices. As a result, image data is projected in one mode only, resulting in poor dimensionality reduction. The less restrictive 2-D PCA algorithm introduced in [20] takes into account the spatial correlation of the image pixels within a localized neighborhood. Two linear transforms are applied to both the left- and the right-hand sides of the input image matrices. Thus, projections in both modes are calculated and better dimensionality reduction results are obtained according to [22]. Similarly to the solutions introduced in [21] and [22], the so-called tensor subspace analysis algorithm of [23] represents the input image as a matrix residing in a tensor space and attempts to detect local geometrical structure in that tensor space by learning a lower dimensional tensor subspace.

For the theoretically inclined reader, it should be noted that there are some recent developments in the analysis of higher order tensors. The higher order singular value decomposition (HOSVD) solution, which extends SVD to higher order tensors, was formulated in [24] and its computation leads to the calculation of N (the order) different matrix SVDs of unfolded matrices. An alternating least square (ALS) algorithm for the best rank-$(R_1, R_2, \ldots, R_N)$ approximation of higher order tensors was studied in [1], where tensor data was projected into a lower dimensional tensor space iteratively. The application of HOSVD truncation and the best rank-$(R_1, R_2, \ldots, R_N)$ approximation to dimensionality reduction in independent component analysis (ICA) was discussed in [25]. These multilinear algorithms have been used routinely for multiple factor analysis [26], [27], where input data such as images are still represented as vectors but with these vectors arranged into a tensor for the subsequent analysis of the multiple factors involved in image/video formation. It should be added that in [25]–[27], the tensor data under consideration is projected in the original coordinate without data centering. However, for classification/recognition applications where eigenproblem solutions are attempted,

the eigendecomposition in each mode can be influenced by the mean (average) of the data set.

Recently, there have been several attempts to develop multilinear subspace algorithms for tensor object feature extraction and classification. In [28], a heuristic MPCA approach based on HOSVD was proposed. The MPCA formulation in [29] targets optimal reconstruction applications (where data is not centered) with a solution built in a manner similar to that of [1]. It should be noted that the solution in [29] was focused on reconstruction not recognition and that it did not cover a number of important algorithmic issues, namely, initialization, termination, convergence, and subspace dimensionality determination. When applied to the problem of tensor object recognition, the methodology described in [29] uses all the entries in the projected tensor for recognition although the discrimination power of these entries varies considerably. There is also a recent work on multilinear discriminant analysis (MLDA) [30], [31], named discriminant analysis with tensor representation (DATER), where an iterative algorithm similar to ALS of [1] is utilized in order to maximize a tensor-based discriminant criterion. Unfortunately, this MLDA variant does not converge and it appears to be extremely sensitive to parameter settings [32]. As the number of possible subspace dimensions for tensor objects is extremely high (e.g., there are $225\,280$ possible subspace dimensions for the gait recognition problem considered in this work), exhaustive testing for determination of parameters is not feasible. Consequently, the algorithmic solution of [30] and [31] cannot be used to effectively determine subspace dimensionality in a comprehensive and systematic manner.

Motivated by the works briefly reviewed here, this paper introduces a new MPCA formulation for tensor object dimensionality reduction and feature extraction. The proposed solution follows the classical PCA paradigm. Operating directly on the original tensorial data, the proposed MPCA is a multilinear algorithm performing dimensionality reduction in all tensor modes seeking those bases in each mode that allow projected tensors to capture most of the variation present in the original tensors.

The main contributions of this paper include the following.
1) The introduction of a new MPCA framework for tensor object dimensionality reduction and feature extraction using tensor representation. The framework is introduced from the perspective of capturing the original tensors' variation. It provides a systematic procedure to determine effective representations of tensor objects. This contrasts to previous work such as those reported in [16], [26], and [27], where vector, not tensor, representation was used, and the works reported in [20], [21], and [23], where matrix representation was utilized. It also differs from the works reported in [1], [24], and [25], where tensor data were processed as part of a reconstruction/regression solution. Furthermore, unlike previous attempts, such as the one in [29], design issues of paramount importance in practical applications, such as the initialization, termination, convergence of the algorithm, and the determination of the subspace dimensionality, are discussed in detail.
2) The definition of eigentensors and n-mode eigenvalues as counterparts of the eigenvectors and eigenvalues in clas-

sical PCA. The geometrical interpretation of these concepts is provided, enabling a deeper understanding of the main principles and facilitating the application of multilinear feature extraction.

3) The presentation of a recognition system that selects discriminative tensor features from tensor objects and uses a novel weighting method for classification. This differs from traditional vector-based object recognition systems [16] that often encounter computational and memory difficulties when dealing with tensor object inputs. It also differs from [29], where all of the projected features were used for recognition.

4) The development of a solution to the gait recognizer by representing gait sequences as tensor samples and extracting discriminative features from them. This is a more natural approach that differs from [3]–[7], where either silhouettes or heuristic features derived from silhouettes were used as features.

The rest of this paper is organized as follows. Section II introduces basic multilinear algebra notations, concepts, and the notion of multilinear projection for dimensionality reduction. In Section III, the problem of MPCA is formulated and an iterative solution is presented. Initialization procedures, termination criteria, convergence, and subspace dimensionality are discussed in detail. The connection to PCA and 2-D PCA is illustrated. The computational aspects of the proposed framework are also discussed in this section. The problem of tensor object recognition is discussed in Section IV. Section V lists experiments on both synthetic data sets and true application data. Synthetic data sets are used to verify the properties of the proposed methodology while gait data sets are used to demonstrate performance on a recognition problem of particular importance. Finally, Section VI summarizes the major findings of this work.

II. MULTILINEAR PROJECTION OF TENSOR OBJECTS

This section briefly reviews some basic multilinear concepts used in the MPCA framework development and introduces the multilinear projection of tensor objects for the purpose of dimensionality reduction.

A. Notations and Basic Multilinear Algebra

Table I lists the fundamental symbols defined in this paper. The notations followed are those decreed by convention in the multilinear algebra, pattern recognition, and adaptive learning literature. Thus, in this paper, vectors are denoted by lowercase boldface letters, e.g., $\mathbf{x}$, matrices by uppercase boldface, e.g., $\mathbf{U}$, and tensors by calligraphic letters, e.g., $\mathcal{A}$. Their elements are denoted with indices in brackets. Indices are denoted by lowercase letters and span the range from 1 to the uppercase letter of the index, e.g., $n = 1, 2, \ldots, N$. To indicate part of a vector/matrix/tensor, ":" denotes the full range of the corresponding index and $n_1 : n_2$ denotes indices ranging from n_1 to n_2. Throughout this paper, the discussion is restricted to real-valued vectors, matrices, and tensors since the targeted applications, such as holistic gait recognition using binary silhouettes, involve real data only. The extension to the complex valued data sets is out of the scope of this work and it will be the focus of a forthcoming paper.

TABLE I
LIST OF SYMBOLS

$\mathcal{X}_m$	the mth input tensor sample, $m = 1, ..., M$
$\tilde{\mathbf{U}}^{(n)}$	the nth projection matrix, $n = 1, ..., N$
$\mathcal{Y}_m$	the projection of $\mathcal{X}_m$ on $\{\tilde{\mathbf{U}}^{(n)}\}$
$\Psi_{\mathcal{Y}}$	the total tensor scatter of $\{\mathcal{Y}_m, m = 1, ..., M\}$
$\mathbf{\Phi}^{(n)}$	the n-mode matrix defined in (5)
$\tilde{\mathbf{U}}_{\mathbf{\Phi}^{(n)}}$	the Kroneker product defined in (6)
$\tilde{\mathcal{X}}_m, \tilde{\mathcal{Y}}_m$	the centered versions of $\mathcal{X}_m, \mathcal{Y}_m$
$\mathbf{\Phi}^{(n)*}, \mathbf{U}^{(n)*}$	$\mathbf{\Phi}^{(n)}$ and $\tilde{\mathbf{U}}^{(n)}$ for full projection
$\mathcal{Y}_{var}^*$	the total scatter tensor for full projection
$\lambda_{i_n}^{(n)*}$	the i_nth n-mode eigenvalue for full projection
$\Psi_{\mathcal{Y}_0}$	the total tensor scatter for the FPT initialization
$\Psi_{\mathcal{X}}$	the input total tensor scatter
$\tilde{\mathcal{Y}}_{var}^*, \hat{\mathcal{Y}}_{var}^*$	truncated versions of $\mathcal{Y}_{var}^*$
k	the iteration step index in the MPCA algorithm
t	the SMT step index
$\delta_t^{(n)}$	the scatter loss rate defined in (11)
$\tilde{\lambda}_{P_n}^{(n)}$	the P_nth n-mode eigenvalue of the reconstructed input tensors
$\mathbf{y}_m$	the vectorized $\mathcal{Y}_m$ with only first $H_{\mathbf{y}}$ component kept, ordered according to $\Gamma_{p_1 p_2 \ldots p_N}$
$\lambda_{p_n}^{(n)}$	the p_nth n-mode eigenvalue corresponding to $\tilde{\mathbf{U}}^{(n)}$
$\mathbf{z}_m$	the projection of $\mathbf{y}_m$ in the LDA space

An Nth-order tensor is denoted as $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$. It is addressed by N indices $i_n, n = 1, \ldots, N$, and each i_n addresses the n-mode of $\mathcal{A}$. The n-mode product of a tensor $\mathcal{A}$ by a matrix $\mathbf{U} \in \mathbb{R}^{J_n \times I_n}$, denoted by $\mathcal{A} \times_n \mathbf{U}$, is a tensor with entries $(\mathcal{A} \times_n \mathbf{U})(i_1, \ldots, i_{n-1}, j_n, i_{n+1}, \ldots, i_N) = \sum_{i_n} \mathcal{A}(i_1, \ldots, i_N) \cdot \mathbf{U}(j_n, i_n)$. The scalar product of two tensors $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is defined as $\langle \mathcal{A}, \mathcal{B} \rangle = \sum_{i_1} \sum_{i_2} \cdots \sum_{i_N} \mathcal{A}(i_1, i_2, \ldots, i_N) \cdot \mathcal{B}(i_1, i_2, \ldots, i_N)$ and the Frobenius norm of $\mathcal{A}$ is defined as $\|\mathcal{A}\|_F = \sqrt{\langle \mathcal{A}, \mathcal{A} \rangle}$. The i_nth "n-mode slice" of $\mathcal{A}$ is an $(N-1)$th-order tensor obtained by fixing the n-mode index of $\mathcal{A}$ to be i_n: $\mathcal{A}(:, \ldots, :, i_n, :, \ldots, :)$. The "$n$-mode vectors" of $\mathcal{A}$ are defined as the I_n-dimensional vectors obtained from $\mathcal{A}$ by varying the index i_n while keeping all the other indices fixed. A rank-1 tensor $\mathcal{A}$ equals to the outer product of N vectors $\mathcal{A} = \mathbf{u}^{(1)} \circ \mathbf{u}^{(2)} \circ \cdots \circ \mathbf{u}^{(N)}$, which means that $\mathcal{A}(i_1, i_2, \ldots, i_N) = \mathbf{u}^{(1)}(i_1) \cdot \mathbf{u}^{(2)}(i_2) \cdot \cdots \cdot \mathbf{u}^{(N)}(i_N)$ for all values of indices. Unfolding $\mathcal{A}$ along the n-mode is denoted as $\mathbf{A}_{(n)} \in \mathbb{R}^{I_n \times (I_1 \times \cdots \times I_{n-1} \times I_{n+1} \times \cdots \times I_N)}$. The column vectors of $\mathbf{A}_{(n)}$ are the n-mode vectors of $\mathcal{A}$. Fig. 1 illustrates the 1-mode (column mode) unfolding of a third-order tensor.

Following standard multilinear algebra, any tensor $\mathcal{A}$ can be expressed as the product

$$\mathcal{A} = \mathcal{S} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times \cdots \times_N \mathbf{U}^{(N)} \tag{1}$$

where $\mathcal{S} = \mathcal{A} \times_1 \mathbf{U}^{(1)^T} \times_2 \mathbf{U}^{(2)^T} \cdots \times_N \mathbf{U}^{(N)^T}$ and $\mathbf{U}^{(n)} = (\mathbf{u}_1^{(n)} \mathbf{u}_2^{(n)} \cdots \mathbf{u}_{I_n}^{(n)})$ is an orthogonal $I_n \times I_n$ matrix. Since $\mathbf{U}^{(n)}$
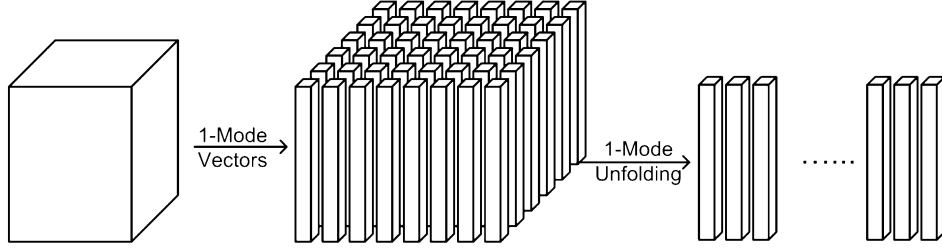
Fig. 1. Visual illustration of the 1-mode unfolding of a third-order tensor.
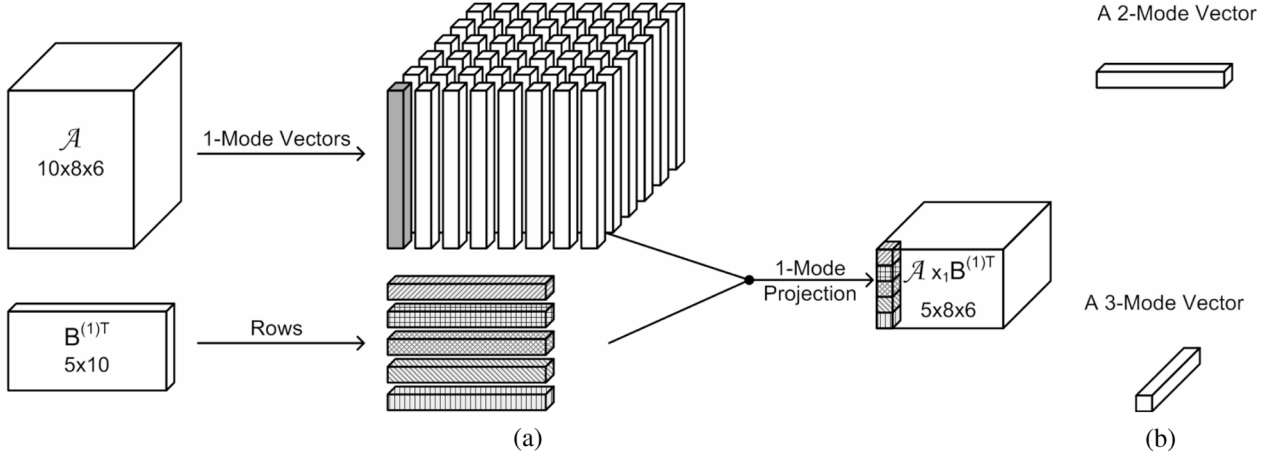

Fig. 2. Visual illustration of multilinear projection: (a) projection in the 1-mode vector space and (b) 2-mode and 3-mode vectors.

has orthonormal columns, $\|\mathcal{A}\|_F^2 = \|\mathcal{S}\|_F^2$ [1]. A matrix representation of this decomposition can be obtained by unfolding $\mathcal{A}$ and $\mathcal{S}$ as

$$\mathbf{A}_{(n)} = \mathbf{U}^{(n)} \cdot \mathbf{S}_{(n)} \cdot \left(\mathbf{U}^{(n+1)} \otimes \mathbf{U}^{(n+2)} \otimes \cdots \right.$$
$$\left. \otimes \mathbf{U}^{(N)} \otimes \mathbf{U}^{(1)} \otimes \mathbf{U}^{(2)} \otimes \cdots \otimes \mathbf{U}^{(n-1)} \right)^T \quad (2)$$

where $\bigotimes$ denotes the Kronecker product. The decomposition can also be written as

$$\mathcal{A} = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} \mathcal{S}(i_1, i_2, \ldots, i_N)$$
$$\times \mathbf{u}_{i_1}^{(1)} \circ \mathbf{u}_{i_2}^{(2)} \circ \cdots \circ \mathbf{u}_{i_N}^{(N)} \quad (3)$$

i.e., any tensor $\mathcal{A}$ can be written as a linear combination of $I_1 \times I_2 \times \cdots \times I_N$ rank-1 tensors. This decomposition is used in the following to formulate multilinear projection for dimensionality reduction.

B. Tensor Subspace Projection for Dimensionality Reduction

An Nth-order tensor $\mathcal{X}$ resides in the tensor (multilinear) space $\mathbb{R}^{I_1} \bigotimes \mathbb{R}^{I_2} \cdots \bigotimes \mathbb{R}^{I_N}$, where $\mathbb{R}^{I_1}, \mathbb{R}^{I_2}, \ldots, \mathbb{R}^{I_N}$ are the N vector (linear) spaces [23]. For typical image and video tensor objects such as 3-D face images and gait sequences, 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 [33]. 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, $P_n < I_n$ orthonormal basis vectors (principle axes) of the n-mode linear space $\mathbb{R}^{I_n}$ are sought for each mode n and a tensor subspace $\mathbb{R}^{P_1} \bigotimes \mathbb{R}^{P_2} \cdots \bigotimes \mathbb{R}^{P_N}$ is formed by these linear subspaces. Let $\tilde{\mathbf{U}}^{(n)}$ denote the $I_n \times P_n$ matrix containing the P_n orthonormal n-mode basis vectors. The projection of $\mathcal{X}$ onto the tensor subspace $\mathbb{R}^{P_1} \bigotimes \mathbb{R}^{P_2} \cdots \bigotimes \mathbb{R}^{P_N}$ is defined as $\mathcal{Y} = \mathcal{X} \times_1 \tilde{\mathbf{U}}^{(1)^T} \times_2 \tilde{\mathbf{U}}^{(2)^T} \cdots \times_N \tilde{\mathbf{U}}^{(N)^T}$.

The projection of an n-mode vector of $\mathcal{X}$ by $\tilde{\mathbf{U}}^{(n)^T}$ is computed as the inner product between the n-mode vector and the rows of $\tilde{\mathbf{U}}^{(n)^T}$. Fig. 2 provides a visual illustration of the multilinear projection. In Fig. 2(a), a third-order tensor $\mathcal{A} \in \mathbb{R}^{10 \times 8 \times 6}$ is projected in the 1-mode vector space by a projection matrix $\mathbf{B}^{(1)^T} \in \mathbb{R}^{5 \times 10}$, resulting in the projected tensor $\mathcal{A} \times_1 \mathbf{B}^{(1)^T} \in \mathbb{R}^{5 \times 8 \times 6}$. In the 1-mode projection, each 1-mode vector of $\mathcal{A}$ of length 10 is projected by $\mathbf{B}^{(1)^T}$ to obtain a vector of length 5, as the differently shaded vectors indicate in Fig. 2(a). Similarly, Fig. 2(b) depicts the 2-mode and 3-mode vectors.

III. MULTILINEAR PRINCIPAL COMPONENT ANALYSIS

In this section, an MPCA solution to the problem of dimensionality reduction for tensor objects is introduced, researched, and analyzed. Before formally stating the objective, the following definition is needed.

Definition 1: Let $\{\mathcal{A}_m, m = 1, \ldots, M\}$ be a set of M tensor samples in $\mathbb{R}^{I_1} \bigotimes \mathbb{R}^{I_2} \cdots \bigotimes \mathbb{R}^{I_N}$. The total scatter of these tensors is defined as $\Psi_{\mathcal{A}} = \sum_{m=1}^{M} \|\mathcal{A}_m - \bar{\mathcal{A}}\|_F^2$, where $\bar{\mathcal{A}}$ is the mean tensor calculated as $\bar{\mathcal{A}} = (1/M) \sum_{m=1}^{M} \mathcal{A}_m$. The n-mode total scatter matrix of these samples is then defined as

Input: A set of tensor samples $\{\mathcal{X}_m \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}, m = 1, ..., M\}$.

Output: Low-dimensional representations $\{\mathcal{Y}_m \in \mathbb{R}^{P_1 \times P_2 \times \dots \times P_N}, m = 1, ..., M\}$ of the input tensor samples with maximum variation captured.

Algorithm:

Step 1 (Preprocessing): Center the input samples as $\{\tilde{\mathcal{X}}_m = \mathcal{X}_m - \bar{\mathcal{X}}, m = 1, ..., M\}$, where $\bar{\mathcal{X}} = \frac{1}{M} \sum_{m=1}^{M} \mathcal{X}_m$ is the sample mean.

Step 2 (Initialization): Calculate the eigen-decomposition of $\mathbf{\Phi}^{(n)*} = \sum_{m=1}^{M} \tilde{\mathbf{X}}_{m(n)} \cdot \tilde{\mathbf{X}}_{m(n)}^T$ and set $\tilde{\mathbf{U}}^{(n)}$ to consist of the eigenvectors corresponding to the most significant P_n eigenvalues, for $n = 1, ..., N$.

Step 3 (Local optimization):

- Calculate $\{\tilde{\mathcal{Y}}_m = \tilde{\mathcal{X}}_m \times_1 \tilde{\mathbf{U}}^{(1)^T} \times_2 \tilde{\mathbf{U}}^{(2)^T} ... \times_N \tilde{\mathbf{U}}^{(N)^T}, m = 1, ..., M\}$.
- Calculate $\Psi_{\mathcal{Y}_0} = \sum_{m=1}^{M} \| \tilde{\mathcal{Y}}_m \|_F^2$ (the mean $\bar{\tilde{\mathcal{Y}}}$ is all zero since $\tilde{\mathcal{X}}_m$ is centered).
- For $k = 1 : K$
  - For $n = 1 : N$
    * Set the matrix $\tilde{\mathbf{U}}^{(n)}$ to consist of the P_n eigenvectors of the matrix $\mathbf{\Phi}^{(n)}$, as defined in (5), corresponding to the largest P_n eigenvalues.
  - Calculate $\{\tilde{\mathcal{Y}}_m, m = 1, ..., M\}$ and $\Psi_{\mathcal{Y}_k}$.
  - If $\Psi_{\mathcal{Y}_k} - \Psi_{\mathcal{Y}_{k-1}} < \eta$, break and go to Step 4.

Step 4 (Projection): The feature tensor after projection is obtained as $\{\mathcal{Y}_m = \mathcal{X}_m \times_1 \tilde{\mathbf{U}}^{(1)^T} \times_2 \tilde{\mathbf{U}}^{(2)^T} ... \times_N \tilde{\mathbf{U}}^{(N)^T}, m = 1, ..., M\}$.

Fig. 3. Pseudocode implementation of the proposed MPCA algorithm.

$\mathbf{S}_{T_\mathcal{A}}^{(n)} = \sum_{m=1}^{M} (\mathbf{A}_{m(n)} - \bar{\mathbf{A}}_{(n)})(\mathbf{A}_{m(n)} - \bar{\mathbf{A}}_{(n)})^T$, where $\mathbf{A}_{m(n)}$ is the n-mode unfolded matrix of $\mathcal{A}_m$.

The previous statement leads to the following formal definition of the problem to be solved.

A set of M tensor objects $\{\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_M\}$ is available for training. Each tensor object $\mathcal{X}_m \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ assumes values in a tensor space $\mathbb{R}^{I_1} \bigotimes \mathbb{R}^{I_2} \dots \bigotimes \mathbb{R}^{I_N}$, where I_n is the n-mode dimension of the tensor. The MPCA objective is to define a multilinear transformation $\{\tilde{\mathbf{U}}^{(n)} \in \mathbb{R}^{I_n \times P_n}, n = 1, \dots, N\}$ that maps the original tensor space $\mathbb{R}^{I_1} \bigotimes \mathbb{R}^{I_2} \dots \bigotimes \mathbb{R}^{I_N}$ into a tensor subspace $\mathbb{R}^{P_1} \bigotimes \mathbb{R}^{P_2} \dots \bigotimes \mathbb{R}^{P_N}$ (with $P_n < I_n$, for $n = 1, \dots, N$): $\mathcal{Y}_m = \mathcal{X}_m \times_1 \tilde{\mathbf{U}}^{(1)^T} \times_2 \tilde{\mathbf{U}}^{(2)^T} \dots \times_N \tilde{\mathbf{U}}^{(N)^T}, m = 1, \dots, M$, such that $\{\mathcal{Y}_m \in \mathbb{R}^{P_1} \bigotimes \mathbb{R}^{P_2} \dots \bigotimes \mathbb{R}^{P_N}, m = 1, \dots, M\}$ captures most of the variations observed in the original tensor objects, assuming that these variations are measured by the total tensor scatter.

In other words, the MPCA objective is the determination of the N projection matrices $\{\tilde{\mathbf{U}}^{(n)} \in \mathbb{R}^{I_n \times P_n}, n = 1, \dots, N\}$ that maximize the total tensor scatter $\Psi_\mathcal{Y}$

$$\{\tilde{\mathbf{U}}^{(n)}, n = 1, \dots, N\} = \arg \max_{\tilde{\mathbf{U}}^{(1)}, \tilde{\mathbf{U}}^{(2)}, \dots, \tilde{\mathbf{U}}^{(N)}} \Psi_\mathcal{Y}. \quad (4)$$

Here, the dimensionality P_n for each mode is assumed to be known or predetermined. Discussions on the adaptive determination of P_n, when it is not known in advance, will be presented in Section III-F.

A. MPCA Algorithm

To the best of the authors' knowledge, there is no known optimal solution which allows for the simultaneous optimization of the N projection matrices. Since the projection to an Nth-order tensor subspace consists of N projections to N vector subspaces, N optimization subproblems can be solved by finding the $\hat{\mathbf{U}}^{(n)}$ that maximizes the scatter in the n-mode vector subspace. This is discussed in Theorem 1.

Theorem 1: Let $\{\tilde{\mathbf{U}}^{(n)}, n = 1, \dots, N\}$ be the solution to (4). Then, given all the other projection matrices $\tilde{\mathbf{U}}^{(1)}, \dots, \tilde{\mathbf{U}}^{(n-1)}, \tilde{\mathbf{U}}^{(n+1)}, \dots, \tilde{\mathbf{U}}^{(N)}$, the matrix $\tilde{\mathbf{U}}^{(n)}$ consists of the P_n eigenvectors corresponding to the largest P_n eigenvalues of the matrix

$$\mathbf{\Phi}^{(n)} = \sum_{m=1}^{M} \left(\mathbf{X}_{m(n)} - \bar{\mathbf{X}}_{(n)}\right) \cdot \tilde{\mathbf{U}}_{\mathbf{\Phi}^{(n)}}$$
$$\cdot \tilde{\mathbf{U}}_{\mathbf{\Phi}^{(n)}}^T \cdot \left(\mathbf{X}_{m(n)} - \bar{\mathbf{X}}_{(n)}\right)^T \quad (5)$$

where

$$\tilde{\mathbf{U}}_{\mathbf{\Phi}^{(n)}} = \left(\tilde{\mathbf{U}}^{(n+1)} \otimes \tilde{\mathbf{U}}^{(n+2)} \otimes \dots \otimes \tilde{\mathbf{U}}^{(N)}\right.$$
$$\left. \otimes \tilde{\mathbf{U}}^{(1)} \otimes \tilde{\mathbf{U}}^{(2)} \otimes \dots \tilde{\mathbf{U}}^{(n-1)}\right). \quad (6)$$

Proof: The proof of Theorem 1 is given in Appendix I-B. ∎

Since the product $\tilde{\mathbf{U}}_{\mathbf{\Phi}^{(n)}} \cdot \tilde{\mathbf{U}}_{\mathbf{\Phi}^{(n)}}^T$ depends on $\tilde{\mathbf{U}}^{(1)}, \dots, \tilde{\mathbf{U}}^{(n-1)}, \tilde{\mathbf{U}}^{(n+1)}, \dots, \tilde{\mathbf{U}}^{(N)}$, the optimization of $\tilde{\mathbf{U}}^{(n)}$ depends on the projections in other modes and there is no closed-form solution to this maximization problem. Instead, from Theorem 1, an iterative procedure can be utilized to solve (4), along the lines of the pseudocode summarized in Fig. 3. The input tensors are centered first: $\{\tilde{\mathcal{X}}_m = \mathcal{X}_m - \bar{\mathcal{X}}, m = 1, \dots, M\}$. With initializations through full projection truncation (FPT), which is to be discussed in details in Section III-C, the projection matrices are computed
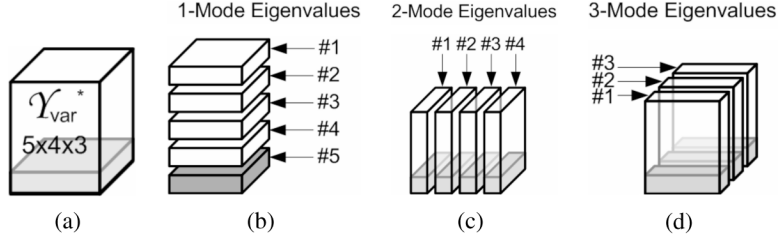
Fig. 4. Visual illustration of (a) total scatter tensor, (b) 1-mode eigenvalues, (c) 2-mode eigenvalues, and (d) 3-mode eigenvalues.

one by one with all the others fixed (local optimization). The local optimization procedure can be repeated, in a similar fashion as the ALS method [34], until the result converges or a maximum number K of iterations is reached.

Remark 1: The issue of centering has been ignored in the existing tensor processing literature. In the authors' opinion, the main reason for the apparent lack of studies on the problem of tensor data centering is due to the fact that previously published works focused predominately on tensor approximation and reconstruction. It should be pointed out that for the approximation/reconstruction problem, centering is not essential, as the (sample) mean is the main focus of attention. However, in recognition applications where the solutions involve eigenproblems, noncentering (in other words, an average different from zero) can potentially affect the per-mode eigendecomposition and lead to a solution that captures the variation with respect to the origin rather than capturing the true variation of the data (with respect to the data center).

Remark 2: The effects of the ordering of the projection matrices to be computed have been studied empirically in this work and simulation results presented in Section V indicate that altering the ordering of the projection matrix computation does not result in significant performance differences in practical situations.

In the following sections, several issues pertinent to the development and implementation of the MPCA algorithm are discussed. First, in-depth understanding of the MPCA framework is provided. The properties of full projection are analyzed, and the geometric interpretation of the n-mode eigenvalues is introduced together with the concept of eigentensor. In the sequence, the initialization method and the construction of termination criteria are described and convergence issues are also discussed. Finally, methods for subspace dimensionality determination are proposed and the connection to PCA and 2-D PCA is discussed, followed by computational issues.

B. Full Projection

With respect to this analysis, the term full projection refers to the multilinear projection for MPCA with $P_n = I_n$ for $n = 1, \ldots, N$. In this case, $\tilde{\mathbf{U}}_{\boldsymbol{\Phi}^{(n)}} \cdot \tilde{\mathbf{U}}_{\boldsymbol{\Phi}^{(n)}}^T$ is an identity matrix, as it can be seen from the pertinent lemma listed in Appendix I-C. As a result, $\boldsymbol{\Phi}^{(n)}$ reduces to $\boldsymbol{\Phi}^{(n)*} = \sum_{m=1}^{M} (\mathbf{X}_{m(n)} - \bar{\mathbf{X}}_{(n)}) \cdot (\mathbf{X}_{m(n)} - \bar{\mathbf{X}}_{(n)})^T$, with $\boldsymbol{\Phi}^{(n)*}$ determined by the input tensor samples only and independent of other projection matrices. The optimal $\tilde{\mathbf{U}}^{(n)} = \mathbf{U}^{(n)*}$ is then obtained as the matrix comprised of the eigenvectors of $\boldsymbol{\Phi}^{(n)*}$ directly without iteration, and the total scatter $\Psi_{\mathcal{X}}$ in the original data is fully captured. However,

there is no dimensionality reduction through this full projection. From the properties of eigendecomposition, it can be concluded that if all eigenvalues (per mode) are distinct, the full projection matrices (corresponding eigenvectors) are also distinct and that the full projection is unique (up to sign) [35].

To interpret the geometric meanings of the n-mode eigenvalues, the total scatter tensor $\mathcal{Y}_{\text{var}}^* \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ of the full projection is introduced as an extension of the total scatter matrix [36]. Each entry of the tensor $\mathcal{Y}_{\text{var}}^*$ is defined as

$$\mathcal{Y}_{\text{var}}^*(i_1, i_2, \ldots, i_N) = \sum_{m=1}^{M} [(\mathcal{Y}_m^* - \bar{\mathcal{Y}}^*)(i_1, i_2, \ldots, i_N)]^2 \quad (7)$$

where $\mathcal{Y}_m^* = \mathcal{X}_m \times_1 \mathbf{U}^{(1)*^T} \cdots \times_N \mathbf{U}^{(N)*^T}$ and $\bar{\mathcal{Y}}^* = (1/M) \sum_{m=1}^{M} \mathcal{Y}_m^*$. Using the previous definition, it can be shown that for the so-called full projection ($P_n = I_n$ for all n), the i_nth n-mode eigenvalue $\lambda_{i_n}^{(n)*}$ is the sum of all the entries of the i_nth n-mode slice of $\mathcal{Y}_{\text{var}}^*$

$$\lambda_{i_n}^{(n)*} = \sum_{i_1=1}^{I_1} \cdots \sum_{i_{n-1}=1}^{I_{n-1}} \sum_{i_{n+1}=1}^{I_{n+1}} \cdots$$
$$\sum_{i_N=1}^{I_N} \mathcal{Y}_{\text{var}}^*(i_1, \ldots, i_{n-1}, i_n, i_{n+1}, \ldots, i_N). \quad (8)$$

In this paper, the eigenvalues are all arranged in a decreasing order. Fig. 4 shows visually what the n-mode eigenvalues represent. In this graph, third-order tensors, e.g., short sequences (three frames) of images with size 5×4, are projected to a tensor space of size $5 \times 4 \times 3$ (full projection) so that a total scatter tensor $\mathcal{Y}_{\text{var}}^* \in \mathbb{R}^{5 \times 4 \times 3}$ is obtained.

Using (3), each tensor $\mathcal{X}_m$ can be written as a linear combination of $P_1 \times P_2 \times \cdots \times P_N$ rank-1 tensors $\tilde{\mathcal{U}}_{p_1 p_2 \cdots p_N} = \tilde{\mathbf{u}}_{p_1}^{(1)} \circ \tilde{\mathbf{u}}_{p_2}^{(2)} \circ \cdots \circ \tilde{\mathbf{u}}_{p_N}^{(N)}$. These rank-1 tensors will be called, hereafter, eigentensors. Thus, the projected tensor $\mathcal{Y}_m^*$ can be viewed as the projection onto these eigentensors, with each entry of $\mathcal{Y}_m^*$ corresponding to one eigentensor. These definitions and illustrations for MPCA help with understanding the MPCA framework in the following discussions.

C. Initialization by Full Projection Truncation

FPT is used to initialize the iterative solution for MPCA, where the first P_n columns of the full projection matrix $\mathbf{U}^{(n)*}$ is kept to give an initial projection matrix $\tilde{\mathbf{U}}^{(n)}$. The corresponding total scatter is denoted as $\Psi_{\mathcal{Y}_0}$ and this initialization is equivalent to the HOSVD-based solution in [28]. Although this FPT initialization is not the optimal solution to (4), it is bounded

and is considered a good starting point for the iterative procedure, as will be discussed in the following.

Remark 3: There are other choices of initialization such as truncated identity matrices [20], [23], [31] (named as pseudoidentity matrices) and random matrices. Simulation studies (reported in Section V) indicate that although in practical applications, the initialization step may not have a significant impact in terms of performance, it can affect the speed of convergence of the iterative solution. Since FPT results in much faster convergence, it is the one utilized throughout this work for initialization purposes.

In studying the optimality, with respect to (4), of the initialization procedure, let us assume, without loss of generality, that the 1-mode eigenvectors are truncated, in other words, only the first $P_1 < I_1$ 1-mode eigenvectors are kept. In this case, Theorem 2 applies.

Theorem 2: Let $\mathbf{U}^{(n)*}$ and $\lambda_{i_n}^{(n)*}, i_n = 1, \ldots, I_n$ be the matrix of the eigenvectors of $\mathbf{\Phi}^{(n)*}$ and the eigenvalues of $\mathbf{\Phi}^{(n)*}$, respectively, and $\tilde{\mathcal{Y}}_m = \check{\mathcal{X}}_m \times_1 \mathbf{U}^{(1)*T} \cdots \times_N \mathbf{U}^{(N)*T}, m = 1, \ldots, M$. Keep only the first $P_1 < I_1$ eigenvectors with $\sum_{i_1=P_1+1}^{I_1} \lambda_{i_1}^{(1)*} > 0$ to get $\check{\mathcal{X}}_m = \tilde{\mathcal{Y}}_m \times_1 \check{\mathbf{U}}^{(1)} \times_2 \mathbf{U}^{(2)*} \cdots \times_N \mathbf{U}^{(N)*}$, where $\check{\mathcal{Y}}_m = \tilde{\mathcal{Y}}_m(1:P_1, :, \ldots, :)$ and $\check{\mathbf{U}}^{(1)} = \mathbf{U}^{(1)*}(:, 1:P_1)$. Let $\check{\mathbf{\Phi}}^{(n)}$ correspond to $\check{\mathcal{X}}_m$, and the matrix of its eigenvectors and its eigenvalues be $\hat{\mathbf{U}}^{(n)}$ and $\hat{\lambda}_{i_n}^{(n)}$, respectively. Then

$$\hat{\lambda}_{i_1}^{(1)} = \begin{cases} \lambda_{i_1}^{(1)*}, & i_1 = 1, \ldots, P_1 \\ 0, & i_1 = P_1+1, \ldots, I_1 \end{cases}.$$

For $n > 1$ (other modes), $\hat{\lambda}_{i_n}^{(n)} \leq \lambda_{i_n}^{(n)*}$. Furthermore, for each mode, at least for one value of $i_n, \hat{\lambda}_{i_n}^{(n)} < \lambda_{i_n}^{(n)*}$.

Proof: The proof is given in Appendix I-D. ∎

It can be seen from Theorem 2 that if a nonzero eigenvalue is truncated in one mode, the eigenvalues in all the other modes tend to decrease in magnitude and the corresponding eigenvectors change accordingly. Thus, the eigendecomposition needs to be recomputed in all the other modes, i.e., the projection matrices in all the other modes need to be updated. Since from Theorem 1 the computations of all the projection matrices are interdependent, the update of a projection matrix $\tilde{\mathbf{U}}^{(n^*)}$ updates the matrices $\{\tilde{\mathbf{U}}_{\mathbf{\Phi}^{(n)}}, n \neq n^*\}$ as well. Consequently, the projection matrices in all the other modes $\{\tilde{\mathbf{U}}^{(n)}, n \neq n^*\}$ are no longer consisting of the eigenvectors of the corresponding (updated) $\mathbf{\Phi}^{(n)}$ and they need to be updated. The update continues until the termination criterion, which is discussed in Section III-D, is satisfied.

Fig. 4 provides a visual illustration of Theorem 2. Removal of a basis vector in one mode results in eliminating a slice of $\mathcal{Y}_{\text{var}}^*$. In Fig. 4, if the last nonzero (fifth) 1-mode eigenvalue is discarded [shaded in Fig. 4(b)], the corresponding (fifth) 1-mode slice of $\mathcal{Y}_{\text{var}}^*$ is removed [shaded in Fig. 4(a)], resulting in a truncated total scatter tensor $\tilde{\mathcal{Y}}_{\text{var}}^* \in \mathbb{R}^{4 \times 4 \times 3}$. Discarding this slice will affect all eigenvalues in the remaining modes, whose corresponding slices have a nonempty overlap with the discarded 1-mode slice. In Fig. 4(c) and (d), the shaded part indicates the removed 1-mode slice corresponding to the discarded eigenvalue.

Having proven the nonoptimality of FPT with respect to the objective function (4), we proceed to derive the bounds for FPT in Theorem 3.

Theorem 3: Let $\lambda_{i_n}^{(n)*}$ denote the i_nth n-mode eigenvalue for the n-mode full projection matrix. The upper and lower bounds for $(\Psi_{\mathcal{X}} - \Psi_{\mathcal{Y}_0})$, the loss of variation due to the FPT (measured by the total scatter), are derived as follows:

$$\Psi_L = \max_n \sum_{i_n=P_n+1}^{I_n} \lambda_{i_n}^{(n)*} \leq (\Psi_{\mathcal{X}} - \Psi_{\mathcal{Y}_0})$$

$$\leq \sum_{n=1}^{N} \sum_{i_n=P_n+1}^{I_n} \lambda_{i_n}^{(n)*} = \Psi_U. \tag{9}$$

Proof: The proof is given in Appendix I-D. ∎

From (9), it can be seen that the tightness of the bounds is determined by the eigenvalues in each mode. The bounds can be observed in Fig. 4. For instance, truncation of the last eigenvector in each of the three modes results in another truncated total scatter tensor $\hat{\mathcal{Y}}_{\text{var}}^* \in \mathbb{R}^{4 \times 3 \times 2}$, and thus the difference between $\Psi_{\mathcal{X}}$ and $\Psi_{\mathcal{Y}_0}$ (the sum of all entries in $\mathcal{Y}_{\text{var}}^*$ and $\hat{\mathcal{Y}}_{\text{var}}^*$, respectively) is upper bounded by the total of the sums of all the entries in each truncated slice and lower bounded by the maximum sum of all the entries in each truncated slice. For FPT, the gap between the actual loss of variation and the upper bound is due to the multiple counts of the overlaps between the discarded slice in one mode and the discarded slices in the other modes of $\mathcal{Y}_{\text{var}}^*$.

The tightness of the bounds Ψ_U and Ψ_L depends on the order N, the eigenvalue characteristics (distribution) such as the number of zero-valued eigenvalues, and the degree of truncation P_n. For example, for $N = 1$, which is the case of PCA, $\Psi_L = \Psi_U$ and the FPT is the optimal solution so no iterations are necessary. Larger N results in more terms in the upper bound and tends to lead to looser bound, and vice versa. In addition, if all the truncated eigenvectors correspond to zero-valued eigenvalues, $\Psi_{\mathcal{Y}_0} = \Psi_{\mathcal{X}}$ since $\Psi_L = \Psi_U = 0$, and the FPT results in the optimal solution.

D. Termination

The termination criterion is to be determined in this paper using the objective function $\Psi_{\mathcal{Y}}$. In particular, the iterative procedure terminates if $(\Psi_{\mathcal{Y}_k} - \Psi_{\mathcal{Y}_{k-1}}) < \eta$, where $\Psi_{\mathcal{Y}_k}$ and $\Psi_{\mathcal{Y}_{k-1}}$ are the resulted total scatter from the kth and $(k-1)$th iterations, respectively, and η is a user-defined small number threshold (e.g., $\eta = 10^{-6}$). In other words, the iterations stop if there is little improvement in the resulted total scatter (the objective function). In addition, the maximum number of iterations allowed is set to K for computational consideration.

E. Convergence of the MPCA Algorithm

The derivation of Theorem 1 (Appendix I-B) implies that per iteration, the total scatter $\Psi_{\mathcal{Y}}$ is a nondecreasing function (as it either remains the same or increases) since each update of the projection matrix $\tilde{\mathbf{U}}^{(n^*)}$ in a given mode n^* maximizes $\Psi_{\mathcal{Y}}$, while the projection matrices in all the other modes $\{\tilde{\mathbf{U}}^{(n)}, n \neq n^*\}$ are considered fixed. On the other hand, $\Psi_{\mathcal{Y}}$ is upper bounded by $\Psi_{\mathcal{X}}$ (the variation in the original samples)

since the projection matrices $\{\tilde{\mathbf{U}}^{(n)}\}$ consist of orthonormal columns. Therefore, MPCA is expected to have good convergence property. Empirical results presented in Section V indicate that the proposed MPCA algorithm converges very fast (within five iterations) for typical tensor objects. Furthermore, when per-mode eigenvalues are all distinct (with multiplicity 1), which is the case for the simulated data as well as the gait data, the projection matrices $\{\tilde{\mathbf{U}}^{(n)}\}$, which maximize the objective function $\Psi_{\mathcal{Y}}$, are expected to converge as well. It should be noted that the claimed convergence regarding the projection matrices $\{\tilde{\mathbf{U}}^{(n)}\}$ is under the condition that the sign for the first component of each n-mode eigenvector is fixed since the eigenvector is unique up to sign. Simulation studies show that the projection matrices $\{\tilde{\mathbf{U}}^{(n)}\}$ do converge well.

F. Determination of the Tensor Subspace Dimensionality

When the targeted dimensionality $\{P_n, n = 1, \ldots, N\}$ is not specified in advance, its value has to be determined. Consequently, the objective function (4) needs to be revised to include a constraint on the desired dimensionality reduction. The revised objective function is as follows:

$$\{\tilde{\mathbf{U}}^{(n)}, P_n, n = 1, \ldots, N\}$$
$$= \arg \max_{\tilde{\mathbf{U}}^{(1)}, \tilde{\mathbf{U}}^{(2)}, \ldots, \tilde{\mathbf{U}}^{(N)}, P_1, P_2, \cdots, P_N} \Psi_{\mathcal{Y}}$$
$$\text{subject to } \frac{\prod_{n=1}^{N} P_n}{\prod_{n=1}^{N} I_n} < \Omega \qquad (10)$$

where the ratio between the targeted (reduced) dimensionality and the original tensor space dimensionality is utilized to measure the amount of dimensionality reduction, and Ω is a threshold to be specified by the user or determined based on empirical studies.

The proposed tensor subspace dimensionality determination solution is called sequential mode truncation (SMT). Starting with $P_n = I_n$ for all n at $t = 0$, at each subsequent step $t = t+1$, the SMT truncates, in a selected mode n, the P_nth n-mode eigenvector of the reconstructed input tensors. The truncation could be interpreted as the elimination of the corresponding P_nth n-mode slice of the total scatter tensor. For the mode selection, the scatter loss rate $\delta_t^{(n)}$ due to the truncation of its P_nth eigenvector is calculated for each mode. $\delta_t^{(n)}$ is defined as follows:

$$\delta_t^{(n)} = \frac{\Psi_{\mathcal{Y}_{(t)}} - \Psi_{\mathcal{Y}_{(t-1)}}}{\left[P_n \cdot \prod_{j=1, j \neq n}^{N} P_j\right] - \left[(P_n - 1) \cdot \prod_{j=1, j \neq n}^{N} P_j\right]}$$
$$= \frac{\tilde{\lambda}_{P_n}^{(n)}}{\prod_{j=1, j \neq n}^{N} P_j} \qquad (11)$$

where $\Psi_{\mathcal{Y}_{(t)}}$ is the scatter obtained at step t, $\prod_{j=1, j \neq n}^{N} P_j$ is the amount of dimensionality reduction achieved, and $\tilde{\lambda}_{P_n}^{(n)}$, which is the corresponding P_nth n-mode eigenvalue, is the loss of variation due to truncating the P_nth n-mode eigenvector. The mode with the smallest $\delta_t^{(n)}$ is selected for the step-t truncation. For the selected mode n, P_n is decreased by 1: $P_n = P_n - 1$

and $(\prod_{n=1}^{N} P_n / \prod_{n=1}^{N} I_n) < \Omega$ is tested. The truncation stops when $(\prod_{n=1}^{N} P_n / \prod_{n=1}^{N} I_n) < \Omega$ is satisfied. Otherwise, the input tensors are reconstructed according to (1) using the current truncated projection matrices and they are used to recompute the n-mode eigenvalues and eigenvectors corresponding to full projection. Since eigenvalues in other modes are affected by the eigenvector truncation in a given mode (see Theorem 2), it is reasonable to assume that a procedure such as the SMT, which takes into account this effect, constitutes a reasonable good choice for determining P_n in the sense of (10).

The Q-based method, a suboptimal, simplified dimensionality determination procedure that requires no recomputation, can be also used in practice. Let us define the ratio $Q^{(n)} = (\sum_{i_n=1}^{P_n} \lambda_{i_n}^{(n)*} / \sum_{i_n=1}^{I_n} \lambda_{i_n}^{(n)*})$ to be the remained portion of the total scatter in the n-mode after the truncation of the n-mode eigenvectors beyond the P_nth, where $\lambda_{i_n}^{(n)*}$ is the i_nth full-projection n-mode eigenvalue. In the proposed Q-based method, the first P_n eigenvectors are kept in the n-mode (for each n) so that $Q^{(1)} = Q^{(2)} = \cdots = Q^{(N)} = Q$ (the equality can hold approximately since it is unlikely to find P_n that gives the exact equality in practice). It should be noted that $\sum_{i_n=1}^{I_n} \lambda_{i_n}^{(n)*} = \Psi_{\mathcal{X}}$ for all n since from Theorem 1, the total scatter for the full projection was given as $\Psi_{\mathcal{Y}}^* = \Psi_{\mathcal{X}} = \sum_{m=1}^{M} \|\mathbf{Y}_{m(n)} - \bar{\mathbf{Y}}_{(n)}\|_F^2 = \sum_{i_n=1}^{I_n} \lambda_{i_n}^{(n)*}, n = 1, \ldots, N$. This method can be viewed as an extension of the dimensionality selection strategy in traditional PCA to the multilinear case. The reason behind this choice is that loss of variation is (approximately) proportional to the sum of the corresponding eigenvalues of the discarded eigenvectors. By discarding the least significant eigenvectors in each mode, the variation loss can be contained and a tighter lower bound for $\Psi_{\mathcal{Y}_0}$ is obtained. The empirical study reported in the experimental section indicated that the Q-based method provides results similar to those obtained by SMT (as measured in terms of the total scatter captured) and thus it can be safely used instead of the more computationally expensive SMT alternative.

G. MPCA Versus PCA and 2-D PCA Solutions

It is not difficult to see that the MPCA framework generalizes not only the classical PCA solution but also a number of the so-called 2-D PCA algorithms.

Indeed, for $N = 1$, the input samples are vectors $\{\mathbf{x}_m \in \mathbb{R}^{I_1}\}$. There is only one mode and MPCA is reduced to PCA. For dimensionality reduction purposes, only one projection matrix $\mathbf{U}$ is needed in order to obtain $\mathbf{y}_m = \mathbf{x}_m \times_1 \mathbf{U} = \mathbf{U}^T \mathbf{x}_m$. In this case, there is only one $\mathbf{\Phi}^{(n)} = \mathbf{\Phi}^{(1)} = \sum_{m=1}^{M} (\mathbf{x}_m - \bar{\mathbf{x}}) \cdot (\mathbf{x}_m - \bar{\mathbf{x}})^T$, which is the total scatter matrix of the input samples in PCA [36]. The projection matrix maximizing the total scatter (variation) in the projected space is determined from the eigenvectors of $\mathbf{\Phi}^{(1)}$. Thus, MPCA subsumes PCA.

In the so-called 2-D PCA solutions, input samples are treated as matrices, in other words second-order tensors. Two (left and right) projection matrices are sought to maximize the captured variation in the projected space. With $N = 2$, the proposed MPCA algorithm (Fig. 3) is equivalent to the 2-D PCA solution of [22], with the exception of the initialization procedure and termination criterion. Other 2-D PCA algorithms such as those discussed in [21] can be viewed as variations of the method in
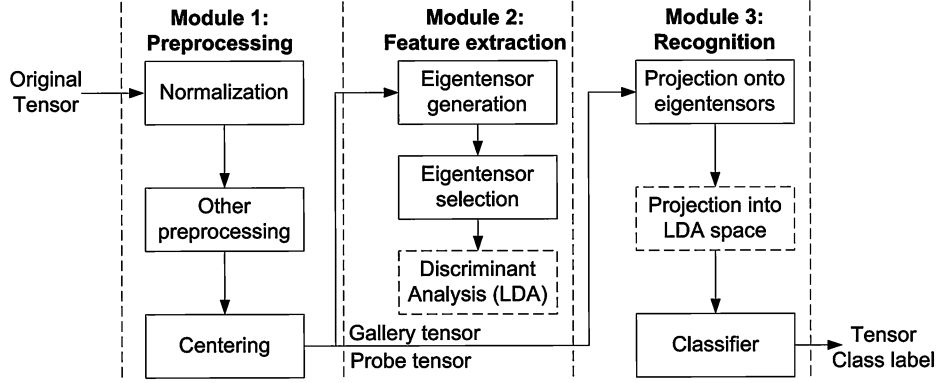
Fig. 5. Block diagram of the MPCA-based tensor object recognition system.

[22] and thus they can be considered special cases of MPCA when second-order tensors are considered.

H. Computational Issues

Apart from the actual performance of any proposed algorithm, its computational complexity, memory requirements, and storage needs are relative measures of its practicality and usefulness as they determine the required computing power and processing (execution) time. To this end, we examine MPCA-related computational issues in a fashion similar to that introduced in [20].

Since this is an iterative solution, the computational complexity analysis is performed for one iteration. For simplicity, it is assumed that $I_1 = I_2 = \cdots = I_N = (\prod_{n=1}^N I_n)^{(1/n)} = I$. From a computational complexity point of view, the most demanding steps are the formation of the matrices $\Phi^{(n)}$, the eigendecomposition of $\Phi^{(n)}$, and the computation of the multilinear projection $\tilde{\mathcal{Y}}_m$. It should be noted that the use of multilinear multiplication and unfolding in order to compute $\Phi^{(n)}$ is more efficient comparing to the use of Kronecker products. The computations needed to determine $\Phi^{(n)}$, the P_n eigenvectors of $\Phi^{(n)}$, and $\tilde{\mathcal{Y}}_m$ are in order of $O(MN \cdot I^{(N+1)})$ (upper bounded), $O(I^3)$, and $O(N \cdot I^{(N+1)})$, respectively. The total complexity is $O((N+1) \cdot MN \cdot I^{(N+1)} + N * I^3)$.

In MPCA, $\bar{\mathcal{X}}$ and $\Phi^{(n)}$ can be computed incrementally by reading $\mathcal{X}_m$ or $\tilde{\mathcal{X}}_m$ sequentially without loss of information. Hence, memory requirements for the MPCA algorithm can be as low as $O(\prod_{n=1}^N I_n)$ as MPCA computes the solution without requiring all data samples in the memory. This is a major advantage that MPCA enjoys over the HOSVD-based solutions [26], [28], which requires the formation of an $(N+1)$th-order tensor when the input tensor samples are of Nth-order. This is of considerable importance in applications with large data sets as the size of the input database may lead to increased complexity and high memory storage requirement.[1]

MPCA compresses each tensor sample of size $\prod_{n=1}^N I_n$ to $\prod_{n=1}^N P_n$, and it needs N matrices $\tilde{U}^{(n)} \in \mathbb{R}^{I_n \times P_n}$ for compression and decompression. Thus, it requires

$(M \cdot \prod_{n=1}^N P_n + \sum_{n=1}^N (I_n \times P_n))$ scalars in the reduced space by MPCA and the compression ratio (CR) is defined as $\text{CR} = M \cdot \prod_{n=1}^N I_n / (M \cdot \prod_{n=1}^N P_n + \sum_{n=1}^N (I_n \times P_n))$. In studying the subspace dimensionality determination performance in the experiments, algorithms are compared under the same CR.

IV. MPCA-BASED TENSOR OBJECT RECOGNITION

The projection matrices $\{\tilde{U}^{(n)}, n = 1, \ldots, N\}$, obtained by maximizing the total scatter $\Psi_{\mathcal{Y}}$ of the projected samples (in a reduced tensor space) from a set of training tensor samples $\{\mathcal{X}_m, m = 1, \ldots, M\}$, can be used to extract features for various applications such as data compression, object retrieval, and pattern recognition. This section presents the MPCA-based tensor object recognition framework depicted in Fig. 5. In typical pattern recognition problems, such as human identification using fingerprints, face, or gait signals, there are two types of data sets: the gallery and the probe [7]. The gallery set contains the set of data samples with known identities and it is used for training. The probe set is the testing set where data samples of unknown identity are to be identified and classified via matching with corresponding entries in the gallery set.

A. Preprocessing

MPCA accepts tensor samples of the same dimensions $I_1 \times I_2 \times \cdots \times I_N$ for feature extraction. However, in practice, tensor object samples are often of different dimensions. Therefore, the input tensors need to be normalized to standard dimensions first, and if they are noisy, a noise-removing preprocessing procedure could follow. The normalized tensor samples are then centered by subtracting the mean obtained from the gallery tensors.

B. Feature Extraction

From the gallery set, a set of eigentensors is obtained, with reduced dimensionality $(P_n \leq I_n)$ determined by a user-specified Q, and each entry of a projected tensor feature can be viewed as a (scalar) feature corresponding to a particular eigentensor. Some of the small variation and noise are removed in the projection. For recognition, it should be noted that MPCA is an unsupervised technique without taking class labels into account. As a result, the variation captured in the projected tensor subspace includes both the within-class variation and the between-class variation. In the task of classification, a larger between-class

---

[1]As an iterative solution, MPCA has a higher input/output (I/O) cost than a noniterative solution. Nevertheless, since solving for the projection matrices using MPCA is only in the training phase of the targeted recognition tasks, it can be done offline and the additional I/O (and computational) cost due to iterations is not considered a disadvantage of the proposed MPCA solution.

Distance	L1	L2	Angle	MMD
$d(\mathbf{a}, \mathbf{b})$	$\sum_{h=1}^{H} \lvert \mathbf{a}(h) - \mathbf{b}(h) \rvert$	$\sqrt{\sum_{h=1}^{H} \left[\mathbf{a}(h) - \mathbf{b}(h) \right]^2}$	$-\dfrac{\sum_{h=1}^{H} \mathbf{a}(h)\cdot\mathbf{b}(h)}{\sqrt{\sum_{h=1}^{H}\mathbf{a}(h)^2 \sum_{h=1}^{H}\mathbf{b}(h)^2}}$	$-\sum_{h=1}^{H} \dfrac{\mathbf{a}(h)\cdot\mathbf{b}(h)}{\mathbf{g}(h)}$
Distance	ML1	ML2	MAD	
$d(\mathbf{a}, \mathbf{b})$	$\sum_{h=1}^{H} \dfrac{\lvert \mathbf{a}(h) - \mathbf{b}(h) \rvert}{\mathbf{g}(h)}$	$\sqrt{\sum_{h=1}^{H} \dfrac{\left[\mathbf{a}(h) - \mathbf{b}(h)\right]^2}{\mathbf{g}(h)}}$	$-\dfrac{\sum_{h=1}^{H} \mathbf{a}(h)\cdot\mathbf{b}(h)}{\mathbf{g}(h)\sqrt{\sum_{h=1}^{H}\mathbf{a}(h)^2 \sum_{h=1}^{H}\mathbf{b}(h)^2}}$	

variation relative to the within-class variation indicates good class separability, while a smaller between-class variation relative to the within-class variation indicates poor class separability. Hence, a feature selection strategy is proposed to select eigentensors according to their class discrimination power [30], [36], [37], which is defined to be the ratio of the between-class scatter over the within-class scatter.

Definition 2: The class discriminability $\Gamma_{p_1 p_2 \cdots p_N}$ for an eigentensor $\tilde{\mathcal{U}}_{p_1 p_2 \cdots p_N}$ is defined as

$$\Gamma_{p_1 p_2 \cdots p_N}$$
$$= \frac{\sum_{c=1}^{C} N_c \cdot [\bar{\mathcal{Y}}_c(p_1, p_2, \ldots, p_N) - \bar{\mathcal{Y}}(p_1, p_2, \ldots, p_N)]^2}{\sum_{m=1}^{M} [\mathcal{Y}_m(p_1, p_2, \ldots, p_N) - \bar{\mathcal{Y}}_{c_m}(p_1, p_2, \ldots, p_N)]^2}$$
$$(12)$$

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 mth gallery sample $\mathcal{X}_m$. $\mathcal{Y}_m$ is the feature tensor of $\mathcal{X}_m$ in the projected tensor subspace. The mean feature tensor $\bar{\mathcal{Y}} = (1/M) \sum_m \mathcal{Y}_m$ and the class mean feature tensor $\bar{\mathcal{Y}}_c = (1/N_c) \sum_{m, c_m=c} \mathcal{Y}_m$.

For the eigentensor selection, the entries in the projected tensor $\mathcal{Y}_m$ (from the gallery set) are rearranged into a feature vector $\mathbf{y}_m$, ordered according to $\Gamma_{p_1 p_2 \cdots p_N}$ in descending order, and only the first $H_{\mathbf{y}}$ most discriminative components of $\mathbf{y}_m$ are kept for classification, with $H_{\mathbf{y}}$ determined empirically or user specified. By this selection, a more discriminating subspace, compared to the MPCA projected tensor subspace that includes both features with good and poor separability, is resulted. Next, a weight tensor $\mathcal{W}$ is formed with entries defined as $\mathcal{W}(p_1, p_2, \ldots, p_N) = \sqrt{\Pi_{n=1}^{N} \lambda_{p_n}^{(n)}}$, where $\lambda_{p_n}^{(n)}$ denotes the p_nth n-mode eigenvalue corresponding to the projection matrix $\tilde{\mathbf{U}}^{(n)}$. $\mathcal{W}$ is rearranged into a vector $\mathbf{w}$ in the same order as $\mathbf{y}_m$ and only the first $H_{\mathbf{y}}$ components of $\mathbf{w}$ will be used in Section IV-C as weights in measuring distances.

The feature vector $\mathbf{y}_m$ can be used directly for recognition, and a classical linear discriminant analysis (LDA) can also be applied to obtain an MPCA+LDA approach for recognition [36], similar to the popular approach of PCA+LDA. LDA seeks a projection $\mathbf{V}$ to maximize the ratio of the between-class scatter matrix $\mathbf{S}_B$ to the within-class scatter matrix $\mathbf{S}_W$, where $\mathbf{S}_W = \sum_{m=1}^{M} (\mathbf{y}_m - \bar{\mathbf{y}}_{c_m})(\mathbf{y}_m - \bar{\mathbf{y}}_{c_m})^T$, $\bar{\mathbf{y}}_c = (1/N_c) \sum_{m, c_m=c} \mathbf{y}_m$, $\mathbf{S}_B = \sum_{c=1}^{C} N_c (\bar{\mathbf{y}}_c - \bar{\mathbf{y}})(\bar{\mathbf{y}}_c - \bar{\mathbf{y}})^T$, and $\bar{\mathbf{y}} = (1/M) \sum_m \mathbf{y}_m$. The solution $\mathbf{V}_{\text{lda}} = \arg\max_{\mathbf{V}}(|\mathbf{V}^T \mathbf{S}_B \mathbf{V}|/|\mathbf{V}^T \mathbf{S}_W \mathbf{V}|) = [\mathbf{v}_1 \ \mathbf{v}_2 \ \cdots \ \mathbf{v}_{H_{\mathbf{z}}}]$, where $\{\mathbf{v}_{h_{\mathbf{z}}}, h_{\mathbf{z}} = 1, \ldots, H_{\mathbf{z}}\}$ is the set of generalized eigenvectors of $\mathbf{S}_B$ and $\mathbf{S}_W$ corresponding to the $H_{\mathbf{z}} (\leq C - 1)$ largest generalized eigenvalues $\{\lambda_{h_{\mathbf{z}}}, h_{\mathbf{z}} = 1, \ldots, H_{\mathbf{z}}\}$ [38]: $\mathbf{S}_B \mathbf{v}_{h_{\mathbf{z}}} = \lambda_{h_{\mathbf{z}}} \mathbf{S}_W \mathbf{v}_{h_{\mathbf{z}}}, h_{\mathbf{z}} = 1, \ldots, H_{\mathbf{z}}$. Thus, the discriminant feature vector $\mathbf{z}_m$ is obtained as $\mathbf{z}_m = \mathbf{V}_{\text{lda}}^T \mathbf{y}_m$.

C. Feature Classification

In classification, the distance between feature vectors is of paramount importance as it determines the performance of the classification module. For the feature vectors discussed previously ($\mathbf{y}_m$ or $\mathbf{z}_m$), seven distance measures are adapted from [39] and tested: L1, L2, angle, modified Mahalanobis distance (MMD), modified L1 (ML1), modified L2 (ML2), and modified angle distance (MAD), as listed in Table II, where $\mathbf{g}$ is a weight vector. The first four measures are commonly used for measuring vector distances and the last three measures can be viewed as the weighted versions of the first three measures. If $\mathbf{z}_m$ is the feature vector, $H = H_{\mathbf{z}}$ and $\mathbf{g}(h) = \sqrt{\lambda_h}$, which is the simplified Mahalanobis distance introduced in [39]. If $\mathbf{y}_m$ is the feature vector, $H = H_{\mathbf{y}}$ and $\mathbf{g} = \mathbf{w}$, where $\mathbf{w}$ was defined previously.

To measure the similarity of one test sample feature $\mathbf{y}$ (or $\mathbf{z}$) against N_c sample features $\mathbf{y}_{n_c}$ (or $\mathbf{z}_{n_c}$) of a class c, the principle of nearest neighbor classifier is applied in this work. The matching score of $\mathbf{y}$ (or $\mathbf{z}$) with class c is obtained as $S(\mathbf{y}, c) = -\min_{n_c} d(\mathbf{y}, \mathbf{y}_{n_c})$ (or $S(\mathbf{z}, c) = -\min_{n_c} d(\mathbf{z}, \mathbf{z}_{n_c})$), using one of the distance measures in Table II. Such a simple classifier is selected to study the performance mainly contributed by the MPCA-based feature extraction algorithm although better classifiers can be investigated.

V. EXPERIMENTAL RESULTS

This section summarizes a number of experiments performed here in support of the following objectives: 1) investigate the various properties of the MPCA algorithm on synthetically generated as well as real application data sets and 2) illustrate the efficacy of MPCA in tensor object classification, by applying MPCA to the very important problem of gait recognition and comparing its performance against state-of-the-art gait recognition algorithms as well as the baseline algorithm.

A. Experimentation Using Synthetic Data

Three synthetic data sets are constructed in order to study the behavior of the MPCA algorithm and, in particular, the following are discussed: 1) the effects of the initial conditions, 2) the effects of the ordering of the projection matrix computation, 3) the evolution of the total scatter $\Psi_{\mathcal{Y}}$ over iterations, 4) the number of iterations needed for convergence, and 5) the performance of the tensor subspace dimensionality determination proposal.
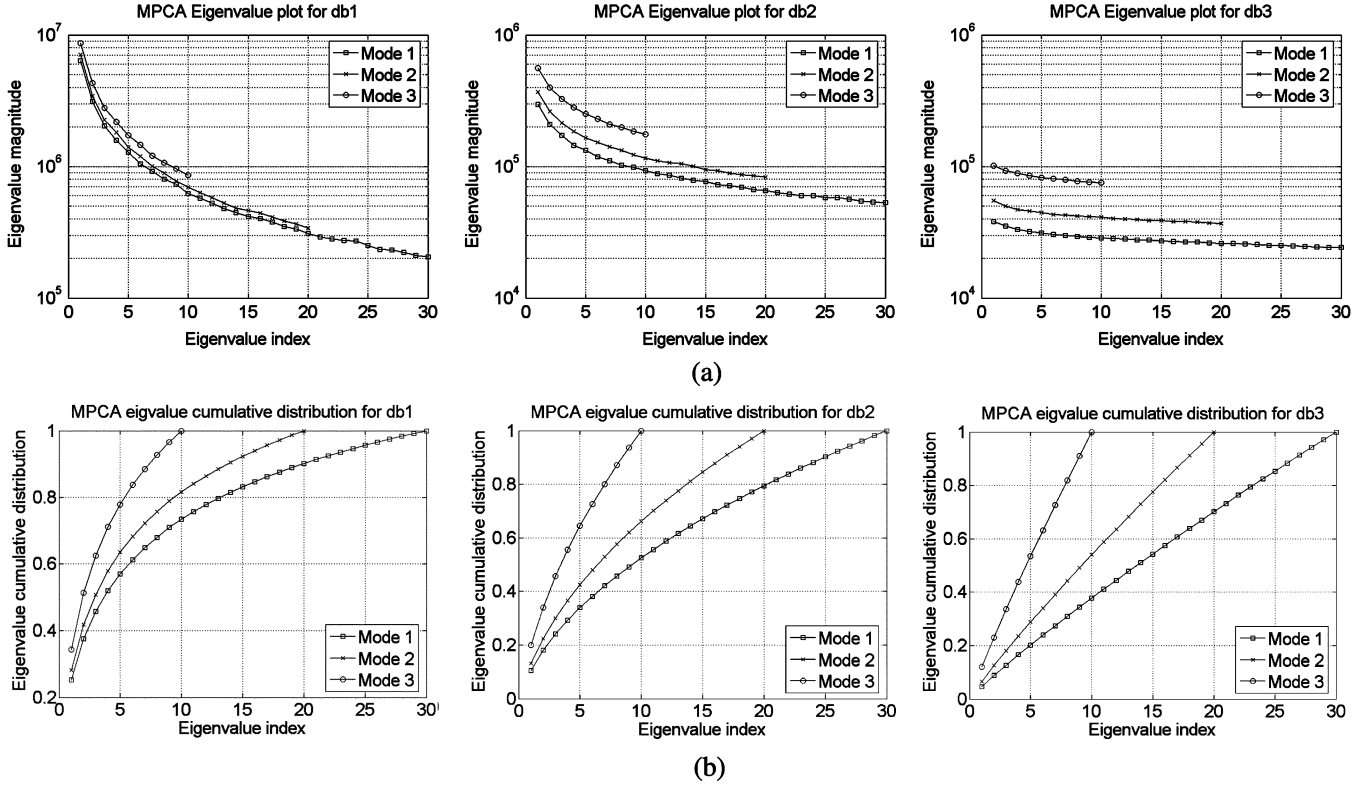
Fig. 6. Eigenvalue magnitudes and their cumulative distributions for the synthetic data sets db1, db2, and db3. (a) Distribution of eigenvalues. (b) Cumulative distribution of eigenvalues.

1) Synthetic Data Generation: The core of the MPCA algorithm is the eigendecomposition in each mode so the distribution of the eigenvalues is expected to impact significantly the performance of the algorithm. To study the MPCA properties on data of different characteristics, three synthetic data sets with eigenvalues in each mode spanning different magnitude ranges are generated. In particular, M third-order tensor samples $\mathcal{A}_m \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ are generated per set according to $\mathcal{A}_m = \mathcal{B}_m \times_1 \mathbf{C}^{(1)} \times_2 \mathbf{C}^{(2)} \times_3 \mathbf{C}^{(3)} + \mathcal{D}_m$, using a core tensor $\mathcal{B}_m \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, n-mode projection matrix $\mathbf{C}^{(n)} \in \mathbb{R}^{I_n \times I_n} (n = 1, 2, 3)$ and a "noise" tensor $\mathcal{D}_m \in \mathbb{R}^{I_1 \times I_2 \times I_3}$. All entries in $\mathcal{B}_m$ are drawn from a zero-mean unit-variance Gaussian distribution and are multiplied by $[(I_1 \cdot I_2 \cdot I_3)/(i_1 \cdot i_2 \cdot i_3)]^f$. In this data generation procedure, f controls the eigenvalue distributions, so that data sets are created having eigenvalues' magnitudes in different ranges. Smaller f results in a narrower range of eigenvalue spread. The matrices $\mathbf{C}^{(n)}(n = 1, 2, 3)$ are orthogonal matrices obtained by applying SVD on random matrices with entries drawn from zero-mean, unit-variance Gaussian distribution. All entries of $\mathcal{D}_m$ are drawn from a zero-mean Gaussian distribution with variance 0.01. Three synthetic data sets db1, db2, and db3 of size $30 \times 20 \times 10$ with $M = 100$ and $f = 1/2, 1/4,$ and $1/16$, respectively, are created. Fig. 6(a) depicts the spread of eigenvalue magnitudes and Fig. 6(b) depicts their eigenvalue cumulative distribution.

2) MPCA Properties: First, the effects of the initial conditions are tested using the synthetic data sets. Both random matrices and pseudoidentity matrices (truncated identity matrices) have been utilized. Typical examples are shown in Fig. 7.

From the simulation studies, it can be observed that despite the different initializations, the algorithm, when applied on sets db1 and db2, converges to the same point within three iterations. On set db3, the algorithm with different initializations converges to the same point within ten iterations for $Q \geq 0.35$. For small value of $Q(<0.35)$ on set db3, the algorithm using random matrices as initialization could converge to a point that is different from (lower than) the point to which the algorithm using the other two initialization methods converges, as shown in Fig. 7(c), indicating that initialization methods could affect the final results on data sets with similar characteristics as db3 when a small Q is used. In summary, initialization has little effect on the final results for synthetic data sets db1 and db2 with all values of Q, and for synthetic data set db3 with $Q \geq 0.35$. In pattern recognition applications, it is often desired to keep most of the variation/energy in the original data, and hence, the proposed algorithm using different initializations is expected to converge well since $Q > 0.35$ is easily satisfied in practice. Since the algorithm using the proposed initialization FPT converges faster than the algorithm using the other initialization methods, the FPT is expected to be closer to the local maximum point and it is used for initialization in this work.

Second, the effects of the ordering of the projection matrix computation are studied through simulations on the synthetic data sets. Simulation results indicate that there is no significant difference in the captured total scatter for db1 and db2, while there is some small difference for db3. For db3, the difference in total scatter captured using different orderings is negligible ($< 0.01\%$) for $Q > 0.5$ and it increases as Q decreases,
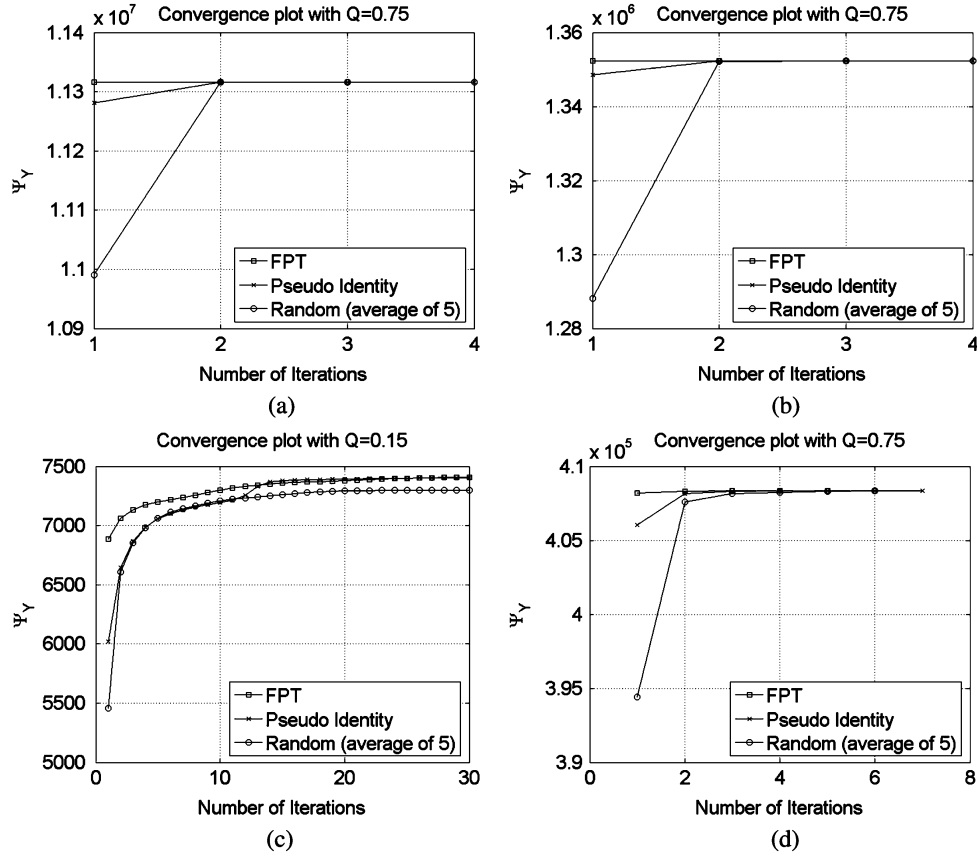
Fig. 7. Convergence plot for different initializations. (a) Convergence plot for db1 with $Q = 0.75$. (b) Convergence plot for db2 with $Q = 0.75$. (c) Convergence plot for db3 with $Q = 0.15$. (d) Convergence plot for db3 with $Q = 0.75$.
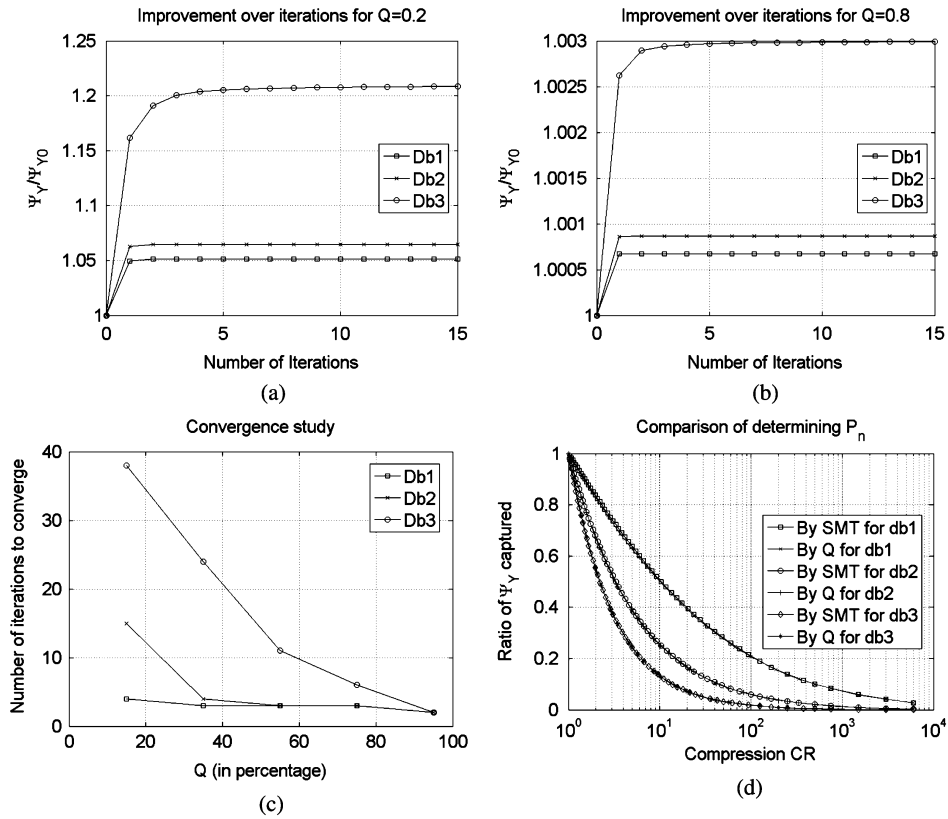

Fig. 8. Illustration of various properties of MPCA on the synthetic data sets. (a) Evolution of $\Psi_{\mathcal{Y}}$ for $Q = 0.2$. (b) Evolution of $\Psi_{\mathcal{Y}}$ for $Q = 0.8$. (c) Number of iterations to converge. (d) SMT versus Q-based selection of P_n.

e.g., the difference is about 1% when $Q = 0.2$. This observation is consistent with the poorer convergence property of db3, especially for a small Q, in other experiments such as the previous initialization study and the convergence study in the following.

To study the evolution of the total scatter $\Psi_{\mathcal{Y}}$ over iterations, the ratio of the value of $\Psi_{\mathcal{Y}_k}$ over the initial value $\Psi_{\mathcal{Y}_0}$ is plotted against the number of iterations, as a function of dimensionality reduction determined by Q. For illustration purpose, results obtained for up to 15 iterations with $Q = 0.2$ and $Q = 0.8$ are shown in Fig. 8(a) and (b), respectively. As it can be seen from the figure, the first iteration results in the greatest increase in $\Psi_{\mathcal{Y}}$ while subsequent iterations result in smaller and smaller increments, especially for data sets db1 and db2. To study the empirical convergence performance, the number of iterations for convergence using a termination value of $\eta = 10^{-6}$ is plotted in Fig. 8(c) as a function of the parameter Q. These figures demonstrate that, in MPCA, the number of iterations needed to converge decreases as the range spanned by the eigenvalues for the data samples or the value of Q increases.

The dependency on Q can be explained from two aspects. Q is closely related to the number of eigenvectors truncated. First, from Theorem 3, the bounds on $\Psi_{\mathcal{Y}_0}$ tend to become looser when the number of eigenvectors truncated increases (Q decreases), and vice versa. Looser (tighter) bounds tend to result in a poorer (better) initialization and it takes more (less) iterations to reach a local optimum. Second, by Theorem 2, more truncation (smaller value of Q) tends to decrease the eigenvalues in the other modes more so that more iterations are needed to converge, and vice versa. The dependency on the range of eigenvalue spread is less obvious. Narrower range in a mode means the variation along each eigenvector in this mode is similar and each of the truncated eigenvectors tends to encode similar amount of variation as each remaining one, which tends to make the convergence harder. On the other hand, broader range of the eigenvalue spread in a mode means that the energy is more concentrated in this mode and, in this case, convergence is expected to be easier.

In practical recognition tasks, Q is commonly set to a large value in order to capture most of the variations. Furthermore, the eigenvalues of practical data samples usually spread a wide range in each mode due to redundancy/correlation. Therefore, in practice, the number of iterations can be set to a small value such as 1 with little sacrifice in the variations captured while enjoying significant gain in empirical efficiency. In the following experiments, there is only one iteration ($K = 1$) in MPCA and, in this case, the calculation of $\Psi_{\mathcal{Y}_k}$ and $\tilde{\mathcal{Y}}_m$ is not needed in the local optimization (Step 3) of Fig. 3.

To investigate the Q-based method and the SMT method for subspace dimensionality determination, P_n is determined for each mode n using these two methods with various degrees of truncation across all the modes. The resulted total scatters are compared under the same CR (the same reduced dimensionality), as shown in Fig. 8(d). The figure indicates that these two methods have very similar results for all the three data sets. Hence, in practice, the Q-based method, which is more efficient, is used to determine the tensor subspace dimensionality. In addition, it is also verified through experiments, which are not in-

cluded here due to space limitation, that truncation in all modes is advantageous against truncation in only one or two modes.

B. MPCA-Based Gait Recognition

The emerging application of video-based gait recognition is of particular interest in the context of surveillance-at-a-distance applications [3]. Gait video is naturally a third-order tensor with the column, row, and time mode. Most of the gait recognition algorithms proposed in the literature extract features using frame-by-frame processing and they did not fully utilize the natural spatial and temporal correlations of the input data. Unfortunately, the linear subspace algorithms that have been successfully applied, adapted, or developed in other biometrics applications, such as faces [16], are not directly applicable to the gait recognition problem because the vectorization of gait video input results in vectors with much higher dimensions compared to that of vectorized face images, which are impossible to process. For example, each gait silhouette sequence from the standard University of South Florida (USF) data sets has a frame size of 128×88 and each sequence contains approximately 180 frames. The dimensionality of such a sequence is approximately two orders of magnitude higher to that of typical image dimensionality encountered in the National Institute of Standards and Technology (NIST) facial recognition technology (FERET) database [40]. By treating a number of frames as a single tensor object from which features are to be extracted, both the spatial and the temporal correlations of the original information are fully preserved, and thus, a cost-effective feature extraction procedure can be obtained under the framework of the MPCA-based tensor object recognition in Section IV. With each gait half cycle treated as a tensor sample, the proposed gait recognition solution introduces a novel representation called EigenTensorGait (ETG) and it can be extended readily to other recognition problems as well. A gait recognition algorithm typically consists of the following steps: algorithm to partition a gait sequence into cycles (or half cycles), feature representation and extraction method, and the matching algorithm. In addition, preprocessing may be applied to further improve the quality of silhouette images for better recognition results, which is not included in this paper.

1) Gait Challenge Data Sets: The experiments are carried out on the USF HumanID "gait challenge" data sets version 1.7 [7]. This database consists of 452 sequences from 74 subjects (persons) walking in elliptical paths in front of the camera. For each subject, there are three covariates: viewpoint (left/right), shoe type (two different types), and surface type (grass/concrete). The gallery set contains 71 sequences (subjects) and seven experiments (probe sets) are designed for human identification as shown in Table III. The capturing condition for each probe set is summarized in brackets after the probe name in Table III, where C, G, A, B, L, and R stand for cement surface, grass surface, shoe type A, shoe type B, left view, and right view, respectively. The capturing condition of the gallery set is grass surface, shoe type A, and right view (GAR). Subjects are unique in the gallery and each probe set and there

TABLE III
SEVEN EXPERIMENTS DESIGNED FOR HUMAN IDENTIFICATION IN USF HUMANID DATA SETS

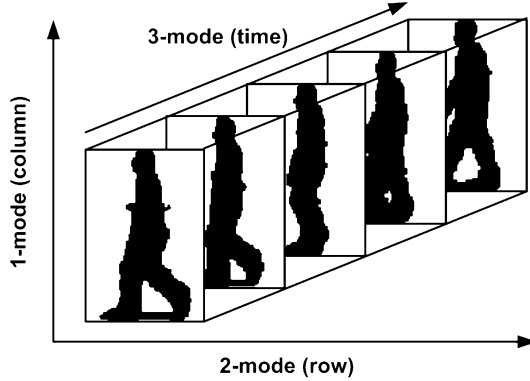| Probe set	A (GAL)	B(GBR)	C(GBL)	D(CAR)	E(CBR)	F(CAL)	G(CBL)
Size	71	41	41	70	44	70	44
Difference	View	Shoe	Shoe, view	Surface	Surface, shoe	Surface, view	Surface, shoe, view

Fig. 9.   Illustration of the gait silhouette sequence as a third-order tensor.

are no common sequences between the gallery set and any of the probe sets. Also, all the probe sets are distinct. A baseline algorithm is proposed in [7] to extract human silhouettes and recognize individuals in this database. The extracted silhouettes data provided by them is widely used in the literature. For this reason, it is considered as inputs to the experiments reported here in order to facilitate a fair comparison among the gait recognition algorithms considered here.

2) ETG Representation: While in many recognition problems, an input sample is unambiguously defined, such as iris, face, or fingerprint images, there is no obvious definition of a gait sample. The proposed method treats each half gait cycle as a data sample. Thus, the input is a third-order tensor and the spatial column space, row space, and the time space account for its three modes, as shown in Fig. 9.

To obtain half cycles, a gait silhouette sequence is partitioned in a way similar to that used in [7]. The number of foreground pixels is counted in the bottom half of each silhouette since legs are the major visible moving body parts from a distance. This number will reach a maximum when the two legs are farthest apart and drop to a minimum when the legs overlap. The sequence of numbers is smoothed with a running average filter and the minimums in this number sequence partition the sequence into several half gait cycles. Following the previous definition, there are 731 gait samples in the gallery set and each subject has an average of roughly ten samples available. This simple partition method may be improved further by taking the periodic nature of the gait cycles into account, such as the more robust cycle partitioning algorithm in [6], which is not investigated in this paper.

Each frame of the gait silhouette sequences from the USF data sets is of standard size 128×88, but the number of frames in each gait sample obtained through half cycle partition has some variation. Before feeding the gait samples to MPCA, the tensorial inputs need to be normalized to the same dimension in each mode. Since the row and column dimensions are normalized by default, only the time mode, i.e., the number of frames in each gait sample, is subject to normalization. The normalized time-mode dimension is chosen to be 20, which is roughly the average number of frames in each gait sample. Thus, each gait sample has a canonical representation of $I_1 \times I_2 \times I_3 = 128 \times 88 \times 20$, and feature vectors are extracted according to the method described in Section IV for the gait samples $\{\mathcal{X}_m, m = 1, \ldots, M\}$. In the following, a simple procedure for this time-mode normalization is described.

Consider one sample $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times D_3}$. While there are sophisticated algorithms available, such as mapping a gait cycle to a unit circle using nonlinear interpolation [26], conventional interpolation algorithms, such as linear interpolation, can be applied to the time-mode normalization as well. In this paper, each 3-mode (time-mode) vector is interpolated linearly to the normal size $I_3 \times 1$, followed by binarization to get binary silhouettes.

The 1-mode unfolding, i.e., concatenating the frames of a sequence sequentially in a row, of a gait sample and the mean gait sample, are shown in Fig. 10(a) and (b), respectively. Fig. 11 shows a plot of the eigenvalues in three modes and their cumulative distributions. The 2-mode eigenvalues drop sharply after the 20th eigenvalue, indicating high redundancy in the 2-mode (row mode). The 1-mode (column mode) eigenvalues decrease gradually except at the last point, which may be due to the centering.

The eigentensors obtained from the gallery gait sequences through MPCA are named ETGs. This approach to gait analysis is called the ETG approach and if LDA is subsequently applied, the approach is called the ETG+LDA (ETGLDA) approach. Fig. 10(c) shows the 1-mode unfolding of the first, second, third, 45th, and 92th most discriminative ETGs for illustration. From Fig. 10(c), it is observed that the ETGs act as a set of multiresolution filters, where each ETG can be viewed as a filter, and the projection using them is very similar to a filtering process applying a filter bank in multiresolution analysis. Using the proposed ETG representation, gait features are extracted from gait sequences for recognition.

3) Gait Feature Matching for Recognition: In the ETG approach, to obtain the matching score of a probe sequence p with N_p samples against a gallery sequence g with N_g samples, the approach in [4] is adopted, which proposed that the distance calculation process should be symmetric with respect to probe and gallery sequences. If the probe and gallery sequences were interchanged, the computed distance would be identical. The details are described as follows: Each probe sample feature $\mathbf{y}_{n_p}$ is matched against
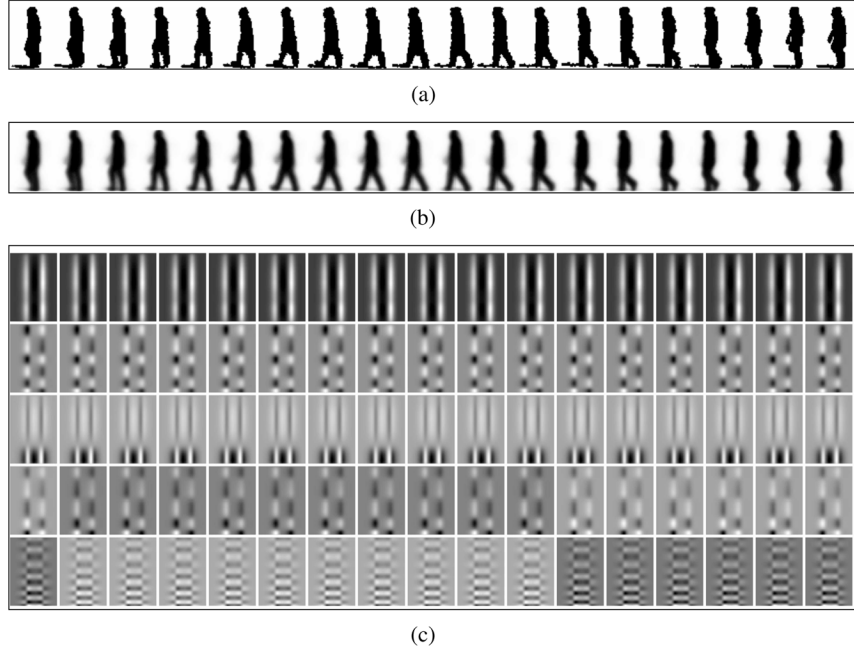
(a)

(b)

(c)

Fig. 10.   Illustration of a gait sample, the mean gait sample, and a number of representative ETGs. (a) The 1-mode unfolding of a gait silhouette sample. (b) The 1-mode unfolding of the mean of the gait silhouette samples. (c) The first, second, third, 45th, and 92nd discriminative ETGs.
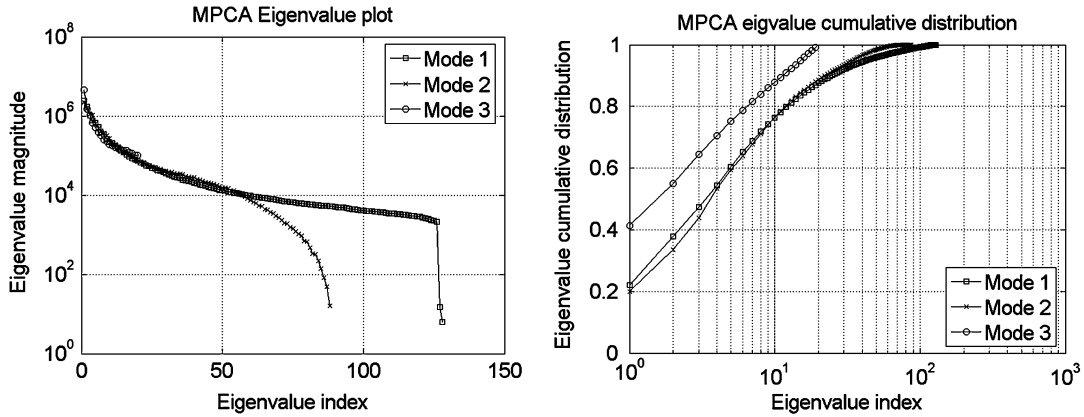

Fig. 11.   Eigenvalue magnitudes and their cumulative distributions for the gallery set.

the gallery sequence to obtain $S(\mathbf{y}_{n_p}, g)$ and each gallery sample feature $\mathbf{y}_{n_g}$ is matched against the probe sequence to obtain $S(\mathbf{y}_{n_g}, p)$. The matching score between the probe sequence p and the gallery sequence g is the sum of the mean matching score of p against g and that of g against p: $S(p, g) = (1/N_p) \sum_{n_p=1}^{N_p} S(\mathbf{y}_{n_p}, g) + (1/N_g) \sum_{n_g=1}^{N_g} S(\mathbf{y}_{n_g}, p)$. The identity of the gallery sequence with the highest matching score $S(p, g)$ is assigned to the probe sequence p.

Feature matching in the ETGLDA approach is similar to that in the ETG approach, with $\mathbf{y}$ replaced by $\mathbf{z}$ in the previous description. The identification performance is measured by the cumulative match characteristic (CMC) [7], which plots identification rates P_I within a given rank k (rank k results report the percentage of probe subjects whose true match in the gallery set was in the top k matches).

4) Effects of Centering on Recognition: To emphasize the fact that centering is an important preprocessing in classifica-

tion/recognition applications, a simple example is shown here. The proposed ETG approach is applied with and without centering of the input gait data. In both cases, the dimensionality in each mode is reduced to half of the original: $P_1 = 64, P_2 = 44, P_3 = 10$. The rank 1 and rank 5 gait recognition results using the angle distance measure and the MAD measure are shown in Fig. 12, with the number of ETGs used $(H_{\mathbf{y}})$ ranging from 20 to 680. Results on centered data are significantly better than the results on the data without centering, implying that variation captured with respect to the data center is more powerful in discriminating classes than variation captured with respect to the origin.

5) Algorithm Comparison: As this paper focuses on feature extraction, different feature extraction methods need to be compared first, with the other algorithmic components fixed, as shown in Table IV. The following features used in the state-of-the-art gait recognition algorithms are compared: silhouettes,
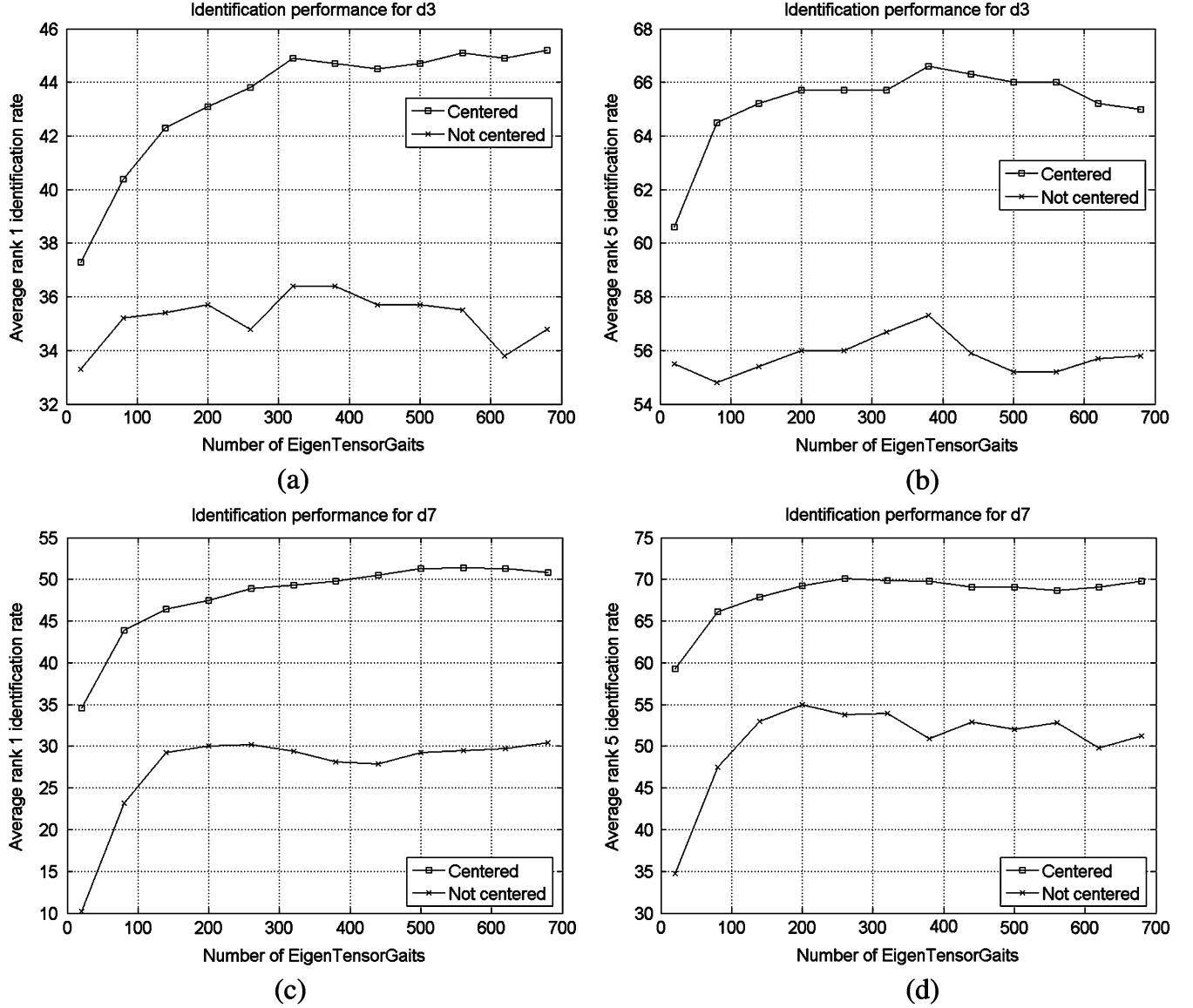
Fig. 12. Gait recognition results with and without centering. (a) Rank 1 results with angle distance. (b) Rank 5 results with angle distance. (c) Rank 1 results with MAD distance. (d) Rank 5 results with MAD distance.

TABLE IV
COMPARISON OF FEATURE EXTRACTION METHODS FOR GAIT RECOGNITION

Approach	Preprocessing	Cycle partition	**Feature extraction**	Matching
Baseline [7]	Baseline (half	Running		L1, L2,
HMM [6]	image size)	average	Silhouettes	or angle
LTN [4]	and time-mode	filtering		with
GEI [5]	linear	and minima	PCA+LDA on averaged silhouette	symmetric
ETGLDA	interpolation	detection	MPCA+LDA on gait tensor sample	matching

features extracted by performing PCA+LDA on averaged silhouettes, and features from the proposed ETGLDA approach. Other heuristic features, such as widths of the outer contour [6] and angular transforms [4], have been shown to be poorer than silhouettes so they are not compared here. As shown in Table IV, starting with the USF silhouettes, half cycles are partitioned by the method described in Section V-B2 and linear interpolation is used for time-mode normalization. The frame

image size is resized to 64×44 for all the feature extraction algorithms in this comparison and the matching algorithm described in Section V-B3 is used with three distance measures: L1, L2, and angle.

Next, the proposed ETG and ETGLDA algorithms are compared against the state-of-the-art gait recognition algorithms, with different processing steps, which is expected to have an impact on the recognition results. Algorithms that underper-

TABLE V
COMPARISON OF THE STATE-OF-THE-ART GAIT RECOGNITION ALGORITHMS

| Approach	Preprocessing	Cycle partition	Feature extraction	Matching
Baseline [7]	Baseline	Period estimation from median of distances between minima	Silhouettes	Spatial-temporal correlation
HMM [6]	Baseline	Adaptive filter	Silhouettes	Viterbi algorithm
LTN [4]	Baseline & silhouette refinement	Autocorrelation, optimal filter, merging	Silhouettes	LTN distance with symmetric matching
GEI [5]	Baseline	Maximum entropy spectrum estimation	PCA+LDA on averaged silhouette	minimum Euclidean distance to class mean
ETG	Baseline & time-mode linear interpolation	Running average filter	EigenTensorGaits	Nearest neighbor with symmetric matching
ETGLDA			ETG+LDA	
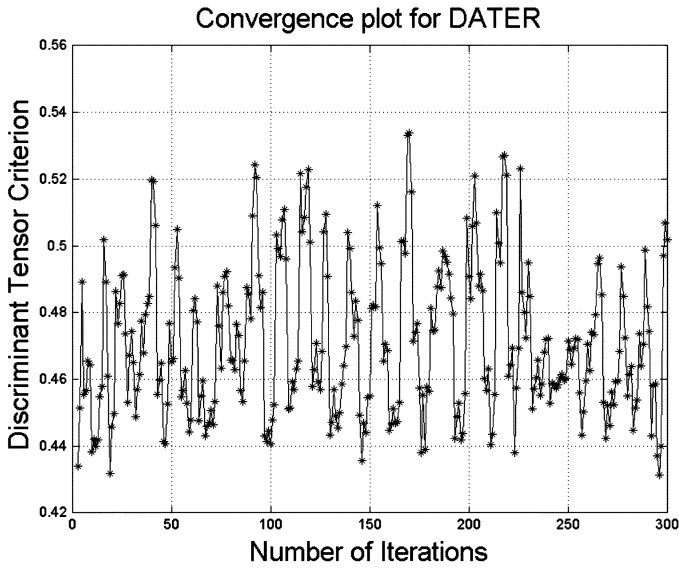
Fig. 13. Evolution of the objective criterion over iterations when the DATER algorithm in [30] is applied on the gait data.

form, i.e., with average rank 1 identification rate $(\bar{P}_I^1)$ lower than 50%, are not included, with the exception of the baseline algorithm [7]. Those included, as summarized in Table V in their original formulations, are the baseline algorithm [7], the hidden Markov model (HMM) framework [6], the linear time normalization (LTN) algorithm [4], and the gait energy image (GEI) algorithm [5]. For the HMM algorithm, the direct approach, which uses the entire silhouette as the feature, is chosen for comparison since the indirect approach, which uses the width of the outer contour of the silhouette as the feature, has $\bar{P}_I^1 = 46\%$. For the LTN algorithm, the LTN using the silhouette feature (LTN-S) is chosen for comparison since the LTN using the angular feature (LTN-A) has $\bar{P}_I^1 = 41\%$. For the GEI method, the results involving the synthetic templates are not included for fair comparison. Table V illustrates the differences between algorithms. It should be pointed out that, in the HMM approach, besides feature extraction and matching, HMM parameter estimation (training) is a major component as well and it is not shown in the table.

In addition, there is a recent work applying the multilinear LDA algorithm in [30] and [31] to the problem of gait recognition using matrix representation (the average of silhouettes) [32], which can be viewed as a second-order tensor representation where the time-mode information is ignored. This is fundamentally different from the natural third-order tensor representation used in the MPCA-based approaches. As mentioned before, the algorithm in [30] does not converge, as shown in Fig. 13, and their results are shown to be very sensitive to parameter settings, as shown in [32, Fig. 4]. Furthermore, exhaustive testing is used to determine the number of iterations and the subspace dimension in [32], which is not practical due to its high computational cost. The authors did their best to faithfully implement the gait recognition algorithm in [32] based on [30] and tested 2125 parameter configurations, evenly sampled from the total 56 320 possible parameter configurations. The best result, obtained after five iterations with projected dimensions of 30×5, has (average) rank 1 and rank 5 recognition rates of 42% and 57%, respectively, on the gait data sets used in this paper. These results are poorer than the recognition rates obtained by the baseline algorithm. Hence, the methods in [32] or [30] are not included in the state-of-the-art gait recognition set used for comparison purposes.

6) Gait Recognition Results and Discussions: In the gait recognition experiments, the number of eigenvectors kept in each mode is $P_1 = 61, P_2 = 42$, and $P_3 = 17$, and they are determined by setting $Q^{(1)} = Q^{(2)} = Q^{(3)} = Q = 0.97$, which captures approximately 92% of the total energy variation of the gallery gait samples in the projected tensor subspace. Although with the same Q, similar amount of variation is kept in each mode, the amount of dimensionality reduction in the image spatial modes (52% for the column mode and 48% for the row mode) is more significant than that in the time mode (15%). This indicates that there are more redundancy in the spatial modes than in the time mode. For the ETGLDA approach, the dimension of the feature vector $\mathbf{z}$ is fixed to be the maximum that can be obtained by classical LDA $H_\mathbf{z} = C - 1 = 70$, where C is the number of gallery subjects (classes).

Two following key design factors are varied to study their effects and to determine the best choices for them: the number of selected ETGs $H_\mathbf{y}$ and the distance measures. For different
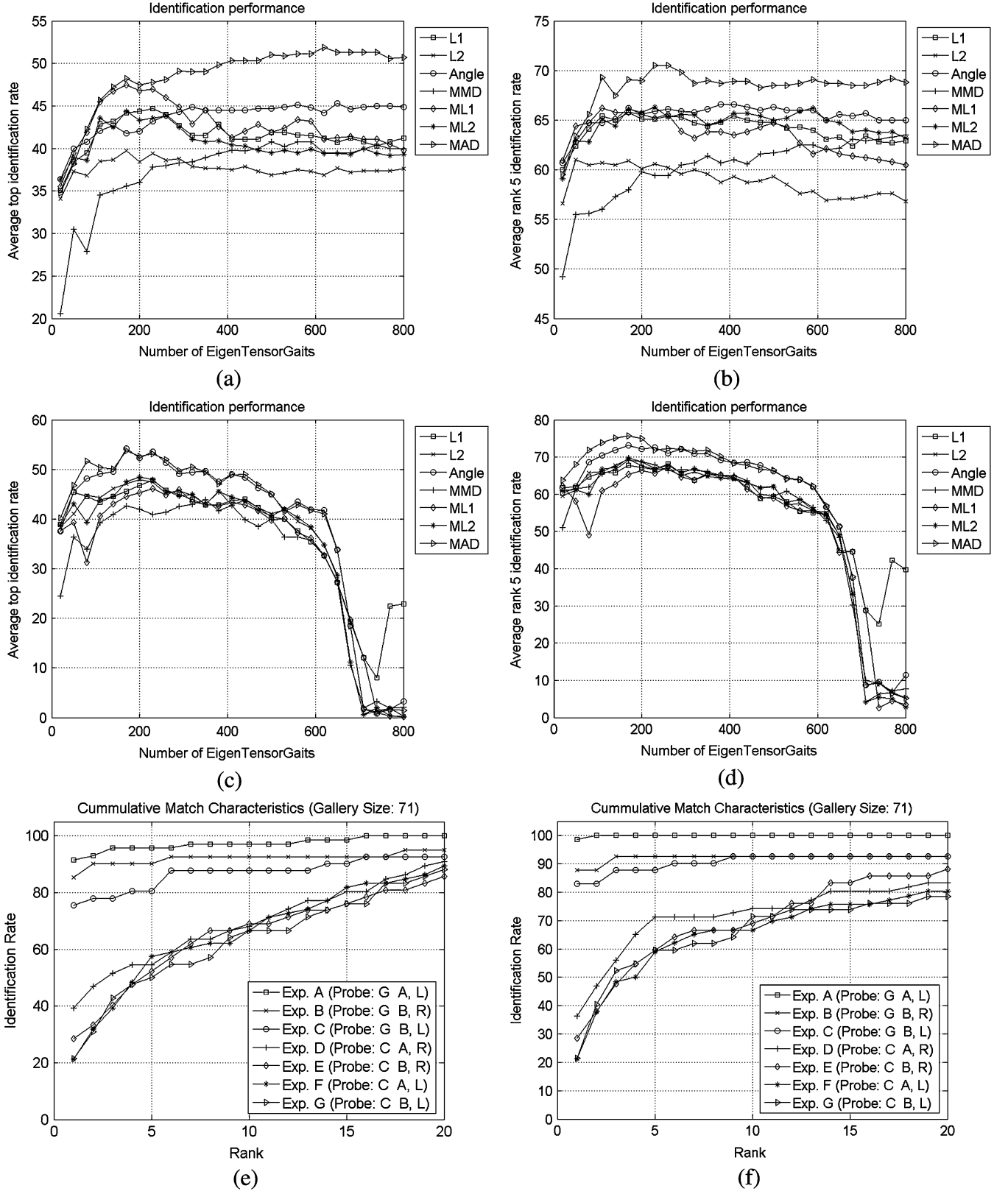
Fig. 14.   Gait recognition results: dependency on $H_{\mathbf{y}}$ and the distance measure used and the CMC curves. (a) Rank 1 identification performance of ETG. (b) Rank 5 identification performance of ETG. (c) Rank 1 identification performance of ETGLDA. (d) Rank 5 identification performance of ETGLDA. (e) CMC curves up to rank 20 for ETG. (f) CMC curves up to rank 20 for ETGLDA.

$H_{\mathbf{y}}$ (up to 800) and for the distance measures listed in Table II, the average rank 1 and rank 5 identification rates are plotted in Fig. 14(a) and (b), respectively, for the ETG approach, and in Fig. 14(c) and (d), respectively, for the ETGLDA approach. For the ETG approach, the MAD measure, with the proposed weight vector $\mathbf{w}$, significantly outperforms all the other dis-

TABLE VI
COMPARISON OF FEATURE EXTRACTION METHODS

Feature extraction method	$\bar{P}_I$ (%) at Rank 1 with			$\bar{P}_I$ (%) at Rank 5 with		
	L1	L2	Angle	L1	L2	Angle
Silhouettes	34	35	33	54	54	53
PCA+LDA on averaged silhouette	45	44	43	68	69	69
MPCA+LDA on gait tensor sample	47	47	53	67	70	73

TABLE VII
COMPARISON OF THE MPCA-BASED AND THE STATE-OF-THE-ART GAIT RECOGNITION ALGORITHMS

Probe	P_I (%) at Rank 1						P_I (%) at Rank 5					
	Baseline	HMM	LTN	GEI	ETG	ETGLDA	Baseline	HMM	LTN	GEI	ETG	ETGLDA
A (GAL)	79	99	94	**100**	92	99	96	**100**	99	**100**	96	**100**
B (GBR)	66	**89**	83	85	85	88	81	90	85	85	90	**93**
C (GBL)	56	78	78	80	76	**83**	76	**90**	83	88	81	88
D (CAR)	29	35	33	30	**39**	36	61	65	65	55	55	**71**
E (CBR)	24	29	24	**33**	29	29	55	65	**67**	55	52	60
F (CAL)	**30**	18	17	21	21	21	46	**60**	58	41	58	59
G (CBL)	10	24	21	**29**	21	21	33	50	48	48	50	**60**
Average	42	53	50	**54**	52	**54**	64	74	72	67	69	**76**

tance measures for $H_\mathbf{y} > 200$. For the ETGLDA approach, the angle and MAD measures outperform all the other measures at rank 1 and the MAD measure is better than the angle at rank 5 for $H_\mathbf{y} < 200$. Thus, both approaches choose MAD as the distance measure. With MAD, when more and more eigentensors are included, i.e., $H_\mathbf{y}$ increases, the ETG identification rate keeps increasing first, except some small fluctuations, and becomes steady beyond a certain point, indicating that most of the ETGs selected first are good features for classification. For the ETGLDA approach, due to the LDA feature extraction process where the feature vector length is $C-1$, the performance against $H_\mathbf{y}$ is different: with the MAD measure, the identification rates (rank 1 and 5) reach maximum around $H_\mathbf{y} = 200$ and drop at a higher rate for $H_\mathbf{y} > 600$, suggesting that more ETGs (more than 200) may lower the performance of ETGLDA.

Table VI compares the feature extraction algorithms in Table IV. The feature vector length is 70 ($C-1$) for both the GEI (PCA+LDA on averaged silhouette) and ETGLDA (MPCA+LDA on gait tensor sample) approaches. The PCA in the GEI approach results in a feature vector of length 142, as recommended by the authors of [5], and the ETGLDA results shown are obtained with $H_\mathbf{y} = 200$, determined empirically. The table shows that both the GEI and ETGLDA features outperform the silhouettes features significantly, and the proposed ETGLDA features outperform the GEI features. Particularly, in the average rank 1 identification rate, the ETGLDA features with the (best-performing) angle distance measure result in about 18% improvement over the GEI features with the (best-performing) L1 distance measure, indicating that the proposed MPCA-based approach is more powerful in recognition than the PCA-based approach.

Based on empirical study, the best gait recognition performance for the ETG and ETGLDA approaches is obtained with

$H_\mathbf{y} = 620$ and $H_\mathbf{y} = 170$, respectively, using MAD. The detailed results are depicted using the CMCs in Fig. 14(e) and (f), and they are compared with the state-of-the-art gait recognition algorithms in Table VII, where the rank 1 and rank 5 identification rates are listed for each probe (A to G) together with their averages. The best results for all the probe and rank combinations are highlighted by boldface font in the table. From Table VII, the HMM, LTN, GEI, ETG, and ETGLDA algorithms have no significant difference in the rank 1 performance, although LTN is slightly poorer, and they outperform the baseline results by more than 19%. For the rank 5 performance, ETGLDA has the best performance.

From the comparisons in Table V, the performance of the HMM framework is mainly contributed to the adaptive filter used for cycle partition, the Viterbi algorithm used for probabilistic matching, and the iterative training of the HMM parameters, while the performance of LTN is mainly contributed to the silhouette refinement, the robust cycle partition procedure, and the LTN distance matching strategy. Besides PCA+LDA in feature extraction, the GEI algorithm utilizes a robust estimator as well for the cycle partition taking into account the periodic nature of the gait signal and it seems that this tends to improve the recognition performance. To summarize, silhouette refinement could be a beneficial step for gait recognition and robust cycle partition seems to be an important component in these start-of-the-art gait recognition algorithms. In addition, robust matching algorithms such as the Viterbi algorithm used in HMM have great potential for gait recognition as well.

Finally, the ETGLDA approach has some small gain (about 4% improvement on average) over the ETG approach in rank 1 performance, especially on probes A and C. In rank 5 performance, the ETGLDA approach has greater improvement (about 10% on average) over the ETG approach. Thus, the combination

of the ETG approach with a classical LDA does result in better recognition and the combination with other linear subspace algorithms are promising future directions as well. On the whole, despite a design without optimizing the preprocessing, cycle partition, and matching algorithms, the MPCA-based approaches to gait recognition are highly competitive against the state-of-the-art gait recognition algorithms. The ETGs, and EigenTensorObjects in general, are very promising features for tensor object recognition.

VI. CONCLUSION

In this paper, a new MPCA framework is proposed for analysis of tensor objects, such as 2-D/3-D images and video sequences. MPCA determines a multilinear projection onto a tensor subspace of lower dimensionality that captures most of the signal variation present in the original tensorial representation. An iterative solution was formulated. Issues including initialization, convergence, and subspace dimensionality determination were discussed in detail. An MPCA-based tensor object recognition framework was then introduced and applied to the problem of gait recognition. Each half cycle of a gait silhouette sequence was treated as a gait sample and the proposed framework was used to devise a novel gait representation called ETG. Comparisons with the start-of-the-art algorithms indicated that the MPCA-based approach is a very promising tool for gait recognition and its performance can be further improved by silhouette refinement, robust cycle partition, and more advanced matching algorithms.

APPENDIX I
PROOFS

A. Proof of Theorem 1

Proof: The mean tensor of all the projected samples is $\bar{\mathcal{Y}} = (1/M) \sum_{m=1}^{M} \mathcal{Y}_m = \bar{\mathcal{X}} \times_1 \tilde{\mathbf{U}}^{(1)^T} \times_2 \tilde{\mathbf{U}}^{(2)^T} \cdots \times_N \tilde{\mathbf{U}}^{(N)^T}, m = 1, \ldots, M$. Write the objective function (4) in terms of the input tensor samples as

$$\Psi_{\mathcal{Y}} = \sum_{m=1}^{M} \|\mathcal{Y}_m - \bar{\mathcal{Y}}\|_F^2 = \sum_{m=1}^{M} \|(\mathcal{X}_m - \bar{\mathcal{X}})$$
$$\times_1 \tilde{\mathbf{U}}^{(1)^T} \times_2 \tilde{\mathbf{U}}^{(2)^T} \cdots \times_N \tilde{\mathbf{U}}^{(N)^T}\|_F^2. \quad (13)$$

From the definition of the Frobenius norm for a tensor and that for a matrix, $\|\mathcal{A}\|_F = \|\mathbf{A}_{(n)}\|_F$, and from (2), $\Psi_{\mathcal{Y}}$ can be expressed using the equivalent matrix representation by n-mode unfolding as follows:

$$\Psi_{\mathcal{Y}} = \sum_{m=1}^{M} \left\| \mathbf{Y}_{m(n)} - \bar{\mathbf{Y}}_{(n)} \right\|_F^2$$
$$= \sum_{m=1}^{M} \left\| \tilde{\mathbf{U}}^{(n)^T} \cdot \left(\mathbf{X}_{m(n)} - \bar{\mathbf{X}}_{(n)} \right) \cdot \tilde{\mathbf{U}}_{\mathbf{\Phi}^{(n)}} \right\|_F^2. \quad (14)$$

Since $\|\mathbf{A}\|_F^2 = \mathrm{trace}(\mathbf{A}\mathbf{A}^T)$, $\Psi_{\mathcal{Y}}$ can be written in terms of the n-mode total scatter matrix of the projected tensor samples

$$\Psi_{\mathcal{Y}} = \sum_{m=1}^{M} \mathrm{trace}\left(\tilde{\mathbf{U}}^{(n)^T} \cdot \left(\mathbf{X}_{m(n)} - \bar{\mathbf{X}}_{(n)} \right) \cdot \tilde{\mathbf{U}}_{\mathbf{\Phi}^{(n)}} \right.$$
$$\left. \cdot \tilde{\mathbf{U}}_{\mathbf{\Phi}^{(n)}}^T \cdot \left(\mathbf{X}_{m(n)} - \bar{\mathbf{X}}_{(n)} \right)^T \cdot \tilde{\mathbf{U}}^{(n)} \right)$$
$$= \mathrm{trace}\left(\tilde{\mathbf{U}}^{(n)^T} \cdot \mathbf{\Phi}^{(n)} \cdot \tilde{\mathbf{U}}^{(n)} \right). \quad (15)$$

Therefore, for given $\tilde{\mathbf{U}}^{(1)}, \ldots, \tilde{\mathbf{U}}^{(n-1)}, \tilde{\mathbf{U}}^{(n+1)}, \ldots, \tilde{\mathbf{U}}^{(N)}$, the total scatter $\Psi_{\mathcal{Y}}$ is maximized if and only if $\mathrm{trace}(\tilde{\mathbf{U}}^{(n)^T} \cdot \mathbf{\Phi}^{(n)} \cdot \tilde{\mathbf{U}}^{(n)})$ is maximized. The maximum of $\mathrm{trace}(\tilde{\mathbf{U}}^{(n)^T} \cdot \mathbf{\Phi}^{(n)} \cdot \tilde{\mathbf{U}}^{(n)})$ is obtained if $\tilde{\mathbf{U}}^{(n)}$ consists of the P_n eigenvectors of the matrix $\mathbf{\Phi}^{(n)}$ corresponding to the largest P_n eigenvalues. ∎

B. Proof of Lemma 4

Lemma 4: When $P_n = I_n$ for $n = 1, \ldots, N$, $\tilde{\mathbf{U}}_{\mathbf{\Phi}^{(n)}} \cdot \tilde{\mathbf{U}}_{\mathbf{\Phi}^{(n)}}^T$ is an identity matrix.

Proof: By successive application of the transpose property of the Kronecker product $(\mathbf{A} \otimes \mathbf{B})^T = \mathbf{A}^T \otimes \mathbf{B}^T$ [38]

$$\tilde{\mathbf{U}}_{\mathbf{\Phi}^{(n)}}^T = \left(\tilde{\mathbf{U}}^{(n+1)^T} \otimes \tilde{\mathbf{U}}^{(n+2)^T} \otimes \cdots \otimes \tilde{\mathbf{U}}^{(N)^T} \right.$$
$$\left. \otimes \tilde{\mathbf{U}}^{(1)^T} \otimes \tilde{\mathbf{U}}^{(2)^T} \otimes \cdots \tilde{\mathbf{U}}^{(n-1)^T} \right). \quad (16)$$

By the Kronecker product theorem $(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = (\mathbf{A}\mathbf{C} \otimes \mathbf{B}\mathbf{D})$ [38]

$$\tilde{\mathbf{U}}_{\mathbf{\Phi}^{(n)}} \cdot \tilde{\mathbf{U}}_{\mathbf{\Phi}^{(n)}}^T = \left(\tilde{\mathbf{U}}^{(n+1)} \tilde{\mathbf{U}}^{(n+1)^T} \otimes \cdots \otimes \tilde{\mathbf{U}}^{(N)} \tilde{\mathbf{U}}^{(N)^T} \right.$$
$$\left. \otimes \tilde{\mathbf{U}}^{(1)} \tilde{\mathbf{U}}^{(1)^T} \otimes \cdots \tilde{\mathbf{U}}^{(n-1)} \tilde{\mathbf{U}}^{(n-1)^T} \right). \quad (17)$$

For all n, when $P_n = I_n$, $\tilde{\mathbf{U}}^{(n)}$ is a square matrix and $\tilde{\mathbf{U}}^{(n)^T} \tilde{\mathbf{U}}^{(n)} = \mathbf{I}_{I_n}$, where $\mathbf{I}_{I_n}$ is an $I_n \times I_n$ identity matrix. Then, $\tilde{\mathbf{U}}^{(n)^{-1}} = \tilde{\mathbf{U}}^{(n)^T}$ and $\tilde{\mathbf{U}}^{(n)} \tilde{\mathbf{U}}^{(n)^T} = \mathbf{I}_{I_n}$. Thus, $\tilde{\mathbf{U}}_{\mathbf{\Phi}^{(n)}} \cdot \tilde{\mathbf{U}}_{\mathbf{\Phi}^{(n)}}^T = \mathbf{I}_{I_1 \times I_2 \times \cdots \times I_{n-1} \times I_{n+1} \times \cdots \times I_N}$. ∎

C. Proof of Theorem 2

The following lemma explains the relationship between the eigenvalues of two covariance matrices that are closely related.

Lemma 5: Let $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2}$ be a sample matrix with I_2 samples. $\tilde{\mathbf{X}} \in \mathbb{R}^{I_1 \times P_2}$ contains $P_2 < I_2$ samples from $\mathbf{X}$, and $\mathbf{E} \in \mathbb{R}^{I_1 \times (I_2 - P_2)}$ contains the rest. Let λ_{i_1} and $\tilde{\lambda}_{i_1}$ denote the i_1th eigenvalues of $\mathbf{X}\mathbf{X}^T$ and $\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T$, respectively. Then, $\tilde{\lambda}_{i_1} \leq \lambda_{i_1}$, for $i_1 = 1, \ldots, I_1$.

Proof: Without loss of generality, let $\tilde{\mathbf{X}} = \mathbf{X}(:, 1 : P_2)$ and $\mathbf{E} = \mathbf{X}(:, (P_2 + 1) : I_2)$. $\mathbf{X}\mathbf{X}^T$ and $\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T$ are both sample covariance matrices and hence are symmetric. $\mathbf{E}$ is related to them by $\mathbf{X}\mathbf{X}^T = \tilde{\mathbf{X}}\tilde{\mathbf{X}}^T + \mathbf{E}\mathbf{E}^T$. $\mathbf{E}\mathbf{E}^T$ is a covariance matrix as well, and hence, it is positive semidefinite. From the Weyl's theorem [35], which is derived from the Courant–Fisher "min–max theorem" [35], the eigenvalues of $\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T$ is not greater than the corresponding eigenvalues of $\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T + \mathbf{E}\mathbf{E}^T = \mathbf{X}\mathbf{X}^T$, i.e., $\tilde{\lambda}_{i_1} \leq \lambda_{i_1}$, for $i_1 = 1, \ldots, I_1$. ∎

The proof of Theorem 2 follows.

Proof: For $n = 1, \check{\mathbf{X}}_{m(1)} = \check{\mathbf{U}}^{(1)} \cdot \check{\mathbf{Y}}_{m(1)} \cdot \mathbf{U}^T_{\Phi^{(1)*}}, m = 1, \ldots, M$. Thus

$$
\begin{aligned}
\check{\Phi}^{(1)} &= \sum_m \check{\mathbf{X}}_{m(1)} \mathbf{U}_{\Phi^{(1)*}} \mathbf{U}^T_{\Phi^{(1)*}} \check{\mathbf{X}}^T_{m(1)} \\
&= \sum_m \check{\mathbf{U}}^{(1)} \cdot \check{\mathbf{Y}}_{m(1)} \cdot \mathbf{U}^T_{\Phi^{(1)*}} \mathbf{U}_{\Phi^{(1)*}} \\
&\quad\quad \cdot \mathbf{U}^T_{\Phi^{(1)*}} \mathbf{U}_{\Phi^{(1)*}} \cdot \check{\mathbf{Y}}^T_{m(1)} \cdot \check{\mathbf{U}}^{(1)T} \\
&= \sum_m \check{\mathbf{U}}^{(1)} \cdot \check{\mathbf{Y}}_{m(1)} \check{\mathbf{Y}}^T_{m(1)} \cdot \check{\mathbf{U}}^{(1)T} \\
&= \check{\mathbf{U}}^{(1)} \cdot \sum_m \left(\check{\mathbf{Y}}_{m(1)} \check{\mathbf{Y}}^T_{m(1)} \right) \cdot \check{\mathbf{U}}^{(1)T}
\end{aligned}
$$

where $\mathbf{U}^T_{\Phi^{(1)*}} \mathbf{U}_{\Phi^{(1)*}}$ results in an identity matrix. Since $\check{\mathbf{Y}}_{m(1)}$ is simply the first P_1 rows of $\tilde{\mathbf{Y}}_{m(1)}, \hat{\lambda}^{(1)}_{i_1} = \lambda^{(1)*}_{i_1}$ for $i_1 = 1, \ldots, P_1$, and $\hat{\lambda}^{(1)}_{i_1} = 0$ for $i_1 = P_1 + 1, \ldots, I_1$.

Similarly, for $n \neq 1, \check{\Phi}^{(n)} = \check{\mathbf{U}}^{(n)} \cdot \sum_m (\check{\mathbf{Y}}_{m(n)} \check{\mathbf{Y}}^T_{m(n)}) \cdot \check{\mathbf{U}}^{(n)T}$. The columns of $\check{\mathbf{Y}}_{m(n)}$ are a subset of the columns of $\tilde{\mathbf{Y}}_{m(n)}$. Therefore, by Lemma 5, $\hat{\lambda}^{(n)}_{i_n} \leq \lambda^{(n)*}_{i_n}$. Since $\sum_{i_1} \hat{\lambda}^{(1)}_{i_1} < \sum_{i_1} \lambda^{(1)*}_{i_1}, \sum_{i_n} \hat{\lambda}^{(n)}_{i_n} = \sum_{i_1} \hat{\lambda}^{(1)}_{i_1} < \sum_{i_n} \lambda^{(n)*}_{i_n} = \sum_{i_1} \lambda^{(1)*}_{i_1}$. Thus, for each mode, at least for one value of $i_n, \hat{\lambda}^{(n)}_{i_n} < \lambda^{(n)*}_{i_n}$. ∎

D. Proof of Theorem 3

Proof: For the lower bound, considering the 1-mode eigenvalues $\lambda^{(1)*}_{i_1}$ first

$$
\begin{aligned}
\Psi_{\mathcal{X}} - \Psi_{\mathcal{Y}_0} &= \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} \mathcal{Y}^*_{\text{var}}(i_1, i_2, \ldots, i_N) \\
&\quad - \sum_{i_1=1}^{P_1} \sum_{i_2=1}^{P_2} \cdots \sum_{i_N=1}^{P_N} \mathcal{Y}^*_{\text{var}}(i_1, i_2, \ldots, i_N) \\
&\geq \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} \mathcal{Y}^*_{\text{var}}(i_1, i_2, \ldots, i_N) \\
&\quad - \sum_{i_1=1}^{P_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} \mathcal{Y}^*_{\text{var}}(i_1, i_2, \ldots, i_N) \\
&= \sum_{i_1=1}^{I_1} \lambda^{(1)*}_{i_1} - \sum_{i_1=1}^{P_1} \lambda^{(1)*}_{i_1} = \sum_{i_1=P_1+1}^{I_1} \lambda^{(1)*}_{i_1}
\end{aligned} \tag{18}
$$

where $\mathcal{Y}^*_{\text{var}}$ is the total scatter tensor (corresponding to the full projection) defined in (7). Inequality (18) can be similarly derived for the other n-mode eigenvalues $\lambda^{(n)*}_{i_n}, \Psi_{\mathcal{X}} - \Psi_{\mathcal{Y}_0} \geq \sum_{i_n=P_n+1}^{I_n} \lambda^{(n)*}_{i_n}$ for $n = 2, \ldots, N$. Therefore

$$
\Psi_{\mathcal{X}} - \Psi_{\mathcal{Y}_0} \geq \max_n \sum_{i_n=P_n+1}^{I_n} \lambda^{(n)*}_{i_n}. \tag{19}
$$

For the upper bound

$$
\begin{aligned}
\Psi_{\mathcal{X}} - \Psi_{\mathcal{Y}_0} &= \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} \mathcal{Y}^*_{\text{var}}(i_1, i_2, \ldots, i_N) \\
&\quad - \sum_{i_1=1}^{P_1} \sum_{i_2=1}^{P_2} \cdots \sum_{i_N=1}^{P_N} \mathcal{Y}^*_{\text{var}}(i_1, i_2, \ldots, i_N)
\end{aligned}
$$

$$
\begin{aligned}
&\leq \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} \mathcal{Y}^*_{\text{var}}(i_1, i_2, \ldots, i_N) \\
&\quad - \sum_{i_1=1}^{P_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} \mathcal{Y}^*_{\text{var}}(i_1, i_2, \ldots, i_N) \\
&\quad + \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} \mathcal{Y}^*_{\text{var}}(i_1, i_2, \ldots, i_N) \\
&\quad - \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{P_2} \cdots \sum_{i_N=1}^{I_N} \mathcal{Y}^*_{\text{var}}(i_1, i_2, \ldots, i_N) \\
&\quad + \cdots \\
&\quad + \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} \mathcal{Y}^*_{\text{var}}(i_1, i_2, \ldots, i_N) \\
&\quad - \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{P_N} \mathcal{Y}^*_{\text{var}}(i_1, i_2, \ldots, i_N) \\
&= \sum_{n=1}^{N} \sum_{i_n=P_n+1}^{I_n} \lambda^{(n)*}_{i_n}.
\end{aligned} \tag{20}
$$

∎

REFERENCES

[1] L. D. Lathauwer, B. D. Moor, and J. Vandewalle, "On the best rank-1 and rank-$(R_1, R_2, \ldots, R_N)$ approximation of higher-order tensors," *SIAM J. Matrix Anal. Appl.*, vol. 21, no. 4, pp. 1324–1342, 2000.

[2] K. N. Plataniotis and A. N. Venetsanopoulos, *Color Image Processing and Applications*. Berlin, Germany: Springer-Verlag, 2000.

[3] N. V. Boulgouris, D. Hatzinakos, and K. N. Plataniotis, "Gait recognition: A challenging signal processing technology for biometrics," *IEEE Signal Process. Mag.*, vol. 22, no. 6, pp. 78–90, Nov. 2005.

[4] N. V. Boulgouris, K. N. Plataniotis, and D. Hatzinakos, "Gait recognition using linear time normalization," *Pattern Recognit.*, vol. 39, no. 5, pp. 969–979, 2006.

[5] J. Han and B. Bhanu, "Individual recognition using gait energy image," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 2, pp. 316–322, Feb. 2006.

[6] A. Kale, A. N. Rajagopalan, A. Sunderesan, N. Cuntoor, A. Roy-Chowdhury, V. Krueger, and R. Chellappa, "Identification of humans using gait," *IEEE Trans. Image Process.*, vol. 13, no. 9, pp. 1163–1173, Sep. 2004.

[7] S. Sarkar, P. J. Phillips, Z. Liu, I. Robledo, P. Grother, and K. W. Bowyer, "The human ID gait challenge problem: Data sets, performance, and analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 2, pp. 162–177, Feb. 2005.

[8] K. W. Bowyer, K. Chang, and P. Flynn, "A survey of approaches and challenges in 3D and multi-modal 3D + 2D face recognition," *Comput. Vis. Image Understand.*, vol. 101, no. 1, pp. 1–15, Jan. 2006.

[9] S. Z. Li, C. Zhao, M. Ao, and Z. Lei, "Learning to fuse 3D+2D based face recognition at both feature and decision levels," in *Proc. IEEE Int. Workshop Anal. Model. Faces Gestures*, Oct. 2005, pp. 43–53.

[10] A. Colombo, C. Cusano, and R. Schettini, "3D face detection using curvature analysis," *Pattern Recognit.*, vol. 39, no. 3, pp. 444–455, Mar. 2006.

[11] R. Chellappa, A. Roy-Chowdhury, and S. Zhou, *Recognition of Humans and Their Activities Using Video*. San Rafael, CA: Morgan & Claypool, 2005.

[12] H. S. Sahambi and K. Khorasani, "A neural-network appearance-based 3-D object recognition using independent component analysis," *IEEE Trans. Neural Netw.*, vol. 14, no. 1, pp. 138–149, Jan. 2003.

[13] C. Nolker and H. Ritter, "Visual recognition of continuous hand postures," *IEEE Trans. Neural Netw.*, vol. 13, no. 4, pp. 983–994, Jul. 2002.

[14] R. D. Green and L. Guan, "Quantifying and recognizing human movement patterns from monocular video images—Part II: Applications to biometrics," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 2, pp. 191–198, Feb. 2004.

[15] E. Shechtman and Y. C. ad M. Irani, "Space-time super-resolution," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 4, pp. 531–545, Apr. 2005.

[16] G. Shakhnarovich and B. Moghaddam, "Face recognition in subspaces," in *Handbook of Face Recognition*, S. Z. Li and A. K. Jain, Eds. New York: Springer-Verlag, 2004, pp. 141–168.

[17] J. Zhang, S. Z. Li, and J. Wang, "Manifold learning and applications in recognition," in *Intelligent Multimedia Processing with Soft Computing*, Y. P. Tan, K. H. Yap, and L. Wang, Eds. New York: Springer-Verlag, 2004, pp. 281–300.

[18] M. H. C. Law and A. K. Jain, "Incremental nonlinear dimensionality reduction by manifold learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 3, pp. 377–391, Mar. 2006.

[19] I. T. Jolliffe, *Principal Component Analysis*, 2nd ed. New York: Springer-Verlag, 2002.

[20] J. Ye, R. Janardan, and Q. Li, "GPCA: An efficient dimension reduction scheme for image compression and retrieval," in *Proc. 10th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2004, pp. 354–363.

[21] 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.

[22] J. Ye, "Generalized low rank approximations of matrices," *Mach. Learn.*, vol. 61, no. 1–3, pp. 167–191, 2005.

[23] X. He, D. Cai, and P. Niyogi, "Tensor subspace analysis," in *Advances in Neural Information Processing Systems 18 (NIPS)*. Cambridge, MA: MIT Press, 2005.

[24] L. D. Lathauwer, B. D. Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM J. Matrix Anal. Appl.*, vol. 21, no. 4, pp. 1253–1278, 2000.

[25] L. D. Lathauwer and J. Vandewalle, "Dimensionality reduction in higher-order signal processing and rank-$(R_1, R_2, \ldots, R_N)$ reduction in multilinear algebra," *Linear Algebra Appl.*, vol. 391, pp. 31–55, Nov. 2004.

[26] C. S. Lee and A. Elgammal, "Towards scalable view-invariant gait recognition: Multilinear analysis for gait," in *Proc. Int. Conf. Audio Video-Based Biometric Person Authentication*, Jul. 2005, pp. 395–405.

[27] M. A. O. Vasilescu and D. Terzopoulos, "Multilinear analysis of image ensembles: Tensorfaces," in *Proc. 7th Eur. Conf. Comput. Vis.*, May 2002, pp. 447–460.

[28] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "Multilinear principal component analysis of tensor objects for recognition," in *Proc. Int. Conf. Pattern Recognit.*, Aug. 2006, pp. 776–779.

[29] D. Xu, S. Yan, L. Zhang, H.-J. Zhang, Z. Liu, and H.-Y. Shum;, "Concurrent subspaces analysis," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, vol. II, pp. 203–208.

[30] S. Yan, D. Xu, Q. Yang, L. Zhang, X. Tang, and H.-J. Zhang, "Discriminant analysis with tensor representation," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, vol. I, pp. 526–532.

[31] 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.

[32] D. Xu, S. Yan, D. Tao, L. Zhang, X. Li, and H.-J. Zhang, "Human gait recognition with matrix representation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 7, pp. 896–903, Jul. 2006.

[33] M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cogn. Neurosci.*, vol. 3, no. 1, pp. 71–86, 1991.

[34] P. Kroonenberg and J. Leeuw, "Principal component analysis of three-mode data by means of alternating least squares algorithms," *Psychometrika*, vol. 45, no. 1, pp. 69–97, 1980.

[35] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1985.

[36] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 711–720, Jul. 1997.

[37] J. Wang, K. N. Plataniotis, and A. N. Venetsanopoulos, "Selecting discriminant eigenfaces for face recognition," *Pattern Recognit. Lett.*, vol. 26, no. 10, pp. 1470–1482, 2005.

[38] T. K. Moon and W. C. Stirling, *Mathematical Methods and Algorithms for Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 2000.

[39] H. Moon and P. J. Phillips, "Computational and performance aspects of PCA-based face recognition algorithms," *Perception*, vol. 30, pp. 303–321, 2001.

[40] P. J. Phillips, H. Moon, S. A. Rizvi, and P. Rauss, "The FERET evaluation method for face recognition algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 10, pp. 1090–1104, Oct. 2000.

Haiping Lu (S'01) received the B.Eng. and M.Eng. degrees in electrical and electronic engineering from Nanyang Technological University, Singapore, in 2001 and 2004, respectively. Currently, he is working towards the Ph.D. degree at The Edward S. Rogers Sr. Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada.

His current research interests include statistical pattern recognition and tensor object processing, with applications in human detection, tracking, and recognition.

Konstantinos N. (Kostas) Plataniotis (S'90–M'92–SM'03) received the B.Eng. degree in computer engineering and informatics from University of Patras, Patras, Greece, in 1988 and the M.S. and Ph.D. degrees in electrical engineering from Florida Institute of Technology (Florida Tech), Melbourne, in 1992 and 1994, respectively.

Currently, he is an Associate Professor at The Edward S. Rogers Sr. Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada.

Dr. Plataniotis is a registered professional engineer in the province of Ontario, Canada, and a member of the Technical Chamber of Greece. He is the 2005 recipient of IEEE Canada's Outstanding Engineering Educator Award "for contributions to engineering education and inspirational guidance of graduate students" and the corecipient of the 2006 IEEE TRANSACTIONS ON NEURAL NETWORKS Outstanding Paper Award for the paper "Face Recognition Using Kernel Direct Discriminant Analysis Algorithms" published in 2003 (with J. Lu and A. N. Venetsanopoulos).

Anastasios N. Venetsanopoulos (S'66–M'69–SM'79–F'88) received the B.S. degree in electrical and mechanical engineering from the National Technical University of Athens (NTU), Athens, Greece, in 1965 and the M.S., M.Phil., and Ph.D. degrees in electrical engineering from Yale University, New Haven, CT, in 1966, 1968, and 1969, respectively.

He has served as a Lecturer in 138 short courses to industry and continuing education programs and as a Consultant to numerous organizations. He is the coauthor of four and a contributor to 35 books, and has published over 800 papers in refereed journals and conference proceedings. He served as the 12th Dean of the Faculty of Applied Science and Engineering, University of Toronto and is currently the Vice President for Research and Innovation at Ryerson University, Toronto, ON, Canada.

Prof. Venetsanopoulos has served as the Chair on numerous boards, councils, and technical conference committees of the IEEE. He is a Fellow of the Engineering Institute of Canada (EIC) and a Fellow of Canadian Academy of Sciences. He was awarded an Honorary Doctorate from the National Technical University of Athens in October 1994. In October 1996, he was awarded the Excellence in Innovation Award of the Information Technology Research Centre of Ontario and Royal Bank of Canada, for innovative work in color image processing and its industrial applications. He is the recipient of the Millennium Medal of IEEE. In 2003, he was the recipient of the highest award of the Canadian IEEE, the MacNaughton Award, and in 2003, he served as the Chair of the Council of Deans of Engineering of the Province of Ontario (CODE). He is the joint recipient of the 2006 IEEE TRANSACTIONS ON NEURAL NETWORKS Outstanding Paper Award.