Real-time Body Tracking Using a Gaussian Process Latent Variable Model

Shaobo Hou Aphrodite Galata Fabrice Caillette Neil Thacker Paul Bromiley

School of Computer Science
The University of Manchester

ISBE
The University of Manchester

Abstract

In this paper, we present a tracking framework for capturing articulated human motions in real-time, without the need for attaching markers onto the subject's body. This is achieved by first obtaining a low dimensional representation of the training motion data, using a nonlinear dimensionality reduction technique called back-constrained GPLVM. A prior dynamics model is then learnt from this low dimensional representation by partitioning the motion sequences into elementary movements using an unsupervised EM clustering algorithm. The temporal dependencies between these elementary movements are efficiently captured by a Variable Length Markov Model. The learnt dynamics model is used to bias the propagation of candidate pose feature vectors in the low dimensional space. By combining this with an efficient volumetric reconstruction algorithm, our framework can quickly evaluate each candidate pose against image evidence captured from multiple views. We present results that show our system can accurately track complex structured activities such as ballet dancing in real-time.

1. Introduction

Tracking of articulated human motion from video sequences has many potential applications including human computer interaction, gesture analysis and computer animation. Expensive, special-purpose motion capture systems are usually required to recover human motion by tracking optical markers attached to an actor's body across multiple views. Therefore, much recent research [3, 11, 1, 29] has been focused on non-intrusive, markerless tracking of articulated human motion. This is a non-trivial problem because human motion resides in high dimensional parameter space and the mapping from the parameter space to the feature space (i.e. image evidence) is complex, nonlinear and multi-modal. Human pose estimation can be poorly constrained by image evidence, which can suffer from self occlusions, noisy signals and low resolution. We propose to resolve such ambiguities by learning a prior model of human motion and using it to constrain pose estimation.

We present a novel markerless system for tracking articulated human motion with multiple calibrated cameras. We exploit the observation that the space of human motions is intrinsically a low dimensional nonlinear subspace embedded in the high dimensional parameter space. In our work, we propose the use of a Back Constrained Gaussian Process Latent Variable Model (BC-GPLVM) [15] to learn a low dimensional embedding of example motions. Tracking is then formulated in a particle filter based framework where each particle is parameterised and propagated in the nonlinear subspace. This allows our framework to efficiently explore the space of human motions. We also learn a dynamic model which captures the local and global dynamics of the low dimensional embedding. The propagation of particles is focused towards the next expected global optimum by using the dynamic model as the motion prior when predicting future particle states. The propagation of global dynamics enables the tracker to escape from any local minima that it may be temporarily trapped in, and allows long and complex motions such as ballet dancing to be tracked robustly.

2. Related Work

One approach to tracking articulated human motion is to treat it as a nonlinear optimisation problem where given an initial estimate, a better pose estimate can be found by using methods based on gradient descent [13]. These frameworks have a number of advantages however, they can get trapped in local optima thus giving poor tracking results. Tracking using particle filtering [2] addresses this limitation by maintaining multiple hypotheses about the current pose. However, particle filtering does not scale well to high dimensional parameter spaces, which can only be properly represented by a very large number of particles [22].

Since the space of human motions is intrinsically low dimensional, it is possible to remove the redundant dimensions and embed the example motions in lower number of dimensions. Principal Component Analysis (PCA) has often been used to learn a subspace of human motions [27, 23]. The main limitation of PCA is that it can only learn

978-1-4244-1631-8/07/$25.00 ©2007 IEEE

linear subspaces and the nonlinearity of human motions is often not well modeled in a linear subspace. Consequently PCA will not be able to effectively reduce the dimensionality of complex and highly non-linear motion data.

Nonlinear dimensionality reduction algorithms such as Isomap [21] and Local Linear Embedding (LLE) [25] can be used to find more effective low dimensional embedding of motion data [31, 16, 6, 19] than PCA. However they lack easily computable mappings from the subspace to the parameter space. Such a mapping is needed in a particle filtering framework because appearance of features needed to be generated from each particle in order to evaluate it against the image evidence. Gaussian Process Latent Variable Model (GPLVM) proposed by Lawrence [14] provides such an efficient probabilistic mapping and also allows the uncertainties of the low dimensional embedding to be estimated. It has been previously applied to human body animation [12] and more recently for tracking simple human motions [29, 28] using deterministic optimisation and particle filtering [26]. In this paper, we use the BC-GPLVM [15] that also guarantees that local distances in the data space are preserved in the latent space, which in turn produces more compact embedding and the mapping from data space to latent space is smooth.

Particle filter based trackers can also be made more accurate by employing a generalisation of the Monte Carlo method such as Simulated Annealing [5] or by partitioning the distribution of human motions into a set of local models [23, 24, 16]. Although our framework also uses a mixture of local models, it has the advantage that the low dimensional representation of human motions learnt by BC-GPLVM is nonlinear and more compact than ones learnt by PCA, allowing for more efficient propagation of particles. Another novelty is that in the estimation of local models, we also take into account the uncertainties associated with the probabilistic mapping of the motion data into its low dimensional representation.

Tracking of simple or cyclical motions can be improved by incorporating local dynamics, however these models have difficulties in handling complex, structured behaviours that exhibit high-order temporal dependencies. A popular approach in human motion analysis is to represent high level behaviours using a Hidden Markov Model (HMM) [1, 11] trained on the distribution of transitions between local motion models. Our framework has the advantage over these works as it models global dynamics using a Variable Length Markov Model [8] which optimises memory length locally to capture temporal dependencies more accurately than fixed order models such as the HMM.

3. Feature Space Representation

In our framework, the human body is represented using a hierarchical articulated model, similar to those used in inverse kinematics systems. Each pose in the motion data is parameterised as $\mathbf{c}_i = [\theta_i, \mathbf{q}_i, \mathbf{p}_i]^T$ where $\theta_i = [\theta_i^0, ..., \theta_i^{18}]^T$ are the 19 Euler angles for the ith frame, each representing a rotational joint with a single Degree of Freedom (DOF). Joints with more than one DOF are represented by multiple one DOF joints with orthogonal axes and zero distance between them. The root orientation $\mathbf{q}_i$ is represented as a quaternion, and $\mathbf{p}_i$ is the root position of the articulated model.

The pose vectors as defined above are not always suitable for learning motion models since the individual vectors do not capture dynamics well. In particular, a motion model learnt from these vectors will generalise poorly to similar motions that have different root trajectories.

A more appropriate feature vector $\mathbf{y}_i = [\theta_i, \dot{\theta}_i, exp(\dot{\mathbf{q}})]_i^T$ is computed from the pose vectors $\mathbf{c}_i$ and $\mathbf{c}_{i-1}$. $\dot{\theta}_i$ is the velocity of the joint angles. The velocity of the root orientation is computed as the difference between the root orientation of the current and the previous frame, expressed in the local coordinate system of the previous frame, $\dot{\mathbf{q}} = \mathbf{q}_{i-1}^{-1}\mathbf{q}_i$. The expression $exp(\dot{\mathbf{q}})$ computes the exponential mapping [10] of the quaternion $\dot{\mathbf{q}}$. Note that we do not incorporate the root positions in the feature vector representation, as in our experience, including such information increases the learning complexity but does not improve the tracking results.

4. Nonlinear Dimensionality Reduction

A BC-GPLVM [15] is used to simultaneously learn a compact low dimensional representation of the training motions and a smooth probabilistic mapping from the subspace of plausible motions to the parameter space. We incorporated the uncertainties associated with the low dimensional representation into a novel clustering algorithm to estimate motion prototypes, as shown in section 5.1.

4.1. Back-constrained GPLVM

Gaussian Process (GP) modelling is a non-parametric approach for solving regression problems which provides an automatic tradeoff between model complexity and data fitness by marginalising over the distribution of functions. Given a training set of q dimensional input points $\{\mathbf{x}_i\}_{i=1}^N$ and the corresponding D dimensional output points $\{\mathbf{y}_i\}_{i=1}^N$, the prediction of the function value at an unseen input position $\mathbf{x}_*$ is a Gaussian distribution conditioned on the training data, the variance/uncertainty of which increases as $\mathbf{x}_*$ moves away from the training data.

GPLVM [14] is a Gaussian Process based nonlinear dimensionality reduction method which represents $\{\mathbf{x}_i\}_{i=1}^N$ as latent variables and assumes their values are initially unknown and should be learnt along with the kernel parameters, specifically by optimising the negative log likelihood of the data modelled as the product of D Gaussian Processes

with a simple Gaussian prior placed on the latent space:

$$L = \frac{D}{2}\ln|\mathbf{K}| + \frac{1}{2}tr(\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^T) + \frac{1}{2}\sum_{i}^{N}||\mathbf{x}_i||^2 + const. \quad (1)$$

where $\mathbf{K}$ is the $N \times N$ kernel matrix computed from the training data using a kernel function such that $K_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j)$. The optimisation of equation 1 is dominated by the repeated inversions of $\mathbf{K}$, therefore in practice, $\mathbf{K}$ is often computed from a much smaller representative set of pseudo points in the latent space [18].

Dimensionality reduction is achieved if $q < D$. The mapping from the latent space to the data space can be non-linearised using an appropriate kernel function such as the radial basis function (RBF):

$$k(\mathbf{x}_i, \mathbf{x}_j) = \alpha_{rbf}exp(-\frac{\gamma}{2}(\mathbf{x}_i - \mathbf{x}_j)^T(\mathbf{x}_i - \mathbf{x}_j)) + \alpha_{bias} + \beta^{-1}\delta_{i,j} \quad (2)$$

where $\alpha_{rbf}, \gamma, \alpha_{bias}$ and β are the kernel parameters and $\delta_{i,j}$ is the Kronecker delta function.

In this work we use a BC-GPLVM [15] which constrains the latent points to be a smooth function g of the data points and optimises the likelihood function 1 with respect to the parameters of g instead of the latent points $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^T$. This allows local distances between data points to be preserved in the latent space and primitive units of motion to be discovered by clustering the constrained latent points.

4.2. Motivation for using BC-GPLVM

We chose to use BC-GPLVM because the basic formulation of GPLVM is ill suited to modelling temporal sequences such as human motions as it is designed to preserve dissimilarity in data space. Although nearby latent points produce similar predictions in the data space, the reverse is not necessarily true, which potentially can create undesirable jumps in the latent space embedding.

This problem can be mitigated by exploiting the inherent temporal ordering in the input sequences and model local dynamics explicitly in the latent space using additional Gaussian Processes [30]. Urtasun et al. [28] proposed a simple approach for balancing between the smoothness of the dynamics mapping and the accuracy of pose reconstructions, and applied it to track walking sequences in monocular view. Moon and Pavlovic[17] showed how to incorporate other types of dynamics into GPLVM. However with these approaches, spatially similar but temporally distant data points can still be placed far apart from each other.

With BC-GPLVM, although temporal ordering is not explicitly modelled, the temporal sequences are still mapped to smooth paths, because consecutive frames of motions tend to be similar.

5. Predictive Dynamic Model

A predictive dynamic model is necessary for ensuring efficient propagation of particles and robust handling of ambiguous image evidence. Such a model is learnt by first clustering the training sequences in latent space into primitive units of motion or motion prototypes. This allows the sequences to be represented at a higher level of abstraction as sequences of motion prototypes. The high level behaviours in the training sequences can then be captured by modelling the temporal dependencies between the motion prototypes using the Variable Length Markov Model [20].

5.1. Clustering with Uncertainty

In order to capture high level behaviours, we estimate motion prototypes from the latent space representation of the training motions. We represent the set of motion prototypes using a Mixture of M Gaussians (MOG). More specifically, we perform clustering in the *augmented* latent space, in which each latent point $\mathbf{x}_i$ is also augmented by its velocity $\dot{\mathbf{x}}_i$. The two issues that need addressing here is choosing automatically the appropriate number of motion prototypes; we solve this by using a similar method to that proposed by Figueiredo and Jain [7]. Also, since BC-GPLVM makes prediction from different latent points with different levels of uncertainty, this suggests that they should not be treated with equal importance in the clustering process, especially if we want to accurately capture the distribution of training data in latent space.

The standard EM clustering algorithm assumes that all data points are equally important and there are no uncertainty associated with the data points. However this is not always a valid assumption, as the measurement process and any intermediate modelling process such as dimensionality reduction (BC-GPLVM or otherwise) can introduce uncertainty into the input data. If the level of uncertainty on each data point can be quantified then they should be incorporated into the clustering process to produce more accurate and representative clusters. The uncertainty about a latent point's position can be estimated by drawing a set of samples (e.g. 100) from its prediction, which are then mapped back into the latent space using the backward mapping g. The covariance matrix of the mapped back samples is then taken to be the uncertainty of that latent point.

Intuitively, data points with lower uncertainties should have greater importance in the clustering process. Therefore we propose a novel clustering algorithm which takes into account of the uncertainties on the data points. For a Gaussian component $N(\mu_m, \mathbf{\Sigma}_m)$ and a data point $\mathbf{x}_n$ with its uncertainty represented by a covariance matrix $\mathbf{C}_n$, the algorithm assumes that the data point is actually drawn from the Gaussian distribution $N(\mu_m, \mathbf{\Sigma}_m + \mathbf{C}_n)$. For a data point with large uncertainty, this spread its contribution more evenly to the nearby clusters and therefore have

less impact on any specific cluster. In practice, this often means that clusters with very narrow covariance will not find sufficient support from the data and is therefore eliminated during the clustering. If all uncertainties are zero, then our algorithm simplifies to standard EM clustering.

Specifically, the contribution of a data point to a Gaussian component is scaled by the inverse of $\mathbf{\Sigma}_m + \mathbf{C}_n$, thus the mean of a Gaussian component is now updated as:

$$\mu_m = (\sum_{n=1}^{N} P_{n,m}\mathbf{A}_{m,n}^{-1})^{-1} \sum_{n=1}^{N} P_{n,m}\mathbf{A}_{m,n}^{-1}\mathbf{x}_n \quad (3)$$

where $P_{n,m}$ is the expected assignment of the nth data point to the mth Gaussian component and $\mathbf{A}_{m,n} = \mathbf{\Sigma}_m + \mathbf{C}_n$. Updating the covariance of a Gaussian component is a little more complicated, by setting the derivative of expected complete log likelihood of the input data w.r.t the covariance $\mathbf{\Sigma}_m$ to $\mathbf{0}$, leads to the following equation:

$$\sum_{n=1}^{N} P_{n,m}\mathbf{A}_{m,n}^{-1}\mathbf{\Sigma}_m\mathbf{A}_{m,n}^{-1} = \quad (4)$$
$$\sum_{n=1}^{N} P_{n,m}\mathbf{A}_{m,n}^{-1}((\mathbf{x}_n - \mu_m)(\mathbf{x}_n - \mu_m)^T - \mathbf{C}_n)\mathbf{A}_{m,n}^{-1}$$

$\mathbf{\Sigma}_m$ can then be found by solving a system of $\frac{(D+1)*D}{2}$ linear equations for the free variables of $\mathbf{\Sigma}_m$. Since updating μ_m and $\mathbf{\Sigma}_m$ depends on an existing estimate of $\mathbf{\Sigma}_m$, the two variables should be alternately updated for a few iterations.

5.2. Learning High Level Behaviour in the Latent Space with VLMM

High level behaviours are captured by learning a Variable Length Markov Model (VLMM) [20] from the abstraction of the training data as sequences of discrete motion prototypes. VLMM is a mathematical framework for modelling complex temporal dependencies at variable temporal scale, this is particular attractive in cases where we need to capture higher order temporal dependencies in some parts of the behaviour and lower order dependencies elsewhere. Therefore VLMM has the advantage over Nth order Markov models as the memory length used for prediction is not fixed but is allowed to vary locally (up to a maximum memory of N) to produce a higher order predictive model. This allows it to disambiguate between complex (or different) activities and provide more accurate predictions by using longer memory length when needed[8]. If there are no higher order temporal dependencies in the training data, the learnt VLMM will simplify to a first order Markov Model. It is worth noting that the size and complexity of a VLMM is automatically learned from the training data.

Although the cost of learning VLMM increases with the size of the training data, the computational cost of prediction using a larger and more complex VLMM remains largely unaffected since a VLMM is conveniently represented by a Probabilistic Finite State Automaton (PFSA). The optimal amount of memory required for prediction at each step is encoded into the states of the PFSA and prediction simply involves traversing from state of the PFSA to the other. A PFSA can be represented as $M = (Q, K, \tau, \gamma, s)$ where K is the alphabet representing the components of the mixture of Gaussians, Q is the set of VLMM states, $\tau : Q \times K \rightarrow Q$ is the transition function, $\gamma : Q \times K \rightarrow [0, 1]$ is the output probability function and $s : Q \rightarrow [0, 1]$ is the probability distribution over the initial states.

5.3. Propagating Global Dynamics

The probability of a latent feature vector $\hat{\mathbf{x}}_i$ given observed image evidence $\mathbf{z}_i$ is:

$$\underbrace{P(\hat{\mathbf{x}}_t|\mathbf{Z}_t)}_{Posterior} = \underbrace{\kappa P(\mathbf{z}_t|\hat{\mathbf{x}}_t)}_{Likelihood} \int \underbrace{P(\hat{\mathbf{x}}_t|\hat{\mathbf{x}}_{t-1:L})}_{MotionPrior} \underbrace{P(\hat{\mathbf{x}}_{t-1}|\mathbf{Z}_{t-1})}_{PreviousPosterior} d\hat{\mathbf{x}}_{t-1} \quad (5)$$

where κ is a normalising constant, $\mathbf{Z}_t = [\mathbf{z}_1, ..., \mathbf{z}_t]$ is the observation history, and $\hat{\mathbf{x}}_{t-1:L}$ are the particle states from frame $t-1$ to $t-L$ where L is less than or equal to the maximum memory of the VLMM. The *posterior* distribution is approximated with a set of particles. Each particle consists of an augmented latent point, a root orientation, a root position, its VLMM state q_t and cluster label k_t. The likelihood of the image evidence, $P(\mathbf{z}_t|\hat{\mathbf{x}}_t)$ is evaluated using an efficient procedure described in section 6.

Since global dynamics are modelled as transitions between motion prototypes using a VLMM, the next motion prototype that governs a particle's dynamics is predicted by sampling from the VLMM function $\gamma(q_t, k_{t+1})$. If a different Gaussian component is predicted for the next frame, i.e. $k_{t+1} \neq k_t$, then a new particle state is sampled from the new cluster. Otherwise the particle's state is propagated using local dynamics as described in the next section.

5.4. Propagating Local Dynamics

A model particle's current state $\hat{\mathbf{x}}_t$ can be predicted from its previous state $\hat{\mathbf{x}}_{t-1}$ and its current Gaussian component $N(\mu_{k_t}, \mathbf{\Sigma}_{k_t})$. Uncertainty in the propagation is modelled by sampling additive noise from the Gaussian component, $[d\mathbf{x}_t, d\dot{\mathbf{x}}_t]^T \sim N(\mathbf{0}, \mathbf{\Sigma}_{k_t})$, with scaling factor λ. The new particle state is $\hat{\mathbf{x}}_t = [\mathbf{x}_t, \dot{\mathbf{x}}_t]^T$, where:

$$\begin{aligned} \dot{\mathbf{x}}_t &= \dot{\mathbf{x}}_{t-1} + \lambda.d\dot{\mathbf{x}}_t \\ \mathbf{x}_t &= \mathbf{x}_{t-1} + \dot{\mathbf{x}}_t + \lambda.d\mathbf{x}_t \end{aligned}$$

In the general case, the current pose of the model particle should be updated by sampling from the prediction distribution at $\mathbf{x}_t$. However, for efficiency reasons, we simply use the mean feature vector $\mathbf{y}_t = [\theta_t, \dot{\theta}_t, exp(\dot{\mathbf{q}}_t)]^T$ predicted by $\mathbf{x}_t$. θ_t are the new joint angles values. The new orientation $\mathbf{q}_t$ is computed by concatenating $\dot{\mathbf{q}}_t$ to the previous

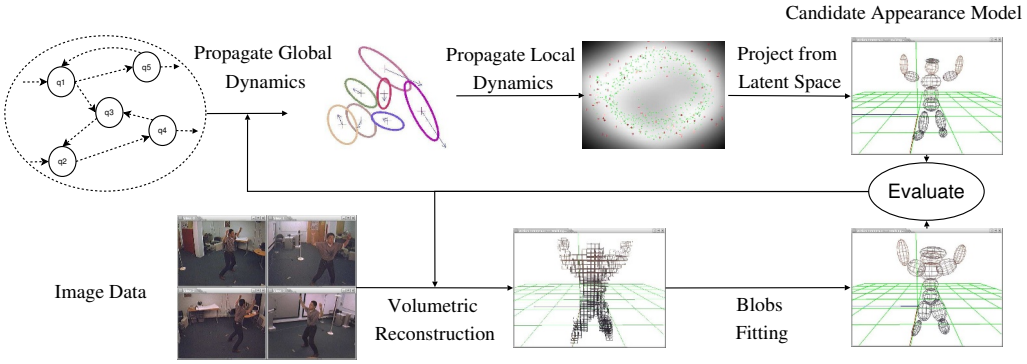

Figure 1. Overview of the runtime system. In the visualisation of the latent space, the red pluses are the pseudo latent points and the green crosses represent the training data. The darker region represents areas with high uncertainty on its predictions.

rotation $\mathbf{q}_{t-1}$, i.e. $\mathbf{q}_t = \mathbf{q}_{t-1}\dot{\mathbf{q}}_t$. Both the root orientation and root position of the model particles are also propagated with Gaussian noise.

6. Fast Likelihood Evaluation

At each new frame, particles are propagated using the dynamic model described in the previous sections and the new posterior likelihood is approximated by re-weighting the particles against the new observed image evidence $\mathbf{z}_t$. $\mathbf{z}_t$ is computed using a hierarchical volumetric reconstruction algorithm which merges information from multiple views to resolve spatial ambiguities [4]. However, the large amount of voxel data prevents the particles from being efficiently evaluated, therefore we summarise the image evidence by fitting a 3D appearance model to the reconstructed voxels using an EM-like procedure [4]. A common appearance model is adopted for both the candidate model configurations and the current observation $\mathbf{z}_t$ in order to make the evaluation process more efficient. This approach is consequently more efficient than evaluating the likelihoods from 2D projections in each camera view.

6.1. Generating Appearance Models

An appearance model is represented by a mixture of non-overlapping Gaussian blobs attached to the bones of the articulated model and can be easily generated from any set of kinematics parameters, as illustrated by the top right image in Figure 1. Each Gaussian blob defines a distribution over 3D position and colour and is defined in the local coordinate system of the corresponding body part.

Initialising the blobs fitting from the last tracked pose can be insufficient for fast movements, causing some blobs to "snap" to incorrect body parts. Therefore we also initialise blobs fitting with appearance models predicted by the VLMM and retain the best result as the "image evidence". An important advantage of this blob fitting procedure is that it can also automatically recover from tracking failures. If the best fitting mixture provides a poor fit then the tracker is deemed lost and the particles are re-initialised according to the distribution s over the initial VLMM states.

6.2. Particle Evaluation with Relative Entropy

A model configuration (particle) is evaluated by first generating an appearance model from the particle state, and then comparing it to the image evidence. Let us denote $F = \sum_i \alpha_i f_i$ as the mixture generated from the model and $G = \sum_i \beta_i g_i$ as the one corresponding to image evidence. The Kullback-Leibler (KL) divergence $D_{KL}(F\|G)$ between the two mixtures can be used to measure their similarity. Since there is no closed form solution for KL divergence between two Mixtures of Gaussians, $D_{KL}(F\|G)$ can be computed using the approximation proposed by [9] for non-overlapping clusters:

$$D_{KL}(F\|G) \approx \sum_{i=1}^{n} \alpha_i \left(D_{KL}(f_i\|g_i) + \ln \frac{\alpha_i}{\beta_i} \right) \quad (6)$$

This approximation can be efficiently computed using the closed form solution of the KL divergence between two Gaussian blobs $f \sim \mathcal{N}(\mu_f, \Sigma_f)$ and $g \sim \mathcal{N}(\mu_g, \Sigma_g)$:

$$\frac{1}{2}\left(\ln \frac{|\Sigma_f|}{|\Sigma_g|} - d + tr(\Sigma_f^{-1}\Sigma_g) + (\mu_g - \mu_f)^T \Sigma_f^{-1}(\mu_g - \mu_f) \right) \quad (7)$$

d is the dimensionality of the blobs f and g. The weight of a particle is inversely proportional to $D_{KL}(F\|G)$. The weights are normalised before resampling.

7. Results

In order to test the tracking framework, we use calibrated cameras to capture video sequences from multiple views, at 30fps with a resolution of 320×240. Learning the dynamic model takes 1 to 3 hours depending on the amount of training data available. As a preprocessing stage, the training sequences used to learn a BC-GPLVM are sub-sampled at 30fps to match their timing with the frame rate of the

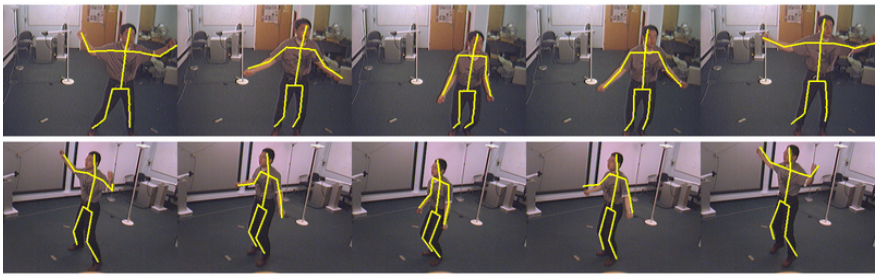

Figure 2. Tracking results (front and side view) for the jumping jacks sequence using 500 particles. Every 5th frame is shown.

cameras. In order to obtain a sufficient number of latent points for clustering, the sequences are also super-sampled to 300fps and mapped into the latent space. The tracking system runs at an average of 10 fps.

We first show our system tracking a jumping jack sequence. The training data was taken from the CMU mocap database (mocap.cs.cmu.edu) of two subjects at 120fps. From the training data, we learned a BC-GPLVM with a 3D latent space and 100 pseudo latent points. Clustering discovered 15 motion prototypes and a VLMM with a maximum memory length of 5 was learned to model dynamics. The test data was captured in a normal office environment using 4 cameras. It can be seen that even though the subject is wearing fairly ordinary clothes, the system tracks well. Figure 2 shows the tracking results superimposed onto the videos captured by 2 of the 4 cameras.

Ballet dancing is a good example of a complex structured activity that can exhibit fast movements. It is difficult to track such activities without a prior motion model, thus it is a good test for our system. We used 5 cameras to capture 1200 frames of a dancer performing 3 repetitions of a ballet sequence at 30fps for training. Additional sequences were also captured and used for testing. We learnt a BC-GPLVM with a 5D latent space and 200 pseudo latent points from the training data and 71 motion prototypes were discovered. A high level model of dance behaviour in the latent space was learned using a VLMM with a maximum memory length of 10. In Figure 3 we compare our system to two particle filter based trackers with annealing.

We also compare our system to two different trackers. The first tracker uses VLMM to propagate particles in the full pose space [3] and the second tracker propagate particles in the latent space of BC-GPLVM using just a First Order Markov Model (FOMM). Each tracker was tested on the ballet sequence 50 times using 200 particles and the result is shown in figure 4. Our system achieves significantly lower mean tracking errors than the other two trackers. The graph also shows our tracker is more robust than the first tracker because we constrain our tracker to only predict valid poses from the low dimensional latent space. Our tracker is also more robust than the second tracker because higher order predictions provided by VLMM allows particles to be focused toward the next expected global optimum, which is especially important for small number of particles.

In figure 5 and 6, we show that by using appropriate image features, our tracking framework can also be modified to track human motions from monocular view. In particular, we compare our system to a gradient descent based tracker using 'balanced GPDM' similar to [28]. We annotated the 2D positions of a set of joints to simulate the results given by a 2D appearance tracker, such as the WSL tracker used in [28]. The likelihood of a particle is then computed as the sum of squared distances between the annotated joint positions and the predicted joint positions projected into the image space. Figure 5 shows a well known walking sequence for which we annotated the positions of the head, the centre of the hip, the left hand and both feet. Both trackers were able to track the sequence quite well. This is because walking is a cyclic activity with relatively simple dynamics which is easier to model (even with PCA) than dancing. The labelled joint features are sufficiently informative to constrain its tracking. Figure 6 shows the beginning of the ballet sequence being tracked from one of the views, the positions of the head, both hands and both feet are annotated. Because the sequence exhibits movements toward and away from the image plane, this presents a good deal of ambiguity in tracking which may not be sufficiently constrained the labelled joints, especially for estimating root position. Therefore in order to obtain some sensible results, we helped both trackers by giving them the ground truth root positions. The screenshots show that our tracker was able to estimate the correct poses despite the sparse image evidence, while the gradient descent based tracker soon became trapped in a local minima. It is also worth noting that even when our tracker is not helped along with ground truth root positions, it can still recover the correct pose, albeit at an incorrect depth from the camera viewpoint.

8. Conclusion

We have presented a markerless tracking system for estimating human poses from images captured by multiple calibrated cameras. We constrain pose estimation by learning

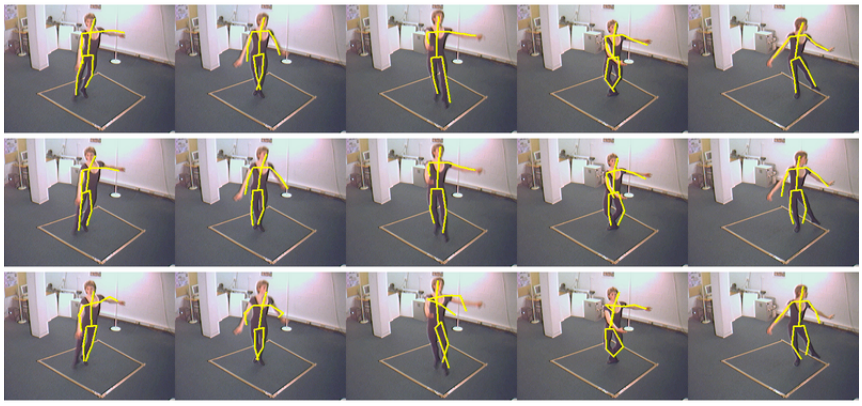

Figure 3. Tracking results for the ballet dance exercise. (top row) Our method with 500 particles. (middle row) Annealed particle propagation in latent space with 5 layers, each has 500 particles. (bottom row) Annealed particle propagation in full pose space with 5 layers, each has 500 particles. Every 40th frame is shown.

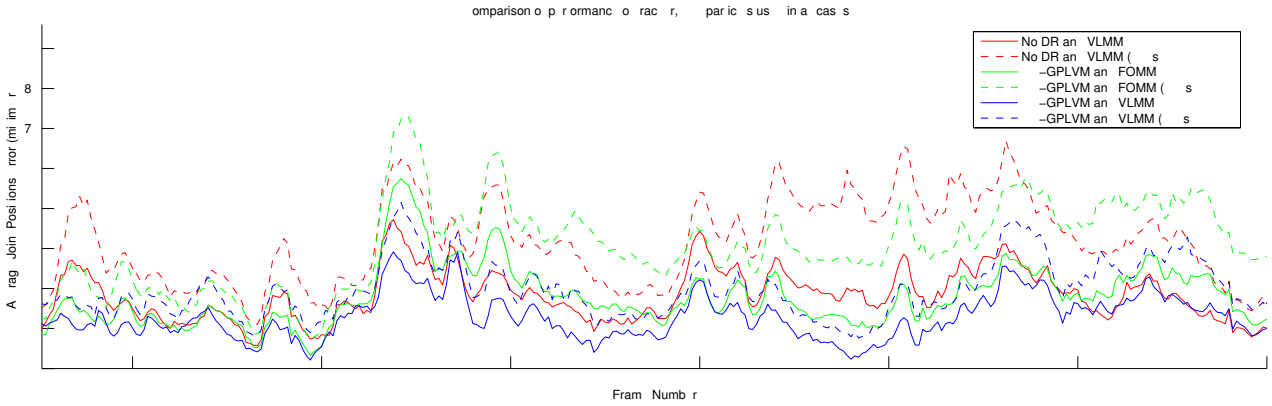

Figure 4. Comparison of our system on the first dance exercise against other systems with dynamics model. 200 particles used in all cases.

a compact low dimensional space of plausible motions using BC-GPLVM, which makes particle propagation more efficient. Accurate tracking of complex structured activities such as ballet dance is achieved by learning a prior motion model by abstracting the training motions as sequences of motion prototypes. The motion prototypes are estimated using a novel clustering algorithm which takes into account of the uncertainties of the latent space embedding of the training motions. Results show that by capturing the temporal dependencies between the motion prototypes in the latent space using VLMM, our tracker is able to accurately predict the high level behaviours of activities and prevent the particle filters from being trapped in poor local minima.

Acknowledgements This work made use of Neil Lawrence's publicly available Fast GPLVM code.

References

[1] A. Agarwal and B. Triggs. Tracking articulated motion using a mixture of autoregressive models. In *Proc. ECCV*, pages 54–65, 2004.

[2] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for on-line non-linear/non-Gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, pages 100–117, 2001.

[3] F. Caillette, A. Galata, and T. Howard. Real-Time 3-D Human Body Tracking using Variable Length Markov Models. In *Proc. BMVC*, volume 1, pages 469–478, 2005.

[4] F. Caillette and T. Howard. Real-Time Markerless Human Body Tracking with Multi-View 3-D Voxel Reconstruction. In *Proc. BMVC*, pages 597–606, 2004.

[5] J. Deutscher, A. Blake, and I. Reid. Articulated body motion capture by annealed particle filtering. *Proc. CVPR*, pages 126–133, 2000.

[6] A. M. Elgammal and C. Lee. Inferring 3d body pose from silhouettes using activity manifold learning. In *Proc. CVPR*, pages 681–688, 2004.

[7] M. A. T. Figueiredo and A. K. Jain. Unsupervised learning of finite mixture models. *IEEE Trans. PAMI.*, 24(3):381–396, 2002.

Figure 5. Simulation of tracking of a walking sequence with BC-GPLVM + VLMM (top) and GPDM + gradient descent (bottom)

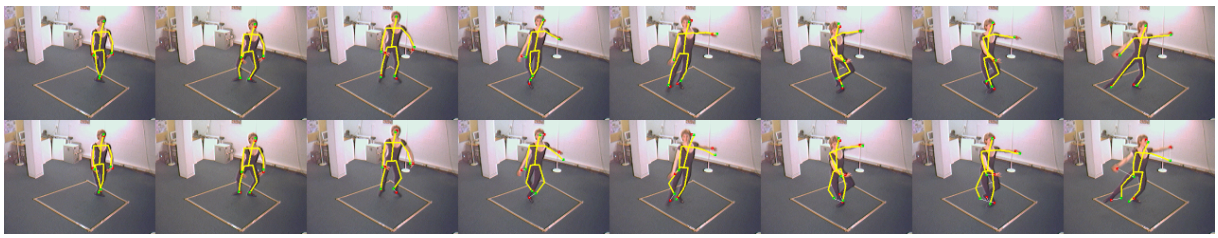

Figure 6. Simulation of tracking of part of a ballet sequence with BC-GPLVM + VLMM (top) and GPDM + gradient descent (bottom)

[8] A. Galata, N. Johnson, and D. Hogg. Learning variable length markov models of behaviour. *CVIU*, 81(3):398–413, 2001.

[9] J. Goldberger, S. Gordon, and H. Greenspan. An efficient image similarity measure based on approximations of kl-divergence between two gaussian mixtures. In *Proc. ICCV*, page 641, 2003.

[10] F. S. Grassia. Practical parameterization of rotations using the exponential map. *J. Graph. Tools*, 3(3):29–48, 1998.

[11] K. Grauman, G. Shakhnarovich, and T. Darrell. Inferring 3d structure with a statistical image-based shape model. In *Proc. ICCV*, page 641, 2003.

[12] K. Grochow, S. L. Martin, A. Hertzmann, and Z. Popović. Style-based inverse kinematics. *ACM Trans. Graph.*, 23(3):522–531, 2004.

[13] N. R. Howe, M. E. Leventon, and W. T. Freeman. Bayesian reconstruction of 3d human motion from single-camera video. In *NIPS 12*, pages 820–826, 2000.

[14] N. D. Lawrence. Probabilistic non-linear principal component analysis with gaussian process latent variable models. *JLMR 6*, pages 1783–1916, 2005.

[15] N. D. Lawrence and J. Q. Candela. Local distance preservation in the gp-lvm through back constraints. In *Proc. ICML*, 2006.

[16] R. Li, M.-H. Yang, S. Sclaroff, and T.-P. Tian. Monocular tracking of 3d human motion with a coordinated mixture of factor analyzers. In *Proc. ECCV*, pages 137–150, 2006.

[17] K. Moon and V. Pavlovic. Impact of dynamics on subspace embedding and tracking of sequences. In *Proc. CVPR*, pages 198–205, 2006.

[18] J. Quinonero-Candela and C. E. Rasmussen. A unifying view of sparse approximate gaussian process regression. *J. Mach. Learn. Res.*, 6:1939–1959, 2005.

[19] A. Rahimi, B. Recht, and T. Darrell. Learning appearance manifolds from video. In *Proc. CVPR*, pages 868–875, 2005.

[20] D. Ron, S. Singer, and N.Tishby. The power of amnesia. In *NIPS*, pages 176–183, 1994.

[21] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.

[22] H. Sidenbladh, M. J. Black, and D. J. Fleet. Stochastic tracking of 3d human figures using 2d image motion. In *Proc. ECCV*, pages 702–718, 2000.

[23] H. Sidenbladh, M. J. Black, and L. Sigal. Implicit probabilistic models of human motion for synthesis and tracking. In *Proc. ECCV*, pages 784–800, 2002.

[24] C. Sminchisescu and A. Jepson. Generative modeling for continuous non-linearly embedded visual inference. In *Proc. ICML*, page 96, 2004.

[25] J. B. Tenenbaum. Mapping a manifold of perceptual observations. In *NIPS 10*, pages 682–688, 1998.

[26] T.-P. Tian, R. Li, and S. Sclaroff. Articulated pose estimation in a learned smooth space of feasible solutions. In *CVPR Learning Workshop*, 2005.

[27] R. Urtasun, D. J. Fleet, and P. Fua. Monocular 3-d tracking of the golf swing. In *Proc. CVPR*, pages 932–938, 2005.

[28] R. Urtasun, D. J. Fleet, and P. Fua. 3d people tracking with gaussian process dynamical models. In *Proc. CVPR*, pages 238–245, 2006.

[29] R. Urtasun, D. J. Fleet, A. Hertzmann, and P. Fua. Priors for people tracking from small training sets. In *Proc. ICCV*, pages 403–410, 2005.

[30] J. Wang, D. Fleet, and A. Hertzmann. Gaussian process dynamical models. In *NIPS 18*, pages 1443–1450, 2006.

[31] Q. Wang, G. Xu, and H. Ai. Learning object intrinsic structure for robust visual tracking. In *Proc. CVPR*, volume 02, page 227, 2003.