Int J Comput Vis
DOI 10.1007/s11263-009-0283-4

3D Human Motion Tracking with a Coordinated Mixture of Factor Analyzers

Rui Li · Tai-Peng Tian · Stan Sclaroff · Ming-Hsuan Yang

Received: 30 January 2008 / Accepted: 24 July 2009

Abstract A major challenge in applying Bayesian tracking methods for tracking 3D human body pose is the high dimensionality of the pose state space. It has been observed that the 3D human body pose parameters typically can be assumed to lie on a low-dimensional manifold embedded in the high-dimensional space. The goal of this work is to approximate the low-dimensional manifold so that a low-dimensional state vector can be obtained for efficient and effective Bayesian tracking. To achieve this goal, a globally coordinated mixture of factor analyzers is learned from motion capture data. Each factor analyzer in the mixture is a "locally linear dimensionality reducer" that approximates a part of the manifold. The global parametrization of the manifold is obtained by aligning these locally linear pieces in a global coordinate system. To enable automatic and optimal selection of the number of factor analyzers and the dimensionality of the manifold, a variational Bayesian formulation of the globally coordinated mixture of factor analyzers is proposed. The advantages of the proposed model are demonstrated in a multiple hypothesis tracker for tracking 3D human body pose. Quantitative comparisons on benchmark datasets show that the proposed method produces more accurate 3D pose estimates over time than those obtained from two previously proposed Bayesian tracking methods.

Keywords 3D human body tracking · Particle filtering · High-dimensional state space · Variational methods

1 Introduction

Tracking articulated human motion is of great interest in various applications: video surveillance, human computer interfaces, computer animation, biometrics and medical applications. Marker-based motion tracking methods are commonly used in computer games and clinical human motion analysis. However, these methods are considered to be too intrusive and/or too expensive for daily deployment because: (1) subjects usually need to bare most of their skin or wear tight-fitting clothes, (2) markers are placed on the subject's skin or clothes, (3) marker placement is time consuming, and (4) a controlled environment is required.

Therefore, vision based tracking methods that require neither special clothing nor markers on the human body have been actively studied. These tracking methods can be broadly categorized as either 2D or 3D, depending on the type of human pose information they can recover.

2D methods track human motion in the image plane by making use of image features and/or a 2D human model (Agarwal and Triggs 2004; Cham and Rehg 1999; Ioffe and Forsyth 2001; Ju et al. 1996; Lan and Huttenlocher 2004; Ramanan et al. 2007), thereby avoiding the need for complex 3D models or camera calibration information. While these methods are usually efficient, only 2D joint locations and angles can be inferred. As a result, the 2D methods have difficulty in handling occlusions and they are less effective

R. Li (✉) · T.-P. Tian · S. Sclaroff
Computer Science Department, Boston University, Boston, MA 02215, USA
e-mail: lir@cs.bu.edu

T.-P. Tian
e-mail: tian@cs.bu.edu

S. Sclaroff
e-mail: sclaroff@cs.bu.edu

M.-H. Yang
Electrical Engineering and Computer Science, University of California, Merced, CA 95344, USA
e-mail: mhyang@ieee.org

Published online: 07 August 2009

for applications where accurate 3D information is required. To better understand human motion, 3D methods resort to detailed 3D articulated models that require significantly more degrees of freedom. Consequently, algorithms that are able to handle high-dimensional, non-linear data efficiently and effectively are essential to the success of the 3D methods. Recent research addresses this issue by combining 3D Bayesian tracking methods with carefully designed strategies for search within the state space (Choo and Fleet 2001; Deutscher et al. 2000; Sidenbladh et al. 2000; Sigal et al. 2004; Sminchisescu and Triggs 2001), as well as methods for dimensionality reduction of the state space (Agarwal and Triggs 2004; Elgammal and Lee 2009; Li et al. 2006, 2007; Sminchisescu and Jepson 2004; Tian et al. 2005b; Urtasun et al. 2006).

In this paper, we exploit the physical constraints of human motion by learning a low-dimensional latent model from high-dimensional motion capture data. Our approach is motivated by the fact that while the representation of human pose is high-dimensional, the intrinsic dimensionality is much lower, as human body movements are highly coordinated (Safonova et al. 2004). Furthermore, not all poses are equally likely given the specific types of motions we aim to track. For our application domain, the preferred dimensionality reduction method should:

(1) minimize information loss during dimensionality reduction so that the low-dimensional representation of the human pose captures key kinematic information;
(2) preserve continuity so that similar poses are mapped to nearby locations in the low-dimensional latent space;
(3) approximate the densities of the training human motion capture data to handle outliers;
(4) provide non-linear bidirectional mapping functions so that data that resides in the latent space can be mapped back to the high-dimensional human pose space, and conversely, poses can be mapped back to the latent space for validation; and
(5) handle large training datasets with ease, as with today's motion capture technology, large training data sets are easily accessible.

To meet these requirements, we employ a globally coordinated mixture of factor analyzers (GCMFA) framework (Verbeek 2006). As has been demonstrated in (Li et al. 2006), the GCMFA is effective in preserving important information when applied to human motion capture data. The GCMFA provides a global parametrization of the low-dimensional manifold. Each factor analyzer in the mixture is a "locally linear dimensionality reducer" that approximates a part of the manifold. The global parametrization of the manifold is obtained by aligning these locally linear pieces in a global coordinate system. The embedded data forms clusters within the globally coordinated low-dimensional space; this makes it possible to derive an efficient multiple hypothesis tracking algorithm based on the distribution of the modes. By tracking in the low-dimensional space, we avoid the sample impoverishment problem and retain the simplicity of the multiple hypothesis tracking algorithm at the same time.

However, even with all the desirable properties that suit our application, for the GCMFA to effectively model the relationship between the high-dimensional observation space and low-dimensional latent space, it still requires careful initialization and model selection, i.e., the number of the factor analyzers in the mixture and the dimensionality of the latent space. In (Roweis et al. 2001; Teh and Roweis 2002; Verbeek 2006), a good embedding algorithm like (Roweis and Saul 2000) was used for initialization and the model structure was fixed before training. In this paper, we take a different approach: we derive a variational Bayesian solution for automatic selection of the optimal model structure. A single factor analyzer is used as the initial model and model splitting is performed iteratively until the optimal model structure is obtained based on the variational Bayesian criterion.

The performance of a 3D tracker that employs the model learned via the variational Bayesian formulation is evaluated on the HumanEvaI benchmark datasets (Sigal and Black 2006). In experiments with real video, the system reliably tracks body motion during self-occlusions and in the presence of motion blur. Given clusters formed in the latent space, our tracker can accurately track large movements of the human limbs in adjacent time steps by propagating each cluster's information over time in the multiple hypothesis tracking algorithm. A quantitative comparison shows that the formulation produces more accurate 3D pose estimates than those obtained using two previously-proposed Bayesian tracking methods (Deutscher et al. 2000; Urtasun et al. 2005) that also employ smart search strategies in the state space. Furthermore, the dimensionality of the state space in our tracker and the number of FA's in the mixture is determined automatically using the proposed variational Bayesian learning method.

Hence, our main contributions in this work are: (1) to propose a variation Bayesian learning approach for model selection of the GCMFA model; and (2) to successfully apply this model in a Bayesian tracking method to achieve accurate 3D human tracking.

2 Related Work

Bayesian tracking methods combined with carefully designed search strategies of the state space have been actively studied. This is especially important in 3D human motion tracking, where the number of parameters needed to represent a 3D body pose is large (usually 20–60 depending on the level of detail). Besides adopting methods like (Cham and Rehg 1999; Deutscher et al. 2000; MacCormick and Blake 1999; Sminchisescu and Triggs 2001; Sullivan and Rittscher 2001) to carefully explore the 20–60 dimensional space, recent efforts (Elgammal and Lee 2004; Li et al. 2006; Sminchisescu and Jepson 2004; Urtasun et al. 2005) have been dedicated to reducing the dimensionality of the state space to achieve efficient tracking. It has been shown in (Elgammal and Lee 2004; Li et al. 2006; Sminchisescu and Jepson 2004; Urtasun et al. 2005) that the relationship between the high-dimensional pose parameters and low-dimensional manifold is non-linear. Hence, before we move on to the discussion of 3D human tracking algorithms that employ a dimensionality reduced state space, we first review non-linear dimensionality reduction (NLDR) algorithms.

2.1 Non-Linear Dimensionality Reduction (NLDR) Algorithms

NLDR techniques can be broadly classified into two categories: embedding techniques vs. mapping techniques. Embedding techniques model the structure of the data that generates the manifold without providing mapping functions between the observation space and the latent space. ISOmetric feature MAPping (ISOMAP) (Tenenbaum et al. 2000) and its variants (Jenkins and Matarić 2004; Silva and Tenenbaum 2003), Locally Linear Embedding (LLE) (Roweis and Saul 2000) and spectral embedding (Belkin and Niyogi 2001) are widely-used NLDR algorithms in this category. These techniques never explicitly learn the mapping functions; therefore, there is no simple method to map data outside the training set to the low-dimensional space or back. In order to use these algorithms for vision problems like tracking, (Elgammal and Lee 2004) and (Sminchisescu and Jepson 2004) have proposed regression methods to learn the mapping functions after embedding.

Mapping-based techniques learn the nonlinear mapping functions either by modeling the nonlinear functions directly (Bishop et al. 1998; Lawrence 2003; Schölkopf et al. 1998) or by using a combination of local linear models (Brand 2002; Roweis et al. 2001; Teh and Roweis 2002) during dimensionality reduction. Mapping functions provide ways to map unseen data to the latent space (we use the term *latent space* and *low-dimensional space* interchangeably) and/or to synthesize new data from the latent space; hence, mapping-based techniques have been widely used in the computer vision. Recent publications demonstrate successful applications in human motion analysis (Li et al. 2006; Tian et al. 2005a; Urtasun et al. 2006, 2005) and image manifold modeling (Verbeek 2006).

In this paper, we make use of the globally coordinated mixture of factor analyzers framework proposed by (Verbeek 2006) and derive a variational Bayesian formulation to infer the number of mixture components and the dimension of the latent space together with the model parameters. This circumvents the problem of fixing the model structure before training as in (Li et al. 2006).

2.2 Human Motion Tracking

There is a broad range of work related to human motion tracking. See (Poppe 2007b; Wang et al. 2003) for recent surveys. Our focus is on the Bayesian tracking techniques that exploit a dimensionality reduced state space.

Recently, researchers have proposed the use of dimensionality reduction techniques on the state space to reduce the size of the body pose state vector. This is justified by the insight that the space of possible human motions is intrinsically low-dimensional (Safonova et al. 2004). Particle filtering in the dimensionality reduced state space is faster because significantly fewer particles are required to adequately approximate the state space posterior distribution.

Three recent works (Sminchisescu and Jepson 2004; Tian et al. 2005b; Urtasun et al. 2005) are most closely related to our proposed algorithm for tracking human motion in a dimensionality-reduced space. In (Sminchisescu and Jepson 2004), different regression algorithms are used for the forward mapping (dimensionality reduction) and inverse mapping. The representatives used in the regression are chosen in an heuristic manner (Sminchisescu and Jepson 2004). In (Urtasun et al. 2005), a Gaussian process latent variable model (GPLVM) and a second order Markov model are used for tracking applications. The learned GPLVM model is used to provide a human pose prior. Tracking is then accomplished by minimizing the cost of 2D image matching, with the negative log-likelihood of the model prior as the regularization term. Both (Sminchisescu and Jepson 2004) and (Urtasun et al. 2005) advocate for the use of gradient descent optimization techniques; thus, the learned low-dimensional space must be smooth. An alternative approach (Tian et al. 2005b) employs the GPLVM in a modified particle filtering algorithm where samples are drawn from the low-dimensional latent space modeled by a trained GPLVM. This approach is similar to our work in the sense that a non-linear dimensionality reduction method is used to attain good particle filter based 3D tracking of human motion in video.

In tracking, if we use the estimate from the previous time step as the initialization for the next step, the gradient descent optimization process may stop at local optima and the samples generated based on the previous estimate may fail to capture an abrupt change of human motion. To combat this problem, global methods like annealing (Deutscher et al.

2000) and hybrid Monte Carlo (Choo and Fleet 2001) can be used.

In a more recent work (Urtasun et al. 2006), dynamical model information has been exploited in constructing the low-dimensional manifold and improved results have been reported. But this method cannot handle large training datasets due to the kernel sparsification problem, as there is no principled way to choose an active set for a dynamic sequence. Without sparsification, the full kernel matrix must be inverted at each iteration of learning. Thus, it is difficult to apply the dynamics extension of GPLVM to large data sets. To avoid the discontinuity problem caused by the use of an active set, Snelson and Ghaharmani propose sparsification techniques that make use of psuedo-inputs (Snelson and Ghahramani 2006). There are still two open problems with (Snelson and Ghahramani 2006): how to choose the number of psuedo-inputs, and how to avoid overfitting. Furthermore, the success of applying such techniques to human tracking has yet to be demonstrated.

There also exists another interesting line of research (Mori and Malik 2002; Poppe 2007a; Shakhnarovich et al. 2003; Stenger et al. 2003) where the methods employ a database that stores pairs of training images and their corresponding 3D poses. For a given input image, the corresponding 3D pose is estimated by searching for similar training images and interpolating using their corresponding 3D poses from the database. Good results on the HumanEvaI datasets have been reported in (Poppe 2007a). Discriminative image features, good similarity functions for comparison, and fast search strategies for large datasets are important for the success of these methods.

The aim of our work is to make Bayesian tracking more efficient and accurate for the task of 3D human tracking. The work presented in this paper is an extension of our previously published work (Li et al. 2006). In (Li et al. 2006), a globally coordinated mixture of factor analyzers is learned from motion capture data to parameterize a low-dimensional manifold where the state of the multiple hypothesis tracker resides. However, the number of factor analyzers and the dimensionality of the latent space are chosen empirically. In this paper, we provide a variational Bayesian learning algorithm so that the optimal model setup is automatically determined during learning. This solves one of the open problems as discussed in (Li et al. 2006).

3 Overview

There are two main components in the proposed tracking algorithm as shown in Fig. 1. The first component is an offline algorithm that learns a bidirectional mapping function between the low-dimensional manifold and the high-dimensional human pose manifold. The second component is an online algorithm for articulated human pose tracking that makes use of a modified multiple hypothesis tracking algorithm; the state space of this multiple hypothesis tracker lies on the low-dimensional manifold.

One key step in the modified multiple hypothesis tracking method is the likelihood computation using the hypotheses generated from the low-dimensional state space. Hence, a mapping function is needed to map the low-dimensional hypotheses to the 3D human body poses lie on the high-dimensional pose manifold. Let $\mathbf{x}$ denote the pose on the high-dimensional pose manifold, and $\mathbf{g}$ be the corresponding point on the low-dimensional manifold. The goal of our offline learning is to find the mapping function: $f_{\mathbf{g}\rightarrow\mathbf{x}}$ such

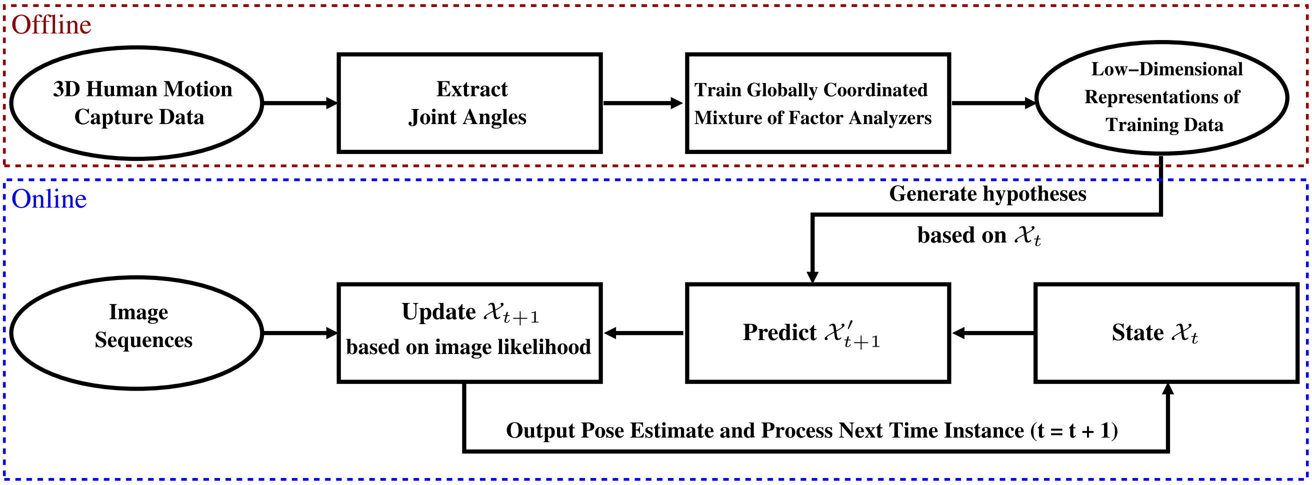
Offline
3D Human Motion Capture Data
Extract Joint Angles
Train Globally Coordinated Mixture of Factor Analyzers
Low–Dimensional Representations of Training Data
Online
Generate hypotheses based on $\mathcal{X}_t$
Image Sequences
Update $\mathcal{X}_{t+1}$ based on image likelihood
Predict $\mathcal{X}'_{t+1}$
State $\mathcal{X}_t$
Output Pose Estimate and Process Next Time Instance (t = t + 1)

Fig. 1 The overview of the system. To update $\mathcal{X}_{t+1}$, first the hypotheses generated from the low-dimensional state space are mapped to pose space using $f_{\mathbf{g}\rightarrow\mathbf{x}}$. The mapped poses are then projected onto the image to compute image likelihood for weighting the hypotheses. The output pose is obtained using $f_{\mathbf{g}\rightarrow\mathbf{x}}$ with the input from the updated state estimate $\mathcal{X}_{t+1}$

that:

$$\mathbf{x} = f_{\mathbf{g}\to\mathbf{x}}(\mathbf{g}) + \mathbf{n}_{\mathbf{x}}, \tag{1}$$

where $\mathbf{n}_{\mathbf{x}}$ is a zero-mean, white Gaussian noise process. One can also view this mapping function as the measurement function in the Kalman filter.

We propose to approximate $f_{\mathbf{g}\to\mathbf{x}}$ (and hence $f_{\mathbf{x}\to\mathbf{g}}$ since the mapping is bidirectional) using piecewise linear functions. It is desirable to have $f_{\mathbf{g}\to\mathbf{x}}$ together with $f_{\mathbf{x}\to\mathbf{g}}$, as $f_{\mathbf{g}\to\mathbf{x}}$ can be used to generate an observation from a point in the latent space, and $f_{\mathbf{x}\to\mathbf{g}}$ can be used to map an observation to a point in the latent space. Our problem of approximating the mapping function using piecewise linear functions can be formulated as a parameter estimation problem for the combination of these linear functions. In order to automatically determine the number of linear functions needed to approximate $f_{\mathbf{g}\to\mathbf{x}}$ and the dimensionality of $\mathbf{g}$, we will derive a variational Bayesian learning method.

In the following sections, we first give a quick review for the globally coordinated mixture of factor analyzers (GCMFA) in Sect. 4. The variational Bayesian learning algorithm of the GCMFA is proposed Sect. 5. Application of the learning algorithm to the HumanEvaI datasets (Sigal and Black 2006) is described in Sect. 6. First we apply this learning algorithm to obtain the globally coordinated mixture of factor analyzers from motion capture data in Sect. 6.1. The multiple hypothesis tracking algorithm that makes use of the resulting dimensionality reduced state space is then described in Sect. 6.2. Experimental evaluation of this approach is given in Sect. 7, where the HumanEvaI datasets are used in 3D human tracking. Finally, we conclude and discuss future work in Sect. 8.

4 Learning the Globally Coordinated Model

The goal of our off-line learning algorithm is to learn mapping functions $f_{\mathbf{g}\to\mathbf{x}}$ and $f_{\mathbf{x}\to\mathbf{g}}$. The function $f_{\mathbf{x}\to\mathbf{g}}$ maps high-dimensional observations to their corresponding low-dimensional representations. The function $f_{\mathbf{g}\to\mathbf{x}}$ does the inverse. Tables 1 and 2 explain the variables and parameters of the GCMFA model.

We start with the basic building block of the GCMFA model, which is called Factor Analyzer (FA). FA performs linear dimensionality reduction by modeling a linear relationship between the observed data $\mathbf{x}$ and the corresponding latent low-dimensional representation $\mathbf{z}$: $\mathbf{x} = \boldsymbol{\Lambda}\mathbf{z} + \boldsymbol{\mu} + \boldsymbol{\nu}$, where the factor loading matrix $\boldsymbol{\Lambda}$ is a $D \times d$ rectangular matrix that lifts the latent representation $\mathbf{z}$ to the observation space; the latent variable $\mathbf{z}$ follows a Gaussian distribution with zero mean and identity covariance matrix, $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$; $\boldsymbol{\mu}$ is the factor mean and $\boldsymbol{\nu}$ is the noise random variable and follows a Gaussian distribution with zero mean and noise covariance matrix $\boldsymbol{\Psi}$, $\boldsymbol{\nu} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Psi})$.

The next step to obtain the GCMFA model is to assemble the multiple linear FAs together. Given the low-dimensional representation $\mathbf{z}^{(s)}$ from factor analyzer s, the linear process to obtain $\mathbf{x}$ now takes the form of:

$$\mathbf{x} = \boldsymbol{\Lambda}^{(s)}\mathbf{z}^{(s)} + \boldsymbol{\mu}^{(s)} + \boldsymbol{\nu}, \tag{2}$$

where we use superscript s in parentheses to index the parameters for different FAs. We treat $\boldsymbol{\nu}$ as the observation noise process and as a result, the same noise covariance matrix $\boldsymbol{\Psi}$ is used for all the FA's.

A straightforward solution would be to use a mixture of factor analyzers (MFA) as proposed in (Ghahramani and

Table 1 Variables used in the GCMFA model. d represents the dimensionality of the low-dimensional representations; D is the dimensionality of the high-dimensional observations (poses); N denotes the length of the training data and S refers to the number of factor analyzers (FA) in the mixture model

Symbol	Size	Description
$\mathbf{x}_n$	$D \times 1$	n-th observation vector
$\mathbf{z}_n^{(s)}$	$d \times 1$	n-th local latent space representation of $\mathbf{x}_n$ in the s-th FA
$\mathbf{g}_n^{(s)}$	$d \times 1$	n-th global latent space representation of $\mathbf{x}_n$ in the s-th FA
$\mathbf{x}_{1:N}$	$D \times N$	training observation sequence

Table 2 Parameters used in the GCMFA model

Symbol	Size	Description
$\boldsymbol{\pi}^{(s)}$	scalar	prior distribution of the s-th FA
$\boldsymbol{\kappa}^{(s)}$	$d \times 1$	the mean of the s-th FA in the latent space
$\boldsymbol{\Sigma}^{(s)}$	$d \times 1$	the covariance of the s-th FA in the latent space
$\boldsymbol{\mu}^{(s)}$	$D \times 1$	the mean of the s-th FA in the observation space
$\boldsymbol{\Lambda}^{(s)}$	$D \times d$	factor loading matrix of the s-th FA
$\boldsymbol{\Psi}$	$D \times D$	observation noise covariance matrix

Hinton 1996). MFA is used to describe a low-dimensional density model of high-dimensional data and it parameterizes a joint distribution over observation $\mathbf{x}$ and hidden variables $\mathbf{z}$:

$$p(\mathbf{x}, s, \mathbf{z}^{(s)}) = p(\mathbf{x}|s, \mathbf{z}^{(s)})p(\mathbf{z}^{(s)}|s)p(s), \tag{3}$$

where $\mathbf{x} \in \mathbb{R}^D$, $\mathbf{z}^{(s)} \in \mathbb{R}^d$ and $D \gg d$. The discrete variable $s \in \{1, \ldots, S\}$ is the index of s-th FA in the mixture and $\mathbf{z}^{(s)}$ is the local coordinate of the low-dimensional representation in s-th FA. In MFA, it is assumed that data is sampled from different FAs in the mixture with prior probability $\boldsymbol{\pi}^{(s)}$. Therefore, the marginal data distribution $p(\mathbf{x})$ can be obtained by integrating over the low-dimensional representation $\mathbf{z}$'s and summing over all the factor analyzers in the mixture. Based on Eqs. 2 and 3, the resulting $p(\mathbf{x})$ is a mixture of Gaussian distributions with parameterized covariance matrices of the form:

$$\begin{aligned} p(\mathbf{x}) = \sum_s \boldsymbol{\pi}^{(s)} \left| \boldsymbol{\Lambda}^{(s)} \left[\boldsymbol{\Lambda}^{(s)}\right]^\mathsf{T} + \boldsymbol{\Psi} \right|^{-1/2} \\ \times \exp\left\{ -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}^{(s)})^\mathsf{T} \left(\boldsymbol{\Lambda}^{(s)} \left[\boldsymbol{\Lambda}^{(s)}\right]^\mathsf{T} + \boldsymbol{\Psi} \right)^{-1} \\ \times (\mathbf{x} - \boldsymbol{\mu}^{(s)}) \right\}. \end{aligned} \tag{4}$$

To estimate the model parameters, $\boldsymbol{\theta}$, where $\boldsymbol{\theta} = \{\{\boldsymbol{\pi}^{(s)}, \boldsymbol{\mu}^{(s)}, \boldsymbol{\Lambda}^{(s)}\}_{s=1}^S, \boldsymbol{\Psi}\}$ for all the FAs in the mixture, an EM algorithm is proposed in (Ghahramani and Hinton 1996) that attempts to maximize the total log likelihood of $\log p(\mathbf{x})$, which is summed over all training samples.

Unfortunately, this type of mixture model is only good for density modeling. It does not describe a single, coherent low-dimensional coordinate system for the data since there is no constraint for the local coordinates $\mathbf{z}^{(s)}$ of each FA to agree. Given that the local coordinates $\mathbf{z}^{(s)}$ follow a Gaussian distribution with zero mean and identity covariance matrix, they are subject to arbitrary rotation with rotation matrix $\mathbf{R}$ as $\mathbf{R}\mathbf{R}^\mathsf{T} = \mathbf{I}$. This means that we could apply an arbitrary rotation to $\mathbf{z}^{(s)}$ without changing the marginal data distribution p($\mathbf{x}$). Thus, maximum likelihood estimation in MFAs does not favor any particular alignment; instead, it would produce models whose internal representations change unpredictably as one traverses connected paths on the manifold.

The graphical model of our globally coordinated MFA is shown in Fig. 2(b) while the model proposed in (Roweis et al. 2001) is shown in Fig. 2(a). As in (Roweis et al. 2001), the global coordination is achieved by maximizing the likelihood of data with an additional variational penalty term to encourage the internal coordinates $\mathbf{z}^{(s)}$of the FAs to agree. This means that we prefer a global coordination scheme to produce a manifold so that as one traverses a connected path on the manifold, the internal coordinates change smoothly and continuously even when the path crosses the domains of many different local models. In our implementation, we choose to use a model that is similar to the one proposed in (Verbeek 2006) by treating the transformations needed to obtain globally consistent coordinates $\mathbf{g}$ from local coordinates $\mathbf{z}^{(s)}$ to be deterministic. As a result, we can take $\mathbf{z}^{(s)}$ out from our graphical model as shown in Fig. 2(b) and we only need to estimate an additional set of alignment parameters $\boldsymbol{\kappa}^{(s)}$ and $\boldsymbol{\Sigma}^{(s)}$, which correspond to the translation and rotation parameters to align all the FAs in the mixture so that a globally consistent representation $\mathbf{g}$ of $\mathbf{x}$ can be obtained. The difference between our model and the model proposed in (Verbeek 2006) is that we treat the observation noise as sensor noise and we use the same noise covariance $\boldsymbol{\Psi}$ for all the FAs in the mixture.

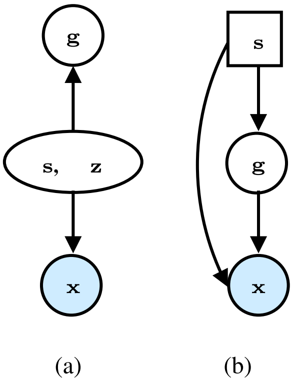

Fig. 2 (**a**) The original model defined in (Roweis et al. 2001). (**b**) The modified model proposed in (Verbeek 2006) for the globally coordinated of mixture of factor analyzers. $\mathbf{x}$, $\mathbf{z}$, $\mathbf{g}$ and s represents the variables used in the model as explained in Table 1. The model shown in (**b**) is different from the model in (**a**) as the local low-dimensional representation $\mathbf{z}$ of the corresponding observed $\mathbf{x}$ does not appear in the model. This is due to the deterministic transformation between the global low-dimensional representation $\mathbf{g}$ and the local coordinate $\mathbf{z}$. We choose to use the model in (**b**) given its computation efficiency. Both the discrete variable s and the continuous variable $\mathbf{g}$ are hidden. Only $\mathbf{x}$ in the *shaded node* is observed

The additional variational penalty term to enforce global consistency is obtained by introducing a family of unimodal distributions of factorized form:

$$q(\mathbf{g}_n, s|\mathbf{x}_n) = q(\mathbf{g}_n|\mathbf{x}_n)q(s|\mathbf{x}_n),$$

where $q(\mathbf{g}_n|\mathbf{x}_n) \sim \mathcal{N}(\hat{\mathbf{g}}_n, \boldsymbol{\Sigma}_{\mathbf{g}_n})$ and $q(s|\mathbf{x}_n) = q_n^{(s)}$ is a scalar. $\hat{\mathbf{g}}_n$ is the expected value of low-dimensional coordinate $\mathbf{g}_n$ for n-th observation $\mathbf{x}_n$. The factorized form of $q(\mathbf{g}_n, s|\mathbf{x}_n)$ implies that $p(\mathbf{g}_n|\mathbf{x}_n, s_1) \approx p(\mathbf{g}_n|\mathbf{x}_n, s_2)$ and $\mathbf{g}_n$ is independent of FA s for a given data point $\mathbf{x}_n$. These exactly are the constraints we want to achieve in order to obtain a globally consistent latent representation $\mathbf{g}_n$ for the corresponding high-dimensional observation data $\mathbf{x}_n$.

The parameters of GCMFA $\boldsymbol{\theta} = \{\{\boldsymbol{\pi}^{(s)}, \boldsymbol{\mu}^{(s)}, \boldsymbol{\Lambda}^{(s)}, \boldsymbol{\kappa}^{(s)}, \boldsymbol{\Sigma}^{(s)}\}_{s=1}^S, \boldsymbol{\Psi}\}$ are estimated by optimizing the following objective function:

$$\Phi = \sum_{n,s} \int q(\mathbf{g}_n, s|\mathbf{x}_n) \log \frac{p(\mathbf{x}_n, \mathbf{g}_n, s)}{q(\mathbf{g}_n, s|\mathbf{x}_n)} d\mathbf{g}. \tag{5}$$

Φ is a lower-bound on the total data log-likelihood $\log p(\mathbf{x})$ using variational distribution $q(\mathbf{g}_n, s|\mathbf{x}_n)$. We estimate the GCMFA parameters $\boldsymbol{\theta}$ together with the variational regularizing parameters $\{\hat{\mathbf{g}}_n, \boldsymbol{\Sigma}_{\mathbf{g}_n}, q_n^{(s)}\}$ by iteratively optimizing Φ via coordinate ascent in learning. The learning algorithm is summarized in Appendix.

As a result of the above formulation, the mapping functions $f_{\mathbf{g}\to\mathbf{x}}$ and $f_{\mathbf{x}\to\mathbf{g}}$ are described by the following probabilistic relations between $\mathbf{x}_n$ and $\mathbf{g}_n$:

$$p(\mathbf{g}_n|\mathbf{x}_n) = \sum_s p(\mathbf{g}_n|\mathbf{x}_n, s) p(s|\mathbf{x}_n), \tag{6}$$

$$p(\mathbf{x}_n|\mathbf{g}_n) = \sum_s p(\mathbf{x}_n|\mathbf{g}_n, s) p(s|\mathbf{g}_n). \tag{7}$$

The main issue with this maximum likelihood learning approach is that the model structure is chosen *a priori* and a coordinate ascent method is used to determine the parameter values. For model selection, one typical approach is to try many possible model setups and the best of these is chosen. However, this tends to compound the training cost as many models need to be tested. Moreover, since the parameter learning is prone to local optima, we might unwittingly end up comparing a "good" GCMFA to a "bad" GCMFA with different numbers of factors and latent dimensions. The "bad" GCMFA might be chosen if the learned local optimal parameters of the "good" GCMFA actually lead to a lower likelihood than those of the "bad" GCMFA.

Because of these problems, we propose to use a top-down variational Bayesian formulation that enables automatic optimal model selection. It is a top-down approach in the sense that the training starts with a single component model and component splitting is performed iteratively until there is no further improvement based on a variational Bayesian criterion. Therefore, our proposed solution allows model selection during parameter learning. This is different from (Verbeek 2006) where a variant of the EM algorithm is used for the learning of the model parameters with a fixed model structure.

5 Variational Bayesian Learning

A Bayesian approach tackles the model selection problem by treating the model parameters $\boldsymbol{\theta}$ as unknown entities and averaging over the ensemble of models they produce. The marginal likelihood $p(\mathbf{x})$ of the MFA for Bayesian learning can be rewritten as:

$$\begin{aligned} p(\mathbf{x}) &= \int d\boldsymbol{\theta}\, p(\boldsymbol{\theta}) p(\mathbf{x}|\boldsymbol{\theta}) \\ &= \int d\boldsymbol{\pi}\, p(\boldsymbol{\pi}) \int d\boldsymbol{\Lambda} p(\boldsymbol{\Lambda}) \int d\boldsymbol{\mu} p(\boldsymbol{\mu}) \\ &\quad \times \prod_{n=1}^{N} \int d\mathbf{z}_n p(\mathbf{z}_n) \\ &\quad \times \left[\sum_{s=1}^{S} p(s|\boldsymbol{\pi}) p(\mathbf{x}_n|s, \mathbf{z}_n^{(s)}, \boldsymbol{\Lambda}^{(s)}, \boldsymbol{\mu}^{(s)}, \boldsymbol{\Psi}) \right]. \end{aligned} \tag{8}$$

By integrating out the model parameters, the principle of Occam's razor is incorporated naturally (Jefferys and Berger 1992; MacKay 1992). From Eq. 8, the factorization of the model parameter distributions are derived from the structure of the graphical model shown in Fig. 2(a). The superscripts (s) for $\boldsymbol{\pi}$, $\boldsymbol{\Lambda}$, $\boldsymbol{\mu}$ and $\mathbf{z}_n$ are dropped because they represent the terms after summing over all factor analyzers. We choose these notations to make the equation more succinct. Instead of using $\boldsymbol{\pi}^{(s)}$, we use $p(s|\boldsymbol{\pi})$ for clear explanation as we later use variational distribution $q(s)$ to approximate $p(s|\boldsymbol{\pi})$. Similar notation conventions have been adopted in the lower bounds defined in Eqs. 10 and 11.

While the Bayesian approach provides a mechanism for automatic model selection, the integrals in Eq. 8 are often computationally intractable. To approximate these integrals, one can use sampling based approaches (Rasmussen 2000; Richardson and Green 1997) or analytical local Gaussian approximation based approaches (Cheeseman and Stutz 1996; Kass and Raftery 1995; Schwarz 1978). However, sampling approaches tend to be slow and it is generally difficult to assess their convergence, while the analytical approximation approaches are based on the large data limit (Beal 2003), and the local Gaussian approximation is not suitable for bounded or positive parameters, such as the mixing proportions of the model.

We adopt a variational Bayesian (VB) approach in our formulation. Using Jensen's inequality, a lower bound can be formulated using the log marginal likelihood $p(\mathbf{x})$ by introducing a variational distribution $q(\boldsymbol{\theta})$ over the model parameters $\boldsymbol{\theta}$:

$$\begin{aligned} \mathcal{L} &\equiv \log p(\mathbf{x}) \\ &= \log \int d\boldsymbol{\theta}\, p(\mathbf{x}, \boldsymbol{\theta}) \\ &\geq \int d\boldsymbol{\theta} q(\boldsymbol{\theta}) \log \frac{p(\mathbf{x}, \boldsymbol{\theta})}{q(\boldsymbol{\theta})} \\ &= -KL(q(\boldsymbol{\theta}) \| p(\boldsymbol{\theta})) + \int d\boldsymbol{\theta} q(\boldsymbol{\theta}) \ln p(\mathbf{x}|\boldsymbol{\theta}) \\ &\equiv \mathcal{F}^{(VB)}. \end{aligned} \tag{9}$$

Based on the above equation, maximizing $\mathcal{F}^{(VB)}$ is equivalent to minimizing the KL-divergence between $q(\boldsymbol{\theta})$ and $p(\boldsymbol{\theta}|\mathbf{x})$, since $p(\mathbf{x}, \boldsymbol{\theta}) = p(\boldsymbol{\theta}) p(\mathbf{x}|\boldsymbol{\theta})$. As a result, we can use a tractable $q(\cdot)$ to approximate the intractable posteriors.

Table 3 MFA model parameter priors. $\mathcal{N}(\cdot,\cdot)$ stands for normal distribution; $\mathcal{G}(\cdot,\cdot)$ refers to Gamma distribution and $\mathcal{D}(\cdot)$ represents Dirichlet distribution. We exclude the prior distribution of the observation noise, $\mathbf{\Psi}$, since it does not grow with the model complexity. The parameters in different components are independent, e.g., $p(\mathbf{\Lambda}|\boldsymbol{\rho},\boldsymbol{\beta}^*) = \prod_{s=1}^{S} p(\mathbf{\Lambda}^{(s)}|\boldsymbol{\rho}^{(s)},\boldsymbol{\beta}^*)$

Parameter	Prior
$p(\boldsymbol{\pi}\|\mathbf{u}^*)$	$\boldsymbol{\pi} \sim \mathcal{D}(\mathbf{u}^*)$, where $\mathbf{u}^* = ((\mathbf{u}^*)^{(1)},\ldots,(\mathbf{u}^*)^{(s)},\ldots,(\mathbf{u}^*)^{(S)})$
$p([\mathbf{\Lambda}^{(s)}]^{\mathsf{T}}_{j\cdot}\|\boldsymbol{\rho}^{(s)},\boldsymbol{\beta}^*_{\mathbf{\Lambda}})$	$[\mathbf{\Lambda}^{(s)}]^{\mathsf{T}}_{j,\cdot} \sim \mathcal{N}(0,[\boldsymbol{\rho}^{(s)}]^{-1}[\mathrm{diag}(\boldsymbol{\beta}^*_{\mathbf{\Lambda}})]^{-1})$, for $j=1,\ldots,D$.
$p(\boldsymbol{\rho}^{(s)}\|a^*,\mathbf{b}^*)$	$[\boldsymbol{\rho}^{(s)}] \sim \mathcal{G}(a^*,\mathbf{b}^*)$, for $i=1,\ldots,d_{max}$.
$p(\boldsymbol{\mu}^{(s)}\|\boldsymbol{\mu}^*,\boldsymbol{\beta}^*_{\boldsymbol{\mu}})$	$\boldsymbol{\mu}^{(s)} \sim \mathcal{N}(\boldsymbol{\mu}^*,[\mathrm{diag}(\boldsymbol{\beta}^*_{\boldsymbol{\mu}})]^{-1})$

Compared to the sampling based and the analytical approximation based approaches, the VB approach has the following advantages:

(1) convergence can be easily assessed by monitoring $\mathcal{F}^{(VB)}$,
(2) a suitable q leads to an efficient solution, and
(3) the form of q can be tailored to each parameter.

In (Beal 2003), a variational Bayesian learning algorithm is derived for the of mixture of factor analyzers (MFA). Our proposed method extends the work of (Beal 2003) for the variational Bayesian learning of the globally coordinated mixture of factor analyzers (VBGCMFA). To help the reader in understanding our derivation of VBGCMFA, we first review the derivation of (Beal 2003) (Sect. 5.1). We then show that the VBMFA algorithm can be easily extended to our VBGCMFA model (Sect. 5.2).

To demonstrate the strength of our variational Bayesian formulation, we apply it on a standard data dimensionality reduction and reconstruction task (Sect. 5.3). We compare its performance with a two-step solution proposed in (Teh and Roweis 2002) which we used in our previous work (Li et al. 2006).

In the following derivations, we use $\boldsymbol{\theta}$ to denote the parameters of the mixture model. The parameters that describe the distribution of the model parameters are referred to as hyperparameters and denoted as $\mathbf{\Theta}$.

5.1 Variational Bayesian Mixture of Factor Analyzers (VBMFA)

In (Beal 2003), conjugate priors for $\boldsymbol{\theta}$ in Eq. 9 are used to simplify inference. We refer the interested reader to (Beal 2003) for the detailed explanation for the choices of prior distributions. We provide an overview of the VB solution for the MFA of (Beal 2003) in this section, so that the VB derivation for GCMFA in Sect. 5.2 is self-contained.

First, we give the prior distributions of the MFA model. The model priors are listed in Table 3. The priors we choose are conjugate to the likelihood terms in the last term of Eq. 4. Variables with superscript "$*$" are hyperparamters to the MFA model. We use $[\cdot]_{j}\cdot$ to indicate the j-th row and $[\cdot]_{ji}$ to index the entry at j-th row and i-th column of the matrix inside the brackets. If the variable inside the brackets is a vector, then single index is used to denote the entries in the vector. We use $[\cdot]^{\mathsf{T}}$ to represent the matrix/vector transpose operation and $[\cdot]^{-1}$ to denote the inverse operation on the matrix or element-wise inverse operation on the vector.

The dimensionality of the latent space is determined through the precision parameter $\boldsymbol{\rho}^{(s)}$ using the automatic relevance determination mechanism proposed in (MacKay 1996). For the precision parameter vector $\boldsymbol{\rho}^{(s)}$, we set the dimensionality of $\boldsymbol{\rho}^{(s)}$ to be d_{max}, i.e., the maximal possible dimensionality of the latent space. If one of these precisions $[\boldsymbol{\rho}^{(s)}]_i \to \infty$ (for $i=1,\ldots,d_{max}$), then the i-th factor of the s-th FA will have to be very close to zero which leads to the analyzer to ignore this factor and thus reduce that dimension of the corresponding latent representation in the s-th FA.

From Eq. 9, we have the following derivation to obtain the lower bound $\mathcal{F}^{(VB)}$:

$$\begin{aligned}
\mathcal{L} &\equiv \ln p(\mathbf{x}) \\
&= \ln\Bigg(\int d\boldsymbol{\pi}\, p(\boldsymbol{\pi}|\mathbf{u}^*) \int d\boldsymbol{\rho}\, p(\boldsymbol{\rho}|a^*, b^*) \\
&\quad \times \int d\mathbf{\Lambda}\, p(\mathbf{\Lambda}|\boldsymbol{\rho},\boldsymbol{\beta}^*_{\mathbf{\Lambda}}) \int d\boldsymbol{\mu}\, p(\boldsymbol{\mu}|\boldsymbol{\mu}^*,\boldsymbol{\beta}^*_{\boldsymbol{\mu}}) \\
&\quad \times \prod_{n=1}^{N} \int d\mathbf{z}_n\, p(\mathbf{z}_n) \\
&\quad \times \Bigg[\sum_{s=1}^{S} p(s|\boldsymbol{\pi})\, p(\mathbf{x}_n|s,\mathbf{z}_n,\mathbf{\Lambda}^{(s)},\boldsymbol{\mu}^{(s)},\mathbf{\Psi})\Bigg]\Bigg) \\
&\geq \int d\boldsymbol{\pi}\, d\boldsymbol{\rho}\, d\mathbf{\Lambda}\, d\boldsymbol{\mu}\; q(\boldsymbol{\pi},\boldsymbol{\rho},\mathbf{\Lambda},\boldsymbol{\mu}) \\
&\quad \times \Bigg(\ln\frac{p(\boldsymbol{\pi}|\mathbf{u}^*)\, p(\boldsymbol{\rho}|a^*, b^*)\, p(\mathbf{\Lambda}|\boldsymbol{\rho},\boldsymbol{\beta}^*_{\mathbf{\Lambda}})\, p(\boldsymbol{\mu}|\boldsymbol{\mu}^*,\boldsymbol{\beta}^*_{\boldsymbol{\mu}})}{q(\boldsymbol{\pi},\boldsymbol{\rho},\mathbf{\Lambda},\boldsymbol{\mu})} \\
&\quad + \sum_{n=1}^{N}\Bigg[\sum_{s=1}^{S}\int d\mathbf{z}_n q(s,\mathbf{z}_n)\Bigg(\ln\frac{p(s|\boldsymbol{\pi})\, p(\mathbf{z}_n)}{q(s,\mathbf{z}_n)}
\end{aligned}$$

Algorithm 1 VB learning algorithm for MFA

```
Initialize parameters. Initialize hidden variables and state priors.
for $n_1 = 1$:max_iter_1 do
    VBM Step:
        Compute the expected natural parameters of $q(\boldsymbol{\theta})$.

    VBE Step:
        Compute the sufficient statistics of hidden variable distributions $q(\mathbf{s})$ and $q(\mathbf{g})$.

    Optimize hyperparameters.
end for
```

$$+ \ln p(\mathbf{x}_n|s, \mathbf{z}_n, \boldsymbol{\Lambda}^{(s)}, \boldsymbol{\mu}^{(s)}, \boldsymbol{\Psi})\Big)\Big]\Big]\Big)$$

$$\equiv \mathcal{F}^{(VB)}(q(\boldsymbol{\theta}), q(s, \mathbf{z}), \boldsymbol{\Theta}). \tag{10}$$

Thus, the resulting lower bound $\mathcal{F}^{(VB)}$ is a functional over the variational posterior distributions of the model parameters $q(\boldsymbol{\theta})$, a functional of the variational posterior distributions of the hidden variables of every data point, $q(s, \mathbf{z})$ (we use $q(s, \mathbf{z})$ to denote the set of distributions), and also a function of the hyperparameters $\boldsymbol{\Theta}$.

The VB learning algorithm aims to optimize the lower bound $\mathcal{F}^{(VB)}$ defined in Eq. 10 with the model parameter priors defined in Table 3.

5.1.1 *Inferring the Number of FAs in the Mixture*

Our learning algorithm already provides an automatic relevance determination (ARD) mechanism to discover the local dimensionality for each FA in the mixture through the precision parameter $\boldsymbol{\rho}$. To infer the optimal number of FAs in the mixture, a top-down approach is used. Therefore, the training process starts with one FA and allows it to split if $\mathcal{F}^{(VB)}$ can be optimized though the splitting. The candidate for splitting is chosen stochastically with probability proportional to $\exp^{-\omega \mathcal{F}_s^{(VB)}}$, where ω is a temperature parameter to be set empirically and $\mathcal{F}_s^{(VB)}$ is the last bracketed term in Eq. 10 normalized by $\sum_{n=1}^{N} q(s_n)$. This splitting process causes $\mathcal{F}^{(VB)}$ to decrease as the newly spawned component is initialized by partitioning the responsibilities of the parent component for the data. If a split is legitimate, after optimization, the spawned components should fit the data better and overcome the penalty of the increasing model complexity. As a result, after the initial decrease, $\mathcal{F}^{(VB)}$ should recover or increase. By monitoring the progress of $\mathcal{F}^{(VB)}$, we can determine whether to accept this splitting or not.

This algorithm can be used to infer the dimensionality of the latent space and the number of FAs automatically. However, as pointed out in Sect. 4, we would prefer a globally coordinated mixture of factor analyzers. Thus, we must extend this formulation to incorporate the prior distributions for the global coordination parameters $\boldsymbol{\Sigma}^{(s)}$ and $\boldsymbol{\kappa}^{(s)}$ so that we can perform variational Bayesian learning for the GCMFA. This formulation will be described in the next section.

5.2 Variational Bayesian GCMFA (VBGCMFA)

The GCMFA described in Sect. 4 has two sets of additional parameters $\{\boldsymbol{\kappa}^{(s)}\}$ and $\{\boldsymbol{\Sigma}^{(s)}\}$ for global coordination. Since the size of these parameters increases with the number of FAs and the dimensionality of the latent space, we introduce hyperparameters for the prior distribution $\boldsymbol{\kappa}^{(s)}$ and $\boldsymbol{\Sigma}^{(s)}$. The prior for $\boldsymbol{\kappa}^{(s)}$ is Gaussian distributed with $p(\boldsymbol{\kappa}^{(s)}|\boldsymbol{\kappa}^{*}, [\mathrm{diag}(\boldsymbol{\beta}^{*}_{\boldsymbol{\kappa}})]^{-1})$ and a Gamma prior is placed on $\boldsymbol{\Sigma}^{(s)}$ such that $p(\boldsymbol{\Sigma}^{(s)}|a^{*}_{\boldsymbol{\Sigma}^{(s)}}, b^{*}_{\boldsymbol{\Sigma}^{(s)}})$. The corresponding lower bound can be derived as:

$$\mathcal{L} \equiv \ln p(\mathbf{x})$$

$$= \ln\Bigg(\int d\boldsymbol{\pi}\, p(\boldsymbol{\pi}|\mathbf{u}^{*}) \int d\boldsymbol{\rho}\, p(\boldsymbol{\rho}|a^{*}, b^{*}) \int d\boldsymbol{\Lambda}\, p(\boldsymbol{\Lambda}|\rho, \beta^{*}_{\boldsymbol{\Lambda}}) \int d\boldsymbol{\mu}\, p(\boldsymbol{\mu}|\boldsymbol{\mu}^{*}, \beta^{*}_{\boldsymbol{\mu}}) \int d\boldsymbol{\kappa}\, p(\boldsymbol{\kappa}|\boldsymbol{\kappa}^{*}, \beta^{*}_{\boldsymbol{\kappa}})$$

$$\times \int d\boldsymbol{\Sigma}\, p(\boldsymbol{\Sigma}|a^{*}_{\boldsymbol{\Sigma}}, b^{*}_{\boldsymbol{\Sigma}}) \sum_{n=1}^{N} \int d\mathbf{g}_n\, p(\mathbf{g}_n) \Bigg[\sum_{s=1}^{S} p(s|\boldsymbol{\pi}) p(\mathbf{x}_n|s, \mathbf{g}_n, \boldsymbol{\Lambda}^{(s)}, \boldsymbol{\mu}^{(s)}, \boldsymbol{\kappa}^{(s)}, \boldsymbol{\Sigma}^{(s)}, \boldsymbol{\Psi})\Bigg]\Bigg)$$

$$\geq \int d\boldsymbol{\pi}\, d\boldsymbol{\rho}\, d\boldsymbol{\Lambda}\, d\boldsymbol{\mu}\, d\boldsymbol{\kappa}\, d\boldsymbol{\Sigma}\; q(\boldsymbol{\pi}, \boldsymbol{\rho}, \boldsymbol{\Lambda}, \boldsymbol{\mu}, \boldsymbol{\kappa}, \boldsymbol{\Sigma})$$

$$
\times \left(\ln \frac{p(\boldsymbol{\pi}|\mathbf{u}^*)p(\boldsymbol{\rho}|a^*, b^*)p(\boldsymbol{\Lambda}|\boldsymbol{\rho}, \beta^*_{\boldsymbol{\Lambda}})p(\boldsymbol{\mu}|\boldsymbol{\mu}^*, \beta^*_{\boldsymbol{\mu}})p(\boldsymbol{\kappa}|\boldsymbol{\kappa}^*, \beta^*_{\boldsymbol{\kappa}})p(\boldsymbol{\Sigma}|a^*_{\boldsymbol{\Sigma}}, b^*_{\boldsymbol{\Sigma}})}{q(\boldsymbol{\pi}, \boldsymbol{\rho}, \boldsymbol{\Lambda}, \boldsymbol{\mu}, \boldsymbol{\kappa}, \boldsymbol{\Sigma})} + \sum_{n=1}^{N} \int d\mathbf{g}_n q(\mathbf{g}_n) \ln \frac{p(\mathbf{g}_n)}{q(\mathbf{g}_n)} \right.
$$

$$
\left. + \sum_{s=1}^{S} q(s) \ln \frac{p(s|\boldsymbol{\pi})}{q(s)} + \sum_{n=1}^{N} \int d\mathbf{g}_n q(\mathbf{g}_n) \left[\sum_{s=1}^{S} q(s) \ln p(\mathbf{x}_n|s, \mathbf{g}_n, \boldsymbol{\Lambda}^{(s)}, \boldsymbol{\mu}^{(s)}, \boldsymbol{\kappa}^{(s)}, \boldsymbol{\Sigma}^{(s)}, \boldsymbol{\Psi}) \right] \right)
$$

$$
\equiv \mathcal{F}^{(VB)}(q(\boldsymbol{\theta}, q(s), q(\mathbf{g}_n), \boldsymbol{\Theta})). \tag{11}
$$

The hidden variable distributions now are $q(s_n)$ and $q(\mathbf{g}_n)$ compared to $q(s_n)$ and $q(\mathbf{z}_n|s_n)$ in the VBMFA formulation. The factorization is from the enforcement of the global alignment constraint such that $q(\mathbf{g}, s_1|\mathbf{x}_n) \approx q(\mathbf{g}, s_1|\mathbf{x}_n)$ described in Sect. 1.

Given these changes, the same derivation procedure for VBMFA can be reused. Furthermore, the same top-down approach for splitting FAs in the mixture can still be used.

5.3 Comparison with the Post-Coordination Solution (Teh and Roweis 2002)

The goal of this experiment is to compare the variational Bayesian formulation of two different coordination schemes. The solution proposed in this paper advocates the idea of learning the coordination parameters in concert with the MFA parameters in a variational Bayesian framework. This is in contrast with (Teh and Roweis 2002), which employs a two-step approach: first a MFA is learned, then the coordination is performed by solving a generalized eigenvalue problem. To realize automatic model selection for the post-coordination solution, the algorithm proposed in (Beal 2003) is used as the first step in the post-coordination approach. The coordination step stays the same. We use the S-CURVE data with added noise of 0.06. 1200 3D data points are used for training. In both approaches, the variational Bayesian formulation gives the same number of factor analyzers and 2 as the dimension of the latent space. The experiment is conducted using Matlab on a 3.46 GHz Intel Pentium PC with 4 GB memory. The training for the post coordination method takes about 5 minutes while our method takes 8 minutes as there are more parameters to optimize in the variational Bayesian learning.

The embedding results are shown in Fig. 3. The first row shows the resulting globally coordinated MFA that is obtained using the post-coordination approach and the second row shows the results obtained using our method. We can see that even with noisy data, our proposed solution is still able to produce a good embedding. In contrast, the post-coordination approach uses many overlapping FAs and the data points in the resulting 2D embedding are not as evenly spaced out as in our method. This is due to the presence of noise in data; the factors in MFA could capture the wrong orientation in the first step since there is no constraint to force it to align with neighboring factors. This problem can be observed in the high curvature areas of the S-CURVE set. Quantitatively, if we look at the reconstruction errors reported in Table 4, the function $f_{\mathbf{g}\to\mathbf{x}}$ learned from our method also provides more accurate mapping from the latent coordinate $\mathbf{g}$ to its corresponding coordinate $\mathbf{x}$.

6 Application Demonstration on the HumanEvaI Data Sets

To demonstrate the application of the proposed variational Bayesian formulation for the one-step solution of the GCMFA, we first apply it to 10 motion capture sequences from HumanEvaI datasets to learn the latent space representation of the joint angles in Sect. 6.1. Then we make use of the latent space representations in a multiple hypothesis tracking algorithm in Sect. 6.2 for tracking corresponding test video sequences from HumanEvaI data set.

We use 10 motion capture sequences from Trial 3 as our training data.[1] These 10 sequences consist of two subjects (S2 and S3) performing various actions. These contains 5 different actions performed by two different subjects $S1$ and $S2$, including interesting and challenging sequences such as Throw/Catch and Box, as the limb movements in these actions are relatively fast and involve abrupt changes in movement directions and velocity.

6.1 Learning the Joint Angle Configurations

In our application for using VBGCMFA to learn the dimensionality reduced space, each training data sample per frame $\mathbf{x}_n$ is a column vector that consists of the exponential map representations of the joint angles computed from the motion capture data. We adopt the same 3D cylindrical model used in (Sidenbladh et al. 2000); we leave out the global translation from training. Hence, the human pose data sample per frame is represented as a 28-dimensional column vector.

[1] We refer interested readers to (Sigal and Black 2006) for the detailed setup of the motion capture sessions.

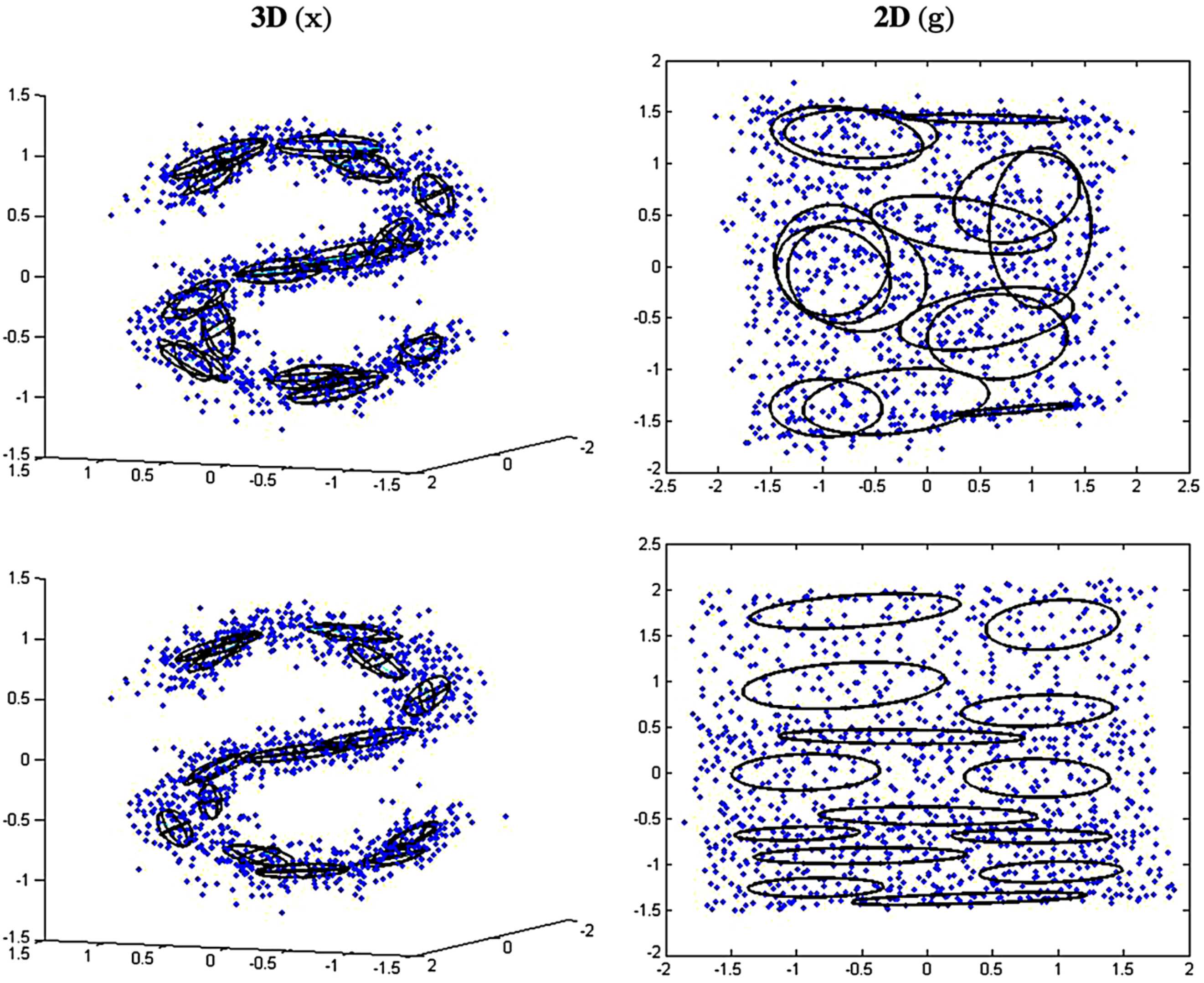

Fig. 3 Comparison of model learning and data embedding results. The data we used for training is the S-CURVE dataset from http://www.cs.toronto.edu/~roweis/lle/code.html. This is a data set of 1200 uniformly sampled data points from the manifold with added noise of 0.06 along each dimension. The visualization of the training dataset is shown in the 3D plots in the first row. The *first column* **a** shows the post-coordination method (Teh and Roweis 2002) where the mixture of factor analyzers (MFA) learning step is replaced by the variational Bayesian MFA (Beal 2003). The results for our method are shown in the *second column*. The ellipsoids in the 3D plots represent the FAs in the mixture and the ellipses are their corresponding 2D projections. It can be seen from the 2D plots that our method produces more evenly spaced out FAs while the FAs in the post-coordination method are mostly overlapped. This is undesirable as some of the FAs might be redundant if there is too much overlap among them. Furthermore, given a fixed number of FAs, overlapping FAs might cause some regions on the manifold uncovered. As evident in the plots for both 3D and 2D, our method is less susceptible to noise thanks to the global coordination constraint. This is especially visible in the high curvature areas of the S-CURVE as the noise has caused some of the FA's to orient towards the orthogonal direction of the tangential direction of the manifold

Table 4 Comparison of the Reconstruction Error. We take the inferred 2D coordinates **g** and make use of the mapping function $f_{\mathbf{g}\to\mathbf{x}}$ to reconstruct the 3D data. The error statistics reported here are the mean and variances of the root mean squared error between the reconstructed data and the training S-CURVE data

	Mean error	Variance of the error
Post-Coordination (Teh and Roweis 2002)	[0.6973, 0.8207, 0.4127]	[0.1919, 0.2186, 0.0665]
Our method	[0.6002, 0.2332, 0.3011]	[0.0863, 0.0211, 0.0373]

Table 5 Number of factor analyzers (S) and dimensionality of the latent space (d) obtained from applying the proposed VBGCMFA to the training motion capture sequences

Subject	Action	Length	Number of factor analyzers	Latent space dimension
S2	Walk	1500	10	4
S2	Jog	1500	12	5
S2	ThrowCatch	3000	9	4
S2	Gesture	3000	14	5
S2	Box	3000	19	6
S3	Walk	2000	13	4
S3	Jog	1500	15	6
S3	ThrowCatch	2000	7	5
S3	Gesture	1500	9	4
S3	Box	1500	11	7

In the VBGCMFA learning, the dimension d for the latent space coordinate $\mathbf{g}$ and the number of mixtures S are determined automatically as described in Sect. 5.2. Clusters in the latent space correspond to factor analyzers in the mixture. This cluster-based representation leads to a straightforward algorithm for multiple hypothesis tracking, as will be described in Sect. 6.2.

In the training, we start with a single factor analyzer and allow it to split during iteration until convergence. On average, training takes about 3–6 hours in Matlab on a 3.46 GHz Intel Pentium PC with 4 GB memory. The results from training are shown Table 5.

In (Li et al. 2006; Sminchisescu and Jepson 2004; Tian et al. 2005b; Urtasun et al. 2005), the dimensionality of the latent space is determined empirically, mostly 2–5 dimensions. The dimensionality of the latent space reported in Table 5 is obtained by examining the posterior distributions over $\boldsymbol{\nu}$, the precisions of each factor analyzer's columns, and thresholding on the mean of each distribution.

6.2 3D Articulated Human Tracking

In the application to 3D articulated human tracking, at each time step t, the tracker state vector is represented by $\mathcal{X}_t = (\mathbf{P}_t, \mathbf{g}_t)$, where $\mathbf{P}_t$ is the $3D$ location of the pelvis (which is the root of the kinematic chain of the 3D human model) and $\mathbf{g}_t$ is the point in latent space. Once the tracker state has been initialized, the basic idea of a filtering based tracking algorithm is to maintain a time-evolving probability distribution over the tracker state. Let $\mathbf{Z}_t$ denote the aggregation of past image observations, i.e., $\mathbf{Z}_t = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_t\}$. Assuming $\mathbf{z}_t$ is independent of $\mathbf{Z}_{t-1}$ given $\mathcal{X}_t$, we have the following standard equation:

$$p(\mathcal{X}_t|\mathbf{Z}_t) \propto p(\mathbf{z}_t|\mathcal{X}_t)p(\mathcal{X}_t|\mathbf{Z}_{t-1}). \qquad (12)$$

Here we use a multiple hypothesis tracker (MHT) together with the learned VBGCMFA model. As VBGCMFA provides clusters in the latent space, it is natural to make use of the centers of the clusters as the initial modes in the MHT, where $p(\mathbf{g}|s)$ follows a Gaussian distribution. Given that in each cluster, the points in the latent space represent the poses that are similar to each other in the original space, we can apply a much simpler dynamical model in the prediction step of the filtering algorithm. The modified MHT is summarized in Algorithm 2.

To compute the likelihood for the current prediction and the input video frame, first the silhouette of the current video frame is extracted through background subtraction. The predicted model is then projected onto the image and the chamfer matching cost between the projected model and the image silhouettes is considered to be proportional to the negative log-likelihood. The reader is referred to (Balan et al. 2005) for a more detailed discussion.

The MHT algorithm proposed here differs from the algorithm proposed in (Cham and Rehg 1999) in the use of the latent space to generate proposals in a principled way. In (Cham and Rehg 1999), the modes were selected empirically and the distributions were assumed to be piecewise Gaussian. In contrast, the output from the off-line learning algorithm (VBGCMFA) forms clusters, where each cluster is described by a Gaussian distribution in the latent space and the samples generated from the latent space are indeed drawn from a piecewise Gaussian distribution. The choice of modes to propagate over time becomes automatic given the statistics of the clusters in the latent space.

7 Experiments

Given the above implementation, we evaluate the proposed tracker's performance in comparison with two competing approaches on the 10 test video sequences from HumanEvaI Trial 2 of subjects S2 and S3 performing the actions as listed in Table 5. In the experiments, we chose to use the videos from cameras $C1$–$C3$. The reason for using multiview information is mainly due to the weak image feature as the silhouettes extracted from the videos are rather noisy.

```
Algorithm 2 A modified multiple hypothesis tracker
for each time instance $t$ do
    Prediction:
    generate the prior density $p(\mathcal{X}_t|\mathbf{Z}_{t-1})$ by passing through the modes of $p(\mathcal{X}_t|\mathbf{Z}_{t-1})$ through a simple constant velocity predictor.

    Likelihood computation:
(1)     Create the initial hypothesis seeds by sampling the distribution of $p(\mathcal{X}_t|\mathbf{Z}_{t-1})$. Note the samples of $\mathbf{g}$ are drawn around the modes of $\mathbf{G}$ in the latent space based on the covariance matrix of each cluster in the latent space.
(2)     Obtain the modes (local maxima) of the likelihood function $p(\mathbf{z}_t|\mathcal{X}_t)$ by computing the matching cost of the samples.
(3)     Measure the local statistics associated with each likelihood mode.

    Posterior density computation:
    The posterior density $p(\mathcal{X}_t|\mathbf{Z}_t)$ is updated through Eq. 12.
end for
```

Table 6 Error statistics of Fig. 4. The means (and standard deviations) of the 3D tracking errors (in millimeters) computed over 200 tracked frames

Action	Subject 2			Subject 3		
	APF (Deutscher et al. 2000)	GPLVMPF (Tian et al. 2005b)	Our Method	APF (Deutscher et al. 2000)	GPLVMPF (Tian et al. 2005b)	Our Method
Walking	107.77(50.58)	88.35(25.66)	68.67(24.66)	110.21(53.53)	87.39(21.69)	69.59(22.22)
Jog	120.53(55.67)	91.69(25.93)	72.14(54.66)	111.82(47.91)	99.05(21.90)	70.13(21.34)
Throw/Catch	107.68(46.71)	85.95(21.25)	68.03(22.18)	104.62(40.42)	84.50(23.01)	59.13(24.00)
Gesture	102.55(45.06)	84.63(18.60)	67.66(23.85)	100.37(33.78)	87.15(11.69)	50.61(18.53)
Box	107.68(34.09)	85.95(18.23)	70.02(22.74)	120.11(58.55)	90.34(25.60)	67.17(23.03)

The quantitative comparisons of our method are carried out against (1) annealed particle filtering (Deutscher et al. 2000) (APF), and (2) the tracking algorithm proposed by (Tian et al. 2005b) where the Gaussian process latent variable model (GPLVM) was used to reduced state space dimensionality of a particle filtering algorithm. We use GPLVMPF to refer this tracking algorithm. APF and GPLVMPF are chosen for comparison as both address the issue of sample impoverishment problem for particle filtering in 3D human tracking. Smart sampling (Deutscher et al. 2000) in the original state space is used in APF and a dimensionality reduced state space is used in GPLVMPF. Our work is similar to GPLVMPF as we also proposed a method to reduce the dimensionality of the state space.

The number of modes and dimensionality of the latent state space for our tracking algorithm are set according to Table 5. For APF, 5 layers and 500 particles for each layer are used. For GPLVMPF, the latent space dimensions for different sequences are set to be the same as the corresponding setup for our method (Table 5). 500 particles are used for our implementation of GPLVMPF.[2]

Three camera views are used for the implementation of all three tracking algorithms. The frame rate for both our proposed method and the method of (Tian et al. 2005b) on a 3.46 GHz machine with 4 GB RAM was approximately 0.6 minutes per frame, while the annealed particle filtering algorithm took 1.5 minutes per frame. In both our proposed algorithm and (Tian et al. 2005b), the global translation was modeled separately by simple linear dynamics learned from motion capture data.

In Table 6, the mean and the standard deviation of tracker error of the three tracking algorithms are reported. As proposed in (Balan et al. 2005), the error is measured as the absolute distance in millimeters between the true and estimated 3D marker positions on the body limbs. Fifteen markers are used, which correspond roughly to the locations of the joints and "ends" of the limbs.

Our method consistently outperforms the competing approaches. Moreover, with our method, the optimal model pa-

[2] Fewer particles were used in (Tian et al. 2005b) as the dimensionality of the latent space was only 2.

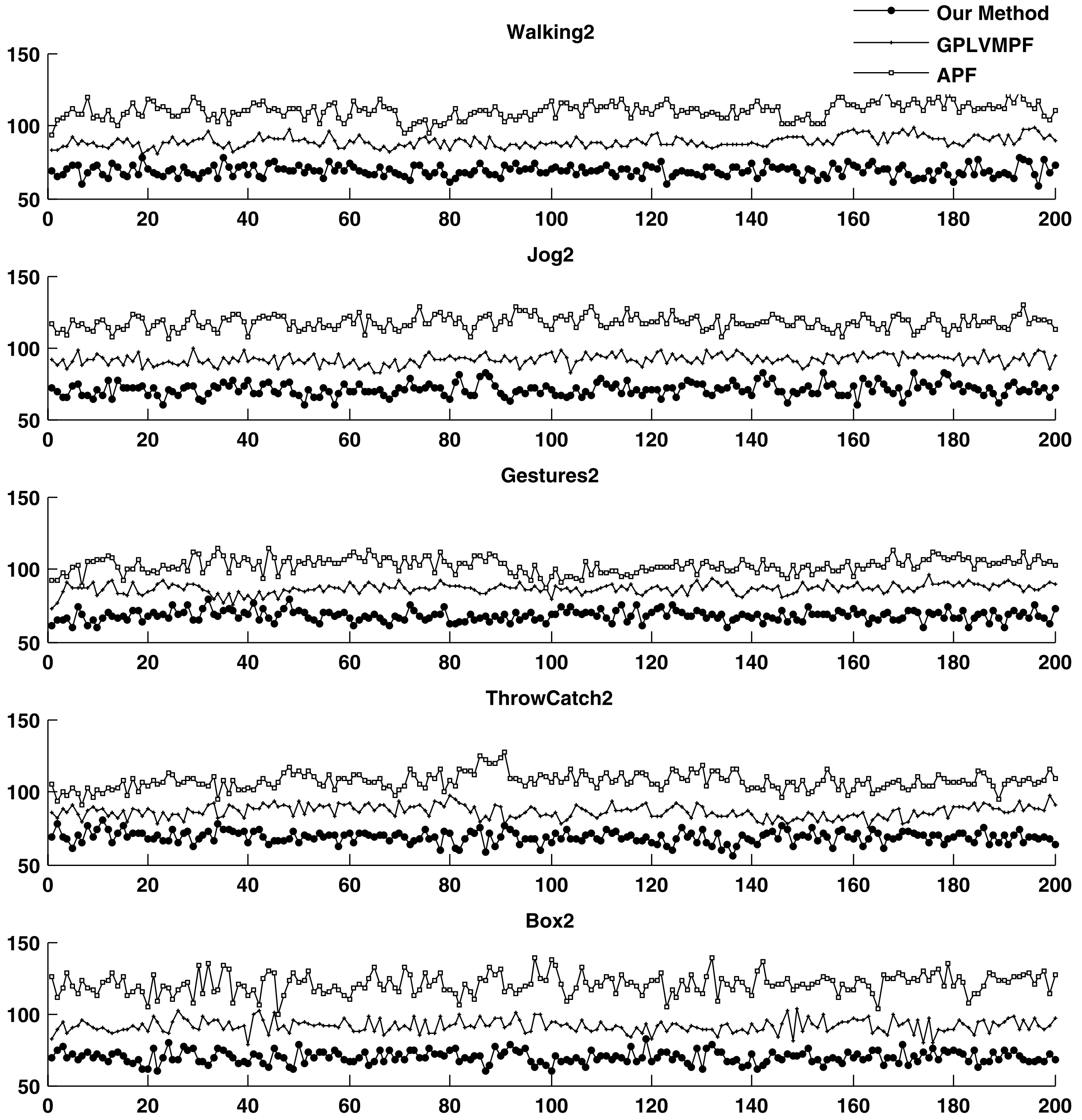

Fig. 4 3D Tracking error of S3 for the 5 actions. The *vertical axis* represents the mean tracker error in millimeters while the *horizontal axis* shows the frame indices

rameters were selected automatically. In Fig. 4, the mean 3D tracking errors of subject S3 for all 5 actions for all three methods are plotted. As can be seen in the plot, our method has the smallest mean error among the three methods. Compared to the results reported in (Poppe 2007b) where an exhaustive search of the (pose, image feature) pair database is used, the mean errors of method on actions like Throw/Catch and Box are about 40 mm better with smaller standard deviation too. For actions like Walking, Jog and Gesture, our method reports comparable numbers. This is because for actions like Throw/Catch and Box, the complex motion variations might be challenging for (Poppe 2007b) as the estimation is based on the interpolation of the top *k* retrieval results.

Figures 5 and 6 show example tracking results and the corresponding estimated 3D poses from the boxing se-

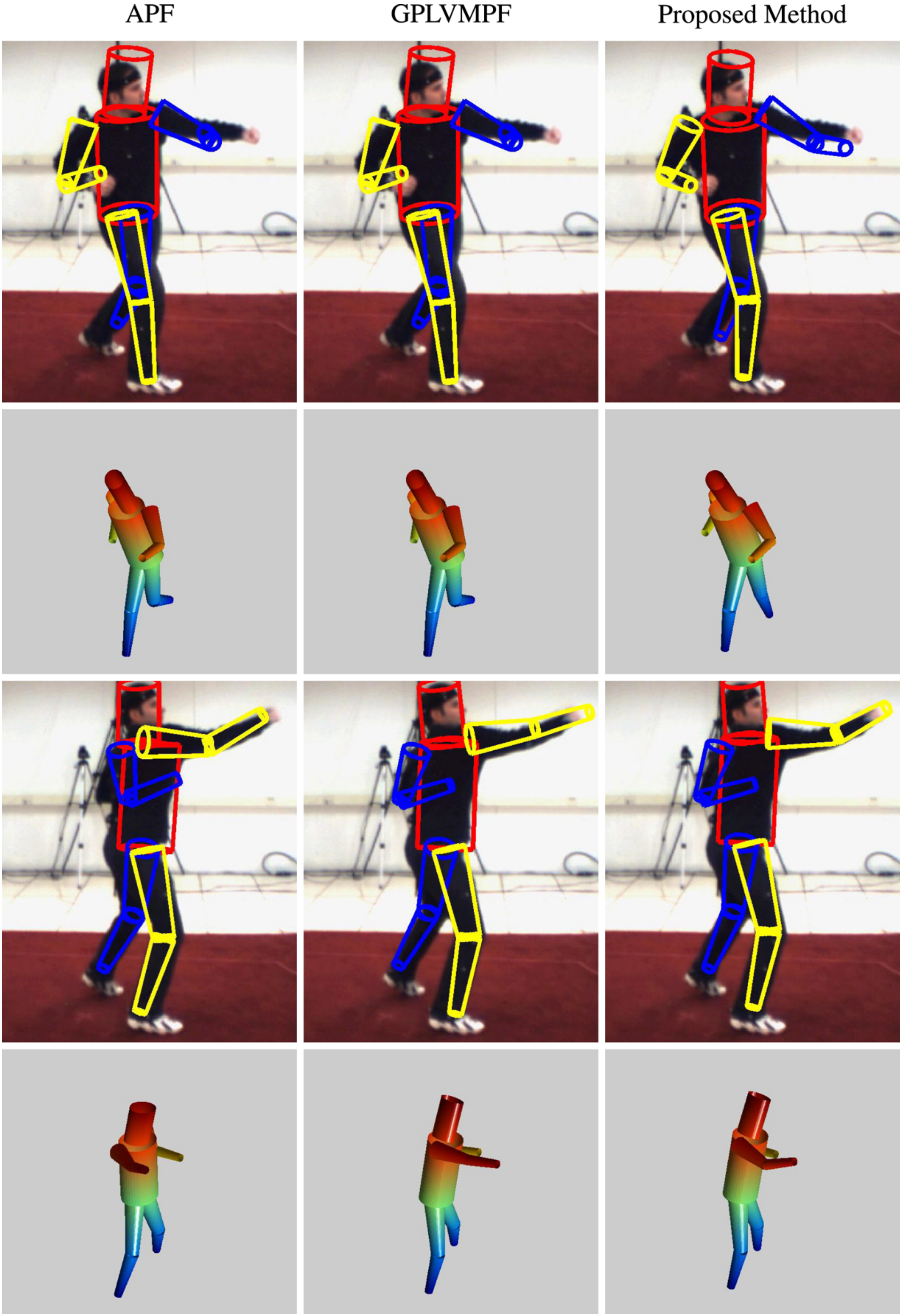

Fig. 5 Sample tracking result. The *first two rows* show the results of frame 35 from the test video sequence of S3 performing boxing. The *first two rows* show the results of frame 55

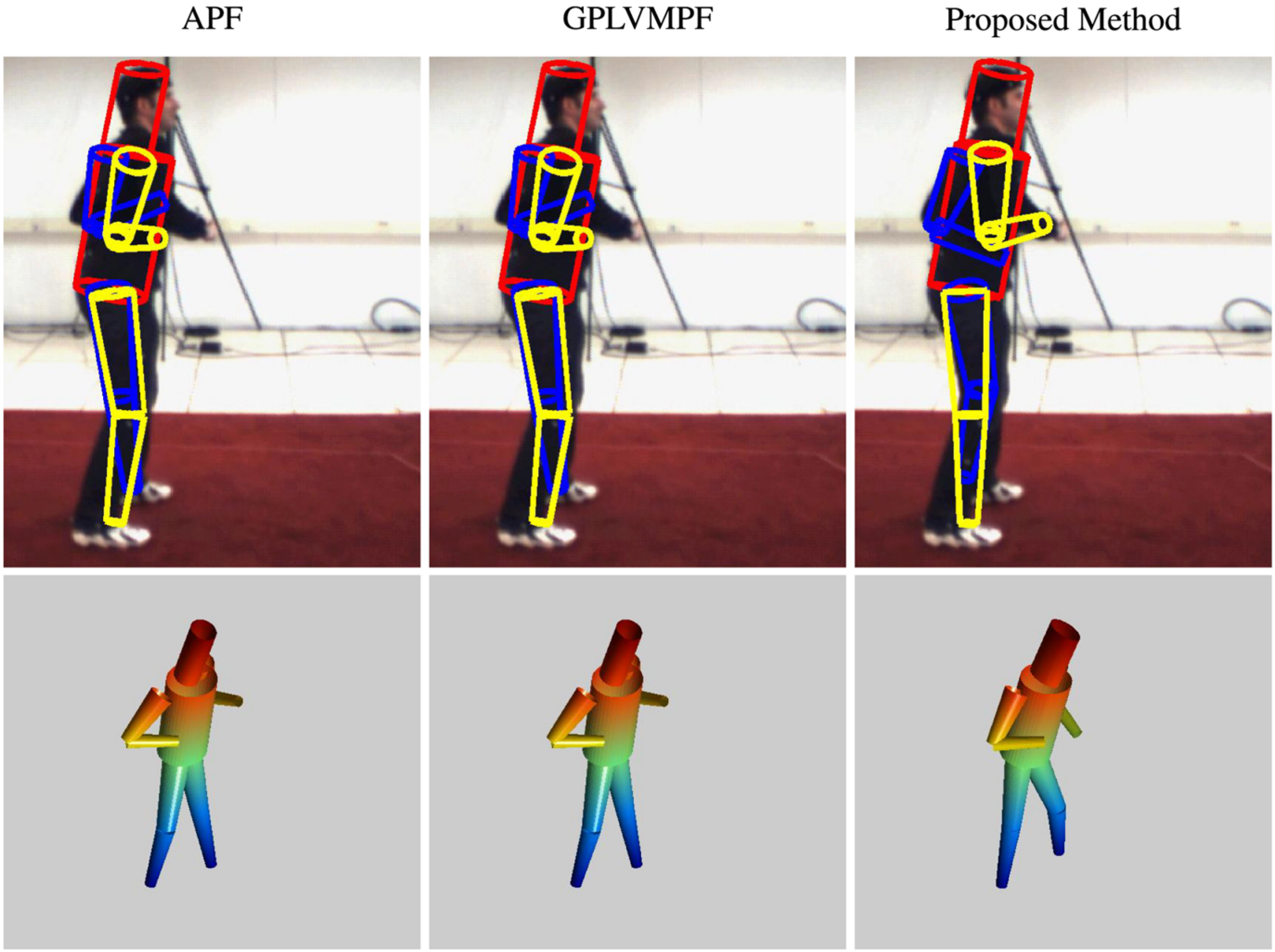

Fig. 6 Sample tracking result of frame 140 from the test video sequence of S3 performing boxing

quence of subject S3. With a learned prior model, both the proposed algorithm and GPLVMPF are able to track reliably when self-occlusion or motion blur occur. In contrast, annealed particle filtering usually loses track of some body limbs. Therefore, smart sampling alone does not do a very good job in tracking given the weak image likelihood function used (chamfer matching between the silhouettes). At frame 35 in Fig. 5 and frame 140 in Fig. 6, GPLVMPF loses track of the subject's left arm. The strength of the GPLVM (global smoothness) in this case may be its weakness. As GPLVM ensures temporal smoothness, it may learn an over-smoothed density function and consequently fail to capture large pose changes over time. In contrast, our method propagates modes over time. At each time step, the samples are generated from each mode separately and temporal smoothness is only enforced on samples drawn from the same cluster; hence, our proposed algorithm tends to capture large movements more accurately.

Additional tracking results for other test sequences are shown in Figs. 7, 8 and 9. It can be seen that our method consistently tracked body limbs over time with the automatically selected model setup. The tracking performance of GPLVMPF is not far from the proposed method, however, the latent dimension must be set manually.

The APF does not require any special training. It only learns a simple linear dynamical model from the training data. The APF does not perform as well as the other two methods, as smart sampling alone does not ensure the sampled hypotheses are similar to the training motion. The weak dynamical model may also cause the relatively large training error. Hence, if we know the motions that we want to track, it is always beneficial to encode such prior information and incorporate it into tracking.

8 Discussion and Future Work

A learning based approach was proposed to reduce the dimensionality of the state space of Bayesian tracking. A variational Bayesian formulation for the one-step solution of

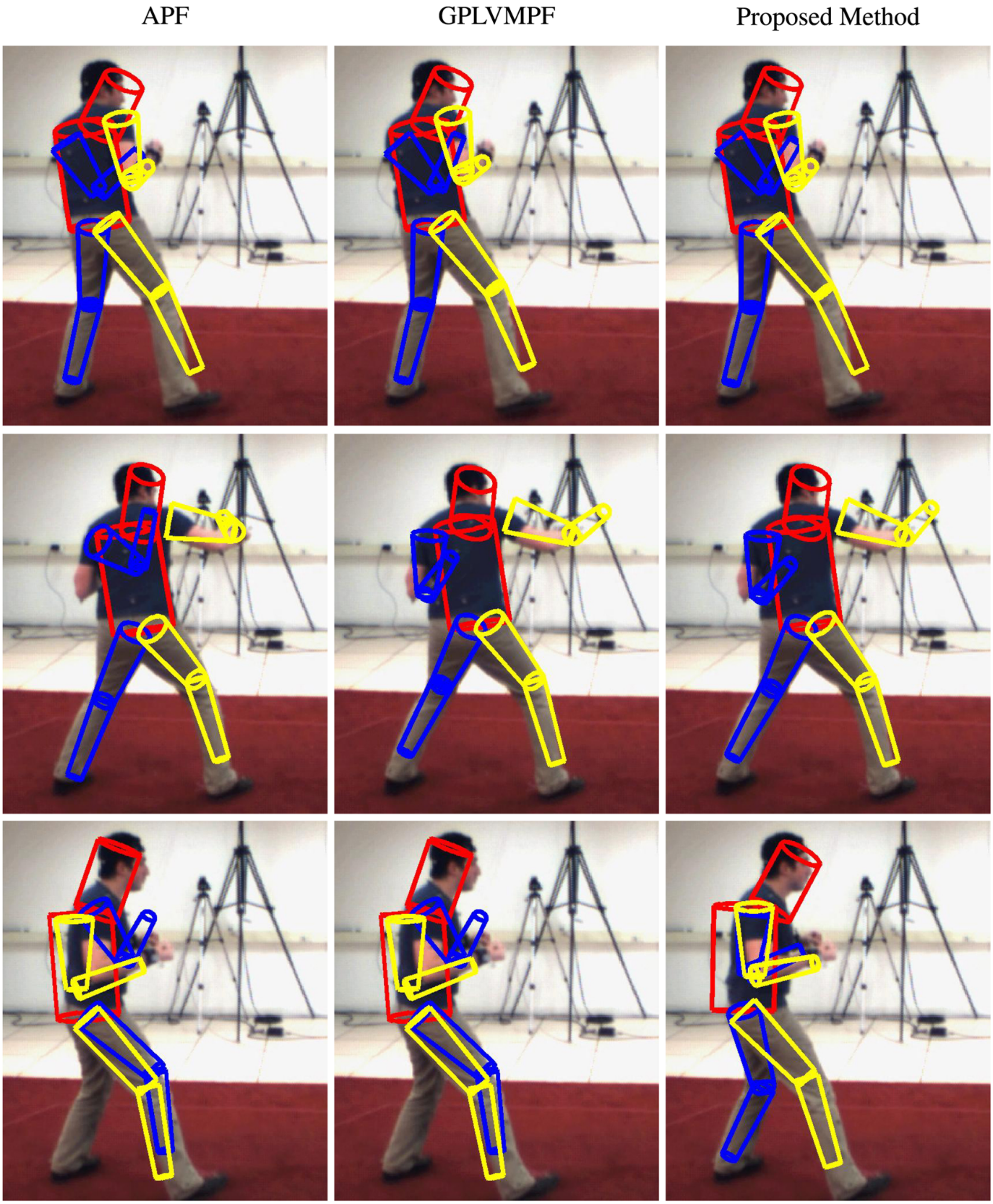

Fig. 7 Sample tracking results from the test video sequence of S2 performing boxing. The *first row* is frame 1, the *second row* is frame 80 and the *last row* is frame 140

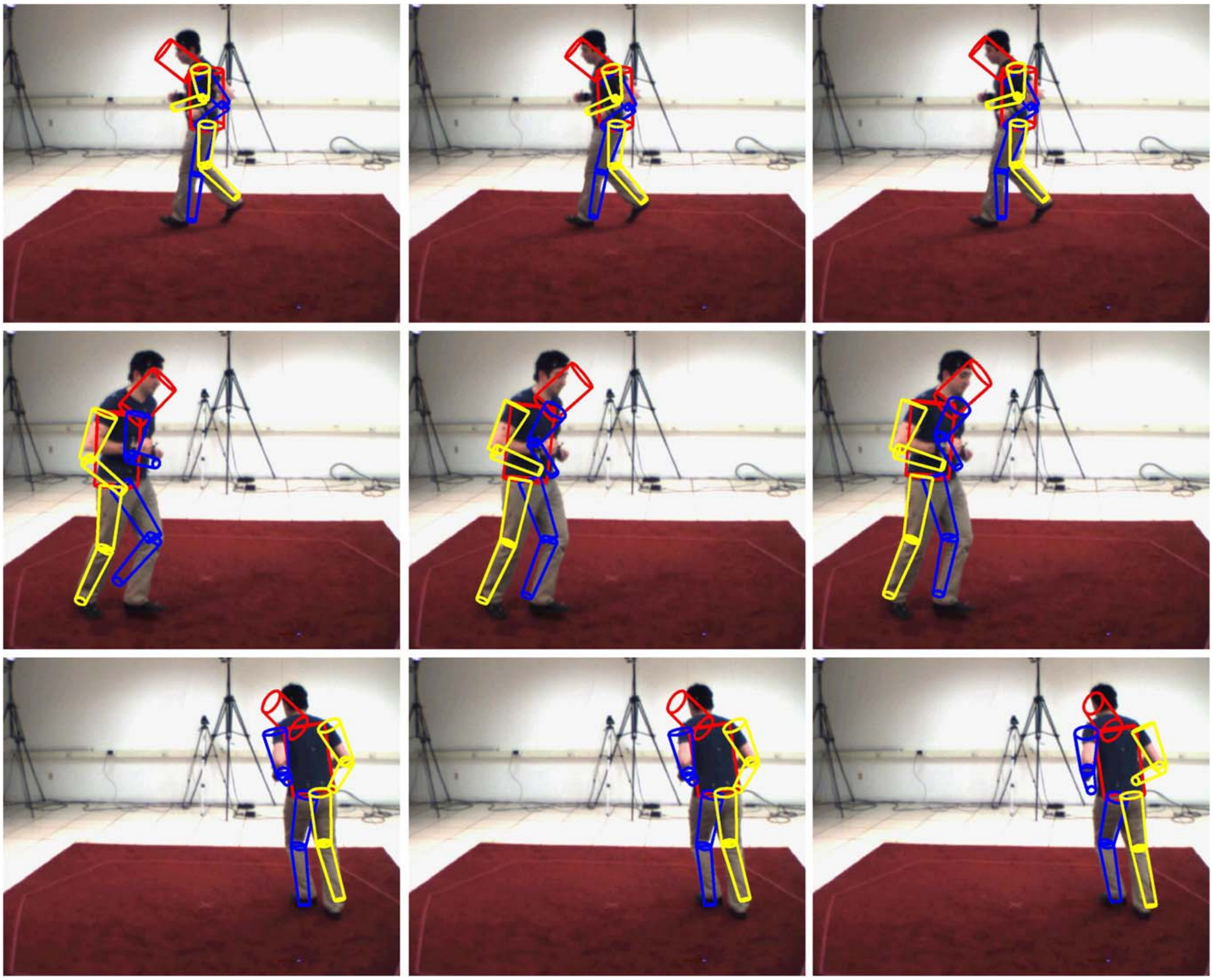

Fig. 8 Sample tracking results from the test video sequence of S2 performing jogging. The *first row* is frame 1, the *second row* is frame 80 and the *last row* is frame 140

the globally coordinated mixture of factor analyzers was derived. Its success in the application to human motion tracking was evaluated on the HumanEvaI benchmark datasets. The variational Bayesian formulation solves the problem of choosing the optimal model setup in a principled way. With the automatically chosen model setup, our tracker demonstrates better performance than the competing approaches (Deutscher et al. 2000; Tian et al. 2005b) in terms of mean and standard deviation of the estimated marker error in the experiments. Hence, the variational Bayesian formulation maintains the advantages of the approach proposed in (Li et al. 2006), but without the trouble of guessing the optimal model setup.

Since tracking involves time series data, one promising direction would be to exploit the temporal information in learning the dimensionality-reduced space. Such temporal extensions have been proposed in (Jenkins and Matarić 2004; Li et al. 2007; Lin et al. 2006; Wang et al. 2008). However, how to choose the optimal model setup still remains an open problem. Thus, our future work would be to derive a variational Bayesian formulation for the methods proposed in (Li et al. 2007; Lin et al. 2006) where the temporal extensions of the mixture of factor analyzers are proposed.

Another interesting direction to explore is to enforce topological constraints in the latent space. As pointed out in (Elgammal and Lee 2009; Urtasun et al. 2008), the latent spaces of certain motions demonstrate specific known topologies. Therefore, such constraints should be exploited when learning the dimensionality-reduced space, and methods to encode such constraints in the mixture models should be investigated.

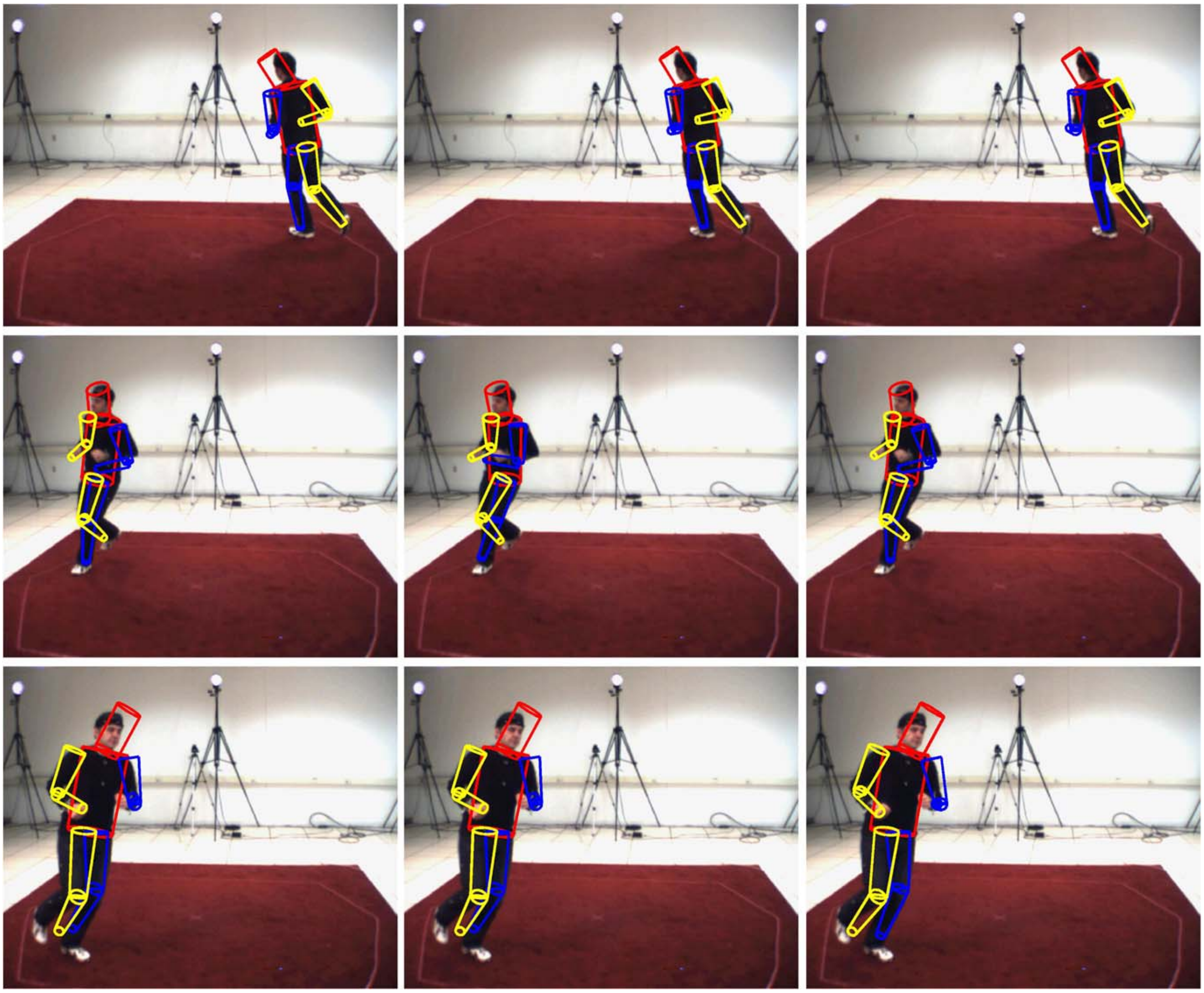

Fig. 9 Sample tracking results from the test video sequence of S3 performing jogging. The *first row* is frame 1, the *second row* is frame 80 and the *last row* is frame 140

Appendix

In GCMFA, the latent coordinate $\mathbf{g}_n^{(s)}$ is normally distributed with each factor analyzer: $p(\mathbf{g}_n|s) = \mathcal{N}(\boldsymbol{\kappa}^{(s)}, \boldsymbol{\Sigma}^{(s)})$. Furthermore, the observation $\mathbf{x}_n$ and its corresponding latent coordinate in factor analyzer s are related by a linear process parameterized by centers $\boldsymbol{\mu}_s$, factor loading matrix $\boldsymbol{\Lambda}^{(s)}$ and sensor noise covariance $\boldsymbol{\Psi}$: $p(\mathbf{x}_n|\mathbf{g}_n, s) = \mathcal{N}(\boldsymbol{\mu}^{(s)} + \boldsymbol{\Lambda}^{(s)}\mathbf{g}_n^{(s)}, \boldsymbol{\Psi})$, where $\mathbf{g}_n^{(s)} = \hat{\mathbf{g}}_n - \boldsymbol{\kappa}_s$. The objective function shown in Eq. 5 can be written as:

$$\Phi = \sum_{n=1}^{N}\sum_{s=1}^{S} q_n^{(s)}(\mathcal{S}_n^{(s)} - \xi_n^{(s)}), \tag{13}$$

where

$$\begin{aligned}
q_n^{(s)} &= q(s_n), \\
\mathcal{S}_n^{(s)} &= \frac{1}{2}\log|\boldsymbol{\Sigma}_{\mathbf{g}_n}| - \log q_n^{(s)} + \frac{d}{2}\log(2\pi), \\
\xi_n^{(s)} &= -\log \boldsymbol{\pi}^{(s)} + \frac{D+d}{2}\log(2\pi) + \frac{1}{2}\log|\boldsymbol{\Sigma}^{(s)}| \\
&\quad + \frac{1}{2}\log|\Psi| + \frac{1}{2}\mathrm{Tr}\{\boldsymbol{\Sigma}^{(s)}(\boldsymbol{\Sigma}_{\mathbf{g}_n} + \mathbf{g}_n^{(s)}\mathbf{g}_n^{(s)T})\} \\
&\quad + \frac{1}{2}\mathrm{Tr}\{\boldsymbol{\Sigma}_{\mathbf{g}_n}[\boldsymbol{\Lambda}^{(s)}]^T\boldsymbol{\Psi}^{-1}\boldsymbol{\Lambda}^{(s)}\}
\end{aligned}$$

Algorithm 3 Learning the globally coordinated mixture of factor analyzers

```
**E-step**: Optimize $\Phi$ with respect to the parameters of $q(s_n)$ and $q(\mathbf{g}_n)$
    $q_n^{(s)} = \frac{\exp^{-\xi_n^{(s)}}}{\sum_{s'=1}^{S} \exp^{-\xi_n^{s'}}}$,
    $\mathbf{\Sigma}_{\mathbf{g}_n} = (q_n^{(s)}\mathbf{V}^{(s)})^{-1}$, where $\mathbf{V}^{(s)} = [\mathbf{\Sigma}^{(s)}]^{-1} + [\mathbf{\Lambda}^{(s)}]^{\mathsf{T}}\mathbf{\Psi}^{-1}\mathbf{\Lambda}^{(s)}$ ,
    $\hat{\mathbf{g}}_n = \mathbf{\Sigma}_{\mathbf{g}_n}\sum_{s=1}^{S} q_n^{(s)}\mathbf{V}^{(s)}\mathbf{m}_n^{(s)}$, where $\mathbf{m}_n^{(s)} = \boldsymbol{\kappa}^{(s)} + [\mathbf{V}^{(s)}]^{-1}[\mathbf{\Lambda}^{(s)}]^{\mathsf{T}}\mathbf{\Psi}^{-1}(\mathbf{x}_n - \boldsymbol{\mu}^{(s)})$.
**M-step**: Optimize $\Phi$ with respect to the parameters of $p(\mathbf{x}_n, \mathbf{g}, s)$
    $\boldsymbol{\pi}^{(s)} = \frac{1}{N}\sum_{n=1}^{N} q_n^{(s)}$,
    $\boldsymbol{\kappa}^{(s)} = \sum_{n=1}^{N}\tilde{q}_n^{(s)}\hat{\mathbf{g}}_n$, where $\tilde{q}_n^{(s)} = q_n^{(s)} / \sum_{n'=1}^{N} q_{n'}^{(s)}$,
    $\boldsymbol{\mu}^{(s)} = \sum_{n=1}^{N}\tilde{q}_n^{(s)}\mathbf{x}_n$,
    $\mathbf{\Sigma}^{(s)} = \sum_{n=1}^{N}\tilde{q}_n^{(s)}[\mathbf{g}_n^{(s)}[\mathbf{g}_n^{(s)}]^{\mathsf{T}} + \mathbf{\Sigma}_{\mathbf{g}_n}]$,
    $\mathbf{\Lambda}^{(s)} = C^{(s)}[\mathbf{\Sigma}^{(s)}]^{-1}$ where $C^{(s)} = \sum_{n=1}^{N}\tilde{q}_n^{(s)}\mathbf{x}_n^{(s)}[\mathbf{g}_n^{(s)}]^{\mathsf{T}}$,
    $[\mathbf{\Psi}]_{ii} = \sum_{s=1}^{S}\sum_{n=1}^{N}\tilde{q}_n^{(s)}([\mathbf{x}_n^{(s)} - \mathbf{\Lambda}^{(s)}\mathbf{g}_n^{(s)}]_i^2 + [\mathbf{\Lambda}^{(s)}\mathbf{\Sigma}_{\mathbf{g}_n}[\mathbf{\Lambda}^{(s)}]^{\mathsf{T}}]_{ii})$,
    where $[\cdot]_{ii}$ and $[\cdot]_i$ denote the $i$-th diagonal entry of a matrix or $i$-th entry of a vector.
```

$$+ \frac{1}{2}(\mathbf{x}_n^{(s)} - \mathbf{\Lambda}^{(s)}\mathbf{g}_n^{s})^T \mathbf{\Psi}^{-1}(\mathbf{x}_n^{(s)} - \mathbf{\Lambda}^{(s)}\mathbf{g}_n^{(s)}),$$

$$\mathbf{g}_n^{(s)} = \hat{\mathbf{g}}_n - \boldsymbol{\kappa}^{(s)} \text{ and } \quad \mathbf{x}_n^{(s)} = \mathbf{x}_n - \boldsymbol{\mu}^{(s)}.$$

We can obtain the GCMFA model parameters $\boldsymbol{\theta}$ together with the variational regularizing parameters $\{\hat{\mathbf{g}}_n, \mathbf{\Sigma}_{\mathbf{g}_n}, q_n^{(s)}\}$ from the variational distributions by iteratively optimizing Φ using an EM-like coordinate ascent algorithm in learning. Φ is maximized in turn with respect to the variational distributions $q(\cdot)$ and the model parameters $\boldsymbol{\theta} = \{\{\mathbf{\Lambda}^{(s)}, \boldsymbol{\mu}^{(s)}, \mathbf{\Sigma}^{(s)}, \boldsymbol{\kappa}^{(s)}\}_{s=1}^{S}, \mathbf{\Psi}\}$ respectively. This process is summarized in Algorithm 3. To initialize the GCMFA, we make use of local linear embedding method (Roweis and Saul 2000).

References

Agarwal, A., & Triggs, B. (2004). Tracking articulated motion with piecewise learned dynamical models. In *Proceedings of the European conference on computer vision (ECCV)* (Vol. 3, pp. 54–65).

Balan, A., Sigal, L., & Black, M. (2005). A quantitative evaluation of video-based 3d person tracking. In *IEEE workshop on VS-PETS* (pp. 349–356).

Beal, M. (2003). *Variational algorithms for approximate Bayesian inference*. PhD thesis, Gatsby Computational Neuroscience Unit, University College London.

Belkin, M., & Niyogi, P. (2001). Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in neural information processing systems (NIPS)* (pp. 585–591).

Bishop, C., Svensén, M., & Williams, C. (1998). GTM: the generative topographic mapping. *Neural Computation*, *10*(1), 215–234.

Brand, M. (2002). Charting a manifold. In *Advances in neural information processing systems (NIPS)* (pp. 961–968).

Cham, T.-J., & Rehg, J. M. (1999). A multiple hypothesis approach to figure tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 239–245).

Cheeseman, P., & Stutz, J. (1996). Bayesian classification (AutoClass: theory and results). In *Advances in knowledge discovery and data mining* (pp. 153–180).

Choo, K., & Fleet, D. (2001). People tracking using hybrid Monte Carlo filtering. In *Proceedings of the IEEE international conference on computer vision (ICCV)* (pp. 321–328).

Deutscher, J., Blake, A., & Reid, I. (2000). Articulated body motion capture by annealed particle filtering. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 126–133).

Elgammal, A., & Lee, C.-S. (2004). Inferring 3D body pose from silhouettes using activity manifold learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 681–688).

Elgammal, A., & Lee, C.-S. (2009). Tracking people on a torus. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *31*(3), 520–538.

Ghahramani, Z., & Hinton, G. (1996). *The EM algorithm for mixtures of factor analyzers* (Technical Report CRG-TR-96-1). University of Toronto.

Ioffe, S., & Forsyth, D. (2001). Human tracking with mixtures of trees. In *Proceedings of the IEEE international conference on computer vision (ICCV)* (pp. 690–695).

Jefferys, W., & Berger, J. (1992). Ockham's Razor and Bayesian analysis. *American Scientist*, *80*, 64–72.

Jenkins, O., & Matarić, M. (2004). A spatio-temporal extension to Isomap nonlinear dimensionality reduction. In *Proceedings of the IEEE international conference on machine learning (ICML)* (pp. 56–73).

Ju, S. X., Black, M., & Yacoob, Y. (1996). Cardboard people: a parameterized model of articulated image motion. In *International conference on automatic face and gesture recognition* (pp. 38–44).

Kass, R., & Raftery, A. (1995). Bayesian factors. *Journal of the American Statistical Association*, *90*, 773–795.

Lan, X., & Huttenlocher, D. (2004). A unified spatio-temporal articulated model for tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 722–729).

Lawrence, N. (2003). Gaussian process latent variable models for visualization of high dimensional data. In *Advances in neural information processing systems (NIPS)* (pp. 329–336).

Li, R., Yang, M.-H., Sclaroff, S., & Tian, T.-P. (2006). Monocular tracking of 3D human motion with a coordinated mixture of factor analyzers. In *Proceedings of the European conference on computer vision (ECCV)* (Vol. 2, pp. 137–150).

Li, R., Tian, T.-P., & Sclaroff, S. (2007). Simultaneous learning of nonlinear manifold and dynamical models for high-dimensional time series. In *Proceedings of the IEEE international conference on computer vision (ICCV)* (pp. 1–8).

Lin, R.-S., Liu, C.-B., Yang, M.-H., Ahuja, N., & Levinson, S. (2006). Learning nonlinear manifolds from time series. In *Proceedings of the European conference on computer vision (ECCV)* (Vol. 3, pp. 239–250).

MacCormick, J., & Blake, A. (1999). A probabilistic exclusion principle for tracking multiple objects. In *Proceedings of the IEEE international conference on computer vision (ICCV)* (pp. 572–578).

MacKay, D. (1992). Bayesian interpolation. *Neural Computation*, *4*(3), 415–417.

MacKay, D. (1996). Bayesian non-linear modelling for the 1993 energy prediction competition. In G. Heidbreder (Ed.), *Maximum entropy and Bayesian methods*, Santa Barbara 1993 (pp. 221–234). Dordrecht: Kluwer.

Mori, G., & Malik, J. (2002). Estimating human body configurations using shape context matching. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 666–680).

Poppe, R. (2007a). Evaluating example-based pose estimation: experiments on the Humaneva sets. In *Online proceedings of the workshop on evaluation of articulated human motion and pose estimation (EHuM) at the international conference on computer vision and pattern recognition (CVPR)*.

Poppe, R. (2007b). Vision-based human motion analysis: an overview. *Computer Vision and Image Understanding*, *108*, 4–18.

Ramanan, D., Forsyth, D. A., & Zisserman, A. (2007). Tracking people by learning their appearance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *29*(1), 65–81.

Rasmussen, C. (2000). The infinite Gaussian mixture model. In *Advances in neural information processing systems (NIPS)* (pp. 554–560).

Richardson, S., & Green, P. (1997). On Bayesian analysis of mixtures with unknown number of components. *Journal of the Royal Statistical Society, Series B*, *59*(4), 731–758.

Roweis, R., & Saul, L. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, *290*(5500), 2323–2326.

Roweis, R., Saul, L., & Hinton, G. (2001). Global coordination of local linear models. In *Advances in neural information processing systems (NIPS)* (pp. 889–896).

Safonova, A., Hodgins, J., & Pollard, N. (2004). Synthesizing physically realistic human motion in low dimensional, behavior-specific spaces. In *ACM computer graphics (SIGGRAPH)* (pp. 514–521).

Schölkopf, B., Smola, A., & Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, *10*(1), 1299–1319.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, *6*(2), 461–464.

Shakhnarovich, G., Viola, P., & Darrel, T. (2003). Fast pose estimation with parameter sensitive hashing. In *Proceedings of the IEEE international conference on computer vision (ICCV)* (pp. 750–757).

Sidenbladh, H., Black, M., & Fleet, D. (2000). Stochastic tracking of 3D human figures using 2D image motion. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 702–718).

Sigal, L., Bhatia, S., Roth, S., Black, M., & Isard, M. (2004). Tracking loose-limbed people. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 421–428).

Sigal, L., & Black, M. (2006). *HumanEva: synchronized video and motion capture dataset for evaluation of articulated human motion* (Technical Report CS-06-08). Brown University.

Silva, V., & Tenenbaum, J. (2003). Global versus local methods in nonlinear dimensionality reduction. In *Advances in neural information processing systems (NIPS)* (pp. 705–712).

Sminchisescu, C., & Jepson, A. (2004). Generative modelling for continuous non-linearly embedded visual inference. In *Proceedings of the IEEE international conference on machine learning (ICML)* (pp. 140–147).

Sminchisescu, C., & Triggs, B. (2001). Covariance scaled sampling for monocular 3D body tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 447–454).

Snelson, E., & Ghahramani, Z. (2006). Sparse Gaussian processes using pseudo-inputs. In *Advances in neural information processing systems (NIPS)* (pp. 1259–1226).

Stenger, B., Thayananthan, A., Torr, P., & Cipolla, R. (2003). Filtering using a tree-based esimator. In *Proceedings of the IEEE international conference on computer vision (ICCV)* (pp. 1063–1070).

Sullivan, J., & Rittscher, J. (2001). Guiding random particles by deterministic search. In *Proceedings of the IEEE international conference on computer vision (ICCV)* (pp. 323–330).

Teh, W.-Y., & Roweis, S. (2002). Automatic alignment of local representations. In *Advances in neural information processing systems (NIPS)* (pp. 841–848).

Tenenbaum, J., Silva, V., & Langford, J. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, *290*, 2319–2323.

Tian, T.-P., Li, R., & Sclaroff, S. (2005a). Articulated pose estimation in a learned smooth space of feasible solutions. In *Learning workshop in conjunction with CVPR*.

Tian, T.-P., Li, R., & Sclaroff, S. (2005b). *Tracking human body pose on a learned smooth space* (Technical Report 2005-029). Boston University.

Urtasun, R., Fleet, D., Hertzmann, A., & Fua, P. (2005). Priors for people tracking from small training sets. In *Proceedings of the IEEE international conference on computer vision (ICCV)* (pp. 403–410).

Urtasun, R., Fleet, D., & Fua, P. (2006). 3D people tracking with Gaussian process dynamical models. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 238–245).

Urtasun, R., Fleet, D., & Lawrence, N. (2008). Topologically-constrained latent variable models. In *Proceedings of the IEEE international conference on machine learning (ICML)*.

Verbeek, J. (2006). Learning non-linear image manifolds by combining local linear models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(10), 1864–1875.

Wang, L., Hu, W., & Tan, T. (2003). Recent development in human motion analysis. *Pattern Recognition*, *36*(3), 585–601.

Wang, J., Fleet, D., & Hertzman, A. (2008). Gaussian process and dynamical models for human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *30*(2), 283–298.