Appearance Modeling on Visual Tracking and Foreground Segmentation by Incremental Tensor-Based Subspace Learning

Journal:	*Transactions on Pattern Analysis and Machine Intelligence*
Manuscript ID:	TPAMI-2007-09-0600
Manuscript Type:	Regular
Keywords:	I.4.8.n Tracking < I.4.8 Scene Analysis < I.4 Image Processing and Computer Vision < I Computing Methodologies, I.4.6.d Pixel classification < I.4.6 Segmentation < I.4 Image Processing and Computer Vision < I Computing Methodologies, I.4.8.c Image models < I.4.8 Scene Analysis < I.4 Image Processing and Computer Vision < I Computing Methodologies

Appearance Modeling on Visual Tracking and Foreground Segmentation by Incremental Tensor-Based Subspace Learning

Xi Li[†], Weiming Hu[†], Zhongfei Zhang[‡], Xiaoqin Zhang[†], Guan Luo[†]

[†]National Laboratory of Pattern Recognition, CASIA, Beijing, China

{lixi, wmhu, xqzhang, gluo}@nlpr.ia.ac.cn

[‡]State University of New York, Binghamton, NY 13902, USA

zhongfei@cs.binghamton.edu

Abstract

Recently, appearance modeling has attracted more and more attention in computer vision and pattern recognition. In this paper, we propose an appearance model based on incremental rank-(R_1, R_2, R_3) tensor-based subspace learning algorithm (referred as *IRTSA*), which models the appearance of an object or a scene by incrementally learning a low-order tensor-based eigenspace representation through adaptively updating the sample mean and eigenbasis. Based on *IRTSA*, two applications to tracking and foreground segmentation are developed. For the tracking application[1], subspace analysis of object appearance is incorporated into the multilinear framework which online constructs a representation of object appearance ensembles using high-order tensors. Compared with existing image-as-vector tracking applications, the developed one better captures the intrinsic spatio-temporal characteristics of object appearance. For the application to foreground segmentation, we construct two *IRTSA*-based background models for grayscale and color images, respectively. In these two models, the spatio-temporal characteristics of the scene are well captured, leading to a robust foreground segmentation result. Theoretic analysis and experimental evaluations against the state-of-the-art methods demonstrate the promise and effectiveness of the proposed *IRTSA* and its two *IRTSA*-based applications to tracking and foreground segmentation.

Index Terms

Appearance modeling, tracking, foreground segmentation, incremental tensor-based subspace learning, tensor decomposition, HOSVD.

I. Introduction

Appearance modeling plays an important role in computer vision and pattern recognition. Typically, a color histogram (CH) [2][3] is used to represent the appearance of an object region, due to the simplicity and robustness (to scaling, rotation, and non-rigid deformation). However, the potential problem with CH is that the spatial layout information of an object appearance is completely ignored. As a result, it is difficult to distinguish two objects with similar colors but different spatial distributions. In order to address this problem of CH, some other appearance models [4][5], based on kernel density estimation, are developed. With the capabilities to better capture the spatial information, they are more robust to noise. Nevertheless, they typically require a high computational complexity and a large storage space. Yet other popular appearance

[1]This work is to appear in ICCV'07. See [1] for details.

models [6][7][8][9][15][17][29] employs the Gaussian mixture model (GMM) to obtain the spatio-temporal statistics of pixels. However, these GMM-based appearance models share the disadvantage that they independently consider the spatial-temporal statistics of each single pixel, and ignore the intrinsic relationships among pixels. Furthermore, the number of Gaussians and a learning rate require setting in advance. Wang *et al.* [10] present an adaptive appearance model based on GMM in a joint spatial-color space (referred as SMOG). SMOG captures rich spatial layout and color information. The downside is that the global spatio-temporal varying information of pixels cannot be effectively captured by SMOG (a spatial weighted version of GMM). Conditional Random Fields are also used in the literature (e.g. [35]) to improve the performance of image modeling. But their training costs are usually very expensive; in addition, they only consider local distribution information of pixels (usually assuming to follow the Markov property), whereas the global information is poorly captured. Recent work utilizes the online subspace learning technique [31] to capture the global statistical information of pixels; due to their image-as-vector representations, the spatial information of pixels is almost lost. As a result, they are sensitive to noise or some global appearance variations (e.g. varying lighting). More recently, multilinear subspace analysis (referred as MSA) is used for appearance modeling. MSA offline constructs a representation of appearance ensembles using high-order tensors. This reduces spatio-temporal redundancies substantially, whereas the task of appearance modeling is done offline, resulting in a high computational complexity.

In this paper, we present an online tensor-based subspace learning algorithm (referred here as *IRTSA*), which models the appearance of an object or a scene by incrementally learning a low-order tensor-based eigenspace representation through adaptively updating the sample mean and eigenbasis. Compared with existing image-as-vector approaches to appearance modeling, the proposed *IRTSA* better captures the intrinsic spatio-temporal characteristics of object appearance. On the other hand, *IRTSA* works online, resulting in a much lower computational complexity than those of the traditional approaches to offline tensor decomposition. Based on *IRTSA*, two specific applications to visual tracking and foreground segmentation are developed.

The remainder of this paper is organized as follows. We briefly review the related work in Section II. In Section III, we introduce the incremental tensor-based subspace learning theory as well as the *IRTSA*. In the following two sections (IV and V), we discuss the specific applications of *IRTSA* to the online tracking and to the foreground segmentation, respectively. In Section VI,

we report the empirical evaluations. We conclude the paper in Section VII.

II. Related work

Tracking and foreground segmentation are common foundations for many computer vision applications such as behavior analysis and event detection. In tracking and foreground segmentation, how to construct an effective and efficient appearance model has become a challenging issue. In this paper, we mainly focus on appearance modeling on tracking and foreground segmentation. In addition, tensor-based appearance models are very popular in recent years. Therefore, it is necessary to give a brief review on tensor-based appearance modeling.

A. Appearance-based tracking

For visual tracking, handling appearance variations of an object is a fundamental and challenging task. In general, there are two types of appearance variations: intrinsic and extrinsic. Pose variation and/or shape deformation of an object are considered as the intrinsic appearance variations while the extrinsic variations are due to the changes resulting from different illumination, camera motion, camera viewpoint, and occlusion. Consequently, effectively modeling such appearance variations plays a critical role in visual tracking.

In recent years, much work has been done in visual tracking based on modeling the appearance of an object. Hager and Belhumeur [11] propose a tracking algorithm which uses an extended gradient-based optical flow method to handle object tracking under varying illumination conditions. They construct a set of illumination basis for a fixed pose with an illumination change. Black *et al.* [12] present a subspace learning based tracking algorithm with the subspace constancy assumption. A pre-trained, view-based eigenbasis representation is used for modeling appearance variations. However, the algorithm does not work well in the clutter with a large lighting change due to the subspace constancy assumption. In [13], curves or splines are exploited to represent the appearance of an object to develop the Condensation algorithm for contour tracking. Due to the simplistic representation scheme, the algorithm is unable to handle the pose or illumination change, resulting in a usually unsuccessful tracking result under a varying lighting condition. Black *et al.* [14] employ a mixture model to represent and recover the appearance changes in consecutive frames. Jepson *et al.* [15] develop a more elaborate mixture model with an online EM algorithm to explicitly model the appearance change during tracking. Comaniciu

et al.[16] propose a new approach to target representation and localization by spatial masking with an isotropic kernel. Zhou *et al.* [17] embed appearance-adaptive models into a particle filter to achieve a robust visual tracking. Yu *et al.* [18] propose a spatial-appearance model which captures non-rigid appearance variations and recovers all motion parameters efficiently. Li *et al.* [19] use a generalized geometric transform to handle the deformation, articulation, and occlusion of appearance. Wong *et al.* [20] present a robust appearance-based tracking algorithm using an online-updating sparse Bayesian classifier. Lee and Kriegman [21] present an online learning algorithm to incrementally learn a generic appearance model from the video. Lim *et al.* [22] present a human tracking framework using robust system dynamics identification and nonlinear dimensiona reduction techniques. Ho *et al.* [23] present a visual tracking algorithm based on linear subspace learning. Li *et al.* [24] propose an incremental PCA algorithm for subspace learning. In [25], a weighted incremental PCA algorithm for subspace learning is presented. Limy *et al.*[27] propose a generalized tracking framework based on the incremental image-as-vector subspace learning methods with a sample mean update.

It is noted that the above tracking methods are unable to fully exploit the spatial redundancies within the image ensembles. This is particularly true for those image-as-vector tracking techniques, as the local spatial information is almost lost. Consequently, the focus has been made on developing the image-as-matrix learning algorithms for effective subspace analysis.

B. Background modeling for foreground segmentation

In recent years, much work has been done in background modeling. Stauffer and Grimson [29] propose an online adaptive background model where a mixture of Gaussians is adopted to model each pixel. The model classifies each pixel by matching the pixel with the Gaussian distribution representing the pixel most effectively. Furthermore, the number of Gaussians is adjusted adaptively to best represent background processes. Sheikh and Shah [32] present an improved nonparametric model combining both temporal and spatial information. In [33], an adaptive background model for grayscale video sequences is presented. The model utilizes local spatio-temporal statistics to detect shadows and highlights. Furthermore, it can adapt to illumination changes. Haritaoglu *et al.* [30] build a statistical background model representing each pixel by three values which are its minimum intensity value, its maximum intensity value, and the maximum intensity difference between consecutive frames during training. In [34], Wang

et al. present a probabilistic method for background subtraction and shadow removal. Their method detects shadows by a combined intensity and edge measure. Tian *et al.* [48] propose an adaptive Gaussian mixture model based on a local normalized cross-correlation metric and a texture similarity metric. The model is used for detecting shadows and illumination changes, respectively. Wang *et al.* [35] present a dynamic conditional random field model for foreground and shadow segmentation. The model utilizes a dynamic probabilistic framework based on the conditional random field (CRF) to capture spatial and temporal statistics of pixels. In [31], PCA is performed on a collection of N images to construct a background model, which is represented by the mean image and the projection matrix comprising the first p significant eigenvectors of PCA. In this way, foreground segmentation is accomplished by computing the difference between the input image and its reconstruction. And then the online PCA is enabled to incrementally learn the background's eigenspace representation.

However, the above foreground segmentation methods are incapable of fully exploiting the spatio-temporal information of a scene. Especially for those techniques based on image-as-vector subspace learning, the local spatial information is almost lost, leading to an incorrect foreground segmentation result. Consequently, it is necessary to develop the learning algorithms which can effectively capture the spatio-temporal characteristics of a scene.

C. Tensor-based appearance modeling

More recent work on modeling the appearance of an object focuses on using high-order tensors to construct a better representation of the object's appearance. The intrinsic spatio-temporal information of the object's appearance is better captured by tensor-based appearance modeling methods due to their image-as-matrix representations. In this case, the problem of tensor-based appearance modeling is reduced to how to make tensor decomposition more accurate and efficient. Yang *et al.* [36] develop a 2-dimensional PCA (2DPCA) for image representation. Based on the original image matrices, 2DPCA constructs an image covariance matrix whose eigenvectors are derived for image feature extraction. Ye *et al.* [37] present a learning method called 2-dimensional linear discriminant analysis (2DLDA). In [38], a novel algorithm, called GLRAM, is proposed for low rank approximations of a collection of matrices. In [39], Ye *et al.* present a new dimension reduction algorithm named GPCA, which constructs the matrix representation of images directly. Wang and Ahuja [40] propose a novel rank-R tensor approximation approach,

which is designed to capture the spatio-temporal redundancies of tensors. In [41], an algorithm named Discriminant Analysis with Tensor Representation (DATER) is proposed. DATER is tensorized from the popular vector-based LDA algorithm. In [42] and [43], the N-mode SVD, multilinear subspace analysis, is applied to construct a compact representation of facial image ensembles factorized by different faces, expressions, viewpoints, and illuminations. He *et al.* [44] present a learning algorithm called Tensor Subspace Analysis (TSA), which learns a lower dimensional tensor-based subspace to characterize the intrinsic local geometric structure of the tensor space. In [45], Wang *et al.* give a convergent solution for general tensor-based subspace learning. Sun *et al.* [46] mine higher-order data streams using dynamic and streaming tensor analysis. Also in [47], Sun *et.al* present a window-based tensor analysis method for representing data streams over the time. All of these tensor-based algorithms share the same problem that they are not allowed for incremental subspace analysis for adaptively updating the sample mean and eigenbasis.

III. Incremental Tensor-based Subspace Learning

Before presenting the proposed online tensor-based subspace learning algorithm, we first give a brief review of the related background as well as the introduction to the notations and symbols we use.

A. Multilinear algebra

The mathematical foundation of multilinear analysis is the tensor algebra. A tensor can be regarded as a multidimensional matrix. We denote an N-order tensor as $\mathcal{A} \in \mathcal{R}^{I_1 \times I_2 \times \ldots \times I_N}$, each element of which is represented as $a_{i_1 \cdots i_n \cdots i_N}$ for $1 \leq i_n \leq I_n$. In the tensor terminology, each dimension of a tensor is associated with a "*mode*". The mode-n unfolding matrix $\mathrm{A}_{(n)} \in \mathcal{R}^{I_n \times (\prod_{i \neq n} I_i)}$ of $\mathcal{A}$ consists of the I_n−dimensional mode-n vectors obtained by varying the nth-mode index i_n while keeping the other mode indices fixed. Namely, the column vectors of $\mathrm{A}_{(n)}$ are just the mode-n vectors. For a better understanding of the tensor unfolding, we take advantage of Fig. 1 to explain the process of the unfolding. The inverse operation of the mode-n unfolding is the mode-n folding, which can restore the original tensor $\mathcal{A}$ from the mode-n unfolding matrix $\mathrm{A}_{(n)}$, i.e. $\mathcal{A} = \mathrm{fold}(\mathrm{A}_{(n)}, n)$. The mode-$n$ product of $\mathcal{A}$ and a matrix $\mathrm{U} \in \mathcal{R}^{J_n \times I_n}$ is denoted as

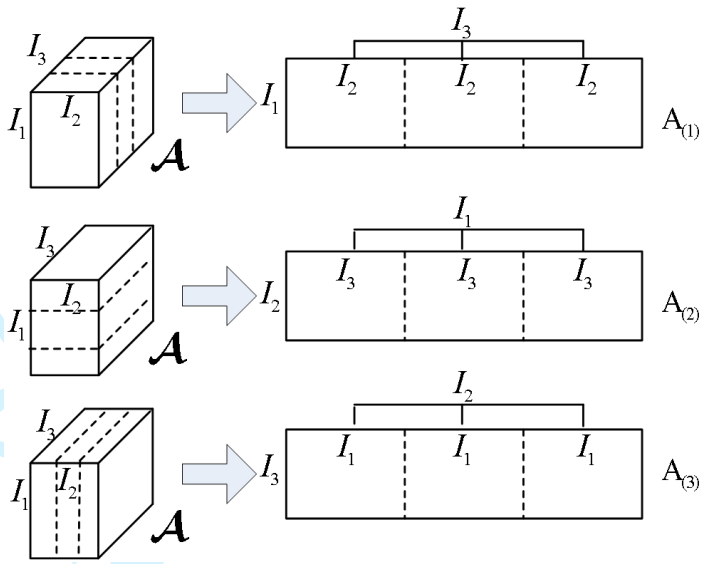

Fig. 1. **Illustration of unfolding a (3-order) tensor.**

$\mathcal{A}\times_n \mathrm{U} \in \mathcal{R}^{I_1\times\ldots\times I_{n-1}\times J_n\times I_{n-1}\times\ldots\times I_N}$ whose entries are as follows:

$$(\mathcal{A}\times_n \mathrm{U})_{i_1\cdots i_{n-1}j_n i_{n+1}\cdots i_N} = \sum_{i_n} a_{i_1\ldots i_n\ldots i_N} u_{j_n i_n} \tag{1}$$

Given a tensor $\mathcal{A}\in\mathcal{R}^{I_1\times I_2\times\ldots\times I_N}$ and the matrices $\mathrm{C}\in\mathcal{R}^{J_n\times I_n}$, $\mathrm{D}\in\mathcal{R}^{K_n\times J_n}$, $\mathrm{E}\in\mathcal{R}^{J_m\times I_m}(n\neq m)$, the mode-$n$ product has the following properties:

$$(\mathcal{A}\times_n C)\times_m E = (\mathcal{A}\times_m E)\times_n C = \mathcal{A}\times_n C\times_m E;\ \ (\mathcal{A}\times_n C)\times_n D = \mathcal{A}\times_n (D\cdot C)$$

The scalar product of two tensors $\mathcal{A},\mathcal{B}$ is defined as:

$$\langle\mathcal{A},\mathcal{B}\rangle = \sum_{i_1}\sum_{i_2}\cdots\sum_{i_N} a_{i_1\ldots i_N} b_{i_1\ldots i_N} \tag{2}$$

The Frobenius norm of $\mathcal{A}\in\mathcal{R}^{I_1\times I_2\times\cdots\times I_N}$ is defined as: $\|\mathcal{A}\| = \sqrt{\langle\mathcal{A},\mathcal{A}\rangle}$. The mode-$n$ rank R_n of $\mathcal{A}$ is defined as the dimension of the space generated by the mode-n vectors: $R_n = \mathrm{rank}(\mathrm{A}_{(n)})$. More details of the tensor algebra are given in [28].

B. *Tensor decomposition*

The Higher-Order Singular Value Decomposition (HOSVD) [42] is a generalized form of the conventional matrix singular value decomposition (SVD). An N-order tensor $\mathcal{A}$ is an N-dimensional matrix composed of N vector spaces. HOSVD seeks for N orthonormal matrices

TABLE I

THE N-MODE HOSVD ALGORITHM

for n=1 to N

1) Compute the SVD of the mode-n unfolding matrix $\mathrm{A}_{(n)} = \widetilde{\mathrm{U}}_n \cdot \widetilde{\mathrm{D}}_n \cdot \widetilde{\mathrm{V}}_n^T$.
2) Set the mode matrix $\mathrm{U}^{(n)}$ as the orthonormal matrix $\widetilde{\mathrm{U}}_n$.

end

Compute the core tensor as:

$$\mathcal{B} = \mathcal{A} \times_1 \mathrm{U}^{(1)^T} \ldots \times_n \mathrm{U}^{(n)^T} \ldots \times_N \mathrm{U}^{(N)^T}$$

$\mathrm{U}^{(1)}, \ldots, \mathrm{U}^{(N)}$ which span these N spaces, respectively. Consequently, the tensor $\mathcal{A}$ can be decomposed as the following form:

$$\mathcal{A} = \mathcal{B} \times_1 \mathrm{U}^{(1)} \times_2 \mathrm{U}^{(2)} \cdots \times_N \mathrm{U}^{(N)} \tag{3}$$

where $\mathcal{B} = \mathcal{A} \times_1 \mathrm{U}^{(1)^T} \times_2 \mathrm{U}^{(2)^T} \cdots \times_N \mathrm{U}^{(N)^T}$ which denotes the core tensor controlling the interaction among the mode matrices $\mathrm{U}^{(1)}, \ldots, \mathrm{U}^{(N)}$. The orthonormal column vectors of $\mathrm{U}^{(n)}$ span the column space of the mode-n unfolding matrix $\mathrm{A}_{(n)}$ $(1 \leq n \leq N)$. In this way, we have the N-mode HOSVD algorithm [42] illustrated in Table I.

In the next two sections (III-C and III-D), we will discuss the proposed incremental rank-(R_1, R_2, R_3) tensor-based subspace learning algorithm (*IRTSA*) for 3-order tensors. *IRTSA* applies the online learning technique (R-SVD [26][27]) to find the dominant projection subspaces of 3-order tensors.

C. Introduction to R-SVD

The classic R-SVD algorithm [26] efficiently computes the SVD of a dynamic matrix with newly added columns or rows, based on the existing SVD. Unfortunately, the R-SVD algorithm [26] is based on the zero mean assumption, leading to the failure of tracking subspace variabilities. Based on [26], [27] extends the R-SVD algorithm to compute the eigenbasis of a scatter matrix with the mean update. The details of R-SVD are given as follows.

Given a matrix $H = \{\mathrm{K}_1, \mathrm{K}_2, \ldots, \mathrm{K}_g\}$ and its column mean K, we let CVD(H) denote the SVD of the matrix $\{\mathrm{K}_1 - \mathrm{K}, \mathrm{K}_2 - \mathrm{K}, \ldots, \mathrm{K}_g - \mathrm{K}\}$. Given the column mean L_p of the existing data matrix $H_p = \{\mathrm{L}_1, \mathrm{L}_2, \ldots, \mathrm{L}_n\}$, CVD$(H_p) = U_p \Sigma_p V_p^T$, the column mean L_q of the new data matrix $F = \{\mathrm{L}_{n+1}, \mathrm{L}_{n+2}, \ldots, \mathrm{L}_{n+m}\}$, and the column mean L_e of the entire data matrix $H_e = (H_p \mid F)$, CVD$(H_e) = U_e \Sigma_e V_e^T$ can be determined as:

1) Compute $\mathrm{L}_e = \frac{n}{n+m}\mathrm{L}_p + \frac{m}{n+m}\mathrm{L}_q$;
2) Compute $\tilde{F} = \left(F - \mathrm{L}_q \mathbb{1}_{1\times m} \mid \sqrt{\frac{nm}{n+m}}\,(\mathrm{L}_p - \mathrm{L}_q)\right)$, where $\mathbb{1}_{1\times m}$ is $(\overbrace{1, 1, \ldots, 1}^{m})$;
3) Apply the classic R-SVD algorithm [26] with $U_p \Sigma_p V_p^T$ and the new data matrix $\tilde{F}$ to obtain $U_e \Sigma_e V_e^T$.

In order to fit the data streams well, the forgetting factor is introduced by [27] to weight the data streams. Typically, recent observations are given more weights than historical ones. For example, the weighted data matrix $H_e^{'}$ of H_e may be formulated as: $H_e^{'} = (\lambda H_p \mid F) = \left(U_p(\lambda\Sigma_p)V_p^T \mid F\right)$ where λ is the forgetting factor. The analytical proof of R-SVD is given in [26][27].

D. *Incremental rank-(R_1, R_2, R_3) tensor-based subspace analysis*

Based on HOSVD [42], *IRTSA* presented below efficiently identifies the dominant projection subspaces of 3-order tensors, and is capable of incrementally updating these subspaces when new data arrive. Given the CVD($\mathrm{A}_{(k)}$) of the mode-k unfolding matrix $\mathrm{A}_{(k)}(1 \le k \le 3)$ for a 3-order tensor $\mathcal{A} \in \mathcal{R}^{I_1\times I_2\times I_3}$, *IRTSA* is able to efficiently compute the CVD($\mathrm{A}^*_{(i)}$) of the mode-i unfolding matrix $\mathrm{A}^*_{(i)}(1 \le i \le 3)$ for $\mathcal{A}^* = (\mathcal{A} \mid \mathcal{F}) \in \mathcal{R}^{I_1\times I_2\times I_3^*}$ where $\mathcal{F} \in \mathcal{R}^{I_1\times I_2\times I_3^{'}}$ is a new 3-order subtensor and $I_3^* = I_3 + I_3^{'}$. To facilitate the description, Fig. 2(b) is used for illustration. In the left half of Fig. 2, three identical tensors are unfolded in three different modes. For each tensor, the white regions represent the original subtensor while the dark regions denote the newly added subtensor. The three unfolding matrices corresponding to the three different modes are shown in the right half of Fig. 2, where the dark regions represent the unfolding matrices of the newly added subtensor $\mathcal{F}$. With the emergence of the new data subtensors, the column spaces of $\mathrm{A}^*_{(1)}$ and $\mathrm{A}^*_{(2)}$ are extended at the same time when the row space of $\mathrm{A}^*_{(3)}$ is extended. Consequently, *IRTSA* needs to track the changes of these three unfolding spaces, and needs to identify the dominant projection subspaces for a compact representation of the tensor. It is noted that $\mathrm{A}^*_{(2)}$ can be decomposed as: $\mathrm{A}^*_{(2)} = \left(\mathrm{A}_{(2)} \mid \mathrm{F}_{(2)}\right)\cdot\mathrm{P} = \mathrm{B}\cdot\mathrm{P}$, where $\mathrm{B} = \left(\mathrm{A}_{(2)} \mid \mathrm{F}_{(2)}\right)$ and P is an orthonormal matrix obtained by column exchange and transpose operations on an $(I_1 \cdot I_3^*)$-order identity matrix G. Let

$$G = (\overbrace{E_1}^{I_3} \mid \overbrace{Q_1}^{I_3^{'}} \mid \overbrace{E_2}^{I_3} \mid \overbrace{Q_2}^{I_3^{'}} \mid \cdots \mid \cdots \mid \overbrace{E_{I_1}}^{I_3} \mid \overbrace{Q_{I_1}}^{I_3^{'}})$$

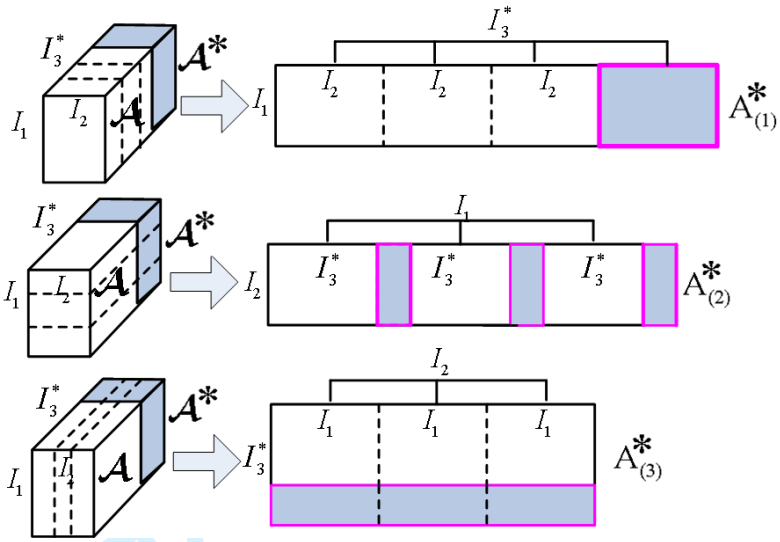

Fig. 2. **Illustration of the incremental rank-(R_1, R_2, R_3) tensor subspace learning of a 3-order tensor.**

which is generated by partitioning G into $2I_1$ blocks in the column dimension. Consequently, the orthonormal matrix P is formulated as:

$$\mathrm{P} = (E_1|E_2|\cdots|E_{I_1}|Q_1|Q_2|\cdots|Q_{I_1})^T. \tag{4}$$

In this way, CVD($\mathrm{A}^*_{(2)}$) is efficiently computed on the basis of P and CVD(B) obtained by applying R-SVD to B. Furthermore, CVD($\mathrm{A}^*_{(1)}$) is efficiently obtained by performing R-SVD on the matrix $\left(\mathrm{A}_{(1)} \mid \mathrm{F}_{(1)}\right)$. Similarly, CVD($\mathrm{A}^*_{(3)}$) is efficiently obtained by performing R-SVD on the matrix $\left(\frac{\mathrm{A}_{(3)}}{\mathrm{F}_{(3)}}\right)^T$. The specific procedure of *IRTSA* is listed in Fig. 3.

E. Complexity analysis of IRTSA and other related methods

Compared with the offline HOSVD, the proposed *IRTSA* based on online tensor decomposition adapts to appearance variations of the object with a much lower complexity. A quantitative complexity analysis of *IRTSA* and HOSVD is given as follows. *IRTSA* requires $O[I_1 \cdot I_2 \cdot (I_3 + I_3') \cdot (R_1 + R_2 + R_3)]$ operations and $O[I_1 \cdot R_1 + I_2 \cdot R_2 + I_1 \cdot I_2 \cdot (R_3 + I_3')]$ memory units. In comparison, HOSVD requires $O[I_1 \cdot I_2 \cdot (I_1 + I_2 + I_3 + I_3') \cdot (I_3 + I_3')]$ operations and $O[I_1 \cdot (I_3 + I_3') \cdot I_2]$ memory units. Consequently, when I_3 $(I_3 \gg I_3')$ is very large, the complexity of HOSVD is much higher than that of *IRTSA*. In addition, if K eigenvectors are maintained during tracking,

Input:

CVD($\mathrm{A}_{(k)}$) of the mode-k unfolding matrix $\mathrm{A}_{(k)}$, i.e. $\mathrm{U}^{(k)}\mathrm{D}^{(k)}\mathrm{V}^{(k)^T}$ $(1 \leq k \leq 3)$ of an original tensor $\mathcal{A} \in \mathcal{R}^{I_1 \times I_2 \times I_3}$, newly-added tensor $\mathcal{F} \in \mathcal{R}^{I_1 \times I_2 \times I_3'}$, column mean $\bar{L}^{(1)}$ of $\mathrm{A}_{(1)}$, column mean $\bar{L}^{(2)}$ of $\mathrm{A}_{(2)}$, row mean $\bar{L}^{(3)}$ of $\mathrm{A}_{(3)}$ and R_1, R_2, R_3.

Output:

CVD($\mathrm{A}^*_{(i)}$) of the mode-i unfolding matrix $\mathrm{A}^*_{(i)}$, i.e. $\hat{\mathrm{U}}^{(i)}\hat{\mathrm{D}}^{(i)}\hat{\mathrm{V}}^{(i)^T}(1 \leq i \leq 3)$ of $\mathcal{A}^* = (\mathcal{A} \mid \mathcal{F}) \in \mathcal{R}^{I_1 \times I_2 \times I_3^*}$ where $I_3^* = I_3 + I_3'$, column mean $\bar{L}^{(1)^*}$ of $\mathrm{A}^*_{(1)}$, column mean $\bar{L}^{(2)^*}$ of $\mathrm{A}^*_{(2)}$ and row mean $\bar{L}^{(3)^*}$ of $\mathrm{A}^*_{(3)}$.

Algorithm:

1. $\mathrm{A}^*_{(1)} = \left(\mathrm{A}_{(1)} \,|\, F_{(1)}\right)$;
2. $\mathrm{A}^*_{(2)} = \left(\mathrm{A}_{(2)} \,|\, F_{(2)}\right) \cdot \mathrm{P} = \mathrm{B} \cdot \mathrm{P}$, where P is defined in (4);
3. $\mathrm{A}^*_{(3)} = \left(\dfrac{\mathrm{A}_{(3)}}{F_{(3)}}\right)$;
4. $[\hat{\mathrm{U}}^{(1)}, \hat{\mathrm{D}}^{(1)}, \hat{\mathrm{V}}^{(1)}, \bar{L}^{(1)^*}]$=R-SVD$(\mathrm{A}^*_{(1)}, \bar{L}^{(1)}, R_1)$;
5. $[\hat{\mathrm{U}}^{(2)}, \hat{\mathrm{D}}^{(2)}, \widetilde{\mathrm{V}}_2, \bar{L}^{(2)^*}]$=R-SVD$(\mathrm{B}, \bar{L}^{(2)}, R_2)$;
6. $\hat{\mathrm{V}}^{(2)} = \mathrm{P}^T \cdot \widetilde{\mathrm{V}}_2$;
7. $[\widetilde{\mathrm{U}}_3, \widetilde{\mathrm{D}}_3, \widetilde{\mathrm{V}}_3, \widetilde{L}_3]$=R-SVD$((\mathrm{A}^*_{(3)})^T, (\bar{L}^{(3)})^T, R_3)$;
8. $\hat{\mathrm{U}}^{(3)} = \widetilde{\mathrm{V}}_3, \;\; \hat{\mathrm{D}}^{(3)} = (\widetilde{\mathrm{D}}_3)^T, \;\; \hat{\mathrm{V}}^{(3)} = \widetilde{\mathrm{U}}_3, \;\; \bar{L}^{(3)^*} = (\widetilde{L}_3)^T$.

Fig. 3. **The incremental rank-(R_1, R_2, R_3) tensor-based subspace analysis algorithm (*IRTSA*). R-SVD$((\mathbb{C} \,|\, \mathbb{E}), L, R)$ represents that the first R dominant eigenvectors are used in R-SVD [27] for the matrix $(\mathbb{C}|\mathbb{E})$ with $\mathbb{C}$'s column mean being L.**

the online PCA technique [27] (referred here as *IAVSL*) requires $O[I_1 \cdot I_2 \cdot (I_3 + I_3') \cdot K]$ operations and $O[I_1 \cdot I_2 \cdot (K + I_3')]$ memory units.

F. Likelihood evaluation for IRTSA

In real applications, it is necessary for a subspace analysis-based algorithm to evaluate the likelihood of the test sample and the learned subspace. In *IRTSA*, the criterion for the likelihood evaluation is given as follows.

Given I_3 existing images represented as $\mathcal{A} \in \mathcal{R}^{I_1 \times I_2 \times I_3}$, a test image denoted as $\mathcal{J} \in \mathcal{R}^{I_1 \times I_2 \times 1}$ and the mode-i column projection matrices $U^{(i)} \in \mathcal{R}^{I_i \times R_i}(1 \leq i \leq 2)$ and the mode-3 row

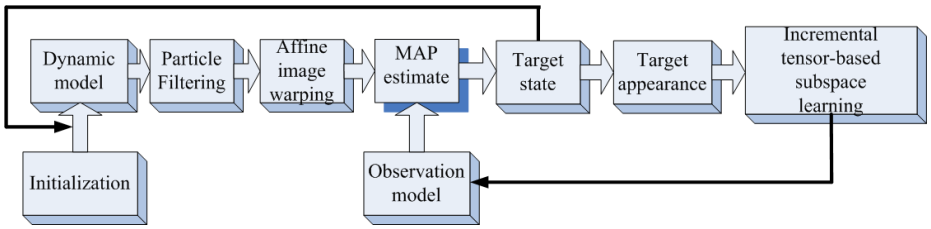

Fig. 4. **The architecture of the proposed tracking application.**

projection matrix $V^{(3)} \in \mathcal{R}^{(I_1 I_2)\times R_3}$ of the learned subspaces of $\mathcal{A}$, the likelihood can be determined by the sum of the reconstruction error norms of the three modes:

$$RE = \sum_{i=1}^{2} \|(\mathcal{J}-\mathcal{M}_i)-(\mathcal{J}-\mathcal{M}_i)\prod_{j=1}^{2}\times_j (U^{(j)}\cdot U^{(j)^T})\|^2 + \|(\mathrm{J}_{(3)}-\mathrm{M}_3)-(\mathrm{J}_{(3)}-\mathrm{M}_3)\cdot(V^{(3)}\cdot V^{(3)^T})\|^2 \quad (5)$$

where $\mathrm{J}_{(i)}$ is the mode-i unfolding matrix of $\mathcal{J}$, $\prod_{k=1}^{K}\times_k D_k = \times_1 D_1 \times_2 D_2 \ldots \times_K D_K$, $\mathrm{M}_3 = \bar{L}^{(3)}$ which is the row mean of the mode-3 unfolding matrix $\mathrm{A}_{(3)}$, $\mathcal{M}_1$ and $\mathcal{M}_2$ are defined as:

$$\mathcal{M}_1 = (\overbrace{\bar{L}^{(1)},\ldots,\bar{L}^{(1)}}^{I_2}) \in \mathcal{R}^{I_1\times I_2\times 1}, \quad \mathcal{M}_2 = (\overbrace{\bar{L}^{(2)},\ldots,\bar{L}^{(2)}}^{I_1})^T \in \mathcal{R}^{I_1\times I_2\times 1} \quad (6)$$

where $\bar{L}^{(1)}$ and $\bar{L}^{(2)}$ are the column means of the mode-$(1,2)$ unfolding matrices $\mathrm{A}_{(1)}$ and $\mathrm{A}_{(2)}$, respectively. The smaller the RE, the larger the likelihood.

IV. Tracking Application

A. *Overview of the tracking application*

The tracking application includes two stages: (a) incremental tensor-based subspace learning; and (b) Bayesian inference for visual tracking. In the first stage, a low dimensional tensor-based eigenspace model is learned online. The model uses the proposed *IRTSA* to identify the dominant projection subspaces of the 3-order tensors (object appearance ensembles). In the second stage, the object locations in consecutive frames are estimated by the Bayesian state inference within the framework in which a particle filter is applied to propagate sample distributions over the time. These two stages are executed repeatedly as time progresses. Moreover, the application has a strong adaptability in the sense that when new image data arrive, the tensor-based eigenspace model follows the updating online. The architecture of the proposed tracking application is shown in Fig. 4.

B. Bayesian inference for the tracking application

For visual tracking, a Markov model with a hidden state variable is generally used for motion estimation. In this model, the object motion between two consecutive frames is usually assumed to be an affine motion. Let X_t denote the state variable describing the affine motion parameters (the location) of an object at time t. Given a set of observed images $\mathcal{O}_t = \{O_1, \ldots, O_t\}$, the posterior probability is formulated by Bayes' theorem as:

$$p(X_t|\mathcal{O}_t) \propto p(O_t|X_t) \int p(X_t|X_{t-1}) p(X_{t-1}|\mathcal{O}_{t-1}) dX_{t-1} \tag{7}$$

where $p(O_t|X_t)$ denotes the likelihood function, and $p(X_t|X_{t-1})$ represents the dynamic model. $p(O_t|X_t)$ and $p(X_t|X_{t-1})$ decide the entire tracking process. A particle filter [13] is used for approximating the distribution over the location of the object with a set of weighted samples. Moreover, the resampling step for the particle filter is executed every three frames.

In the tracking application, we apply an affine image warping to model the object motion of two consecutive frames. The six parameters of the affine transform are used to model $p(X_t|X_{t-1})$ of a tracked object. Let $X_t = (x_t, y_t, \eta_t, s_t, \beta_t, \phi_t)$ where $x_t, y_t, \eta_t, s_t, \beta_t, \phi_t$ denote the x, y translations, the rotation angle, the scale, the aspect ratio, and the skew direction at time t, respectively. We employ a Gaussian distribution to model the state transition distribution $p(X_t|X_{t-1})$. Also the six parameters of the affine transform are assumed to be independent. Consequently, $p(X_t|X_{t-1})$ is formulated as:

$$p(X_t|X_{t-1}) = \mathcal{N}(X_t; X_{t-1}, \Sigma) \tag{8}$$

where Σ denotes a diagonal covariance matrix whose diagonal elements are $\sigma_x^2, \sigma_y^2, \sigma_\eta^2, \sigma_s^2, \sigma_\beta^2, \sigma_\phi^2$, respectively. The observation model $p(O_t|X_t)$ reflects the probability that a sample is generated from the subspace. In this paper, RE, defined in (5), is used to measure the distance from the sample to the center of the subspace. Consequently, $p(O_t|X_t)$ is formulated as:

$$p(O_t|X_t) \propto exp(-RE) \tag{9}$$

For MAP (maximum a posterior) estimate, we just use the affinely warped image region associated with the highest weighted hypothesis to update the tensor-based eigenspace model.

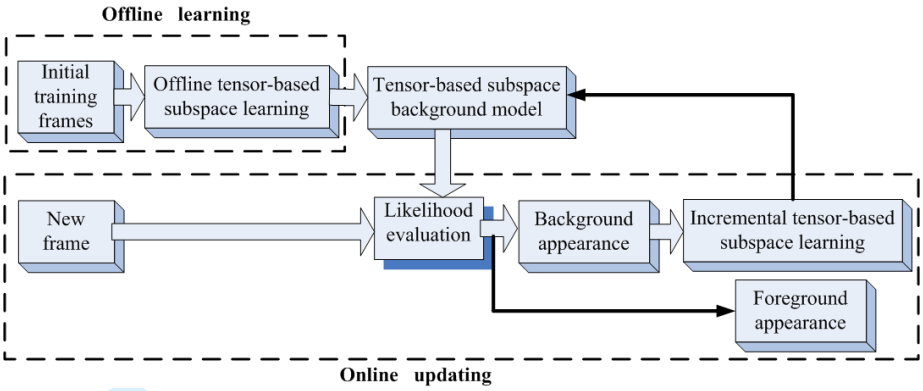

Fig. 5. **The architecture of the proposed application to foreground segmentation.**

C. Summary of the contributions of the tracking application

First, the application does not need to know any prior knowledge of the object. A low dimensional eigenspace representation is learned online, and is updated incrementally over the time. The application only assumes that the initialization of the object region is provided. Second, while the Condensation algorithm [13] is used for propagating the sample distributions over the time, we develop an effective probabilistic likelihood function based on the learned tensor-based eigenspace model. Third, while R-SVD [27] is applied to update both the sample mean and eigenbasis online as new data arrive, an incremental multilinear subspace analysis is enabled to capture the appearance characteristics of the object during the tracking.

V. Foreground Segmentation Application

A. Overview of the application to foreground segmentation

The application to foreground segmentation includes two stages: (a) offline learning; and (b) online updating. In the first stage, a low dimensional tensor-based eigenspace background model is learned by the offline HOSVD (referred to in Table I) over several initial frames for background training. In the second stage, two steps need to be executed. At step one, consecutive frames are evaluated by the learned tensor-based eigenspace background model to detect moving regions over the time. At step two, *IRTSA* is applied to online update the tensor-based eigenspace background model. These two steps are executed repeatedly as time progresses. The architecture of the application to foreground segmentation is shown in Fig. 5.

Now we are ready to discuss the two proposed background models (*IRTSA-GBM* and *IRTSA-CBM*) respectively in the next two sections (V-B and V-C).

B. *Grayscale background model (*IRTSA-GBM*)*

For a given matrix $X = (x_{ij})_{M\times N}$, let abs(X) be the matrix $Y = (y_{ij})_{M\times N}$ with the entry y_{ij} being the absolute value $|x_{ij}|$ of x_{ij}. Given the learned eigenspaces of an existing tensor $\mathcal{A} \in \mathcal{R}^{I_1\times I_2\times t}$ consisting of t background appearance matrices $BM_{1:t}$, i.e. the mode-i column projection matrices $U^{(i)} \in \mathcal{R}^{I_i\times R_i}(1 \leq i \leq 2)$ and the mode-3 row projection matrix $V^{(3)} \in \mathcal{R}^{(I_1 I_2)\times R_3}$, and a new frame $\mathcal{J}_{t+1} \in \mathcal{R}^{I_1\times I_2\times 1}$, the distance between $\mathcal{J}_{t+1}$ and the learned eigenspaces measured by the sum of the reconstruction difference matrices of the three modes is formulated as:

$$\begin{aligned} RM =& \text{fold}[\text{abs}(Q_3), 3] + \textstyle\sum_{i=1}^{2} \text{abs}(Q_i); \\ Q_i =& (\mathcal{J}_{t+1} - \mathcal{M}_i) - (\mathcal{J}_{t+1} - \mathcal{M}_i) \textstyle\prod_{j=1}^{2} \times_j (U^{(j)} \cdot U^{(j)^T}), \;\; i = 1, 2; \\ Q_3 =& (\mathrm{J}_{(3)} - \mathrm{M}_3) - (\mathrm{J}_{(3)} - \mathrm{M}_3) \cdot (V^{(3)} \cdot V^{(3)^T}); \end{aligned} \tag{10}$$

where fold$(\cdot)$ denotes tensor folding referred to in Section III-A, $\mathrm{J}_{(i)}$ is the mode-i unfolding matrices of $\mathcal{J}_{t+1}$, $\prod_{k=1}^{K} \times_k D_k = \times_1 D_1 \times_2 D_2 \ldots \times_K D_K$, $\mathrm{M}_3 = \bar{L}^{(3)}$ which is the row mean of the mode-3 unfolding matrix $\mathrm{A}_{(3)}$, $\mathcal{M}_1$ and $\mathcal{M}_2$ are defined as:

$$\mathcal{M}_1 = (\overbrace{\bar{L}^{(1)}, \ldots, \bar{L}^{(1)}}^{I_2}) \in \mathcal{R}^{I_1\times I_2\times 1}, \quad \mathcal{M}_2 = (\overbrace{\bar{L}^{(2)}, \ldots, \bar{L}^{(2)}}^{I_1})^T \in \mathcal{R}^{I_1\times I_2\times 1} \tag{11}$$

where $\bar{L}^{(1)}$ and $\bar{L}^{(2)}$ are the column means of the mode-$(1, 2)$ unfolding matrices $\mathrm{A}_{(1)}$ and $\mathrm{A}_{(2)}$, respectively. Let p_{ij} be the pixel corresponding to the entry $RM(i, j)$ of RM. In this way, the criterion for foreground segmentation is defined as:

$$p_{ij} = \begin{cases} \text{background} & \text{if } exp\left(-\frac{RM^2(i,j)}{\sigma^2}\right) > T_{gray} \\ \text{foreground} & \text{otherwise,} \end{cases} \tag{12}$$

where T_{gray} denotes a threshold. Let $BM_{t+1} \in \mathcal{R}^{I_1\times I_2}$ be the background matrix at time $t + 1$, whose entry $BM_{t+1}(i, j)$ is defined as:

$$BM_{t+1}(i, j) = \begin{cases} (1 - \alpha^*)\text{med}\left[BM_{t-2:t}(i, j)\right] + \alpha^* \mathcal{J}_{t+1}(i, j) & \text{if } p_{ij} \text{ belongs to foreground} \\ \mathcal{J}_{t+1}(i, j) & \text{otherwise} \end{cases} \tag{13}$$

where α^* is a learning rate factor, $BM_{t-2:t}(i,j)$ denotes $\{BM_{t-2}(i,j), BM_{t-1}(i,j), BM_t(i,j)\}$, and med $[\cdot]$ represents the median value of its vector argument. Subsequently, $IRTSA$ is applied to incrementally update the tensor-based eigenspace model of the background appearance ensembles $BM_{1:t}$ as t increases. In the next section (V-C), we discuss the proposed color background model, which is an extension to the proposed *IRTSA-GBM*.

C. *Color background model (*IRTSA-CBM*)*

In *IRTSA-CBM*, the RGB color space is transformed into the scaled one (r, g, s), where $r = R/(R+G+B)$, $g = G/(R+G+B)$, and $s = (R+G+B)/3$. Let $\mathcal{A}^r \in \mathcal{R}^{I_1 \times I_2 \times t}$ be the r-component image ensemble composed of t background appearance matrices $BM^r_{1:t}$, $\mathcal{A}^g \in \mathcal{R}^{I_1 \times I_2 \times t}$ be the g-component image ensemble composed of t background appearance matrices $BM^g_{1:t}$, $\mathcal{A}^s \in \mathcal{R}^{I_1 \times I_2 \times t}$ be the s-component image ensemble composed of t background appearance matrices $BM^s_{1:t}$, $\mathcal{J}^r_{t+1} \in \mathcal{R}^{I_1 \times I_2 \times 1}$ be the r-component frame at time $t+1$, $\mathcal{J}^g_{t+1} \in \mathcal{R}^{I_1 \times I_2 \times 1}$ be the g-component frame at time $t+1$, and $\mathcal{J}^s_{t+1} \in \mathcal{R}^{I_1 \times I_2 \times 1}$ be the s-component frame at time $t+1$. In this way, we have three 3-order tensors $BM^r_{1:t}$, $BM^g_{1:t}$, and $BM^s_{1:t}$ corresponding to the (r, g, s) components. The eigenspaces of these three tensors are obtained by applying *IRTSA* to them. The (r, g, s)-component distance matrices between the new frame and the learned subspace are respectively represented as RM^r, RM^g and RM^s, which are referred to in (10). Let p_{ij} be the pixel at the ith row and jth column. The criterion for the foreground segmentation is defined as:

$$p_{ij} = \begin{cases} \text{background} & \text{if } exp\left[-\left(\frac{RM^r(i,j)}{\sigma_r}\right)^2 - \left(\frac{RM^g(i,j)}{\sigma_g}\right)^2 - \left(\frac{RM^s(i,j)}{\sigma_s}\right)^2\right] > T_{color} \\ \text{foreground} & \text{otherwise,} \end{cases} \tag{14}$$

where σ_r, σ_g and σ_s are three scaling factors, and T_{color} is a threshold. Let $BM^r_{t+1} \in \mathcal{R}^{I_1 \times I_2}$, $BM^g_{t+1} \in \mathcal{R}^{I_1 \times I_2}$, and $BM^s_{t+1} \in \mathcal{R}^{I_1 \times I_2}$ respectively be the (r, g, s)-component background matrices at time $t+1$, whose entries $BM^r_{t+1}(i,j)$, $BM^g_{t+1}(i,j)$, and $BM^s_{t+1}(i,j)$ are respectively defined as:

$$BM^r_{t+1}(i,j) = \begin{cases} (1-\alpha_r)\text{med}\left[BM^r_{t-2:t}(i,j)\right] + \alpha_r \mathcal{J}^r_{t+1}(i,j) & \text{if } p_{ij} \text{ belongs to foreground} \\ \mathcal{J}^r_{t+1}(i,j) & \text{otherwise} \end{cases} \tag{15}$$

$$BM^g_{t+1}(i,j) = \begin{cases} (1-\alpha_g)\text{med}\left[BM^g_{t-2:t}(i,j)\right] + \alpha_g \mathcal{J}^g_{t+1}(i,j) & \text{if } p_{ij} \text{ belongs to foreground} \\ \mathcal{J}^g_{t+1}(i,j) & \text{otherwise} \end{cases} \tag{16}$$

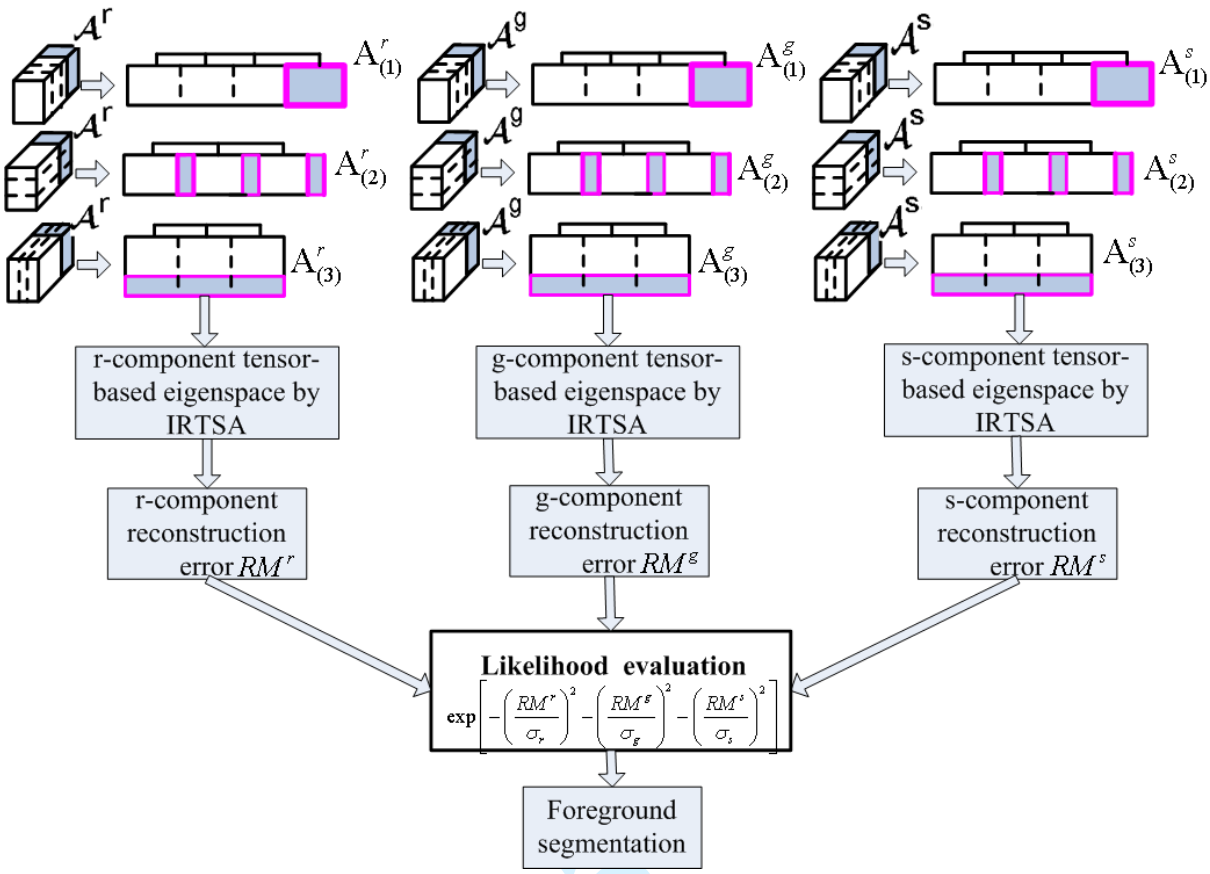

Fig. 6. **Illustration of the foreground segmentation process using IRTSA-CBM**

$$BM^s_{t+1}(i,j) = \begin{cases} (1-\alpha_s)\text{med}\left[BM^s_{t-2:t}(i,j)\right] + \alpha_s \mathcal{J}^s_{t+1}(i,j) & \text{if } p_{ij} \text{ belongs to foreground} \\ \mathcal{J}^s_{t+1}(i,j) & \text{otherwise} \end{cases} \tag{17}$$

where med $[\cdot]$ represents the median value of its vector argument, α_r, α_g and α_s are three learning rate factors. Subsequently, $IRTSA$ is applied to incrementally update the tensor-based eigenspace models of the background appearance ensembles $BM^r_{1:t}$, $BM^g_{1:t}$, and $BM^s_{1:t}$ as t increases. For a better understanding, Fig. 6 is used to illustrate the foreground segmentation process by IRTSA-CBM.

D. Summary of the contributions of the foreground segmentation application

The application online constructs a low-order tensor-based eigenspace background model, in which the sample mean and the eigenbasis are updated adaptively. As a result, the spatio-temporal information of a scene is well captured by our tensor-based eigenspace background

model. Moreover, the model is available for modeling both color and grayscale images.

VI. Experimental results

In this section, two *IRTSA*-based applications respectively to tracking and foreground segmentation are evaluated under many different circumstances.

A. Experimental evaluations of the tracking application

In order to evaluate the performance of the proposed tracking application, five videos are used in the experiments. Videos 1, 4, and 5 are captured indoor while videos 2 and 3 are recorded outdoor. Furthermore, videos 1, 3, and 5 are taken from moving cameras in different scenes while videos 2 and 4 are recorded by stationary cameras. Each frame in these videos is a 8-bit gray scale image. In the first video[2], a man walks in a room changing his pose and facial expression over the time with varying lighting conditions. In the second video, a pedestrian as a small object moves down a road in a dark and blurry scene. In the third video, a man walks from left to right in a bright road scene; his body pose varies over the time, with a drastic motion and pose change (bowing down to reach the ground and standing up back again) in the middle of the video stream. The fourth video consists of dark and motion-blurring gray scale images, where many motion events take place, including wearing and taking off the glasses, head shaking, and hands occluding the face from time to time. In the last video[2], a man moves in an office changing his pose and facial expression over the time. In the middle of the video stream, his face is completely occluded by his hand. Each frame in the last video contains seven benchmark points, which characterize the location and the shape of his face.

For the tensor-based eigenspace representation, the size of each object region is normalized to 20×20 pixels. The settings of the (R_1, R_2, R_3) in *IRTSA* and the eigenspace dimensionality in IAVSL are selected experimentally to produce optimal tracking results. The forgetting factor λ in R-SVD is set as 0.99. The tensor-based subspace is updated every three frames. For the particle filtering in the visual tracking, the number of particles is set to be 300. The six diagonal elements $(\sigma_x^2, \sigma_y^2, \sigma_\eta^2, \sigma_s^2, \sigma_\beta^2, \sigma_\phi^2)$ of the covariance matrix Σ in (8) are assigned as $(5^2, 5^2, 0.03^2, 0.03^2, 0.005^2, 0.001^2)$, respectively.

[2]http://www.cs.toronto.edu/∼dross/ivt/

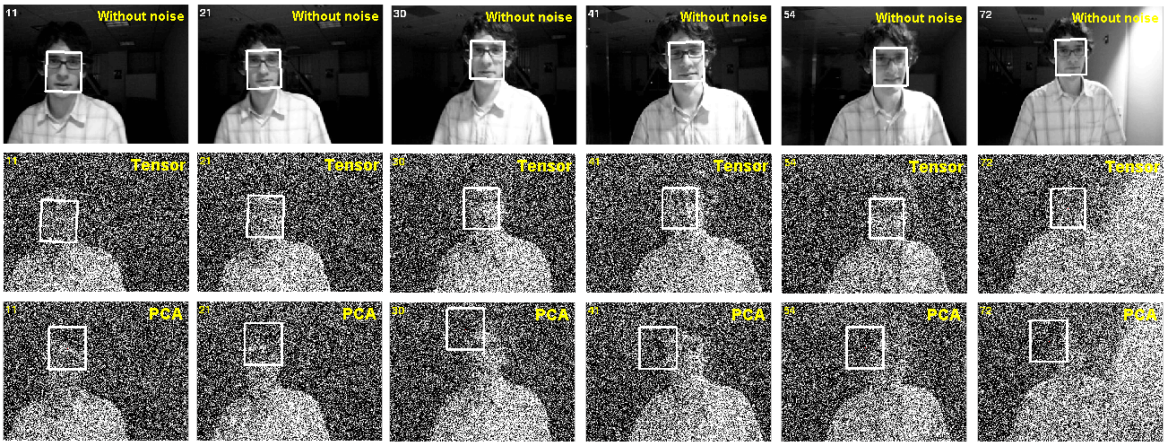

Fig. 7. **The tracking results of *IRTSA* and *IAVSL*, respectively, under the disturbance of a strong noise. Row 1 is the reference tracking result with no noise. Rows 2 and 3 correspond to the tracking results of *IRTSA* and *IAVSL*, respectively.**

Five experiments are conducted to demonstrate the claimed contributions of the proposed *IRTSA*. These five experiments are to compare tracking results of *IRTSA* with those of a state-of-the-art image-as-vector subspace learning based tracking algorithm [27], referred here as *IAVSL* in this paper, in different scenarios including noise disturbance, scene blurring, small object tracking, object pose variation, and occlusion. *IAVSL* is a representative image-as-vector linear subspace learning algorithm which incrementally learns a low dimensional eigenspace representation of the object appearance by the online PCA. Compared with most existing tracking algorithms, based on constructing an invariant object appearance representation, *IAVSL* is able to online track appearance changes of the object, resulting in a better tracking result. In contrast to image-as-vector *IAVSL*, the proposed *IRTSA* relies on image-as-matrix tensor-based subspace analysis to reflect the appearance changes of an object. Consequently, it is very significant to make a comparison between *IAVSL* and *IRTSA*. Moreover, the parameter settings for the comparing methods is conducted to make them perform best simultaneously.

The first experiment is conducted to evaluate the performances of the two subspace analysis based tracking techniques—*IAVSL* and *IRTSA* on investigating their tracking capabilities under the disturbance of strong noise. The video used in this experiment is obtained by manually adding Gaussian random noise to Video 1. The process of adding the noise is formulated as: $I'(x,y)=\mathcal{G}\left(I(x,y)+s\cdot Z\right)$, where $I(x,y)$ denotes the original pixel value, $I'(x,y)$ represents the pixel value after adding noise, Z follows the standard normal distribution $\mathcal{N}(0,1)$, s is a scaling

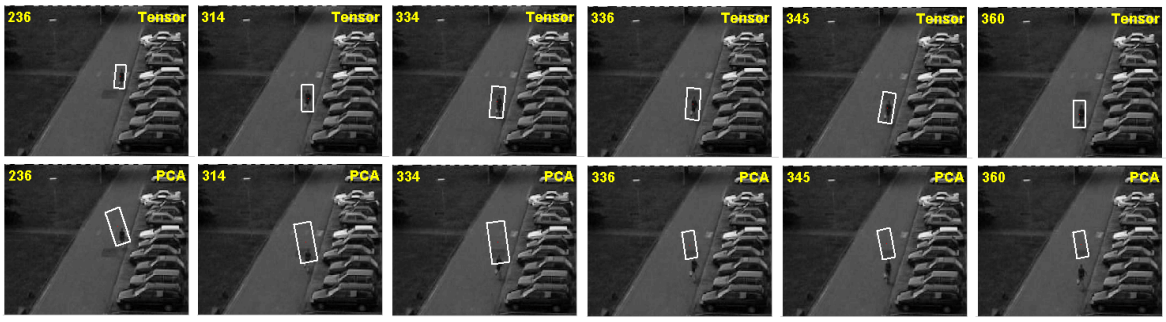

Fig. 8. **Tracking results of *IRTSA* and *IAVSL*, respectively, in the scenarios of small object and blurring scenes. Rows 1 and 2 correspond to *IRTSA* and *IAVSL*, respectively.**

factor controlling the amplitude of the noise, and the function $\mathcal{G}(\cdot)$ is defined as:

$$\mathcal{G}(x) = \begin{cases} 0 & x < 0 \\ 255 & x > 255 \\ [x] & 0 \leq x \leq 255 \end{cases} \tag{18}$$

where $[x]$ stands for the floor of the element x. In this experiment, s is set as 200. R_1, R_2 and R_3 in *IRTSA* are assigned as 3,3 and 5, respectively. For *IAVSL*, 5 eigenvectors are maintained during the tracking, and the remaining eigenvectors are discarded at each subspace updating. The final tracking results of *IRTSA* and *IAVSL* are shown in Fig. 7. For a better visualization, we just show the tracking results of six representative frames 11, 21, 30, 41, 54 and 72. In Fig. 7, the first row corresponds to the tracking results of the reference frames without noise using *IRTSA*. The remaining two rows are for the tracking results of *IRTSA* and *IAVSL*, respectively, under the disturbance of the noise. From Fig. 7, we see that the proposed tracking algorithm exhibits a robust tracking result while *IAVSL* fails to track the face under the disturbance of strong noise. This is due to the fact that since the spatial correlation information is ignored in *IAVSL*, the noise disturbance substantially changes the vector eigenspace representation of the object's appearance. In comparison, *IRTSA* relies on a robust tensor-based eigenspace model which makes full use of the spatio-temporal distribution information of the image ensembles in the three modes. Consequently, *IRTSA* has a strong error-tolerating capability.

The second experiment aims to compare the tracking performance of *IRTSA* with that of *IAVSL* in handling scene blurring and small object scenarios using Video 2. R_1, R_2 and R_3 in *IRTSA* are set as 5, 5 and 8, respectively. For *IAVSL*, 16 eigenvectors are maintained during the tracking, and the remaining eigenvectors are discarded at each subspace updating. We show the final tracking

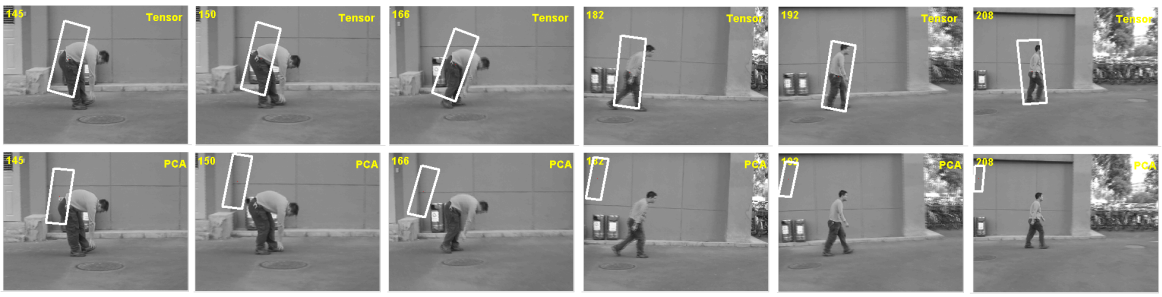

Fig. 9. **The tracking results of *IRTSA* and *IAVSL* in the scenarios of drastic pose change. Rows 1 and 2 correspond to *IRTSA* and *IAVSL*, respectively.**

results for *IRTSA* and *IAVSL* in Fig. 8, where the first and the second rows correspond to the performances of *IRTSA* and *IAVSL*, respectively, in which six representative frames (236, 314, 334, 336, 345 and 360) of the video stream are shown. Clearly, *IRTSA* succeeds in tracking the moving object while *IAVSL* fails. The reasons are explained as follows. *IRTSA* takes an image as a matrix, in comparison with the image-as-vector representation in *IAVSL*. Consequently, *IRTSA* makes a more compact object representation capable of reducing potentially substantial spatio-temporal redundancy of the image ensembles while *IAVSL* must solve for a high-dimensional data learning problem. This becomes particularly true for tracking a small object and/or with a blurring scene; here the spatial correlation information of the object's appearance is critical. Due to this loss of the spatial correlation information, *IAVSL* fails to track the object in these scenarios.

The third experiment is for a comparison between *IRTSA* and *IAVSL* in the scenarios of pose variation using Video 3. In this experiment, R_1, R_2 and R_3 are assigned as 8,8 and 10, respectively. For *IAVSL*, 16 eigenvectors are maintained during the tracking, and the remaining eigenvectors are discarded at each subspace updating. The final tracking results are demonstrated in Fig. 9, where rows 1 and 2 correspond to *IRTSA* and *IAVSL*, respectively, in which six representative frames (145, 150, 166, 182, 192, and 208) of the video stream are shown. From Fig. 9, it is clear that *IRTSA* is capable of tracking the object successfully even with a drastic pose and motion change while *IAVSL* gets lost in tracking the object after this drastic pose and motion change.

The fourth experiment is to compare the performances of the two methods *IRTSA* and *IAVSL* in handling partial occlusions using Video 4. In this experiment, R_1, R_2 and R_3 are set as 3,3 and

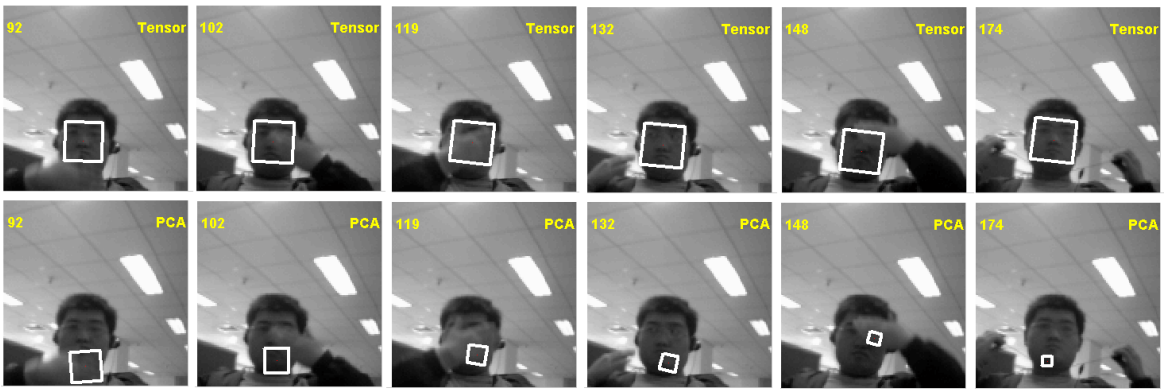

Fig. 10. **The tracking results of *IRTSA* and *IAVSL* in the scenarios of partial occlusions. Rows 1 and 2 show the tracking results of *IRTSA* and *IAVSL*, respectively.**

TABLE II

COMPARISON BETWEEN *IRTSA* AND *IAVSL* IN THE TRACKING MEAN LOCALIZATION DEVIATION WITH THE GROUND TRUTH. EXP k CORRESPONDS TO EXPERIMENT k $(1 \leq k \leq 4)$, AND THE LOCALIZATION DEVIATION IS MEASURED IN PIXELS. IT IS CLEAR THAT THE PROPOSED *IRTSA* PERFORMS MUCH BETTER THAN *IAVSL* .

Method \ Exp	Exp 1	Exp 2	Exp 3	Exp 4
IRTSA	5.12	2.54	3.26	2.52
IAVSL	31.71	28.65	77.19	28.61

5, respectively. For *IAVSL*, 10 eigenvectors are maintained during the tracking, and the remaining eigenvectors are discarded at each subspace updating. The final tracking results are demonstrated in Fig. 10, where rows 1 and 2 are the performance results of *IRTSA* and *IAVSL*, respectively, in which six representative frames (92, 102, 119, 132, 148 and 174) of the video stream are shown. From Fig. 10, we see that *IRTSA* is capable of tracking the object all the time even though the object is occluded partially from time to time in a poor lighting condition. On the other hand, *IAVSL* gets completely lost in tracking the object.

From the results in the third and the fourth experiments, we note that *IRTSA* is robust to pose variation and occlusion. The reason is that the dominant subspace information of the three modes is incorporated into *IRTSA*. Even if the subspace information of some modes is partially lost or drastically varies, *IRTSA* is capable of recovering the information using the cues of the subspace information from other modes.

Since there are no benchmark databases in the first four experiments, we have to provide a

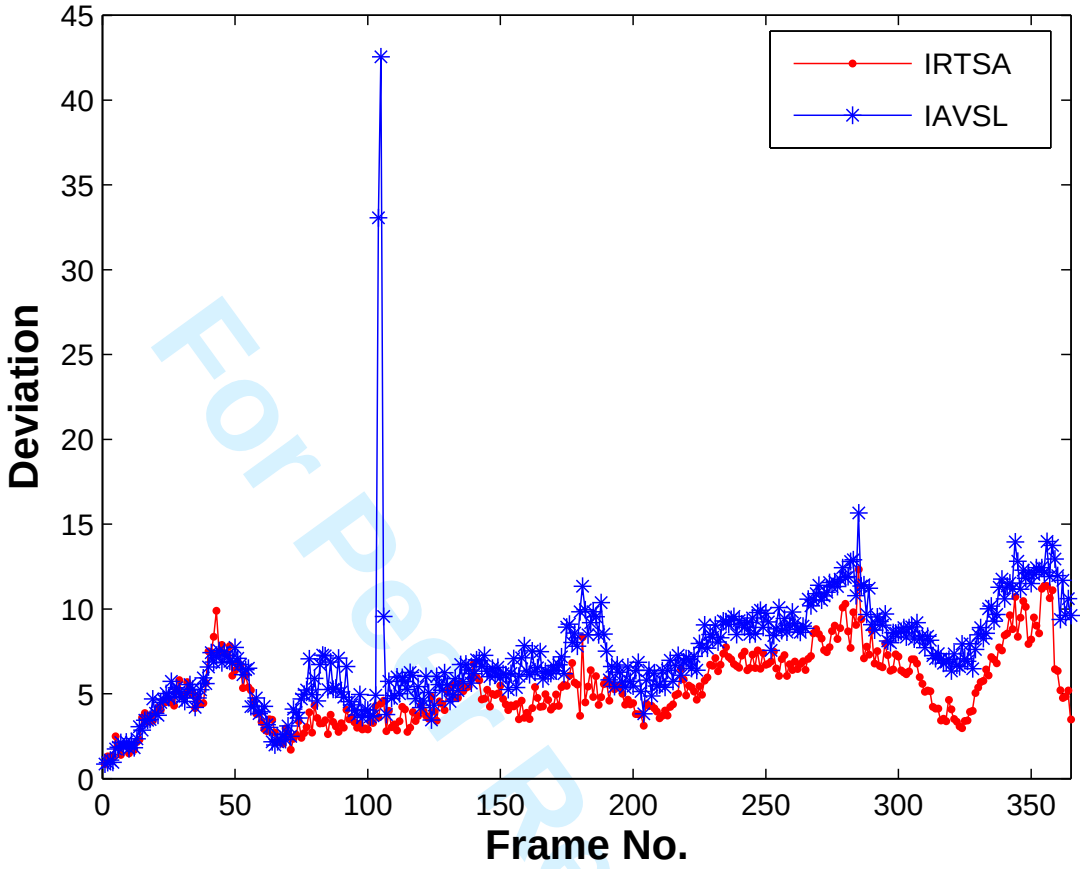

Fig. 11. **The quantitative comparison between *IRTSA* and *IAVSL* over the benchmark Video 5. The x-axis corresponds to the frame number while the y-axis is associated with the average location deviation between the validation points and the benchmark points.**

quantitative comparison between *IRTSA* and *IAVSL* using some representative frames. The object center locations in the representative frames used by the above four experiments are labeled manually as the ground truth. Thus, we can quantitatively evaluate the tracking performances of *IRTSA* and *IAVSL* by computing their corresponding pixel-based mean localization deviations between tracking results and the ground truth. The less the deviation, the higher the localization accuracy. The final comparison results are listed in Table II. From Table II, we see that the object localization accuracy of *IRTSA* is much higher than that of *IAVSL*.

The last experiment is to provide a quantitative comparison between *IRTSA* and *IAVSL* over the benchmark database (namely Video 5). In this experiment, R_1, R_2 and R_3 are set as 8, 8 and 8, respectively. For *IAVSL*, 13 eigenvectors are maintained during the tracking, and the remaining eigenvectors are discarded at each subspace updating. During the tracking, seven validation

Fig. 12. **The tracking results of *IRTSA* and *IAVSL* over two representative frames (106 and 107) from the benchmark Video 5. Rows 1 and 2 correspond to the tracking results of *IRTSA* and *IAVSL*, respectively.**

points, corresponding to the seven benchmark points, are obtained according to the object's affine motion parameters at each frame. In this way, we can use the average location deviation between the validation points and the benchmark ones to evaluate the tracking performance. The quantitative evaluation results are demonstrated in Fig. 11, where the dot-marked curve and the star-marked one correspond to *IRTSA* and *IAVSL*, respectively. From Fig. 11, it is clear that the average location deviation of *IRTSA* is lower than that of *IAVSL*. For a better visualization, we just show the final tracking results of two representative frames (106 and 107) with very high location deviations (see the highest point in Fig. 11) are shown in Fig. 12, where rows 1 and 2 are the performance results of *IRTSA* and *IAVSL*, respectively. From Fig. 12, we see that *IRTSA* performs better than *IAVSL* in the case of occlusions.

In summary, we observe that *IRTSA* outperforms *IAVSL* in the scenarios of noise disturbance, blurring scenes, small objects, drastic object pose change, and occlusions. Consequently, *IRTSA* is an effective online tensor-based subspace learning algorithm which performs well in modeling

appearance changes of an object in many complex scenarios.

B. Experimental evaluations of the application to foreground segmentation

In order to evaluate the performance of the proposed foreground segmentation application, four videos are used in the experiments. The first two videos consist of 8-bit grayscale images while the last two videos are composed of 24-bit color images. In the first video (selected from PETS2001[2]), a person and vehicles enter or leave a bright road scene. In the second video, three persons are walking in a scene containing a building wall, two lightly swaying trees, two cars and so on. The occlusion event, in which these three persons are overlapped, takes place in the middle of the video stream. In the third video, two cars are moving in a dark and blurry traffic scene. In the last video (selected from CAVIAR[3]), several people are walking along a corridor. They come into or leave the corridor from time to time. For the tensor-based eigenspace representation, the settings of the ranks R_1, R_2 and R_3 in *IRTSA* are obtained from the experiments. The forgetting factor λ in R-SVD is set as 0.96. The tensor-based subspace is updated every three frames.

Four experiments are conducted to demonstrate the claimed contributions of the proposed *IRTSA*. The first two experiments are performed to evaluate the foreground segmentation performances of the two subspace analysis based foreground segmentation techniques—the one proposed in [31] (referred here as IRSL) and the proposed *IRTSA-GBM* using videos 1 and 2, respectively. The last two experiments are performed to evaluate the foreground segmentation performances of the algorithm developed in [35] (referred here as DCRF) and the proposed *IRTSA-CBM* using videos 3 and 4, respectively. IRSL [31] is a representative image-as-vector linear subspace learning algorithm which incrementally learns a low dimensional eigenspace representation of a real scene by online PCA. However, it is only available for modeling grayscale images. On the other hand, DCRF [35] employs the dynamic conditional random field to model the spatio-temporal statistics of the pixels from color images. It has been proven in the literature that IRSL and DCRF are able to obtain a visually feasible foreground segmentation results. Thus, it is very significant for the proposed *IRTSA-GBM* and *IRTSA-CBM* to make a comparison with them. Furthermore, the parameter settings for the comparing methods is conducted to make them perform best simultaneously.

[2]http://www.cvg.cs.rdg.ac.uk/slides/pets.html

[3]http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/

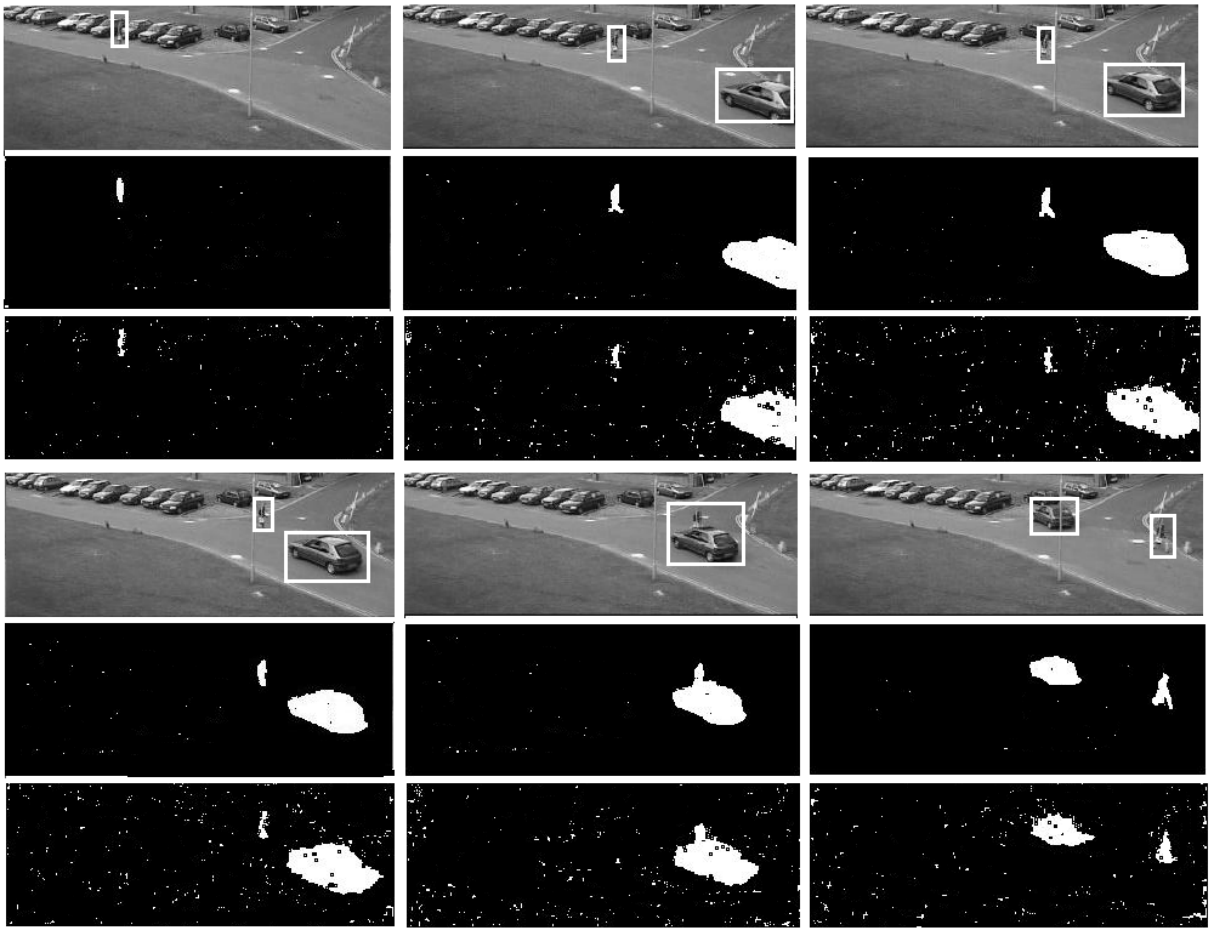

Fig. 13. **The foreground segmentation results of *IRTSA-GBM* and IRSL [31] using the first video. In rows 1 and 4, the moving regions are highlighted by white boxes. Rows 2 and 5 correspond to *IRTSA-GBM* while rows 3 and 6 are associated with IRSL.**

In the first experiment, R_1, R_2 and R_3 in *IRTSA-GBM* are assigned as 10, 10, and 10, respectively. The scaling factor σ in *IRTSA-GBM* is set as 15. The threshold T_{gray} is chosen as 0.72. The learning rate factor α^* is assigned as 0.01. For IRSL [31], the PCA dimensionality $p = 25$, the update rate $\alpha = 0.96$, and the coefficient $\beta = 11$. The final foreground segmentation results are shown in Fig. 13, where the second and the fifth rows correspond to *IRTSA-GBM* while the third and the sixth ones are associated with the IRSL. For a better visualization, we just show the segmentation results of six representative frames 2, 43, 68, 86, 117, and 154.

In the second experiment, R_1, R_2 and R_3 in *IRTSA-GBM* are assigned as 15, 15, and 15, respectively. The scaling factor σ in *IRTSA-GBM* is set as 20. The threshold T_{gray} is chosen as 0.73. The learning rate factor α^* is assigned as 0.01. For IRSL, the PCA dimensionality $p = 26$,

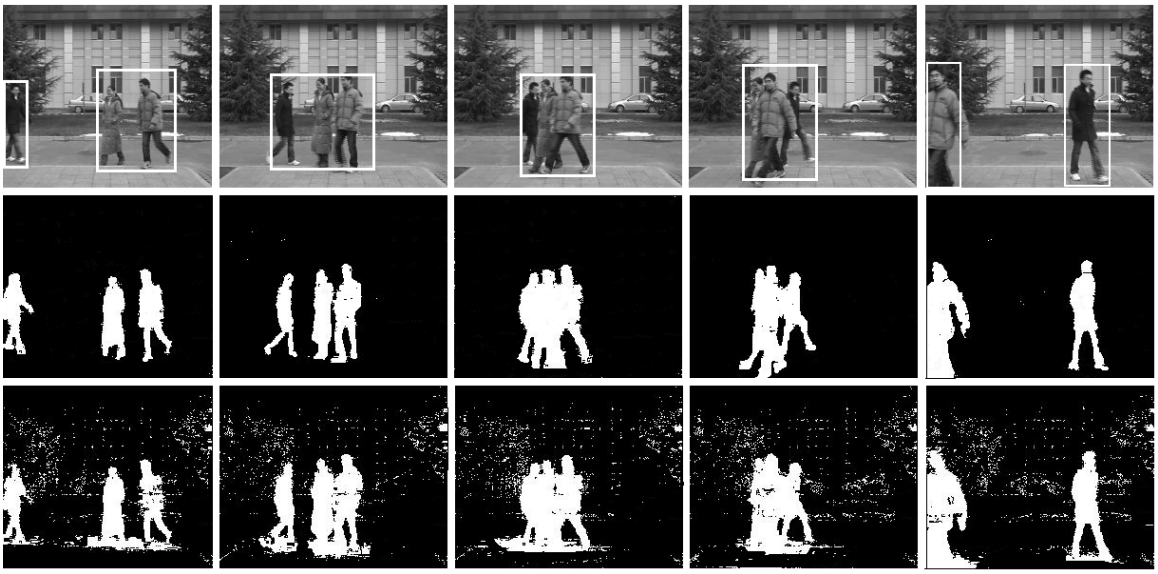

Fig. 14. **The foreground segmentation results of *IRTSA-GBM* and IRSL [31] using the second video. In row 1, the moving regions are highlighted by white boxes. Rows 2 and 3 correspond to *IRTSA-GBM* and IRSL, respectively.**

Fig. 15. **The foreground segmentation results of *IRTSA-CBM* and DCRF [35] using the third video. In row 1, the moving regions are highlighted by white boxes. Rows 2 and 3 correspond to *IRTSA-CBM* and DCRF, respectively.**

the update rate $\alpha = 0.95$, and the coefficient $\beta = 9$. The final foreground segmentation results are shown in Fig. 14, where the second row corresponds to *IRTSA-GBM* while the third one is associated with IRSL. The segmentation results of five representative frames 7, 26, 32, 44, and 72 are displayed.

From the results in the first and the second experiments, we note that *IRTSA-GBM* demonstrates a better foreground segmentation result than IRSL. Specifically, *IRTSA-GBM*'s segmen-

Fig. 16. **The foreground segmentation results of *IRTSA-CBM* and DCRF [35] using the fourth video. In row 1, the moving regions are highlighted by white boxes. Rows 2 and 3 correspond to *IRTSA-CBM* and DCRF, respectively.**

tation results are cleaner, more connected, and less noisy, and more shadow-free. This is due to the fact that since the spatial correlation information is ignored in IRSL, the global or local variations of a scene substantially change the vector eigenspace representation of IRSL.

In the third experiment, (R_1^r, R_2^r, R_3^r), (R_1^g, R_2^g, R_3^g), and (R_1^s, R_2^s, R_3^s), corresponding to three components in the (r, g, s) color space, are respectively assigned as $(11, 11, 11)$, $(11, 11, 11)$ and $(15, 15, 15)$. The learning rate factors α_r, α_g and α_s are all assigned as 0.01. The scaling factors σ_r, σ_g and σ_s in (14) are set as 0.092, 0.092, and 16, respectively. The threshold T_{color} is chosen as 0.51. DCRF is initialized with $\gamma = 3.5$, $\tau = 4.5$, the 24-pixel spatial neighborhood, and the 81-pixel temporal neighborhood. The final foreground segmentation results are demonstrated in Fig. 15, where rows 2 and 3 correspond to *IRTSA-CBM* and DCRF, respectively, in which five representative frames (3, 20, 30, 34, and 38) of the video stream are shown.

In the fourth experiment, (R_1^r, R_2^r, R_3^r), (R_1^g, R_2^g, R_3^g), and (R_1^s, R_2^s, R_3^s), corresponding to the three components in the rgs color space, are respectively assigned as $(9, 9, 9)$, $(9, 9, 9)$, and $(13, 13, 13)$. The learning rate factors α_r, α_g, and α_s are all assigned as 0.01. The scaling factors σ_r, σ_g and σ_s in (14) are set as 0.1, 0.1, and 20, respectively. The threshold T_{color} is chosen as 0.53. DCRF is initialized with $\gamma = 3$, $\tau = 4.2$, the 24-pixel spatial neighborhood, and the 81-pixel temporal neighborhood. The final foreground segmentation results are demonstrated in Fig. 16, where rows 2 and 3 correspond to *IRTSA-CBM* and DCRF, respectively, in which five representative frames (296, 312, 472, 790, and 814) of the video stream are shown.

From the results in the third and the fourth experiments, we note that *IRTSA-CBM* secures a

better foreground segmentation result than DCRF. Compared with DCRF, *IRTSA-CBM* is able to fully exploit the spatio-temporal redundancies within the image ensembles by image-as-matrix tensor-based subspace analysis, resulting in a more robust foreground segmentation result.

In summary, we observe that *IRTSA-GBM* and *IRTSA-CBM* perform well in complex scenarios. Consequently, *IRTSA-GBM* and *IRTSA-CBM* are two effective models for foreground segmentation.

VII. Conclusion

In this paper, we have developed an appearance model based on an incremental rank-(R_1, R_2, R_3) tensor-based subspace learning algorithm (referred as *IRTSA*), which models the appearance of an object or a scene by incrementally learning a low-order tensor-based eigenspace representation through adaptively updating the sample mean and eigenbasis. Compared with existing image-as-vector approach to image modeling, the developed *IRTSA* better captures the intrinsic spatio-temporal characteristics of object appearance. On the other hand, the *IRTSA*) works online, resulting in a much lower computational cost against the traditional offline approaches to tensor decomposition. Based on *IRTSA*, two applications to tracking and foreground segmentation are developed. The main contributions of these two applications are three-fold. (1) A novel online tensor-based subspace learning algorithm, which enables subspace analysis within a multilinear framework, is proposed to effectively model the appearance of an object. (2) A novel likelihood evaluation function, based on the tensor reconstruction error norm, is developed to measure the similarity between the test image and the learned tensor-based subspace model during the tracking. (3) Two novel background models (*IRTSA-GBM* and *IRTSA-CBM*), based on the tensor reconstruction error norm, is developed to measure the similarity between the test image and the learned tensor-based subspace model. Compared with the image-as-vector tracking methods in the literature, the proposed image-as-matrix tracking application is more robust to noise or low quality images, occlusion, scene blurring, small object, and object pose variation. Furthermore, the proposed foreground segmentation application exhibits a better foreground segmentation result than the existing foreground segmentation methods in the literature. Experimental results have demonstrated the robustness and promise of the proposed *IRTSA* and its applications (to tracking and foreground segmentation).

REFERENCES

[1] X. Li, W. Hu, Z. Zhang, X. Zhang, and G. Luo, "Robust visual tracking based on incremental tensor subspace learning," to appear in ICCV'07.

[2] K. Nummiaroa, E. Koller-Meierb, and L.V. Gool, "An adaptive color-based particle filter," *Image and Vision Computing,* 21:pp.99-110, 2003.

[3] P. Perez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," in *Proc. ECCV,* pp.661-675, 2002.

[4] A. Elgammal, R. Duraiswami, D. Harwood, and L.S. Davis, "Background and foreground modeling using nonparametric kernel density estimation for visual surveillance," *Proceedings of IEEE,* Iss. 7, Vol. 90, pp.1151-1163, 2002.

[5] C. Yang, R. Duraiswami, and L.S. Davis, "Efficient mean-shift tracking via a new similarity measure," in *Proc. CVPR,* pp.176-183, 2005.

[6] S. J. McKennaa, Y. Rajab, and S. Gong, "Tracking colour objects using adaptive mixture models," *Image and Vision Computing,* 17:pp.225-231, 1999.

[7] B. Han, and L. Davis, "On-line density-based appearance modeling for object tracking.," in *Proc. ICCV,* pp.1492-1499, 2005.

[8] Y. Wu and T.S. Huang, "Robust visual tracking by integrating multiple cues based on co-inference learning," *IJCV,* Iss. 1, Vol. 58, pp.55-71, 2004.

[9] S. Khan and M. Shah, "Tracking people in presence of occlusion," in *Proc. ACCV,* pp.263-266, 2000.

[10] H. Wang, D. Suter, and K. Schindler, "Effective appearance model and similarity measure for particle filtering and visual tracking," in *Proc. ECCV,* pp. 606-618, 2006.

[11] G. Hager and P. Belhumeur, "Real-time tracking of image regions with changes in geometry and illumination," in *Proc. CVPR'96,* pp.430-410, 1996.

[12] M. J. Black and A. D. Jepson, "Eigentracking: Robust matching and tracking of articulated objects using view-based representation," in *Proc. ECCV'96,* pp.329-342, 1996.

[13] M. Isard and A. Blake, "Contour tracking by stochastic propagation of conditional density," in *Proc. ECCV'96,* Vol. 2, pp.343-356, 1996.

[14] M. J. Black, D. J. Fleet, and Y. Yacoob, "A framework for modeling appearance change in image sequence," in *Proc. ICCV'98,* pp.660-667, 1998.

[15] A. D. Jepson, D. J. Fleet, and T. F. El-Maraghi, "Robust Online Appearance Models for Visual Tracking," in *Proc. CVPR'01,* Vol. 1, pp.415-422, 2001.

[16] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Trans. on PAMI.,* Vol. 25,

pp.564-577 , May 2003.

[17] S. K. Zhou, R. Chellappa, and B. Moghaddam, "Visual Tracking and Recognition Using Appearance-Adaptive Models in Particle Filters," *IEEE Trans. on Image Processing,* Vol. 13, pp.1491-1506 , November 2004.

[18] T. Yu and Y. Wu, "Differential Tracking based on Spatial-Appearance Model(SAM)," in *Proc. CVPR'06,* Vol. 1, pp.720-727, June 2006.

[19] J. Li, S. K. Zhou, and R. Chellappa, "Appearance Modeling under Geometric Context," in *Proc. ICCV'05,* Vol. 2, pp.1252-1259, 2005.

[20] S. Wong, K. K. Wong, and R. Cipolla, "Robust Appearance-based Tracking using a sparse Bayesian classifier," in *Proc. ICPR'06,* Vol. 3, pp.47-50, 2006.

[21] K. Lee and D. Kriegman, "Online Learning of Probabilistic Appearance Manifolds for Video-based Recognition and Tracking," in *Proc. CVPR'05,* Vol. 1, pp.852-859 , 2005.

[22] H. Lim, V. I. Morariu3, O. I. Camps, and M. Sznaier1, "Dynamic Appearance Modeling for Human Tracking," in *Proc. CVPR'06,* Vol. 1, pp.751-757, 2006.

[23] J. Ho, K. Lee, M. Yang, and D. Kriegman, "Visual Tracking Using Learned Linear Subspaces ," in *Proc. CVPR'04,* Vol. 1, pp.782-789, 2004.

[24] Y. Li, L. Xu, J. Morphett, and R. Jacobs, "On Incremental and Robust Subspace Learning," *Pattern Recognition,* 37(7), pp. 1509-1518, 2004.

[25] D. Skocaj, A. Leonardis, "Weighted and Robust Incremental Method for Subspace Learning," in *Proc. ICCV'03,* pp.1494-1501, 2003.

[26] A. Levy and M. Lindenbaum, "Sequential Karhunen-Loeve Basis Extraction and Its Application to Images," *IEEE Trans. on Image Processing,* Vol. 9, pp.1371-1374, 2000.

[27] J. Limy, D. Ross, R. Lin, and M. Yang, "Incremental Learning for Visual Tracking," *NIPS'04,* pp.793-800, MIT Press, 2005.

[28] 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 Journal of Matrix Analysis and Applications,* Vol. 21, Iss. 4, pp.1324-1342, 2000.

[29] C. Stauffer, and W.E.L. Grimson, "Adaptive Background Mixture Models for Real-Time Tracking," in *Proc. CVPR'99,* Vol. 2, 1999.

[30] I. Haritaoglu, D. Harwood, and L.S. Davis, "W^4: Real-Time Surveillance of People and Their Activities," *IEEE Trans. PAMI.,* Vol. 22, Iss. 8, pp.809-830, 2000.

[31] Y. Li, "On Incremental and Robust Subspace Learning," *Pattern Recognition,* Vol. 37, Iss. 7, pp.1509-1518, 2004.

[32] Y. Sheikh, and M. Shah, "Bayesian Object Detection in Dynamic Scenes," in *Proc. CVPR'05,* Vol. 1, pp.74-79,

2005.

[33] J. Cezar Silveira Jacques, C. Rosito Jung, and S.R. Musse, "A Background Subtraction Model Adapted to Illumination Changes," in *Proc. ICIP'06,* pp.1817-1820, 2006.

[34] Y. Wang, T. Tan, K.F. Loe, and J.K. Wu, "A Probabilistic Approach for Foreground and Shadow Segmentation in Monocular Image Sequences," *Pattern Recognition,* Vol. 38, Iss. 11, pp.1937-1946, Nov. 2005.

[35] Y. Wang, K. Loe, and J. Wu, "A Dynamic Conditional Random Field Model for Foreground and Shadow Segmentation ," *IEEE Trans. PAMI.,* Vol. 28, Iss. 2, pp.279-289, 2006.

[36] J. Yang, D. Zhang, A. F. Frangi, and J. Yang, "Two-dimensional PCA: A New Approach to Appearance-based Face Representation and Recognition," in *IEEE Trans. PAMI.,* Vol. 26, Iss. 1, pp.131-137, Jan. 2004.

[37] J. Ye, R. Janardan, and Q. Li, "Two-Dimensional Linear Discriminant Analysis," *NIPS'04,* pp.1569-1576, MIT Press,2004.

[38] J. Ye, "Generalized low rank approximations of matrices," *ICML'04,* July 2004.

[39] J. Ye, R. Janardan, and Q. Li, "GPCA: An Efficient Dimension Reduction Scheme for Image Compression and Retrieval," *ACM KDD'04,* pp.354-363, August 2004.

[40] H. Wang and N. Ahuja, "Rank-R Approximation of Tensors Using Image-as-matrix Representation ," in *Proc. CVPR'05,* Vol. 2, pp.346-353, 2005.

[41] S. Yan, D. Xu, Q. Yang, L. Zhang, X. Tang, and H. Zhang, "Discriminant analysis with tensor representation," in *Proc. CVPR'05,* Vol. 1, pp.526-532, June 2005.

[42] M. A. O. Vasilescu and D. Terzopoulos, "Multilinear Subspace Analysis of Image Ensembles," in *Proc. CVPR'03,* Vol. 2, pp.93-99, June 2003.

[43] M. A. O. Vasilescu and D. Terzopoulos, "Multilinear Subspace Analysis of Image Ensembles: TensorFaces," in *Proc. ECCV'02,* pp.447-460, May 2002.

[44] X. He, D. Cai, and P. Niyogi, "Tensor Subspace Analysis," *NIPS'05,* Dec. 2005.

[45] H. Wang, S. Yan, T. Huang, and X. Tang, "A Convergent Solution to Tensor Subspace Learning," in *Proc. IJCAI'07,* 2007.

[46] J. Sun, D. Tao, and C. Faloutsos, "Beyond Streams and Graphs: Dynamic Tensor Analysis," *ACM KDD'06,* Aug. 2006.

[47] J. Sun, S. Papadimitriou, and P. S. Yu, "Window-based Tensor Analysis on High-dimensional and Multi-aspect Streams," in *Proc. ICDM'06,* Dec. 2006.

[48] Y. Tian, M. Lu, and A. Hampapur, "Robust and Efficient Foreground Analysis for Real-Time Video Surveillance," in *Proc. CVPR'05,* Vol. 1, pp.1182-1187, 2005.

Summary of Changes

Compared with the ICCV'07 paper, the changes in this manuscript are briefly summarized as follows:

1. This manuscript focuses on object/scene appearance modeling while the ICCV'07 paper deals with appearance-based tracking. Specifically, the ICCV'07 work is just one of the two IRTSA-based applications in this manuscript.
2. The proposed IRTSA is used for foreground segmentation in this manuscript. We construct two IRTSA-based background models (namely IRTSA-GBM and IRTSA-CBM) for grayscale and color images, respectively. In these two models, the spatiotemporal characteristics of the scene are well captured, leading to a robust foreground segmentation result. (See Sections V and VI-B of this manuscript.)
3. Theoretic analysis or empirical evaluation of some representative appearance models (like GMM [29], IRSL [31], DCRF [35] etc.) against the IRTSA-based ones (IRTSA-GBM and IRTSA-CBM) is given in this manuscript. (See Sections I, II, and VI of this manuscript.)
4. The quantitative complexity analysises of IRTSA and other related work are given in this manuscript. (See Section III-E of this manuscript.)
5. One quantitative comparing experiment is supplemented in this manuscript. (See the last experiment in Section VI-A of this manuscript.)

You can see the ICCV'07 paper in the following pages.

Robust Visual Tracking Based on Incremental Tensor Subspace Learning

Xi Li†, Weiming Hu†
†National Laboratory of Pattern Recognition,
Institute of Automation, CAS, Beijing, China

Zhongfei Zhang‡, Xiaoqin Zhang†, Guan Luo†
‡State University of New York,
Binghamton, NY 13902, USA

Abstract

Most existing subspace analysis-based tracking algorithms utilize a flattened vector to represent a target, resulting in a high dimensional data learning problem. Recently, subspace analysis is incorporated into the multilinear framework which offline constructs a representation of image ensembles using high-order tensors. This reduces spatio-temporal redundancies substantially, whereas the computational and memory cost is high. In this paper, we present an effective online tensor subspace learning algorithm which models the appearance changes of a target by incrementally learning a low-order tensor eigenspace representation through adaptively updating the sample mean and eigenbasis. Tracking then is led by the state inference within the framework in which a particle filter is used for propagating sample distributions over the time. A novel likelihood function, based on the tensor reconstruction error norm, is developed to measure the similarity between the test image and the learned tensor subspace model during the tracking. Theoretic analysis and experimental evaluations against a state-of-the-art method demonstrate the promise and effectiveness of this algorithm.

1. Introduction

For visual tracking, handling appearance variations of a target is a fundamental and challenging task. In general, there are two types of appearance variations: intrinsic and extrinsic. Pose variation and/or shape deformation of a target object are considered as the intrinsic appearance variations while the extrinsic variations are due to the changes resulting from different illumination, camera motion, camera viewpoint, and occlusion. Consequently, effectively modeling such appearance variations plays a critical role in visual tracking.

In recent years, much work has been done in visual tracking based on modeling the appearance of a target. Hager and Belhumeur [1] propose a tracking algorithm which uses an extended gradient-based optical flow method to handle object tracking under varying illumination conditions. They construct a set of illumination basis for a fixed pose with illumination change. Black *et al.* [2] present a subspace learning based tracking algorithm with the subspace constancy assumption. A pre-trained, view-based eigenbasis representation is used for modeling appearance variations. However, the algorithm does not work well in the clutter with a large lighting change due to the subspace constancy assumption. In [3], curves or splines are exploited to represent the appearance of a target to develop the Condensation algorithm for contour tracking. Due to the simplistic representation scheme, the algorithm is unable to handle the pose or illumination change, resulting in a usually unsuccessful tracking result under a varying lighting condition. Black *et al.* [4] employ a mixture model to represent and recover the appearance changes in consecutive frames. Jepson *et al.* [5] develop a more elaborate mixture model with an online EM algorithm to explicitly model the appearance change during tracking. Zhou *et al.* [6] embed appearance-adaptive models into a particle filter to achieve a robust visual tracking. Yu *et al.* [7] propose a spatial-appearance model which captures non-rigid appearance variations and recovers all motion parameters efficiently. Li *et al.* [8] use a generalized geometric transform to handle the deformation, articulation, and occlusion of appearance. Wong *et al.* [9] present a robust appearance-based tracking algorithm using an online-updating sparse Bayesian classifier. Lee and Kriegman [10] present an online learning algorithm to incrementally learn a generic appearance model from the video. Lim *et al.* [11] present a human tracking framework using robust system dynamics identification and nonlinear dimensiona reduction techniques. Ho *et al.* [12] present a visual tracking algorithm based on linear subspace learning. Li *et al.* [13] propose an incremental PCA algorithm for subspace learning. In [14], a weighted incremental PCA algorithm for subspace learning is presented. Limy *et al.*[15] propose a generalized tracking framework based on the incremental image-as-vector subspace learning methods with a sample mean update. It is noted that all the above tracking methods are unable to fully exploit the spatial redundancies within the image ensembles. This is particularly true for those image-as-vector tracking techniques, as the local spatial information is almost lost. Consequently, the focus has been made on developing the image-as-matrix learning al-

Accepted by ICCV 2007

gorithms for effective subspace analysis. Yang *et al.* [16] develop a 2-dimensional PCA (2DPCA) for image representation. Based on the original image matrices, 2DPCA constructs an image covariance matrix whose eigenvectors are derived for image feature extraction. Ye *et al.* [17] present a learning method called 2-dimensional linear discriminant analysis (2DLDA). In [18], a novel algorithm, called GLRAM, is proposed for low rank approximations of a collection of matrices. In [19], Ye *et al.* present a new dimension reduction algorithm named GPCA, which constructs the matrix representation of images directly,

More recent work on modeling the appearance of a target focuses on using high-order tensors to construct a better representation of the target's appearance. In this case, the problem of modeling the appearance of a target is reduced to how to make tensor decomposition more accurate and efficient. Wang and Ahuja [20] propose a novel rank-R tensor approximation approach, which is designed to capture the spatio-temporal redundancies of tensors. In [21], an algorithm named Discriminant Analysis with Tensor Representation (DATER) is proposed. DATER is tensorized from the popular vector-based LDA algorithm. In [22, 23], the N-mode SVD, multilinear subspace analysis, is applied to constructing a compact representation of facial image ensembles factorized by different faces, expressions, viewpoints, and illuminations. He *et al.* [24] present a learning algorithm called Tensor Subspace Analysis (TSA), which learns a lower dimensional tensor subspace to characterize the intrinsic local geometric structure of the tensor space. In [25], Wang *et al.* give a convergent solution for general tensor-based subspace learning. Sun *et al.* [26] mine higher-order data streams using dynamic and streaming tensor analysis. Also in [27], Sun *et.al* present a window-based tensor analysis method for representing data streams over the time. All of these tensor-based algorithms share the same problem that they are not allowed for incremental subspace analysis for adaptively updating the sample mean and eigenbasis.

In this paper, we develop a tracking framework based on an incremental tensor subspace learning. The main contributions of the framework are as follows. First, the proposed framework does not need to know any prior knowledge of the object. A low dimensional eigenspace representation is learned online, and is updated incrementally over the time. The framework only assumes that the initialization of the object region is provided. Second, while the Condensation algorithm [3] is used for propagating the sample distributions over the time, we develop an effective probabilistic likelihood function based on the learned tensor eigenspace model. Third, while R-SVD [15, 28] is applied to update both the sample mean and eigenbasis online as new data arrive, an incremental multilinear subspace analysis is enabled to capture the appearance characteristics of the object during the tracking.

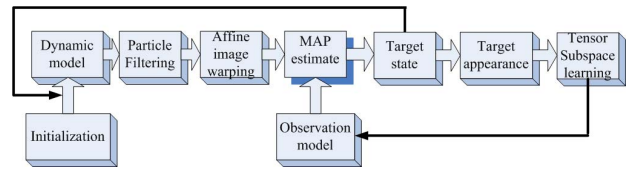

Figure 1. **The architecture of the proposed tracking framework**

2. The framework for visual tracking

2.1. Overview of the framework

The tracking framework includes two stages: (a) tensor subspace learning; and (b) Bayesian inference for visual tracking. In the first stage, a low dimensional tensor eigenspace model is learned online. The model uses the proposed incremental rank-(R_1, R_2, R_3) tensor subspace analysis (thus called *IRTSA*) to find the dominant projection subspaces of the 3-order tensors (image ensembles). In the second stage, the target locations in consecutive frames are estimated by the Bayesian state inference within the framework in which a particle filter is applied to propagate sample distributions over the time. These two stages are executed repeatedly as time progresses. Moreover, the framework has a strong adaptability in the sense that when new image data arrive, the tensor eigenspace model follows the updating online. The architecture of the framework is shown in Figure 1.

2.2. Dynamic tensor subspace analysis

Before we present the proposed online tensor subspace learning method, we first give a brief review of the related background as well as the introduction to the notations and symbols we use.

2.2.1 Multilinear algebra

The mathematical foundation of multilinear analysis is the tensor algebra. A tensor can be regarded as a multidimensional matrix. We denote an N-order tensor as $\mathcal{A} \in \mathcal{R}^{I_1 \times I_2 \times \ldots \times I_N}$, each element of which is represented as $a_{i_1 \cdots i_n \cdots i_N}$ for $1 \leq i_n \leq I_n$. In the tensor terminology, each dimension of a tensor is associated with a "*mode*". The mode-n unfolding matrix $\mathrm{A}_{(n)} \in \mathcal{R}^{I_n \times (\prod_{i \neq n} I_i)}$ of $\mathcal{A}$ consists of the I_n-dimensional mode-n vectors obtained by varying the nth-mode index i_n while keeping the other mode indices fixed. Namely, the column vectors of $\mathrm{A}_{(n)}$ are just the mode-n vectors. For a better understanding of the tensor unfolding, we take advantage of Figure 2 to explain the process of the unfolding. The inverse operation of the mode-n unfolding is the mode-n folding, which can restore the original tensor $\mathcal{A}$ from the mode-n unfolding matrix $\mathrm{A}_{(n)}$. The mode-n product of $\mathcal{A}$ and a matrix $\mathrm{U} \in \mathcal{R}^{J_n \times I_n}$ is denoted as $\mathcal{A} \times_n \mathrm{U} \in \mathcal{R}^{I_1 \times \ldots \times I_{n-1} \times J_n \times I_{n-1} \times \ldots \times I_N}$ whose entries are as follows:

$$(\mathcal{A} \times_n \mathrm{U})_{i_1 \cdots i_{n-1} j_n i_{n+1} \cdots i_N} = \sum_{i_n} a_{i_1 \ldots i_n \ldots i_N} u_{j_n i_n} \quad (1)$$

Accepted by ICCV 2007

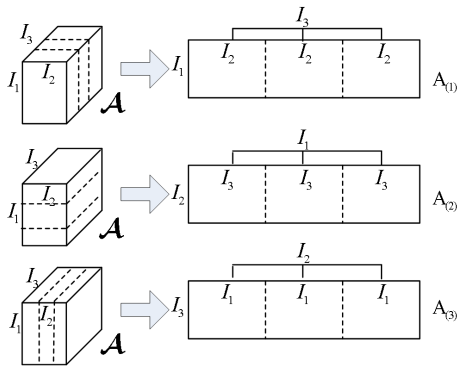

Figure 2. **Illustration of unfolding a (3-order) tensor.**

Given a tensor $\mathcal{A} \in \mathcal{R}^{I_1 \times I_2 \times \ldots \times I_N}$ and the matrices $\mathrm{C} \in \mathcal{R}^{J_n \times I_n}$, $\mathrm{D} \in \mathcal{R}^{K_n \times J_n}$, $\mathrm{E} \in \mathcal{R}^{J_m \times I_m} (n \neq m)$, the mode-$n$ product has the following properties:

1. $(\mathcal{A} \times_n \mathrm{C}) \times_m E = (\mathcal{A} \times_m \mathrm{E}) \times_n C = \mathcal{A} \times_n \mathrm{C} \times_m \mathrm{E}$
2. $(\mathcal{A} \times_n \mathrm{C}) \times_n \mathrm{D} = \mathcal{A} \times_n (\mathrm{D} \cdot \mathrm{C})$

The scalar product of two tensors $\mathcal{A}, \mathcal{B}$ is defined as:

$$\langle \mathcal{A}, \mathcal{B} \rangle = \sum_{i_1} \sum_{i_2} \cdots \sum_{i_N} a_{i_1 \ldots i_N} b_{i_1 \ldots i_N} \tag{2}$$

The Frobenius norm of $\mathcal{A} \in \mathcal{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is defined as: $\|\mathcal{A}\| = \sqrt{\langle \mathcal{A}, \mathcal{A} \rangle}$. The mode-$n$ rank R_n of $\mathcal{A}$ is defined as the dimension of the space generated by the mode-n vectors: $R_n = \mathrm{rank}(\mathrm{A}_{(n)})$. More details of the tensor algebra are given in [29].

2.2.2 Tensor decomposition

The Higher-Order Singular Value Decomposition (HOSVD) [22] is a generalized form of the conventional matrix singular value decomposition (SVD). An N-order tensor $\mathcal{A}$ is an N-dimensional matrix composed of N vector spaces. HOSVD seeks for N orthonormal matrices $\mathrm{U}^{(1)}, \ldots, \mathrm{U}^{(N)}$ which span these N spaces, respectively. Consequently, the tensor $\mathcal{A}$ can be decomposed as the following form:

$$\mathcal{A} = \mathcal{B} \times_1 \mathrm{U}^{(1)} \times_2 \mathrm{U}^{(2)} \cdots \times_N \mathrm{U}^{(N)} \tag{3}$$

where $\mathcal{B} = \mathcal{A} \times_1 \mathrm{U}^{(1)^T} \times_2 \mathrm{U}^{(2)^T} \cdots \times_N \mathrm{U}^{(N)^T}$ which denotes the core tensor controlling the interaction among the mode matrices $\mathrm{U}^{(1)}, \ldots, \mathrm{U}^{(N)}$. The orthonormal column vectors of $\mathrm{U}^{(n)}$ span the column space of the mode-n unfolding matrix $\mathrm{A}_{(n)}$ $(1 \leq n \leq N)$. In this way, we have the N-mode HOSVD algorithm [22] illustrated in Table 1.

In real applications, dimension reduction is necessary for a compact representation of tensors. In [29], Lathauwer *et al.* propose the best rank-$(R_1, R_2, \ldots, R_N)$ approximation algorithm for dimension reduction. The algorithm applies the Alternative Least Squares (ALS) to find the dominant projection subspaces. However, its computational cost is very expensive.

for n=1 to N

1. Compute the SVD of the mode-n unfolding matrix $\mathrm{A}_{(n)} = \widetilde{\mathrm{U}}_n \cdot \widetilde{\mathrm{D}}_n \cdot \widetilde{\mathrm{V}}_n^T$.
2. Set the mode matrix $\mathrm{U}^{(n)}$ as the orthonormal matrix $\widetilde{\mathrm{U}}_n$.

end

Compute the core tensor as:

$$\mathcal{B} = \mathcal{A} \times_1 \mathrm{U}^{(1)^T} \ldots \times_n \mathrm{U}^{(n)^T} \ldots \times_N \mathrm{U}^{(N)^T}$$

Table 1. **The N-mode HOSVD algorithm**

In the next two sections (2.2.3 and 2.2.4), we will discuss the proposed incremental rank-(R_1, R_2, R_3) tensor subspace analysis (*IRTSA*) method for 3-order tensors. *IRTSA* applies the online learning technique (R-SVD [15, 28]) to find the dominant projection subspaces of 3-order tensors.

2.2.3 Introduction to R-SVD

The classic R-SVD algorithm [28] efficiently computes the SVD of a dynamic matrix with newly added columns or rows, based on the existing SVD. Unfortunately, the R-SVD algorithm [28] is based on the zero mean assumption, leading to the failure of tracking subspace variabilities. Based on [28], [15] extends the R-SVD algorithm to compute the eigenbasis of a scatter matrix with the mean update. The details are described as follows.

Given a matrix $H = \{\mathrm{K}_1, \mathrm{K}_2, \ldots, \mathrm{K}_g\}$ and its column mean K, we let CVD(H) denote the SVD of the matrix $\{\mathrm{K}_1 - \mathrm{K}, \mathrm{K}_2 - \mathrm{K}, \ldots, \mathrm{K}_g - \mathrm{K}\}$. Given the column mean L_p of the existing data matrix $H_p = \{\mathrm{L}_1, \mathrm{L}_2, \ldots, \mathrm{L}_n\}$, CVD$(H_p) = U_p \Sigma_p V_p^T$, the column mean L_q of the new data matrix $F = \{\mathrm{L}_{n+1}, \mathrm{L}_{n+2}, \ldots, \mathrm{L}_{n+m}\}$, and the column mean L_e of the entire data matrix $H_e = (H_p \mid F)$, CVD$(H_e) = U_e \Sigma_e V_e^T$ can be determined as:

1. Compute $\mathrm{L}_e = \frac{n}{m+n} \mathrm{L}_p + \frac{m}{m+n} \mathrm{L}_q$;
2. Compute $\tilde{F} = \left(F - \mathrm{L}_q \mathbb{1}_{1 \times m} \mid \sqrt{\frac{mn}{m+n}} (\mathrm{L}_p - \mathrm{L}_q) \right)$, where $\mathbb{1}_{1 \times m}$ is $(\overbrace{1, 1, \ldots, 1}^{m})$;
3. Apply the classic R-SVD algorithm [28] with $U_p \Sigma_p V_p^T$ and the new data matrix $\tilde{F}$ to obtain $U_e \Sigma_e V_e^T$.

In order to fit the data streams well, the forgetting factor is introduced by [15] to weight the data streams. Typically, recent observations are given more weights than historical ones. For example, the weighted data matrix $H_e^{'}$ of H_e may be formulated as: $H_e^{'} = (\lambda H_p \mid F) = \left(U_p (\lambda \Sigma_p) V_p^T \mid F \right)$ where λ is the forgetting factor. The analytical proof of R-SVD is given in [15, 28].

2.2.4 Incremental rank-(R_1, R_2, R_3) tensor subspace analysis

Based on HOSVD [22], *IRTSA* presented below efficiently identifies the dominant projection subspaces of 3-order tensors, and is capable of incrementally updating

Accepted by ICCV 2007

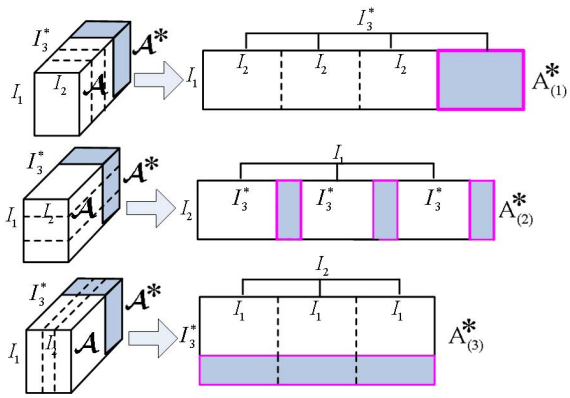

Figure 3. **Illustration of the incremental rank-(R_1, R_2, R_3) tensor subspace learning of a 3-order tensor.**

these subspaces when new data arrive. Given the CVD of the mode-k unfolding matrix $\mathrm{A}_{(k)} (1 \leq k \leq 3)$ for a 3-order tensor $\mathcal{A} \in \mathcal{R}^{I_1 \times I_2 \times I_3}$, *IRTSA* is able to efficiently compute the CVD of the mode-i unfolding matrix $\mathrm{A}^*_{(i)} (1 \leq i \leq 3)$ for $\mathcal{A}^* = (\mathcal{A} \mid \mathcal{F}) \in \mathcal{R}^{I_1 \times I_2 \times I_3^*}$ where $\mathcal{F} \in \mathcal{R}^{I_1 \times I_2 \times I_3^{'}}$ is a new 3-order subtensor and $I_3^* = I_3 + I_3^{'}$. To facilitate the description, Figure 3 is used for illustration. In the left half of Figure 3, three identical tensors are unfolded in three different modes. For each tensor, the white regions represent the original subtensor while the dark regions denote the newly added subtensor. The three unfolding matrices corresponding to the three different modes are shown in the right half of Figure 3, where the dark regions represent the unfolding matrices of the newly added subtensor $\mathcal{F}$. With the emergence of the new data subtensors, the column spaces of $\mathrm{A}^*_{(1)}$ and $\mathrm{A}^*_{(2)}$ are extended at the same time when the row space of $\mathrm{A}^*_{(3)}$ is extended. Consequently, *IRTSA* needs to track the changes of these three unfolding spaces, and needs to identify the dominant projection subspaces for a compact representation of the tensor. It is noted that $\mathrm{A}^*_{(2)}$ can be decomposed as: $\mathrm{A}^*_{(2)} = \left(\mathrm{A}_{(2)} \mid \mathrm{F}_{(2)}\right) \cdot \mathrm{P} = \mathrm{B} \cdot \mathrm{P}$, where $\mathrm{B} = \left(\mathrm{A}_{(2)} \mid \mathrm{F}_{(2)}\right)$ and P is an orthonormal matrix obtained by column exchange and transpose operations on an $(I_1 \cdot I_3^*)$-order identity matrix G. Let

$$G = (\overbrace{E_1}^{I_3} \mid \overbrace{Q_1}^{I_3^{'}} \mid \overbrace{E_2}^{I_3} \mid \overbrace{Q_2}^{I_3^{'}} \mid \cdots \mid \cdots \mid \overbrace{E_{I_1}}^{I_3} \mid \overbrace{Q_{I_1}}^{I_3^{'}})$$

which is generated by partitioning G into $2I_1$ blocks in the column dimension. Consequently, the orthonormal matrix P is formulated as:

$$\mathrm{P} = (E_1 | E_2 | \cdots \mid E_{I_1} | Q_1 | Q_2 | \cdots | Q_{I_1})^T. \qquad (4)$$

In this way, $\mathrm{A}^*_{(2)}$'s CVD is efficiently computed on the basis of P and B's CVD obtained by applying R-SVD to B. Furthermore, $\mathrm{A}^*_{(1)}$'s CVD is efficiently obtained by performing R-SVD on the matrix $\left(\mathrm{A}_{(1)} \mid \mathrm{F}_{(1)}\right)$. Similarly, $\mathrm{A}^*_{(3)}$'s CVD is efficiently obtained by performing R-SVD on the matrix $\left(\frac{\mathrm{A}_{(3)}}{\mathrm{F}_{(3)}}\right)^T$. The specific procedure of *IRTSA* is listed in Table 2.

Input:
CVD of the mode-k unfolding matrix $\mathrm{A}_{(k)}$, i.e. $\mathrm{U}^{(k)}\mathrm{D}^{(k)}\mathrm{V}^{(k)^T}$ $(1 \leq k \leq 3)$ of an original tensor $\mathcal{A} \in \mathcal{R}^{I_1 \times I_2 \times I_3}$, newly-added tensor $\mathcal{F} \in \mathcal{R}^{I_1 \times I_2 \times I_3^{'}}$, column mean $\bar{L}^{(1)}$ of $\mathrm{A}_{(1)}$, column mean $\bar{L}^{(2)}$ of $\mathrm{A}_{(2)}$, row mean $\bar{L}^{(3)}$ of $\mathrm{A}_{(3)}$ and R_1, R_2, R_3.
Output:
CVD of the mode-i unfolding matrix $\mathrm{A}^*_{(i)}$, i.e. $\hat{\mathrm{U}}^{(i)}\hat{\mathrm{D}}^{(i)}\hat{\mathrm{V}}^{(i)^T} (1 \leq i \leq 3)$ of $\mathcal{A}^* = (\mathcal{A} \mid \mathcal{F}) \in \mathcal{R}^{I_1 \times I_2 \times I_3^*}$ where $I_3^* = I_3 + I_3^{'}$, column mean $\bar{L}^{(1)^*}$ of $\mathrm{A}^*_{(1)}$, column mean $\bar{L}^{(2)^*}$ of $\mathrm{A}^*_{(2)}$ and row mean $\bar{L}^{(3)^*}$ of $\mathrm{A}^*_{(3)}$.
Algorithm:
1. $\mathrm{A}^*_{(1)} = \left(\mathrm{A}_{(1)} | F_{(1)}\right)$;
2. $\mathrm{A}^*_{(2)} = \left(\mathrm{A}_{(2)} | F_{(2)}\right) \cdot \mathrm{P} = \mathrm{B} \cdot \mathrm{P}$, where P is defined in (4);
3. $\mathrm{A}^*_{(3)} = \left(\frac{\mathrm{A}_{(3)}}{F_{(3)}}\right)$;
4. $[\hat{\mathrm{U}}^{(1)}, \hat{\mathrm{D}}^{(1)}, \hat{\mathrm{V}}^{(1)}, \bar{L}^{(1)^*}]$=R-SVD$(\mathrm{A}^*_{(1)}, \bar{L}^{(1)}, R_1)$;
5. $[\hat{\mathrm{U}}^{(2)}, \hat{\mathrm{D}}^{(2)}, \widetilde{\mathrm{V}}_2, \bar{L}^{(2)^*}]$=R-SVD$(\mathrm{B}, \bar{L}^{(2)}, R_2)$;
6. $\hat{\mathrm{V}}^{(2)} = \mathrm{P}^T \cdot \widetilde{\mathrm{V}}_2$;
7. $[\widetilde{\mathrm{U}}_3, \widetilde{\mathrm{D}}_3, \widetilde{\mathrm{V}}_3, \widetilde{L}_3]$=R-SVD$((\mathrm{A}^*_{(3)})^T, (\bar{L}^{(3)})^T, R_3)$;
8. $\hat{\mathrm{U}}^{(3)} = \widetilde{\mathrm{V}}_3$, $\hat{\mathrm{D}}^{(3)} = (\widetilde{\mathrm{D}}_3)^T$, $\hat{\mathrm{V}}^{(3)} = \widetilde{\mathrm{U}}_3$, $\bar{L}^{(3)^*} = (\widetilde{L}_3)^T$.

Table 2. **The incremental rank-(R_1, R_2, R_3) tensor subspace analysis algorithm (*IRTSA*). R-SVD$((\mathbb{C} \mid \mathbb{E}), L, R)$ represents that the first R dominant eigenvectors are used in R-SVD [15] for the matrix $(\mathbb{C}|\mathbb{E})$ with $\mathbb{C}$'s column mean being L.**

In real tracking applications, it is necessary for a subspace analysis-based algorithm to evaluate the likelihood of the test sample and the learned subspace. In *IRTSA*, the criteria for the likelihood evaluation are given as follows.

Given I_3 existing images represented as $\mathcal{A} \in \mathcal{R}^{I_1 \times I_2 \times I_3}$, a test image denoted as $\mathcal{J} \in \mathcal{R}^{I_1 \times I_2 \times 1}$ and the mode-i column projection matrices $U^{(i)} \in \mathcal{R}^{I_i \times R_i} (1 \leq i \leq 2)$ and the mode-3 row projection matrix $V^{(3)} \in \mathcal{R}^{(I_1 I_2) \times R_3}$ of the learned subspaces of $\mathcal{A}$, the likelihood can be determined by the sum of the reconstruction error norms of the three modes:

$$RE = \sum_{i=1}^{2} \|(\mathcal{J} - \mathcal{M}_i) - (\mathcal{J} - \mathcal{M}_i) \prod_{j=1}^{2} \times_j (U^{(j)} \cdot U^{(j)^T})\|^2 + \|(\mathrm{J}_{(3)} - \mathrm{M}_3) - (\mathrm{J}_{(3)} - \mathrm{M}_3) \cdot (V^{(3)} \cdot V^{(3)^T})\|^2 \qquad (5)$$

where $\mathrm{J}_{(i)}$ is the mode-i unfolding matrix of $\mathcal{J}$, $\prod_{k=1}^{K} \times_k D_k = \times_1 D_1 \times_2 D_2 \ldots \times_K D_K$, $\mathrm{M}_3 = \bar{L}^{(3)}$ which is the row mean of the mode-3 unfolding matrix $\mathrm{A}_{(3)}$, $\mathcal{M}_1$ and $\mathcal{M}_2$ are defined as:

Accepted by ICCV 2007

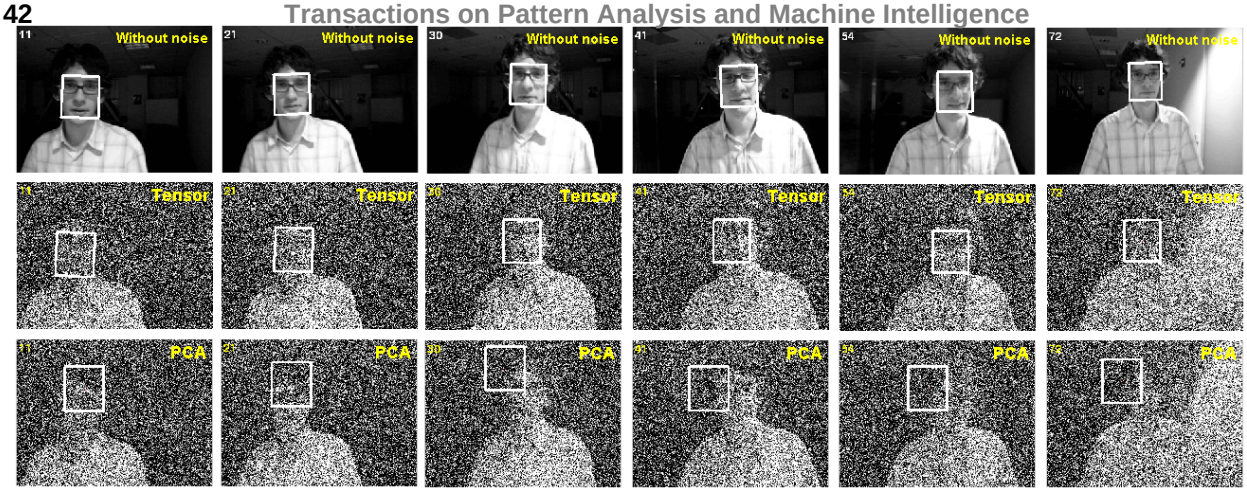

Figure 4. **The tracking results of *IRTSA* and *IAVSL*, respectively, under the disturbance of a strong noise. Row 1 is the reference tracking result with no noise. Rows 2 and 3 correspond to the tracking results of *IRTSA* and *IAVSL*, respectively.**

$$\begin{aligned} \mathcal{M}_1 &= (\overbrace{\bar{L}^{(1)}, \ldots, \bar{L}^{(1)}}^{I_2}) \in \mathcal{R}^{I_1 \times I_2 \times 1} \\ \mathcal{M}_2 &= (\underbrace{\bar{L}^{(2)}, \ldots, \bar{L}^{(2)}}_{I_1})^T \in \mathcal{R}^{I_1 \times I_2 \times 1} \end{aligned} \quad (6)$$

where $\bar{L}^{(1)}$ and $\bar{L}^{(2)}$ are the column means of the mode-$(1,2)$ unfolding matrices $\mathrm{A}_{(1)}$ and $\mathrm{A}_{(2)}$, respectively. The smaller the RE, the larger the likelihood.

2.3. Bayesian inference for visual tracking

For visual tracking, a Markov model with a hidden state variable is generally used for motion estimation. In this model, the target motion between two consecutive frames is usually assumed to be an affine motion. Let X_t denote the state variable describing the affine motion parameters (the location) of a target at time t. Given a set of observed images $\mathcal{O}_t = \{O_1, \ldots, O_t\}$, the posterior probability is formulated by Bayes' theorem as:

$$p(X_t|\mathcal{O}_t) \propto p(O_t|X_t) \int p(X_t|X_{t-1}) p(X_{t-1}|\mathcal{O}_{t-1}) dX_{t-1} \quad (7)$$

where $p(O_t \mid X_t)$ denotes the likelihood function, and $p(X_t|X_{t-1})$ represents the dynamic model. $p(O_t|X_t)$ and $p(X_t|X_{t-1})$ decide the entire tracking process. A particle filter [3] is used for approximating the distribution over the location of the target using a set of weighted samples.

In the tracking framework, we apply an affine image warping to model the target motion of two consecutive frames. The six parameters of the affine transform are used to model $p(X_t \mid X_{t-1})$ of a tracked target. Let $X_t = (x_t, y_t, \eta_t, s_t, \beta_t, \phi_t)$ where $x_t, y_t, \eta_t, s_t, \beta_t, \phi_t$ denote the x, y translations, the rotation angle, the scale, the aspect ratio, and the skew direction at time t, respectively. We employ a Gaussian distribution to model the state transition distribution $p(X_t \mid X_{t-1})$. Also the six parameters of the affine transform are assumed to be independent. Consequently, $p(X_t|X_{t-1})$ is formulated as:

$$p(X_t|X_{t-1}) = \mathcal{N}(X_t; X_{t-1}, \Sigma) \quad (8)$$

where Σ denotes a diagonal covariance matrix whose diagonal elements are $\sigma_x^2, \sigma_y^2, \sigma_\eta^2, \sigma_s^2, \sigma_\beta^2, \sigma_\phi^2$, respectively. The observation model $p(O_t|X_t)$ reflects the probability that a sample is generated from the subspace. In this paper, RE, defined in (5), is used to measure the distance from the sample to the center of the subspace. Consequently, $p(O_t|X_t)$ is formulated as:

$$p(O_t|X_t) \propto exp(-RE) \quad (9)$$

For MAP estimate, we just use the affinely warped image region associated with the highest weighted hypothesis to update the tensor-based eigensapace model.

3. Experiments

In order to evaluate the performance of the proposed tracking framework, four videos are used in the experiments. Videos 1 and 4 are captured indoor while videos 2 and 3 are recorded outdoor. Furthermore, videos 1 and 3 are taken from moving cameras in different scenes while videos 2 and 4 are recorded by stationary cameras. Each frame in these videos is a 8-bit gray scale image. In video 1, a man walks in a room changing his pose and facial expression over the time with varying lighting conditions. In video 2, a pedestrian as a small target moves down a road in a dark and blurry scene. In video 3, a man walks from left to right in a bright road scene; his body pose varies over the time, with a drastic motion and pose change (bowing down to reach the ground and standing up back again) in the middle of the video stream. Video 4 consists of dark and motion-blurring gray scale images, where many motion events take place, including wearing and taking off the glasses, head shaking, and hands occluding the face from time to time. For the tensor eigenspace representation, the size of each target region is normalized to 20×20 pixels. The settings of the ranks R_1, R_2 and R_3 in *IRTSA* are obtained from the experiments. The forgetting factor λ in R-SVD is set as 0.99. The tensor subspace is updated every three frames. For the particle filtering in the visual tracking,

Accepted by ICCV 2007

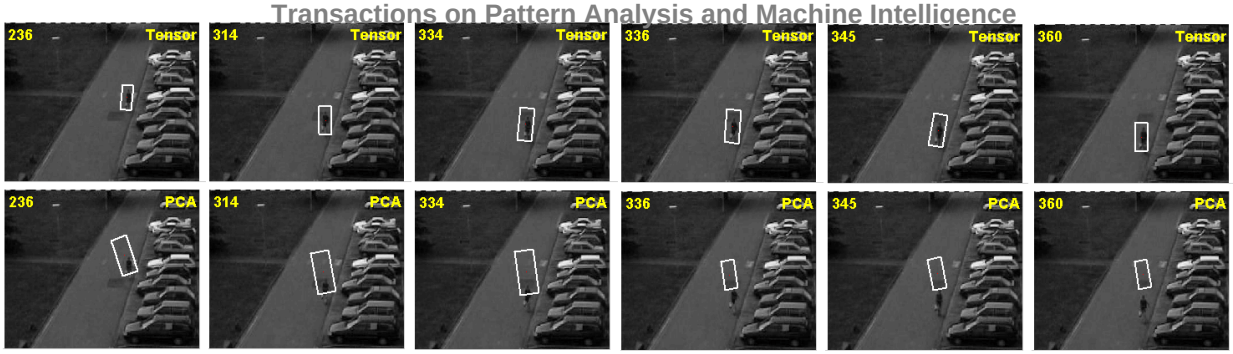

Figure 5. **The tracking results of *IRTSA* and *IAVSL*, respectively, in the scenarios of small target and blurring scenes. Rows 1 and 2 correspond to *IRTSA* and *IAVSL*, respectively.**

the number of particles is set to be 300. The six diagonal elements $(\sigma_x^2, \sigma_y^2, \sigma_\eta^2, \sigma_s^2, \sigma_\beta^2, \sigma_\phi^2)$ of the covariance matrix Σ in (8) are assigned as $(5^2, 5^2, 0.03^2, 0.03^2, 0.005^2, 0.001^2)$, respectively.

Four experiments are conducted to demonstrate the claimed contributions of the proposed *IRTSA*. These four experiments are to compare tracking results of *IRTSA* with those of a state-of-the-art image-as-vector subspace learning based tracking algorithm [15], referred as *IAVSL* in this paper, in different scenarios including noise disturbance, scene blurring, small target tracking, target pose variation, and occlusion. *IAVSL* is a representative image-as-vector linear subspace learning algorithm which incrementally learns a low dimensional eigenspace representation of the target appearance by online PCA. Compared with most existing tracking algorithms, based on constructing an invariant target appearance representation, *IAVSL* is able to online track appearance changes of the target, resulting in a better tracking result. In contrast to image-as-vector *IAVSL*, our proposed *IRTSA* relies on image-as-matrix tensor subspace analysis to reflect the appearance changes of a target. Consequently, it is very significant to make a comparison between *IAVSL* and *IRTSA*.

The first experiment is performed to evaluate the performances of the two subspace analysis based tracking techniques—*IAVSL* and *IRTSA* on investigating their tracking performances under the disturbance of strong noise. The video used in this experiment is obtained by manually adding Gaussian random noise to Video 1. The process of adding the noise is formulated as: $I'(x,y) = \mathcal{G}\left(I(x,y) + s \cdot Z\right)$, where $I(x,y)$ denotes the original pixel value, $I'(x,y)$ represents the pixel value after adding noise, Z follows the standard normal distribution $\mathcal{N}(0,1)$, s is a scaling factor controlling the amplitude of the noise, and the function $\mathcal{G}(\cdot)$ is defined as:

$$\mathcal{G}(x) = \begin{cases} 0 & x < 0 \\ 255 & x > 255 \\ [x] & 0 \le x \le 255 \end{cases} \tag{10}$$

where $[x]$ stands for the floor of the element x. In this experiment, s is set as 200. R_1, R_2 and R_3 in *IRTSA* are assigned as 3,3 and 5, respectively. For *IAVSL*, 5 eigenvectors are maintained during the tracking, and the remaining eigenvectors are discarded at each subspace updating. The final tracking results of *IRTSA* and *IAVSL* are shown in Figure 4. For a better visualization, we just show the tracking results of six representative frames 11,21,30,41,54 and 72. In Figure 4, the first row corresponds to the tracking results of the reference frames without noise using *IRTSA*. The remaining two rows are for the tracking results of *IRTSA* and *IAVSL*, respectively, under the disturbance of the noise. From Figure 4, we see that the proposed tracking algorithm exhibits a robust tracking result while *IAVSL* fails to track the face under the disturbance of strong noise. This is due to the fact that since the spatial correlation information is ignored in *IAVSL*, the noise disturbance substantially changes the vector eigenspace representation of the target's appearance. In comparison, *IRTSA* relies on a robust tensor eigenspace model which makes a full use of the spatio-temporal distribution information of the image ensembles in the three modes. Consequently, *IRTSA* has a strong error-tolerating capability. (Please see the supplementary video "Experiment1.mpg" for the first experiment.)

The second experiment aims to compare the tracking performance of *IRTSA* with that of *IAVSL* in handling scene blurring and small target scenarios using Video 2. R_1, R_2 and R_3 in *IRTSA* are set as 5,5 and 8, respectively. For *IAVSL*, 16 eigenvectors are maintained during the tracking, and the remaining eigenvectors are discarded at each subspace updating. We show the final tracking results for *IRTSA* and *IAVSL* in Figure 5, where the first and the second rows correspond to the performances of *IRTSA* and *IAVSL*, respectively, in which six representative frames (236,314,334,336,345 and 360) of the video stream are shown. Clearly, *IRTSA* succeeds in tracking while *IAVSL* fails. The reasons are explained as follows. *IRTSA* takes an image as a matrix, in comparison with the image-as-vector representation in *IAVSL*. Consequently, *IRTSA* makes a more compact target representation capable of reducing potentially substantial spatio-temporal redundancy of the image ensembles while *IAVSL* must solve for a high-dimensional data learning problem. This becomes particularly true for tracking a small target and/or with a blurring

Accepted by ICCV 2007

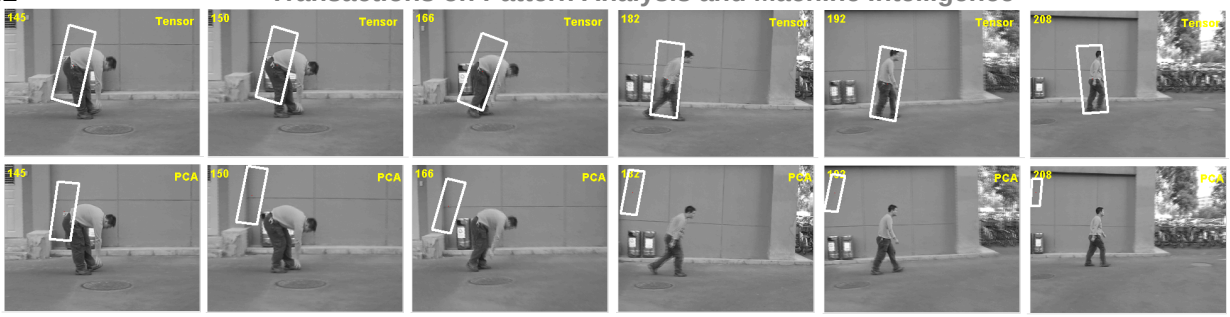

Figure 6. **The tracking results of *IRTSA* and *IAVSL* in the scenarios of drastic pose change. Rows 1 and 2 correspond to *IRTSA* and *IAVSL*, respectively.**

scene; here the spatial correlation information of the target's appearance is critical. Due to this loss of the spatial correlation information, *IAVSL* fails to track the target in these scenarios. (Please see the supplementary video "Experiment2.mpg" for the second experiment.)

The third experiment is for a comparison between *IRTSA* and *IAVSL* in the scenarios of pose variation using Video 3. In this experiment, R_1, R_2 and R_3 are assigned as 8,8 and 10, respectively. For *IAVSL*, 16 eigenvectors are maintained during the tracking, and the remaining eigenvectors are discarded at each subspace updating. The final tracking results are demonstrated in Figure 6, where rows 1 and 2 correspond to *IRTSA* and *IAVSL*, respectively, in which six representative frames (145, 150, 166, 182, 192 and 208) of the video stream are shown. From Figure 6, it is clear that *IRTSA* is capable of tracking the target successfully even with a drastic pose and motion change while *IAVSL* gets lost in tracking the target after this drastic pose and motion change. (Please see the supplementary video "Experiment3.mpg" for the third experiment.)

The fourth experiment is to compare the performances of the two methods *IRTSA* and *IAVSL* in handling partial occlusions using Video 4. In this experiment, R_1, R_2 and R_3 are set as 3,3 and 5, respectively. For *IAVSL*, 10 eigenvectors are maintained during the tracking, and the remaining eigenvectors are discarded at each subspace updating. The final tracking results are demonstrated in Figure 7, where rows 1 and 2 are the performance results of *IRTSA* and *IAVSL*, respectively, in which six representative frames (92, 102, 119, 132, 148 and 174) of the video stream are shown. From Figure 7, we see that *IRTSA* is capable of tracking the target all the time even though the target is occluded partially from time to time in a poor lighting condition. On the other hand, *IAVSL* gets completely lost in tracking the target. (Please see the supplementary video "Experiment4.mpg" for the fourth experiment.)

From the results in the third and the fourth experiments, we note that *IRTSA* is robust to pose variation and occlusion. The reason is that the dominant subspace information of the three modes is incorporated into *IRTSA*. Even if the subspace information of some modes is partially lost or drastically varies, *IRTSA* is capable of recovering the information using the cues of the subspace information from other modes.

Method \ Exp	Exp 1	Exp 2	Exp 3	Exp 4
IRTSA	5.12	2.54	3.26	2.52
IAVSL	31.71	28.65	77.19	28.61

Table 3. **Comparison between *IRTSA* and *IAVSL* in the tracking mean localization deviation with the ground truth. Exp k corresponds to experiment k $(1 \leq k \leq 4)$, and the localization deviation is measured in pixels. It is clear that the proposed *IRTSA* performs much better than *IAVSL*.**

Since there are no benchmark databases in the experiments, we have to provide a quantitative comparison between *IRTSA* and *IAVSL* using some representative frames. The object center locations in the representative frames used by the above four experiments are labeled manually as the ground truth. In this way, we can quantitatively evaluate the tracking performances of *IRTSA* and *IAVSL* by computing their corresponding pixel-based mean localization deviations between tracking results and the ground truth. The less the deviation, the higher the localization accuracy. The final comparing results are listed in Table 3. From Table 3, we see that the target localization accuracy of *IRTSA* is much higher than that of *IAVSL*.

In summary, we observe that *IRTSA* outperforms *IAVSL* in the scenarios of noise disturbance, blurring scenes, small targets, drastic target pose change, and occlusions. Consequently, *IRTSA* is an effective online tensor subspace learning algorithm which performs well in modeling appearance changes of a target in many complex scenarios.

4. Conclusion

In this paper, we have developed a visual tracking framework based on the incremental tensor subspace learning. The main contribution of this framework is two-fold. (1) A novel online tensor subspace learning algorithm, which enables subspace analysis within a multilinear framework, is proposed to reflect the appearance changes of a target. (2) A novel likelihood function, based on the tensor reconstruction error norm, is developed to measure the similarity between the test image and the learned tensor subspace model

Accepted by ICCV 2007

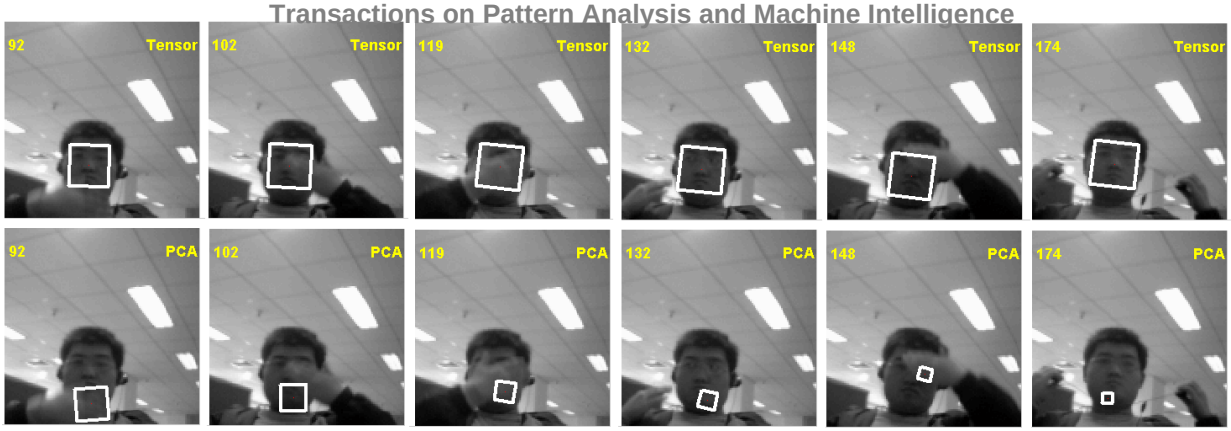

Figure 7. **The tracking results of *IRTSA* and *IAVSL* in the scenarios of partial occlusions. Rows 1 and 2 show the tracking results of *IRTSA* and *IAVSL*, respectively.**

during the tracking. Compared with the image-as-vector tracking methods in the literature, our proposed image-as-matrix tracking method is more robust to noise or low quality images, occlusion, scene blurring, small target, and target pose variation. Experimental results have demonstrated the robustness and promise of the proposed framework.

5. Acknowledgment

This work is partly supported by NSFC (Grant No. 60520120099 and 60672040) and the National 863 High-Tech R&D Program of China (Grant No. 2006AA01Z453). ZZ is partly supported by NSF (IIS-0535162), AFRL (FA8750-05-2-0284), and AFOSR (FA9550-06-1-0327).

References

[1] G. Hager and P. Belhumeur, "Real-time tracking of image regions with changes in geometry and illumination," in *Proc. CVPR'96,* pp.430-410, 1996.

[2] M. J. Black and A. D. Jepson, "Eigentracking: Robust matching and tracking of articulated objects using view-based representation," in *Proc. ECCV'96,* pp.329-342, 1996.

[3] M. Isard and A. Blake, "Contour tracking by stochastic propagation of conditional density," in *Proc. ECCV'96,* Vol. 2, pp.343-356, 1996.

[4] M. J. Black, D. J. Fleet, and Y. Yacoob, "A framework for modeling appearance change in image sequence," in *Proc. ICCV'98,* pp.660-667, 1998.

[5] A. D. Jepson, D. J. Fleet, and T. F. El-Maraghi, "Robust Online Appearance Models for Visual Tracking," in *Proc. CVPR'01,* Vol. 1, pp.415-422, 2001.

[6] S. K. Zhou, R. Chellappa, and B. Moghaddam, "Visual Tracking and Recognition Using Appearance-Adaptive Models in Particle Filters," *IEEE Trans. on Image Processing,* Vol. 13, pp.1491-1506 , November 2004.

[7] T. Yu and Y. Wu, "Differential Tracking based on Spatial-Appearance Model(SAM)," in *Proc. CVPR'06,* Vol. 1, pp.720-727, June 2006.

[8] J. Li, S. K. Zhou, and R. Chellappa, "Appearance Modeling under Geometric Context," in *Proc. ICCV'05,* Vol. 2, pp.1252-1259, 2005.

[9] S. Wong, K. K. Wong and R. Cipolla, "Robust Appearance-based Tracking using a sparse Bayesian classifier," in *Proc. ICPR'06,* Vol. 3, pp.47-50, 2006.

[10] K. Lee and D. Kriegman, "Online Learning of Probabilistic Appearance Manifolds for Video-based Recognition and Tracking," in *Proc. CVPR'05,* Vol. 1, pp.852-859 , 2005.

[11] H. Lim, V. I. Morariu3, O. I. Camps, and M. Sznaier1, "Dynamic Appearance Modeling for Human Tracking," in *Proc. CVPR'06,* Vol. 1, pp.751-757, 2006.

[12] J. Ho, K. Lee, M. Yang and D. Kriegman, "Visual Tracking Using Learned Linear Subspaces ," in *Proc. CVPR'04,* Vol. 1, pp.782-789, 2004.

[13] Y. Li, L. Xu, J. Morphett and R. Jacobs, "On Incremental and Robust Subspace Learning," *Pattern Recognition,* 37(7), pp. 1509-1518, 2004.

[14] D. Skocaj, A. Leonardis, "Weighted and Robust Incremental Method for Subspace Learning," in *Proc. ICCV'03,* pp.1494-1501, 2003.

[15] J. Limy, D. Ross, R. Lin and M. Yang, "Incremental Learning for Visual Tracking," *NIPS'04,* pp.793-800, MIT Press, 2005.

[16] J. Yang, D. Zhang, A. F. Frangi, and J. Yang, "Two-dimensional PCA: A New Approach to Appearance-based Face Representation and Recognition," in *IEEE Trans. PAMI.,* Vol. 26, Iss. 1, pp.131-137, Jan. 2004.

[17] J. Ye, R. Janardan, and Q. Li, "Two-Dimensional Linear Discriminant Analysis," *NIPS'04,* pp.1569-1576, MIT Press,2004.

[18] J. Ye, "Generalized low rank approximations of matrices," *ICML'04,* July 2004.

[19] J. Ye, R. Janardan, and Q. Li, "GPCA: An Efficient Dimension Reduction Scheme for Image Compression and Retrieval," *ACM KDD'04,* pp.354-363, August 2004.

[20] H. Wang and N. Ahuja, "Rank-R Approximation of Tensors Using Image-as-matrix Representation ," in *Proc. CVPR'05,* Vol. 2, pp.346-353, 2005.

[21] S. Yan, D. Xu, Q. Yang, L. Zhang, X. Tang and H. Zhang, "Discriminant analysis with tensor representation," in *Proc. CVPR'05,* Vol. 1, pp.526-532, June 2005.

[22] M. A. O. Vasilescu and D. Terzopoulos, "Multilinear Subspace Analysis of Image Ensembles," in *Proc. CVPR'03,* Vol. 2, pp.93-99, June 2003.

[23] M. A. O. Vasilescu and D. Terzopoulos, "Multilinear Subspace Analysis of Image Ensembles: TensorFaces," in *Proc. ECCV'02,* pp.447-460, May 2002.

[24] X. He, D. Cai and P. Niyogi, "Tensor Subspace Analysis," *NIPS'05,* Dec. 2005.

[25] H. Wang, S. Yan, T. Huang and X. Tang, "A Convergent Solution to Tensor Subspace Learning," in *Proc. IJCAI'07,* 2007.

[26] J. Sun, D. Tao and C. Faloutsos, "Beyond Streams and Graphs: Dynamic Tensor Analysis," *ACM KDD'06,* Aug. 2006.

[27] J. Sun, S. Papadimitriou and P. S. Yu, "Window-based Tensor Analysis on High-dimensional and Multi-aspect Streams," in *Proc. ICDM'06,* Dec. 2006.

[28] A. Levy and M. Lindenbaum, "Sequential Karhunen-Loeve Basis Extraction and Its Application to Images," *IEEE Trans. on Image Processing,* Vol. 9, pp.1371-1374, 2000.

[29] 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 Journal of Matrix Analysis and Applications,* Vol. 21, Iss. 4, pp.1324-1342, 2000.

Accepted by ICCV 2007