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

Rui Li[1], Ming-Hsuan Yang[2], Stan Sclaroff[1], and Tai-Peng Tian[1]

[1] Boston University, Boston, MA 02215, USA
{lir, sclaroff, tian}@cs.bu.edu
[2] Honda Research Institute, Mountain View, CA 94041, USA
myang@honda-ri.com

Abstract. Filtering based algorithms have become popular in tracking human body pose. Such algorithms can suffer the curse of dimensionality due to the high dimensionality of the pose state space; therefore, efforts have been dedicated to either smart sampling or reducing the dimensionality of the original pose state space. In this paper, a novel formulation that employs a dimensionality reduced state space for multi-hypothesis tracking is proposed. During off-line training, a mixture of factor analyzers is learned. Each factor analyzer can be thought of as a "local dimensionality reducer" that locally approximates the pose manifold. Global coordination between local factor analyzers is achieved by learning a set of linear mixture functions that enforces agreement between local factor analyzers. The formulation allows easy bidirectional mapping between the original body pose space and the low-dimensional space. During online tracking, the clusters of factor analyzers are utilized in a multiple hypothesis tracking algorithm. Experiments demonstrate that the proposed algorithm tracks 3D body pose efficiently and accurately , even when self-occlusion, motion blur and large limb movements occur. Quantitative comparisons show that the formulation produces more accurate 3D pose estimates over time than those that can be obtained via a number of previously-proposed particle filtering based tracking algorithms.

1 Introduction

Tracking articulated human motion is of interest in numerous applications: video surveillance, gesture analysis, human computer interfaces, computer animation, etc. Various tracking algorithms have been proposed that require neither special clothing nor markers on the human body. A number of algorithms track body motion in the image plane (2D), 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 inutile for applications where accurate 3D information is required. To better understand human motion, 3D tracking algorithms resort to detailed 3D articulated models which 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 3D human tracking algorithms.

In this paper, we propose an efficient and accurate algorithm for tracking 3D articulated human motion given monocular video sequences. We exploit the physical constraints of human motion by learning a low-dimensional latent model from high-dimensional motion capture data. A probabilistic algorithm is employed to perform non-linear dimensionality reduction and clustering concurrently within a global coordinate system. The projected 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 modes. By tracking in low-dimensional space, we avoid the sample impoverishment problem [1] and retain the simplicity of the multiple hypothesis tracking algorithm at the same time. Given clusters formed in the latent space, temporal smoothness is only enforced within each cluster. In experiments with real video, the system reliably tracks body motion during self-occlusions and in the presence of motion blur. The system 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 via a number of previously-proposed particle filtering based tracking algorithms.

2 Related Work

In this section, we first outline recent progress in particle filtering based tracking algorithms. We then give a quick review of dimensionality reduction algorithms, followed by a discussion of algorithms that solve the tracking problem in the dimensionality reduced space.

2.1 Particle Filtering

Particle filtering methods have been applied widely in tracking applications. Unfortunately, the number of particles needed to sufficiently approximate the state posterior distribution can explode when the state vector is high dimensional. Various approaches have been proposed to alleviate this problem. One common approach is to reposition the particles according to some importance function [2] to ensure a high survival rate [3]. For example, particles can be resampled using weighted resampling [3] or repositioned using deterministic search [4, 5] to localize the set of particles around significant maxima of the importance function. Other methods employ a coarse to fine search on the weighting function, e.g., the annealed particle filter [6] or layered sampling [7]. If the particle dynamics can be factored into independent components, then partitioned sampling [3] can be used to improve the performance of the particle filter.

2.2 Non-Linear Dimensionality Reduction Algorithms

Dimensionality reduction algorithms are popular techniques to discover compact representations of high-dimensional data. As a classic dimensionality reduction algorithm, Principal Component Analysis (PCA) is inadequate to handle the non-linear behavior inherent to our problem domain. Locally Linear Embedding (LLE) [8], Isomap [9] and Laplacian Eigenmaps [10] are some representative non-linear dimensionality reduction algorithms – but unfortunately, these techniques are typically not invertible. Inverse mapping of particles (proposals) back to the original human pose space is needed in order to reweight the particles given the image measurements. A number of existing dimensionality reduction methods provide inverse mappings, such as Charting [11], Locally Linear Coordination (LLC) [12] and the Gaussian Process Latent Variable Model (GPLVM) [13]. In principle, any dimensionality reduction technique that provides an inverse mapping will be applicable. LLC is chosen in our algorithm because it is a probabilistic algorithm that performs non-linear dimensionality reduction and clustering concurrently within a global coordinate system. The projected data forms clusters within the globally coordinated low-dimensional space; this makes it possible to derive an efficient multiple hypothesis tracking algorithm based on distribution modes.

2.3 Human Motion Tracking

There is a broad range of work related to human motion tracking. See [14] for a recent survey as our focus is on the subclass of stochastic tracking algorithms.

Following the seminal work of [15], the CONDENSATION algorithm has been adapted for human motion tracking [4, 6]. Multiple Hypothesis Tracking [16] tracks modes in a simpler piece-wise Gaussian distribution. In [17], exemplars are incorporated into the CONDENSATION algorithm. A more specific motion model and accurate background modelling using learning are used in [18, 19].

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 [20–22]. Particle filtering with the reduced state space will be faster because significantly fewer particles are required to adequetely approximate the state space posterior distribution. Recent works [23–25] are most closely related to our proposed algorithm for tracking human in a dimensionality-reduced space. In [23], two 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 [23]. In [25], GPLVM and a second order Markov model are used for tracking applications. The learned GPLVM model is used to provide model prior. Tracking is then done by minimizing a cost of 2D image matching, with the negative log-likelihood of the model prior as the regularization term. Both [23] and [25] advocate the use of gradient descent optimization techniques; hence, the low-dimensional space learned has to be smooth. An alternative approach [24] employs the GPLVM in a modified particle filtering algorithm where samples are drawn from the low-dimensional latent space. The GPLVM model in this case is used as a good non-linear dimensionality reduction algorithm. The smoothness enforced in the low-dimensional space by the learning algorithms in these three papers works well for tracking small limb movements, but may fail when large limb movements occur over time. In the case of using gradient descent optimization techniques, good initialization is required for the success of such techniques.

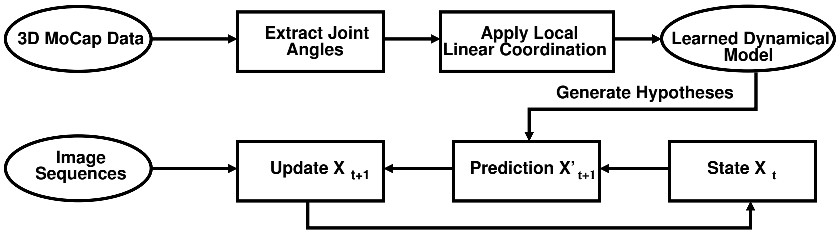

Fig. 1: Overview of our algorithm.

As will be shown in the rest of this paper, Locally Linear Coordination (LLC) leads to a principled way of solving the embedding and inverse mapping problems. Instead of enforcing smoothness everywhere in the latent space, this algorithm preserves the clustering behavior of similar high-dimensional data points and separates different clusters in the global coordinate system. The model learned from the LLC is then used in the algorithm for multiple hypothesis tracking of 3D human body motion.

3 Formulation

There are two main components in the proposed tracking algorithm as shown in Fig. 1. The first component is an off-line algorithm that learns a bidirectional mapping function between the low-dimensional space and the original pose space. The second component is an on-line algorithm for articulated human pose tracking that makes use of a modified multiple hypothesis tracking algorithm; the modes of this multiple hypothesis tracker are propagated over time in the embedded space.

3.1 Learning the Global Coordination Model

Roweis *et. al.* [26] proposed a model which performs a global coordination of local coordinate systems in a mixture of factor analyzers (MFA). Each factor analyzer (FA) can also be regarded as a local dimensionality reducer. The assumption is that both the high-dimensional data $\mathbf{y}$ and its global coordinate $\mathbf{g}$ are generated from the same set of latent variables s and $\mathbf{z}_s$, where each discrete hidden variable s refers to the s-th FA and each continuous hidden variable $\mathbf{z}_s$ represents the low-dimensional local coordinates in the s-th FA.

In the MFA model, data generated from s-th FA with prior probability $P(s)$, and the distribution of $\mathbf{z}_s$ are Gaussian: $\mathbf{z}_s|s \sim \mathcal{N}(0, \mathbf{I})$, where $\mathbf{I}$ is the identity matrix. Given s and $\mathbf{z}_s$, $\mathbf{y}$ and the global coordinate $\mathbf{g}$ are generated by the following linear equations:

$$\begin{aligned} \mathbf{y} &= \mathbf{T}_{L_s}\mathbf{z}_s + \boldsymbol{\mu}_s + \mathbf{u}_s, \\ \mathbf{g} &= \mathbf{T}_{G_s}\mathbf{z}_s + \boldsymbol{\kappa}_s + \mathbf{v}_s, \end{aligned} \tag{1}$$

where $\mathbf{T}_{L_s}$ and $\mathbf{T}_{G_s}$ are the transformation matrices, $\boldsymbol{\mu}_s$ and $\boldsymbol{\kappa}_s$ are uniform translations between the coordinate systems, $\mathbf{u}_s \sim \mathcal{N}(0, \Lambda_{u_s})$ and $\mathbf{v}_s \sim \mathcal{N}(0, \Lambda_{v_s})$ are independent zero mean Gaussian noise terms. The following probability distributions can be derived from Eq. 1, 1:

$$\begin{aligned} \mathbf{y}|s, \mathbf{z}_s &\sim \mathcal{N}(\mathbf{T}_{L_s}\mathbf{z}_s + \boldsymbol{\mu}_s, \Lambda_{\mathbf{u}_s}) \\ \mathbf{g}|s, \mathbf{z}_s &\sim \mathcal{N}(\mathbf{T}_{G_s}\mathbf{z}_s + \boldsymbol{\kappa}_s, \Lambda_{\mathbf{v}_s}). \end{aligned} \tag{2}$$

With $\mathbf{z}_s$ being integrated out, we have

$$\begin{aligned} \mathbf{y}|s &\sim \mathcal{N}(\boldsymbol{\mu}_s, \Lambda_{\mathbf{u}_s} + \mathbf{T}_{L_s}\mathbf{T}_{L_s}^T) \\ \mathbf{g}|s &\sim \mathcal{N}(\boldsymbol{\kappa}_s, \Lambda_{\mathbf{v}_s} + \mathbf{T}_{G_s}\mathbf{T}_{G_s}^T). \end{aligned} \tag{3}$$

The inference of global coordinate $\mathbf{g}$ conditioned on a data point $\mathbf{y}_n$ can be rewritten as

$$p(\mathbf{g}|\mathbf{y}_n) = \sum_s p(\mathbf{g}|\mathbf{y}_n, s)p(s|\mathbf{y}_n), \tag{4}$$

where

$$p(\mathbf{g}|\mathbf{y}_n, s) = \int p(\mathbf{g}|s, \mathbf{z}_s)p(\mathbf{z}_s|s, \mathbf{y}_n)d\mathbf{z}_s. \tag{5}$$

Given Eq. 1, both $p(\mathbf{g}|s, \mathbf{z}_s)$ and $p(\mathbf{z}_s|s, \mathbf{y}_n)$ are Gaussian distributions, $p(\mathbf{g}|\mathbf{y}_n, s)$ also follows a Gaussian distribution. Since $p(s|\mathbf{y}_n) \propto p(\mathbf{y}_n|s)p(s)$ can be computed and viewed as a weight, $p(\mathbf{g}|\mathbf{y}_n)$ is essentially a mixture of Gaussians. Though the mappings from $\{s, \mathbf{z}_s\}$ to $\mathbf{y}$ and $\mathbf{g}$ are linear, the mappings between them are not. An EM algorithm is proposed in [26] to learn this global coordination by maximizing the likelihood of the data, with an additional variational penalty term to encourage consistency of internal coordinates of each factor analyzer. This algorithm requires a user given trade-off parameter between modeling data and having consistent local coordinate systems. This algorithm suffers from the same problems of standard EM approaches, i.e., inefficiency and local minima.

Teh and Roweis came up with an efficient two stage model learning algorithm in [12]. By leveraging on the mixture of local models to collapse large groups of points together, their proposed algorithm works only with the groups rather than individual data points in the global coordination. In the two-stage model learning process, first the MFA between $\mathbf{y}$ and $(s, \mathbf{z}_s)$ is learned as proposed in [27]. Given the learned MFA model with S factor analyzers, $\mathbf{z}_{ns}$ is the expected local coordinate in the s-th FA for each data point $\mathbf{y}_n$. Let r_{ns} denote the likelihood, $p(\mathbf{y}_n|s)$. From Eqs. 1 and 2, $\mathbf{g}_n$, the expected global coordinate of $\mathbf{y}_n$ is defined as:

$$\mathbf{g}_n = \sum_s r_{n_s}(\mathbf{T}_{G_s}\mathbf{z}_{n_s} + \kappa_s) = \mathbf{L}\mathbf{u}_n, \tag{6}$$

where

$$\mathbf{L} = [\mathbf{T}_{G_1}, \kappa_1, \mathbf{T}_{G_2}, \kappa_2 \ldots, \mathbf{T}_{G_S}, \kappa_S]$$

and

$$\mathbf{u}_n^T = [r_{n_1}\mathbf{z}_{n_1}^T, r_{n_1}, r_{n_2}\mathbf{z}_{n_2}^T, r_{n_2}, \ldots, r_{n_S}\mathbf{z}_{n_S}^T, r_{n_S}].$$

Let $\mathbf{G} = [\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_N]^T$ be the global coordinates of the whole data set (the rows of $\mathbf{G}$ corresponding to the coordinated data points) and $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_N]^T$,

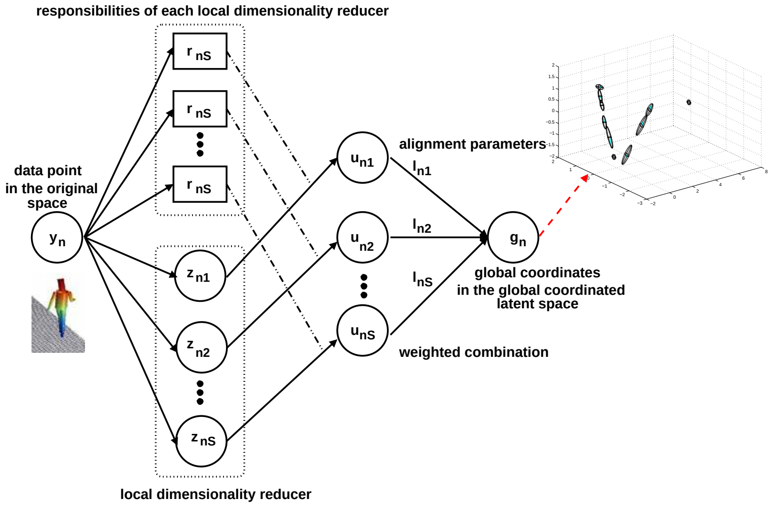

Fig. 2: The two stage learning process of [12].

we then have a compact representation $\mathbf{G} = \mathbf{U}\mathbf{L}$. We want to estimate $\mathbf{L}$. To determine $\mathbf{L}$, we need to minimize a cost function that incorporates the topological constraints that govern $\mathbf{g}_n$. The cost function used here is based on LLE [8]. For each data point $\mathbf{y}_n$, we denote its nearest neighbors as $\mathbf{y}_m$ $(m \in N_n)$ and minimize the following:

$$\begin{aligned}\xi(\mathbf{Y}, \mathbf{W}) &= \sum_n \|\mathbf{y}_n - \sum_{m \in N_n} w_{nm}\mathbf{y}_m\|^2 \\ &= Tr(\mathbf{Y}^T(\mathbf{I} - \mathbf{W}^T)(\mathbf{I} - \mathbf{W})\mathbf{Y}), \qquad (7)\end{aligned}$$

with respect to $\mathbf{W}$ and subject to the constraint $\sum_{m \in N_n} w_{nm} = 1$. Here the set of training data points is $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N]^T$ where each row of $\mathbf{Y}$ corresponds to a training data point. The weights w_{nm} are unique and can be obtained via constrained least squares. These weights represent the locally linear relationships between $\mathbf{y}_n$ and its neighbors. In a similar fashion, we can define the following cost function:

$$\begin{aligned}\xi(\mathbf{G}, \mathbf{W}) &= \sum_n \|\mathbf{g}_n - \sum_{m \in N_n} \mathbf{g}_m\|^2 \\ &= Tr(\mathbf{G}^T(\mathbf{I} - \mathbf{W}^T)(\mathbf{I} - \mathbf{W})\mathbf{G}) \\ &= Tr(\mathbf{L}^T\mathbf{A}\mathbf{L}), \qquad (8)\end{aligned}$$

where $A = \mathbf{U}(\mathbf{I} - \mathbf{W}^T)(\mathbf{I} - \mathbf{W})\mathbf{U}^T$. To ensure $\mathbf{G}$ is invariant to translations, rotations and scaling, the following constraints are defined,

$$\frac{1}{N}\sum_n \mathbf{g}_n = 0 \qquad (9)$$

and

$$\frac{1}{N}\sum_{n}\mathbf{g}_n\mathbf{g}_n^T = \frac{1}{N}\mathbf{G}^T\mathbf{G} = \mathbf{L}^T\mathbf{B}\mathbf{L} = \mathbf{I}, \tag{10}$$

where $\mathbf{I}$ is the identity matrix and $\mathbf{B} = \frac{1}{N}\mathbf{U}^T\mathbf{U}$. Both the cost function Eq. 8 and the constraints Eq. 10 are quadratic and the optimal $\mathbf{L}$ is determined by solving a generalized eigenvalue problem [12]. Let $d \ll D$ be the dimensionality of the underlying manifold that $\mathbf{y}$ is generated from. The 2^{nd} to $(d+1)^{th}$ smallest generalized vectors solved from $\mathbf{A}\mathbf{v} = \lambda\mathbf{B}\mathbf{v}$ form the columns of $\mathbf{L}$. The whole process is summarized in Fig. 2.

3.2 Learning the Joint Angle Configurations

In our application for using LLC to learn the dimensionality reduced space, each training data $\mathbf{y}$ is a column vector that consists of joint angles computed from motion capture data. We adopt the same 3D cylindrical model used in [18]; we ignore the global translation. The dimension of $\mathbf{y}$ is 28 and 1900 frames from a motion capture sequence with a person walking are used for training. $\mathbf{Y}$ is used to represent the collection of training data $\mathbf{y}_n,\ n = 1,\dots N$ and $N = 1900$. In the LLC learning, the dimension for variables $\mathbf{z}$ in each MFA is 3 and the number of mixtures $S = 10$. In the global coordination stage, the dimension of the latent space variable $\mathbf{g}$ is 3 (these parameters were determined empirically). The learning algorithm is summarized in Algorithm 1. Clusters are obtained through the two stage learning process described above. Each cluster is modeled as a Gaussian distribution in the latent space with its own mean vector and covariance matrix as shown in Fig. 3. This cluster-based representation leads to a straightforward algorithm for multiple hypothesis tracking, as described in Section 3.3.

3.3 3D Articulated Human Tracking

In the application to 3D articulated human tracking, at each time instance, the tracker state vector is represented by $\mathcal{X}_t = (\mathbf{P}_t, \mathbf{g}_t)$. $\mathbf{P}_t$ 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 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

Algorithm 1 Learning the globally coordinated space of human motion

```
Compute local linear reconstruction weights $w_{nm}$ based on Eq. 7 using $\mathbf{Y}$
Train a mixture of local dimensionality reducers.
    Apply this mixture to training human motion poses $\mathbf{Y}$.
    Obtain a local representation $\mathbf{z}_{ns}$ and responsibility $r_{ns}$ for each submodel $s$ and
    each data point $\mathbf{y}_n$.
Form the matrix $\mathbf{U}$ and compute $\mathbf{A}$ and $\mathbf{B}$ from Eq. 8, Eq. 9 and Eq. 10.
Solve the generalized eigenvalue problem $\mathbf{A}\mathbf{v} = \lambda\mathbf{B}\mathbf{v}$ and form $\mathbf{L}$ as described in
Section 3.1.
Return $\mathbf{G} = \mathbf{U}\mathbf{L}$ and $\mathbf{L}$.
```

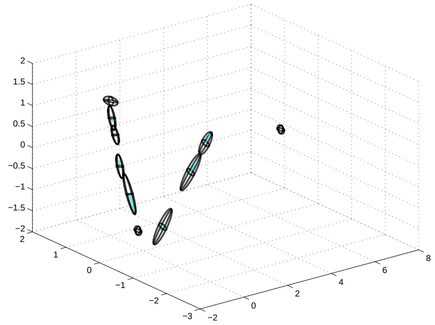

Fig. 3: The learned globally coordinated latent space. Each ellipsoid represents a cluster in the latent space, where mean is the centroid and covariances are the axes of the ellipsoids.

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:

$$p(\mathcal{X}_t|\mathbf{Z}_t) \propto p(\mathbf{z}_t|\mathcal{X}_t)p(\mathcal{X}_t|\mathbf{Z}_{t-1}) \tag{11}$$

Here we use a multiple hypothesis tracker (MHT) together with the learned LLC model for the tracking task. As LLC provides clusters in the latent space as a step in the global coordination, it is natural to make use the centers of the clusters as the initial modes in the MHT ($p(\mathbf{g}|\mathbf{z}_s, s)$ 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

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. 11.
end for
```

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. We use the same model proposed by [28], which consists of a group of cylinders. The MHT algorithm proposed here differs from the algorithm proposed in [16] in the use of the latent space to generate proposals in a principled way. This is in contrast with [16], where the modes were selected empirically and the distributions were assumed to be piecewise Gaussian. While in the proposed algorithm, the output from the off-line learning algorithm (LLC) forms clusters (each cluster is described by a Gaussian distribution in latent space), the samples generated from the latent space are indeed drawn from a piecewise Gaussian distribution. The choice of modes to propagate over time becomes straightforward given the statistics of the clusters in the latent space.

4 Experiments

The proposed algorithm has been tested in tracking walking humans. The data set and calibration information were obtained from [28]. The video data set shows a person walking, as captured simultaneously from four different viewpoints. Sigal *et. al.* have used the multiple view information for 3D tracking [28]. We only need monocular sequences, and so we use each of the four videos as an individual test sequence for our algorithm. Our proposed tracker is able to track reliably over 400 frames for all four test sequences (there are 596 frames in each sequence).

We conducted a quantitative comparison of our method (where 10 modes are used) against (1) simple particle filtering, (2) annealed particle filtering [6], and (3) the tracking algorithm proposed by [24] where the GPLVM was used for non-linear dimensionality reduction. We used 2000 particles for the simple particle filtering algorithm and 200 particles for our implementation of [24]. Ten layers and 100 particles for each layer are used in the annealed particle filtering algorithm, following the setup of [6]. The frame rate for both our proposed method and the method of [24] on a 1.6GHz machine with 512MB RAM was approximately one minute per frame, while the annealed particle filtering algorithm took two minutes per frame. The frame rate of the simple particle filtering was about four minutes per frame due to the large number of particles. In both our proposed algorithm and [24], the global translation was modeled separately by simple linear dynamics learned from motion capture data.

Fig. 4 shows the accuracy of the four different tracking algorithms. As proposed in [28], the error is measured as the absolute distance in millimeters between the true and estimated marker positions on the body limbs. 15 markers are chosen which correspond roughly to the locations of the joints and “ends” of the limbs. As can be seen in the graph of Fig. 4, our proposed method is consistently more accurate and the simple particle filtering algorithm does much worse than all other methods. Smart sampling, or a dimensionality reduction method can greatly improve the performance of particle filtering based tracking. Based on the performance reported in [28] (up to 50 frames), our proposed algorithm is

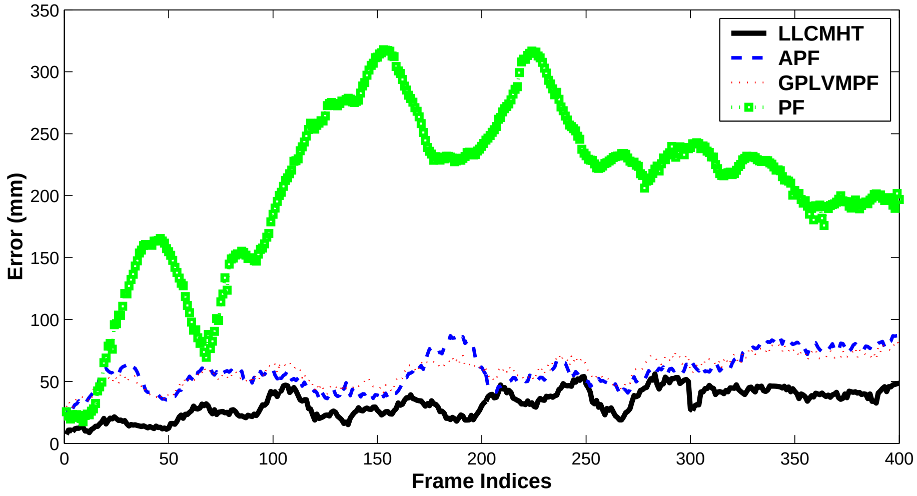

Fig. 4: Estimation Error.

able to track reliably over a longer time on monocular video sequences (all 400 frames, for all four sequences).

Figs. 5 and 6 show example tracking results and the corresponding 3D poses. The results of particle filtering are not shown here due to the large error. With a learned prior model, both the proposed algorithm and particle filtering with GPLVM are able to track reliably when self-occlusion or motion blur occurred. In contrast, annealed particle filtering usually loses track of some body limbs. At frame 183 in Fig. 5, particle filtering with GPLVM 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 a over-smoothed density function and consequently fail to capture large pose change over time. This over-smoothing effect is also demonstrated in the tracking result of frame 70 in Fig. 6, where the left leg movement was underestimated. 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 is able to capture large movements accurately.

5 Conclusions and Future Work

We have proposed a algorithm for tracking 3D body poses. The proposed algorithm is able to track long sequences of video robustly. The experiments demonstrate that our tracker performs much better than the recent tracking algorithms proposed by [6] and [24]. It is also shown that our tracker is capable of handling self-occlusions, motion blur and large movement over time. The tracking algorithm is tested on sequences that contain similar motion with respect to the training data set. Currently we only learned the model of human walking. Essentially, with the proposed learning algorithm, multiple motions can be clustered

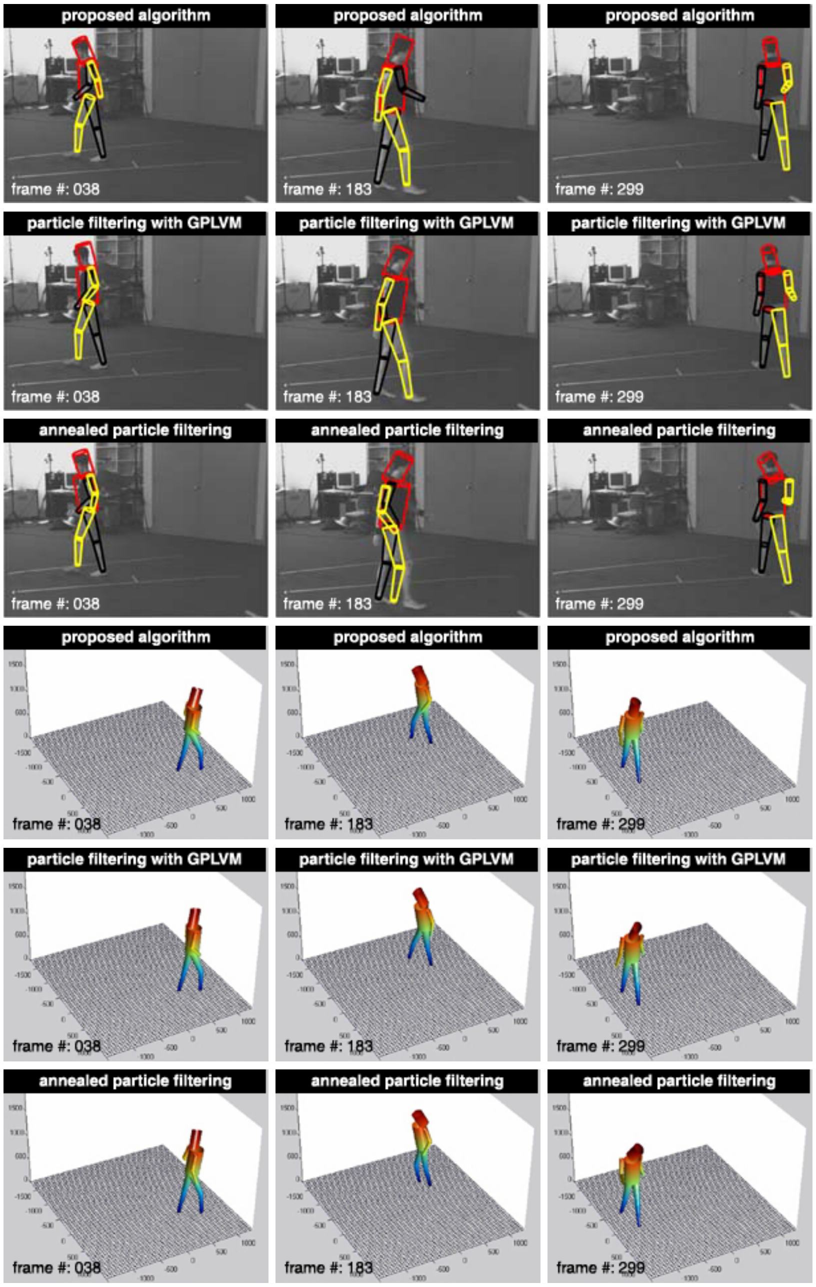
proposed algorithm
frame #: 038
proposed algorithm
frame #: 183
proposed algorithm
frame #: 299
particle filtering with GPLVM
frame #: 038
particle filtering with GPLVM
frame #: 183
particle filtering with GPLVM
frame #: 299
annealed particle filtering
frame #: 038
annealed particle filtering
frame #: 183
annealed particle filtering
frame #: 299
proposed algorithm
frame #: 038
proposed algorithm
frame #: 183
proposed algorithm
frame #: 299
particle filtering with GPLVM
frame #: 038
particle filtering with GPLVM
frame #: 183
particle filtering with GPLVM
frame #: 299
annealed particle filtering
frame #: 038
annealed particle filtering
frame #: 183
annealed particle filtering
frame #: 299

Fig. 5: Selected frames of the tracking results from one of the four sequences and the corresponding 3D poses. The proposed algorithm was able to track the pose reliably while the other two failed to track the movement of the limbs, e.g., forearm (frame 38 and 299) and legs (frame 183). See `http://cs-people.bu.edu/lir/tracking` for videos.

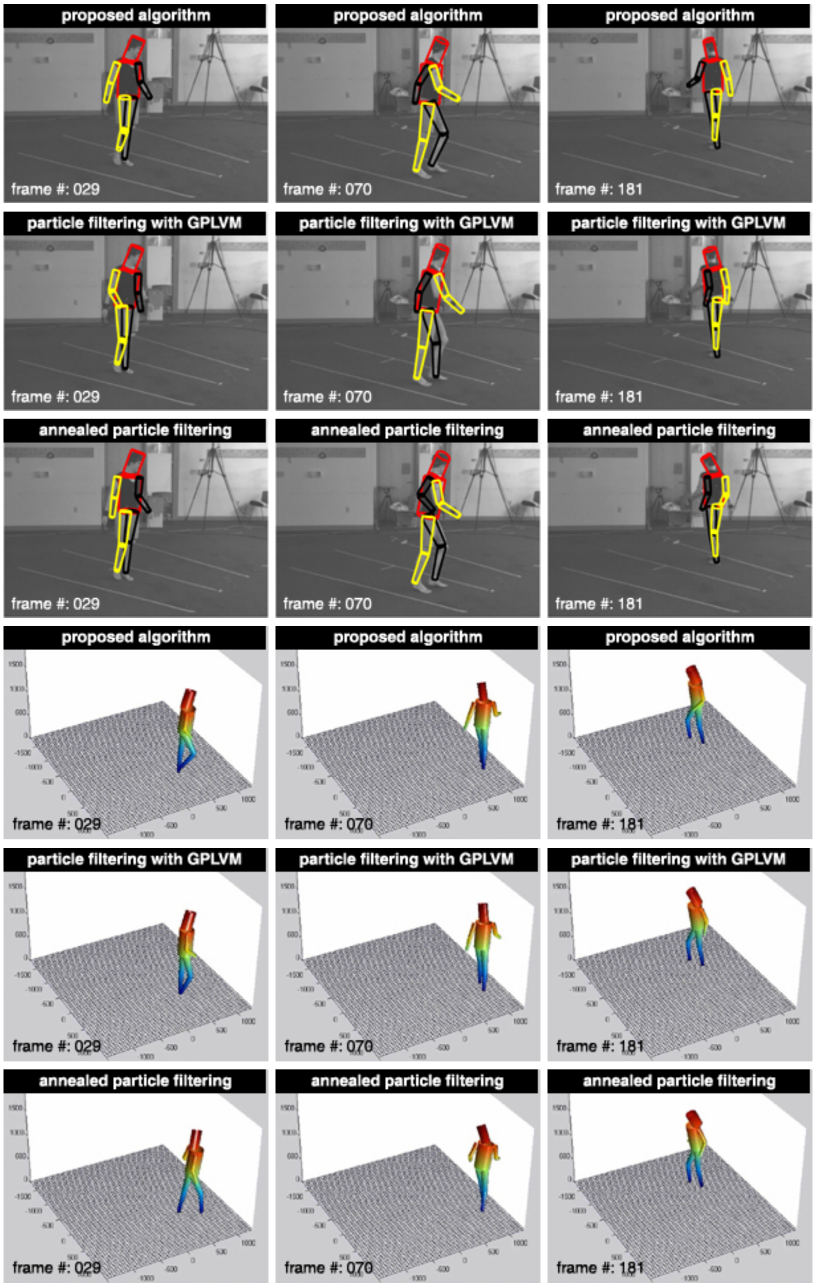
proposed algorithm
frame #: 029
proposed algorithm
frame #: 070
proposed algorithm
frame #: 181
particle filtering with GPLVM
frame #: 029
particle filtering with GPLVM
frame #: 070
particle filtering with GPLVM
frame #: 181
annealed particle filtering
frame #: 029
annealed particle filtering
frame #: 070
annealed particle filtering
frame #: 181
proposed algorithm
frame #: 029
proposed algorithm
frame #: 070
proposed algorithm
frame #: 181
particle filtering with GPLVM
frame #: 029
particle filtering with GPLVM
frame #: 070
particle filtering with GPLVM
frame #: 181
annealed particle filtering
frame #: 029
annealed particle filtering
frame #: 070
annealed particle filtering
frame #: 181

Fig. 6: Selected frames of the tracking results from another sequence and the corresponding 3D poses. The proposed algorithm was able to track the pose reliably while the other two failed to track the movement of the certain limbs, this is similar to what has been observed in Fig.5. See `http://cs-people.bu.edu/lir/tracking` for videos.

in the globally coordinated system; hence, more complicated tracking tasks can be accomplished using the same tracking algorithm when more data becomes available. Another promising direction is to recognize activities during tracking by analyzing the mode jumping in the latent space. It is likely that motions from the same category will form clusters together in the latent space, so whenever a mode jumping occurs, there is likely a change of activity.

Acknowledgments

This work was conducted while the first author was an intern at Honda Research Institute in 2005. Work at Boston University was supported in part through NSF grants IIS-0308213, IIS-0208876, EIA-0202067 and ONR grant N00014-03-1-0108. The authors thank A. Balan, M. Black and L. Sigal at Brown University for providing the motion capture data set and code for performance comparison.

References

1. King, O., Forsyth, D.A.: How does CONDENSATION behave with a finite number of samples? In: Proc. European Conf. on Computer Vision (ECCV). (2000) 695–709
2. Isard, M., Blake, A.: ICondensation : Unifying low-level and high-level tracking in a stochastic framework. In: Proc. European Conf. on Computer Vision (ECCV). (1998) 893–908
3. MacCormick, J., Blake, A.: A probabilistic exclusion principle for tracking multiple objects. In: Proc. IEEE International Conf. on Computer Vision (ICCV). (1999) 572–578
4. Sminchisescu, C., Triggs, B.: Covariance scaled sampling for monocular 3D body tracking. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). (2001) 447–454
5. Sullivan, J., Rittscher, J.: Guiding random particles by deterministic search. In: Proc. IEEE International Conf. on Computer Vision (ICCV). (2001) 323–330
6. Deutscher, J., Blake, A., Reid, I.: Articulated body motion capture by annealed particle filtering. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). (2000) 126–133
7. Sullivan, J., Blake, A., Isard, M., MacCormick, J.: Bayesian object localization in images. International Journal of Computer Vision (IJCV) **44** (2001) 111–135
8. Roweis, R., Saul, L.: Nonlinear dimensionality reduction by locally linear embedding. Science **290** (2000) 2323–2326
9. Tenenbaum, J., Silva, V., Langford, J.: A global geometric framework for nonlinear dimensionality reduction. Science **290** (2000) 2319–2323
10. Belkin, M., Niyogi, P.: Laplacian eigenmaps and spectral techniques for embedding and clustering. In: Advances in Neural Information Processing Systems (NIPS). (2001) 585–591
11. Brand, M.: Charting a manifold. In: Advances in Neural Information Processing Systems (NIPS). (2002) 961–968

12. Teh, W.Y., Roweis, S.: Automatic alignment of local representations. In: Advances in Neural Information Processing Systems (NIPS). (2002) 841–848
13. Lawrence, N.D.: Gaussian process models for visualization of high dimensional data. In: Advances in Neural Information Processing Systems (NIPS). (2003)
14. Wang, L., Hu, W.M., Tan, T.N.: Recent development in human motion analysis. Pattern Recognition **36** (2003) 585–601
15. Isard, M., Blake, A.: CONDENSATION : conditional density propagation for visual tracking. International Journal of Computer Vision (IJCV) **29** (1998) 5–28
16. Cham, T.J., Rehg, J.M.: A multiple hypothesis approach to figure tracking. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). (1999) 239–245
17. Toyama, K., Blake, A.: Probabilistic tracking in a metric space. In: Proc. IEEE International Conf. on Computer Vision (ICCV). (2001) 5057
18. Sidenbladh, H., Black, M.J., Fleet, D.J.: Stochastic tracking of 3D human figures using 2D image motion. In: Proc. European Conf. on Computer Vision (ECCV). (2000) 702–718
19. Sidenbladh, H., Black, M.J.: Learning image statistics for Bayesian tracking. In: Proc. IEEE International Conf. on Computer Vision (ICCV). (2001) 709–716
20. Elgammal, A., Lee, C.S.: Inferring 3D body pose from silhouettes using activity manifold learning. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). (2004) 681–688
21. Grochow, K., Martin, S.L., Hertzmann, A., Popovic, Z.: Style-based inverse kinematics. In: ACM Computer Graphics (SIGGRAPH). (2004) 522–531
22. Safonova, A., Hodgins, J.K., Pollard, N.S.: Synthesizing physically realistic human motion in low dimensional, behavior-specific spaces. In: ACM Computer Graphics (SIGGRAPH). (2004) 514 – 521
23. Sminchisescu, C., Jepson, A.: Generative modelling for continuous non-linearly embedded visual inference. In: Proc. IEEE International Conf. on Machine Learning (ICML). (2004) 140–147
24. Tian, T.P., Li, R., Sclaroff, S.: Tracking human body pose on a learned smooth space. Technical Report 2005-029, Boston University (2005)
25. Urtasun, R., Fleet, D.J., Hertzmann, A., Fua, P.: Priors for people tracking from small training sets. In: Proc. IEEE International Conf. on Computer Vision (ICCV). (2005) 403–410
26. Roweis, R., Saul, L., Hinton, G.E.: Global coordination of local linear models. Advances in Neural Information Processing Systems (NIPS) (2001) 889–896
27. Ghahramani, Z., Hinton, G.E.: The EM algorithm for mixtures of factor analyzers. Technical Report CRG-TR-96-1, University of Toronto (1996)
28. Sigal, L., Bhatia, S., Roth, S., Black, M.J., Isard, M.: Tracking loose-limbed people. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). (2004) 421–428