

Tracking Human Body Pose on a Learned Smooth Space

Tai-Peng Tian Rui Li Stan Sclaroff
Computer Science Department
Boston University
Boston, MA 02215

Abstract

Particle filtering is a popular method used in systems for tracking human body pose in video. One key difficulty in using particle filtering is caused by the curse of dimensionality: generally a very large number of particles is required to adequately approximate the underlying pose distribution in a high-dimensional state space. Although the number of degrees of freedom in the human body is quite large, in reality, the subset of allowable configurations in state space is generally restricted by human biomechanics, and the trajectories in this allowable subspace tend to be smooth. Therefore, a framework is proposed to learn a low-dimensional representation of the high-dimensional human poses state space. This mapping can be learned using a Gaussian Process Latent Variable Model (GPLVM) framework. One important advantage of the GPLVM framework is that both the mapping to, and mapping from the embedded space are smooth; this facilitates sampling in the low-dimensional space, and samples generated in the low-dimensional embedded space are easily mapped back into the original high-dimensional space. Moreover, human body poses that are similar in the original space tend to be mapped close to each other in the embedded space; this property can be exploited when sampling in the embedded space. The proposed framework is tested in tracking 2D human body pose using a Scaled Prismatic Model. Experiments on real life video sequences demonstrate the strength of the approach. In comparison with the Multiple Hypothesis Tracking and the standard Condensation algorithm, the proposed algorithm is able to maintain tracking reliably throughout the long test sequences. It also handles singularity and self occlusion robustly.

1. Introduction

Tracking and analysis of human body motion has been an active research area in recent years. Efficient and effective solutions would bring great advancement in many important applications, e.g. video surveillance, gesture analysis, human computer interfaces, computer animation, etc.

Particle filtering is one of the most popular algorithms for tracking human body pose. Given the degrees of freedom in the body pose space is usually ≥ 20 , it is almost impractical to apply particle filtering directly as a large number of particles is required to adequately approximate the underlying probability distribution in the body pose space. Hence, a significant amount of effort has been put into the research of making the particle filtering algorithm more efficient [6, 25, 26].

In this paper, we exploit the observation that most dynamic human motion is intrinsically low-dimensional [7, 8, 21]. We learn a low-dimensional representation of human motion and use it for tracking articulated bodies. The tracking framework is based on the Condensation algorithm and the particle state is represented using the low-dimensional representation. By making the transition from tracking in high-dimensional space to low-dimensional space, we avoid the sample impoverishment [12] problem and at the same time retain the simplicity of the Condensation algorithm.

2. Previous Work

There is a broad range of related work in the areas of particle filtering, non-linear dimensionality reduction, and human motion tracking.

2.1 Particle Filtering

Many approaches have been proposed to overcome the curse of dimensionality inherent in the Particle Filtering framework. The key idea is to reposition the particles according to some *importance function* [10] to ensure a high *survival rate* [16]. For example the particles are resampled using weighted resampling [16] or repositioned using deterministic search [25, 27] to localize the set of particles around significant maxima of the importance function. Others apply a coarse to fine search on the weighting function such as the Annealed Particle Filter [6] or Layered Sampling [26]. If the particle dynamics can be factored into independent components, then Partitioned Sampling [16] can be used to improve the particle filter.

One way to avoid the curse of dimensionality is to apply tracking on a dimensionality reduced space. Given that human motion has been observed to be intrinsically low-dimensional [7, 8, 21], tracking on such a dimensionality reduced space is a viable solution. Thus, in contrast with previous approaches that attempt to reposition sampled particles in an optimal way, we instead sample directly from a dimensionality reduced space. Furthermore, by choosing to learn an embedding from a training set of feasible poses, we can form a more “focused” prior for sampling the poses. The prior is focused in the sense that the probability of sampling a biomechanically impossible pose is much more remote when compared to sampling in the original space. Therefore, instead of considering the entire space of possible state configurations, we learn a low-dimensional representation for the human body poses from a set of training data that encompasses the state configurations we are interested in tracking.

2.2 Non-Linear Dimensionality Reduction

Dimensionality reduction is an active research area. Given its wide applications, there is also a large amount of work done in this area, such as Principal Component Analysis (PCA), Local Linear Embedding (LLE) [20], Isomap [29] and Laplacian Eigenmaps [3]. PCA is inadequate to handle non-linear behavior. Primarily the works of LLE, Isomap and Laplacian Eigenmaps are representative. These embedding techniques are typically not invertible. For the purpose of our algorithm, invertible embedding is a crucial property. In particular, the inverse mapping is required during the measurement phase of our modified Condensation algorithm where a weight is assigned to each particle based on how well the state configuration explains the image measurements. Since the particles are expressed in the low-dimensional space and the measurement is done in the original space, the inverse mapping is then used to recover the original state configurations from the low-dimensional representation.

There are existing works that provide inverse mappings, such as Charting [4] and the Gaussian Process Latent Variable Model (GPLVM)[13]. In principle, any dimensionality reduction technique that provides an inverse mapping will be applicable in our framework but in particular, GPLVM is chosen as it has been shown to perform well for human pose synthesis and animation [8].

2.3 Human Motion Tracking

There is a broad range of work related to human motion tracking and a recent survey is available in [31]. We will focus on the class of stochastic tracking algorithms.

After the seminal work of Isard and Blake [9], the Condensation algorithm has been adapted for human motion tracking [6, 25]. Cham and Rehg’s Multiple Hypothesis Tracking [5] modified the algorithm to track modes in the distribution with a simpler Piece-Wise Gaussian (PWG) distribution. Toyama and Blake [30] demonstrated how to incorporate exemplars into such a probabilistic tracking framework. Sidenbladh and co-workers incorporate a more specific motion model and accurate background modelling through learning [22, 23].

The work of Sminchisescu and Jepson [24] is closely related to ours. They also proposed the use of dimensionality reduction techniques to cope with the high dimensional state spaces encountered during 3D tracking. A sophisticated algorithm is developed to facilitate optimization over the embedded manifold in the lower-dimensional space. They solved the two problems of dimensionality reduction and inverse mapping recovery using separate algorithms. They chose to use a Gaussian Mixture Model (GMM) for learning the prior and the GMM has been known to exhibit undesirable artifacts such as skinny Gaussians and “bumpy” surfaces [8]. In contrast, the Gaussian Process Latent Variable Model (GPLVM) smoothly interpolates the underlying data. Secondly, the representatives used in the kernel regression [24] are chosen in an heuristic manner. The representatives are selected from the centroids learned in the GMM and do not take into account the underlying intrinsic space curvature (for example, areas of high curvature will require a denser sampling). In contrast for the GPLVM, the Information Vector Machine [13] is used to select representatives based on an information theoretic framework.

The choice of using the GPVLM leads to a principled way of solving the embedding and reverse mapping problem. It also leads to a simpler algorithm for 2D human motion tracking.

3. Formulation

Here we consider the problem of tracking 2D human body pose in a dimensionality reduced space using a particle filter. We assume that the set of valid 2D poses can be described as a manifold $\mathcal{Y} \subset \mathbb{R}^n$. Given a 2D human body pose $\mathbf{y} \in \mathcal{Y}$ we would like to express it using a lower-dimensional representation $\mathbf{x} \in \mathcal{X}$ (where $\mathcal{X} = \mathbb{R}^m, m < n$) and apply tracking in the lower-dimensional space \mathcal{X} . This motivates the need to learn the mappings $\mu : \mathcal{X} \rightarrow \mathcal{Y}$ and $\mu^{-1} : \mathcal{Y} \rightarrow \mathcal{X}$.

The proposed framework consists of two components. The first component is an off-line algorithm to learn the mappings μ and μ^{-1} . The second component is the on-line algorithm for tracking using the lower-dimensional representation with a modified Condensation algorithm.

Algorithm 1 GPLVM Learning Algorithm

Initialize size of active set D , number of iterations T .
Initialize the \mathbf{X} from \mathbf{Y} through ISOMAP [29].
for T iterations **do**
 Select a new active set based on [15].
 Optimize L (Eq 1) using scaled conjugate gradient (SCG).
 Select a new active set.
 for each component j not in the active set, **do**
 Optimize L' (Eq 2) with respect to \mathbf{x}_j using SCG.
 end for
end for

3.1 Learning the Low-Dimensional Representation

In the training stage, the goal is to learn smooth mappings μ and μ^{-1} between the manifold of feasible 2D body poses \mathcal{Y} and the lower dimensional space \mathcal{X} .

The learning algorithm used is Gaussian Process Latent Variable Model (GPLVM). Given training poses $\{\mathbf{y}_i\}$ as inputs, we learn a low-dimensional space called the *latent space* (we will use both terms interchangeably). During learning, we estimate a *latent space variable* \mathbf{x}_i for each corresponding input training example \mathbf{y}_i . The learning process is formulated as an optimization problem.

For a detailed tutorial on Gaussian Process and the GPLVM, see [13, 17]. We only describe the basic mechanism and adaptation of the GPLVM to our specific problem. The presentations here follows [8].

3.1.1 Kernel Matrix

The kernel matrix, \mathbf{K} , is the core of the GPLVM model. We use the Radial Basis Function (RBF) kernel function because it smoothly interpolates the latent space. The RBF kernel takes the form:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \alpha \exp(-\frac{\gamma}{2}(\mathbf{x}_i - \mathbf{x}_j)^T(\mathbf{x}_i - \mathbf{x}_j)) + \delta_{ij}\beta^{-1},$$

where $k(\mathbf{x}_i, \mathbf{x}_j)$ is the element in i -th row and j -th column of the kernel matrix \mathbf{K} , α controls the scale of the output functions, γ is the inverse width parameter. The scalar $k(\mathbf{x}_i, \mathbf{x}_j)$ models the proximity between two points \mathbf{x}_i and \mathbf{x}_j .

3.1.2 GPLVM Learning

GPLVM learning is the process of learning the kernel parameters (α , β and γ) and latent variables \mathbf{x}_i 's. The reverse mapping $\mu(\cdot)$ is constructed from the latent variables \mathbf{x}_i 's.

Within the training data $\{\mathbf{y}_i\}$, each \mathbf{y}_i is a M dimensional vector. We collect the m -th dimension of

input \mathbf{y}_i 's into \mathbf{Y}_m . Then we maximize the posterior $p(\{\mathbf{x}_i\}, \alpha, \gamma | \{\mathbf{y}_i\})$, which corresponds to minimizing the following objective function:

$$L = \frac{M}{2} \ln |\mathbf{K}| + \frac{1}{2} \sum_m \mathbf{Y}_m^T \mathbf{K}^{-1} \mathbf{Y}_m + \frac{1}{2} \sum_i \|\mathbf{x}_i\|^2, \quad (1)$$

with respect to the α , γ and \mathbf{x}_i 's.

The intuition and derivation of L can be found in [14]. This optimization process is realized through the Scaled Conjugate Gradient (SCG) method. The gradients needed for optimization are listed in the Appendix A.

To speed up the training, \mathbf{K} is only learned on a subset of the training data. This selected subset is called *active set* and denoted by \mathbf{I} . The active set can be considered as a sparse representation of the training data. The process of selecting the active set is described in [15]. The remaining points are denoted by \mathbf{J} . Active set selection allows us to optimize each point in \mathbf{J} independently [32]. We solve for each \mathbf{x}_j in \mathbf{J} by minimizing the following objective function:

$$L'(\mathbf{x}_j, \mathbf{y}_j) = \frac{\|\mathbf{y}_j - \mu(\mathbf{x}_j)\|^2}{2\sigma^2(\mathbf{x}_j)} + \frac{M}{2} \ln \sigma^2(\mathbf{x}_j) + \frac{1}{2} \|\mathbf{x}_j\|^2, \quad (2)$$

where

$$\mu(\mathbf{x}_j) = \mathbf{Y}^T \mathbf{K}_{I,I}^{-1} \mathbf{k}_{I,j}, \quad (3)$$

and $\mathbf{K}_{I,I}$ denotes the kernel matrix learned from the *active set*. The vector $\mathbf{k}_{I,j}$ is made up of the rows in \mathbf{I} from the j -th column of \mathbf{K} , and the variance is

$$\sigma^2(\mathbf{x}_j) = k(\mathbf{x}_j, \mathbf{x}_j) - \mathbf{k}_{I,j}^T \mathbf{K}_{I,I}^{-1} \mathbf{k}_{I,j}. \quad (4)$$

Taking gradients of L' with respect to \mathbf{x}_j does not depend on other data in \mathbf{J} . The gradients of L' with respect to \mathbf{x} and \mathbf{y} are listed in Appendix A. The learning process is summarized in Algorithm 1.

The mapping μ^{-1} is given by

$$\mu^{-1}(\mathbf{y}) = \arg \max_{\mathbf{x}} L'(\mathbf{x}, \mathbf{y}). \quad (5)$$

SCG is used to solve the above optimization problem.

3.2 Modified Condensation Tracker

To describe the proposed modified Condensation Tracker, we follow the notations of Isard and Blake [9]. The Condensation algorithm [9] represents the conditional probability at each time step using a set of particles $\{s_t^{(n)}\}$. This set of particles is propagated through time. At each time step the particles undergo *drifting*, *diffusion* and *measurement reinforcement*.

In the modified Condensation tracker, the particles $\{s_t^{(n)}\}$ are represented using the low-dimensional space.

Algorithm 2 Modified Condensation Tracker

The sample set $\{s_t^{(n)}, \pi_t^{(n)}, n = 1, \dots, N\}$ is from the previous time step. The purpose is to construct the time $t + 1$ sample set.

Construct the n^{th} of N new samples as follows:

Select a sample $s_{t+1}^{(n)}$ from $\{s_t^{(n)}\}$ with replacement and probability proportional to their weights $\{\pi_t^{(n)}\}$

Predict by sampling $s_{t+1}^{(n)}$ from

$$p(x_{t+1}|x_t = s_{t+1}^{(n)}).$$

Measure and weight the new position in terms of the measured features z_{t+1} :

$$\pi_{t+1}^{(n)} = p\left(z_{t+1}|y_{t+1} = \mu(s_{t+1}^{(n)})\right)$$

Normalize such that $\sum_N \pi_{t+1}^{(n)} = 1$.

Estimate the state using the mean

$$\mathbf{E}[x_{t+1}] \approx \sum_{n=1}^N \pi_{t+1}^{(n)} s_{t+1}^{(n)}$$

The state configuration is given by $\mu(\mathbf{E}[x_{t+1}])$

Thus sampling is performed in the low-dimensional space as well. The original space configuration is only used when evaluating the observation likelihood $p(\mathbf{z}|\mathbf{y} = \mu(\mathbf{x}))$, where the inverse function $\mu(\cdot)$ maps the low-dimensional representation back into the original configuration space and \mathbf{z} are the image features. The modified Condensation tracker is described in Algorithm 2.

One way to model the state dynamics is using a random walk in the latent space described by

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \omega. \quad (6)$$

Where $\omega \sim N(0, \nu^2)$, and ν^2 can be learned from training sequences. A motivation for using such simple dynamics is due to the way the latent space is arranged as similar poses are packed close together in the latent space [8]. Hence if we assume that the frame to frame pose difference is not too large, we can model the dynamics using a random walk.

4. Implementation

We demonstrate the proposed approach on the problem of tracking upper body hand gestures. The upper body model is described using the Scaled Prismatic Model [5]. The marker positions used are show in Figure 1. The state \mathbf{y} is defined in terms of the Scaled Prismatic Model as well as an overall scaling parameter and a 2D translation. The input state vector is denoted by \mathbf{y} , where $|\mathbf{y}| = 25$.

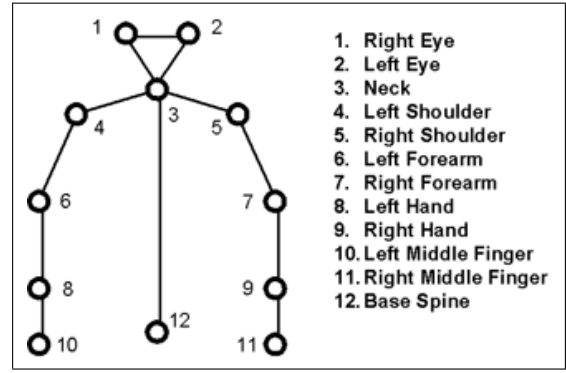


Figure 1: *Upper Body Marker Positions*

We learn a 2D latent space using the GPLVM model learning. Based on the results of Growchow *et al.*'s work [8], a 3D latent space is empirically found to be sufficient for 3D pose synthesis. Whereas for our application of 2D human pose tracking, we found that 2D latent space is sufficient to give good results though using a high dimension (e.g. three or four) will improve the accuracy but it requires more samples to adequately sample the higher-dimensional space.

The training data of upper body poses of a male character are synthesized in a manner similar to [19]. Computer graphics model of a human is used to create poses taken from the gestures used in aircraft signals [2]. The 3D poses are rendered using Poser 5 and the projected marker positions are used to compute the Scaled Prismatic Model in the training set. The poses are rendered from different viewpoints on the view-sphere to account for viewpoint changes.

Training the GPLVM with 5000 samples takes around five hours to complete on an Intel 2.2GHz PC with the GPLVM implemented in Matlab [1]. The learned latent space is shown in Figure 2 together with the corresponding silhouette images for easy visualization. It can be seen that the silhouette images of similar poses are placed close to each other and there are smooth transitions between different body poses.

For the modified Condensation tracker, the observation likelihood is modeled as using

$$p(\mathbf{z}|\mathbf{y}) \propto NCC(f(\mathbf{u}), T(\mathbf{u}, y)). \quad (7)$$

The function $NCC(\cdot, \cdot)$ computes the normalized correlation coefficient, $T(\mathbf{u}, y)$ represents the template pixel value when the state is y at pixel location \mathbf{u} and $f(\mathbf{u})$ represents the image intensity. The template $T(\mathbf{u}, y)$ is a small neighborhood around each marker location. In our implementation, we set this neighborhood value to be 1/10 of the value of the height from the base of spine (joint 12) to the neck (joint 3) specified in Figure 1. The ratio 1/10 is chosen empirically and it gives a rough estimate between the ration of

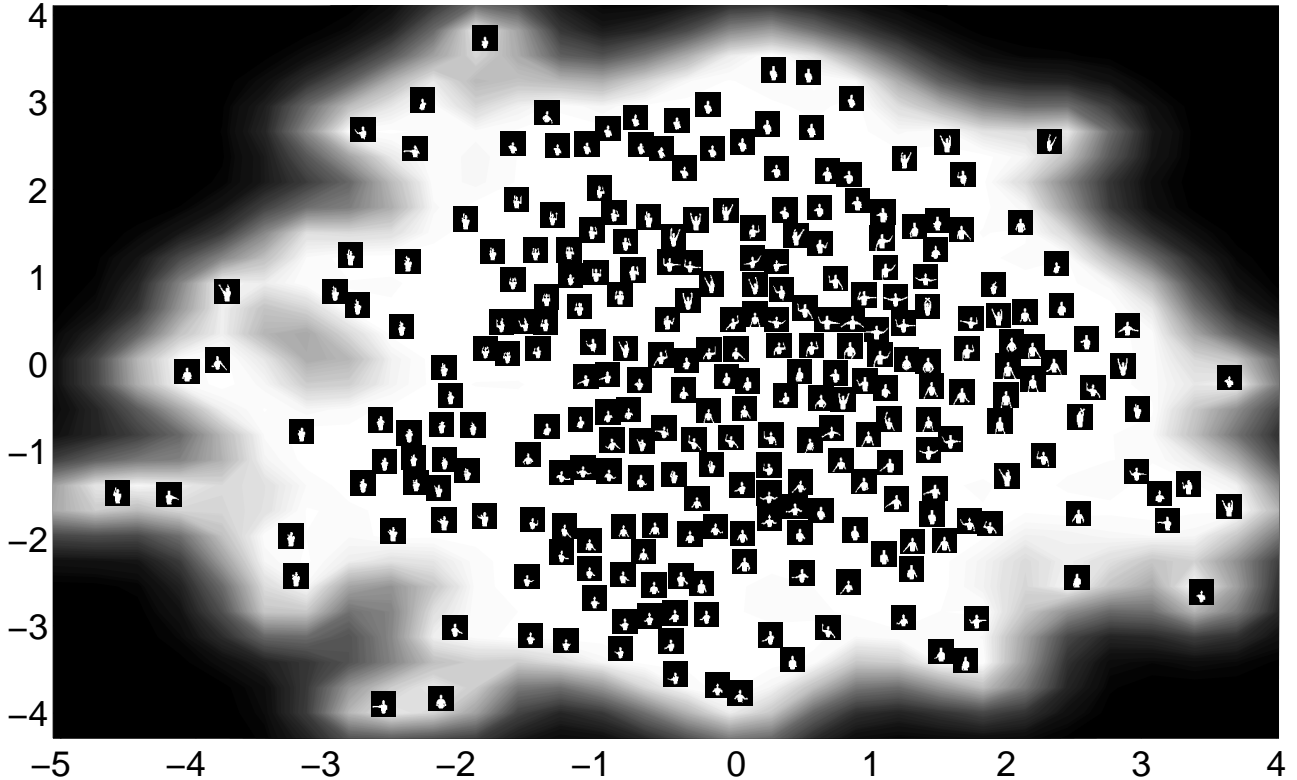


Figure 2: *Visualizing the latent space.*

the length of the spine and the thickness of the arm. The value u ranges over all image pixels that overlap with the templates.

The state dynamics is modeled as a random walk (see Eq. 6) in 2D space. The parameter ν^2 can be learned from training sequences. We project training sequences onto the latent space and compute the variance of all the distances between pose transitions. Currently, for our data set, ν_1 is found to be 0.01 and ν_2 is 0.02. Note that the dynamics is described in terms of the low-dimensional representation, hence the training poses have to be projected down onto the low-dimensional space. The projection of a point \mathbf{y} is given by μ^{-1} (Eq. 5).

We found that 200 particles were sufficient to obtain good performance in the modified Condensation algorithm. The Processing time for each frame is 0.3 sec on a 1.7GHz Intel processor running un-optimized Matlab code.

5. Experiments

The modified Condensation algorithm is used to track upper body gestures. We present the tracking results from two sequences that are typical of our tracker (they are available in the submitted avi file `res_2seqs.avi`.) The first se-

quence contains a female subject and is 500 frames long, the second sequence contains a male subject and it is 250 frames long. For all the experiments described here, the body pose is manually initialized in the first frame. Auto-initialization is currently not handled and left as a topic for future work.

5.1 Singularity and Self Occlusion

The strength of the algorithm is demonstrated in Figure 3 and Figure 4, where the case singularity [18] and self occlusion occurred. With the strong prior, our tracker is able to handle the singularity and self occlusion gracefully. In the case of singularity, the joints of subject's forearm, upper arm and hand become almost one single point, our algorithm is still able to track the arm. In Figure 4, when the subject's hand moved across the face causing self occlusion, though there is a bit confusion due to the similar appearance of the face and the hand, our tracker is able to maintain the track once the hand is moving away from the face.

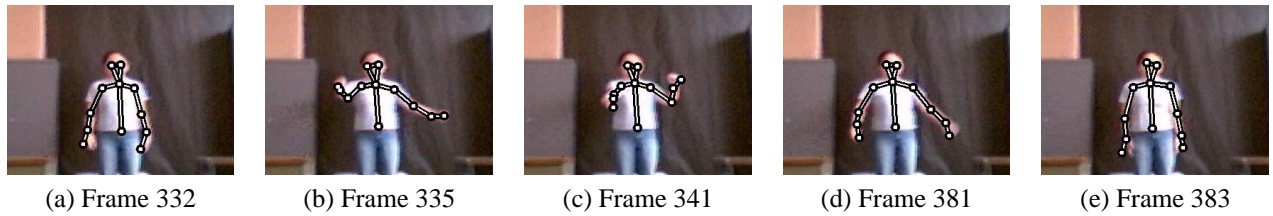


Figure 3: *Tracking through a singularity. The right hand is lifted up and a singularity is formed in (b). The singularity persists through frames 336 to 370 (c). The tracker manages to recover in the later frames as shown in (d) and (e).*

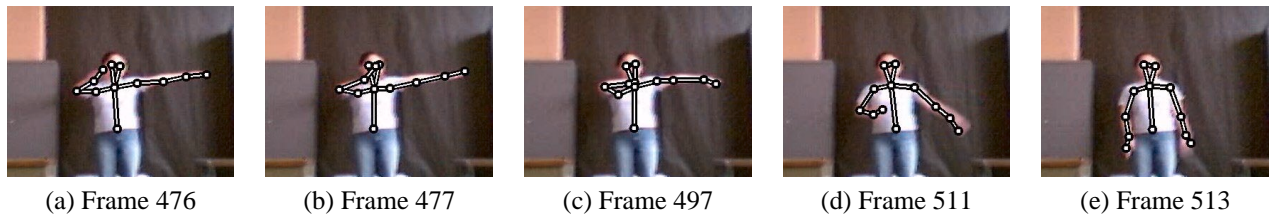


Figure 4: *Tracking through self-occlusion. The right hand is raised and begins to occlude the face in (a). The right hand crosses the face in (b) and (c) and the tracker still maintains tracking. The tracker is a bit distracted by the same appearance patch of the face and but manages to maintain tracking in frame (d) and (e).*



Figure 5: *Comparison of Tracking Performance. (a) Particle Filtering, (b) Multiple Hypothesis Tracking, (c) Our modified Particle Filtering*

5.2 Comparison with Multiple Hypothesis Tracker and Condensation

We implemented the Multiple Hypothesis Tracker [5] (MHT) and the original Condensation tracker [9]. Both algorithms are unable to track upper body poses in our test sequences beyond the first few frames. To improve the performance of both algorithms, we used a simpler kinematic chain model anchored at the shoulders. Hence the number of states is greatly reduced to 6 when compared to the full upper body model. A second order auto-regressive (AR) model is learned from the first 30 frames for the prediction step in both algorithms. For the standard Condensation, 1000 particles are used. During the tracking using Condensation, we predict 1000 states (particles) for the next state using the second order auto-regressive model we have learned, then we match the appearance template for each of these predicted states and set its weights according to the matching cost. The processing time per frame is about 7 seconds. With the MHT, we use 10 modes. A general Damped Gauss-Newton Method with Armijo rule is used to search the local minima of the matching cost. The parameters of the hypotheses are updated accordingly and the weights of the hypotheses are computed in the same way as Condensation. The processing time per frame for MHT is about 50 seconds due to the optimization process. Both algorithms are implemented in Matlab and are not optimized. Both trackers failed to track successfully beyond the first few frames even when a second order AR model is learned from the test sequences. In comparison, our tracker uses the simpler random walk model learned on the latent space and is able to track successfully through the frames. The comparison results are shown in Figure 5.

6. Conclusions and Future Work

We have propose a framework for tracking 2D body poses that is able to robustly track a long sequences of video. Our experiments show that our tracker performs significantly better than Multiple Hypothesis tracking and the original Condensation algorithm. It is also shown that our tracker is capable of handling singularities and self occlusions. As the performance of the tracker depends on the latent space learnt, it will fail when the poses to track deviates significantly from the training data. The tracker can only track poses that are similar to those in the training data.

Currently, the tracker is being initialized manually, we can automatically initialize the tracker by searching the image for a face using a face detector, similar to the work of Taycher and Darrell [28]. Secondly, there is also jittering in the poses tracked. This can be dealt with using smoothing filters for example [11]. Such filters condition the state distribution on both the past and future measurements thus reducing the variance of the estimates.

We will also consider extending this work to recognizing a larger set of gestures involving the full body and hands.

A. Appendix

The following gradients are used in the optimizing L :

$$\begin{aligned}\frac{\partial L}{\partial \mathbf{K}} &= -\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^T\mathbf{K}^{-1} + \frac{1}{2}M\mathbf{K}^{-1}, \\ \frac{\partial k(\mathbf{x}, \mathbf{x}')}{\partial \mathbf{x}} &= -\gamma(\mathbf{x} - \mathbf{x}')k(\mathbf{x}, \mathbf{x}'), \\ \frac{\partial k(\mathbf{x}, \mathbf{x}')}{\partial \alpha} &= \exp\left(-\frac{\gamma}{2}(\mathbf{x} - \mathbf{x}')^T(\mathbf{x} - \mathbf{x}')\right), \\ \frac{\partial k(\mathbf{x}, \mathbf{x}')}{\partial \gamma} &= -\frac{1}{2}(\mathbf{x} - \mathbf{x}')^T(\mathbf{x} - \mathbf{x}')k(\mathbf{x}, \mathbf{x}').\end{aligned}$$

The following gradients are used in optimizing L' :

$$\begin{aligned}\frac{\partial L'}{\partial \mathbf{y}} &= -\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^T\mathbf{K}^{-1} + \frac{1}{2}M\mathbf{K}^{-1}, \\ \frac{\partial \mu(\mathbf{x})}{\partial \mathbf{x}} &= \mathbf{Y}_{I,I}\mathbf{K}_{I,I} \frac{\partial k_I(\mathbf{x})}{\partial \mathbf{x}}, \\ \frac{\partial \sigma^2(\mathbf{x})}{\partial \alpha} &= -2k_I(\mathbf{x})^T\mathbf{K}_{I,I}^{-1} \frac{\partial k_I(\mathbf{x})}{\partial \mathbf{x}}.\end{aligned}$$

References

- [1] <http://www.dcs.shef.ac.uk/~neil/gplvm/>.
- [2] *NAVAIR 00-80T-113 Aircraft Signals NATOPS Manual*.
- [3] M. Belkin and P. Niyogi. Laplacian Eigenmaps and spectral techniques for embedding and clustering. In *NIPS*, 2002.
- [4] Matthew Brand. Charting a manifold. In *NIPS*, 2002.
- [5] Tat-Jen Cham and James M. Rehg. A multiple hypothesis approach to figure tracking. In *CVPR*, 1999.
- [6] J. Deutscher, A. Blake, and I. Reid. Articulated body motion capture by annealed particle filtering. In *CVPR*, 2000.
- [7] Ahmed Elgammal and Chan Su Lee. Inferring 3D body pose from silhouettes using activity manifold learning. In *CVPR*, 2004.
- [8] K. Grochow, S.L. Martin, A. Hertzmann, and Z. Popovic. Style-Based Inverse Kinematics. In *ACM Computer Graphics (SIGGRAPH)*, 2004.
- [9] M. Isard and A. Blake. CONDENSATION – conditional density propagation for visual tracking. *IJCV*, 29(1):5–28, 1998.

- [10] Michael Isard and Andrew Blake. ICondensation : Unifying low-level and high-level tracking in a stochastic framework. In *ECCV*, 1998.
- [11] Michael Isard and Andrew Blake. A smoothing filter for Condensation. In *ECCV*, 1998.
- [12] O. King and D. A. Forsyth. How does CONDENSATION behave with a finite number of samples? In *ECCV*, 2000.
- [13] N. Lawrence. Gaussian Process Latent Variable Models for Visualisation of High dimensional Data. In *NIPS*, 2004.
- [14] N. Lawrence. Probabilistic non-linear principal component analysis with Gaussian process latent variable models. Technical Report CS-04-8, Department of Computer Science, Univeristy of Sheffield, 2004.
- [15] N. Lawrence, M. Seeger, and R. Herbrich. Fast sparse Gaussian process methods: The informative vector machine. In *NIPS*, 2003.
- [16] J. MacCormick and A. Blake. A probabilistic exclusion principle for tracking multiple objects. In *Proc. ICCV*, 1999.
- [17] D. Mackay. Introduction to Gaussian processes. In C. Bishop, editor, *Neural Networks and Machine Learning*, NATO ASI Series, pages 133–166. Kluwer Academic Press, 1998.
- [18] Daniel D. Morris and Jim Rehg. Singularity analysis for articulated object tracking. 1998.
- [19] R. Rosales and S. Sclaroff. Specialized mappings and the estimation of human body pose from a single image. In *IEEE Workshop on Human Motion*, 2000.
- [20] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290, 2000.
- [21] Alla Safonova, Jessica K. Hodgins, and Nancy S. Pollard. Synthesizing physically realistic human motion in low dimensional, behavior-specific spaces. In *SIGGRAPH*, 2004.
- [22] Hedvig Sidenbladh and Michael J. Black. Learning image statistics for bayesian tracking. In *ICCV*, 2001.
- [23] Hedvig Sidenbladh, Michael J. Black, and David J. Fleet. Stochastic tracking of 3D human figures using 2d image motion. In *ECCV*, 2000.
- [24] C. Sminchisescu and A. Jepson. Generative modeling for continuous non-linearly embedded visual inference. In *Proc. ICML*, 2004.
- [25] C. Sminchisescu and B. Triggs. Covariance scaled sampling for monocular 3D body tracking. In *CVPR*, 2001.
- [26] Josephine Sullivan, Andrew Blake, Michael Isard, and John MacCormick. Bayesian object localisation in images. *IJCV*, 44(2):111–135, 2001.
- [27] Josephine Sullivan and Jens Rittscher. Guiding random particles by deterministic search. 2001.
- [28] L. Taycher and T. Darrell. Bayesian articulated tracking using single frame pose sampling constraints. In *3rd Int'l Workshop on Statistical and Computational Theories of Vision*, 2003.
- [29] J. Tenenbaum, V. Silva, and J. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290, 2000.
- [30] K. Toyama and A. Blake. Probabilistic tracking in a metric space. In *Proc. ICCV*, 2001.
- [31] L. Wang, W.M. Hu, and T.N. Tan. Recent developments in human motion analysis. *PR*, 36(3):585–601, March 2003.
- [32] C. Williams. Prediction with Gaussian processes: From linear regression to linear prediction and beyond. In M. Jordan, editor, *Learning Graphical Models*, volume 89 of *Series D: Behavioural and Social Sciences*. Kluwer, 1998.