3D HUMAN TRACKING WITH GAUSSIAN PROCESS ANNEALED PARTICLE FILTER

Leonid Raskin, Ehud Rivlin and Michael Rudzsky
Computer Science Department,Technion Israel Institute of Technology, Technion City, Haifa, Israel

Keywords: Tracking, Annealed particle filter, Gaussian fields, Latent space.

Abstract: We present an approach for tracking human body parts with prelearned motion models in 3D using multiple cameras. We use an annealed particle filter to track the body parts and a Gaussian Process Dynamical Model in order to reduce the dimensionality of the problem, increase the tracker's stability and learn the motion models. We also present an improvement for the weighting function that helps to its use in occluded scenes. We compare our results to the results achieved by a regular annealed particle filter based tracker and show that our algorithm can track well even for low frame rate sequences.

1 INTRODUCTION

This paper presents an approach to 3D people tracking that enables reduction in the complexity of this model. We propose a novel algorithm, Gaussian Process Annealed Particle Filter (GPAPF). In this algorithm we use nonlinear dimensionality reduction with the help a Gaussian Process Dynamical Model (GPDM), (Lawrence (2004), Wang et al. (2005)), and an annealed particle filter proposed by Deutscher and Reid (2004). The annealed particle filter has good performance when working on videos which were shot with a high frame rate (60 fps, as reported by Balan et al. (2005)), but performance drops when the frame rate is lower (30fps). We show that our approach provides good results even for the low frame rate (30fps). An additional advantage of our tracking algorithm is the capability to recover after temporal loss of the target.

2 RELATED WORKS

One of the common approaches for tracking is using a Particle Filtering. Particle Filtering uses multiple predictions, obtained by drawing samples of the pose and location prior and then propagating them using the dynamic model, which are refined by comparing them with the local image data (the likelihood) (see, for example Isard (1998) or Bregler et al. (1998)). The prior is typically quite diffused (because motion can be fast) but the likelihood function may be very peaky, containing multiple local maxima which are hard to account for in detail. For example, if an arm swings past an “arm-like” pole, the correct local maximum must be found to prevent the track from drifting (Sidenbladh (2000)). Annealed particle filter (Deutscher and Reid (2004)) or local searches are ways to attack this difficulty.

There exist several possible strategies for reducing the dimensionality of the configuration space. Firstly it is possible to restrict the range of movement of the subject. This approach has been pursued by Rohr et al. (1997). The assumption is that the subject is performing a specific action. Agarwal et al. (2004) assume a constant angle of view of the subject. Because of the restricting assumptions the resulting trackers are not capable of tracking general human poses.

Another way to cope with high-dimensional data is to learn low-dimensional latent variable models. Urtasun et al. (2006) uses a form of probabilistic dimensionality reduction by Gaussian Process Dynamical Model (GPDM) (Lawrence (2004), and Wang et al. (2005)) formulate the tracking as a nonlinear least-squares optimization problem.

Our approach is similar in spirit to the one proposed by Urtasun et al. (2006), but we perform a two stage process. The first stage is annealed particle filtering in a latent space of low dimension. The particles obtained after this step are transformed into the data space by GPDM mapping. The second stage is annealed particle filtering with these particles in the data space.

Raskin L., Rivlin E. and Rudzsky M. (2007).
3D HUMAN TRACKING WITH GAUSSIAN PROCESS ANNEALED PARTICLE FILTER.
In *Proceedings of the Second International Conference on Computer Vision Theory and Applications - IU/MTSV*, pages 459-465

The article is organized as follows. In Section 3 and Section 4 we give short descriptions of particle filtering and Gaussian fields. In Section 5, we describe our algorithm. Section 6 contains our results. The conclusions are in Section 7.

3 FILTERING

The particle filter algorithm was developed for tracking objects, using the Bayesian inference framework. Let us denote x_n as a hidden state vector and y_n as a measurement in time n. The algorithm builds an approximation of maximum a posterior estimate of the filtering distribution: $p(x_n|y_{1:n})$ where $y_{1:n} \triangleq (y_1,...,y_n)$ is the history of the observation. This distribution is represented by a set of pairs $\{x_n^{(i)},\pi_n^{(i)}\}_{i=1}^{N_p}$, where $\pi_n^{(i)} \propto p(y_n|x_n^{(i)})$. The main problem is that the distribution $p(y_n|x_n)$ may be very picky. Often a weighting function $w(y_n,x)$ can be constructed in a way that it provides a good approximation of $p(y_n|x_n)$, and is also easy to calculate. The main idea in the annealed particle filter is to use a set of weighting functions instead of using a single one. A series of $\{w_n(y_n,x)\}_{n=0}^{M}$ is used, where $w_{n+1}(y_n,x)$ represents a smoothed version of $w_n(y_n,x)$. The usual method to achieve this is by using $w_n(y_n,x)=w_0(y_n,x)^{\beta_n}$, where $w_0(y_n,x)$ is equal to the original weighting function and $1=\beta_0>...>\beta_M$. Therefore, each iteration of the annealed particle filter algorithm consists of M steps, in each of these the appropriate weighting function is used and a set of pairs is constructed $\{x_{n,m}^{(i)},\pi_{n,m}^{(i)}\}_{i=1}^{N_p}$. For details see Deutscher et al. (2004) and Raskin et. al (2007).

4 GAUSSIAN FIELDS

The Gaussian Process Dynamical Model (GPDM) represents a mapping from the latent space to the data: $y=f(x)$, where $\mathbf{x} \in \mathbb{R}^d$ denotes a vector in a d-dimensional latent space and $\mathbf{y} \in \mathbb{R}^D$ is a vector that represents the corresponding data in a D-dimensional space. Gaussian processes stem from Bayesian formulation, in which the GPDM is obtained by marginalizing out the parameters and optimizing the latent coordinates x of the trained data y. The model that is used to derive the GPDM is a mapping with first-order Markov Dynamics:

$$x_t = \sum_i a_i \varphi_i(x_{t-1}) + n_{x,t} \qquad (1)$$

$$y_t = \sum_j b_j \psi_j(x_t) + n_{y,t}$$

where $n_{x,t}$ and $n_{y,t}$ are zero-mean Gaussian noise processes, $A=[a_1,a_2,...]$ and $B=[b_1,b_2,...]$ are weights and φ_i and ψ_j are basis functions.

For the Bayesian perspective A and B should be marginalized out through model average with an isotropic Gaussian prior on B in closed form to yield:

$$p(Y|X,\bar{\beta}) = \frac{|W|^N}{\sqrt{(2\pi)^{ND}|K_y|^D}} \exp\left(-\tfrac{1}{2}\mathrm{tr}\left(K_y^{-1}YW^2Y^T\right)\right) \qquad (2)$$

where Y is a matrix of training vectors, X contains corresponding latent vectors and K_y is the kernel matrix:

$$(K_y)_{i,j} = \beta_1 \exp\left(-\frac{\beta_2}{2}\|x_i - x_j\|^2\right) + \frac{\delta_{x_i,x_j}}{\beta_3} \qquad (3)$$

W is a scaling diagonal matrix. It is used to account for the different variances in different data elements. The hyper parameter β_1 represents the scale of the output function, β_2 represents the inverse of the RBF's and β_3^{-1} represents the variance of $n_{y,t}$.

For the dynamic mapping of the latent coordinates X the joint probability density over the latent coordinate system and the dynamics weights A are formed with an isotropic Gaussian prior over the A, it can be shown (see Wang et al. (2005)) that

$$p(X|\bar{\alpha})=\frac{p(x_1)}{\sqrt{(2\pi)^{(N-1)d}|K_x|^d}}\exp\left(-\frac{1}{2}tr\left(K_x^{-1}X_{out}X_{out}^T\right)\right) \quad (4)$$

where $X_{out}=[x_2,...,x_N]^T$, K_x is a kernel constructed from $[x_1,...,x_{N-1}]^T$ and x_1 has an isotropic Gaussian prior. GPDM uses a "linear+RBF" kernel with parameter α_i:

$$(K_x)_{i,j}=\alpha_1\exp(-\frac{\alpha_2}{2}\|x_i-x_j\|^2)+\alpha_2 x_i^T x_j+\frac{\delta_{x_i,x_j}}{\alpha_4} \quad (5)$$

Following Wang et al. (2005)

$$p(X,\bar{\alpha},\bar{\beta}|Y)\propto p(Y|X,\bar{\beta})\,p(X|\bar{\alpha})\,p(\bar{\alpha})\,p(\bar{\beta}) \quad (6)$$

The latent positions and hyper parameters are found by maximizing this distribution or minimizing the negative log posterior:

$$\begin{aligned}\Lambda=&\frac{d}{2}\ln|K_x|+\frac{1}{2}tr\left(K_X^{-1}X_{out}X_{out}^T\right)+\sum_i\ln\alpha_i\\&-N\ln|W|+\frac{D}{2}\ln|K_y|+\frac{1}{2}tr\left(K_Y^{-1}YW^2Y^T\right)+\sum_j\ln\beta_j\end{aligned} \quad (7)$$

5 GPAPF FILTERING

5.1 The Model

In our work we use a model similar to the one proposed by Deutscher et al (2004) with some differences in the annealed schedule and weighting function. The body model is defined by a pair $M=\{L,\Gamma\}$, where L are the limbs lengths and Γ are the angles between limbs and the global location of the body in 3D. The limbs parameters are constant, and represent the actual size of the tracked person. The angles represent the body pose and, therefore, are dynamic. The state is a vector of dimensionality 29: 3 DoF for the global 3D location, 3 DoF for the global rotation, 4 DoF for each leg, 4 DoF for the torso, 4 DoF for each arm and 3 DoF for the head. The whole tracking process estimates the angles in such a way that the resulting body pose will match the actual pose. This is done by minimizing the weighting function which is explained next.

5.2 The Weighting Function

In order to evaluate how well the body pose matches the actual pose using the particle filter tracker we have to define a weighting function $w(\Gamma,Z)$, where Γ is the model's configuration (i.e. angles) and Z stands for visual content (the captured images). Our function is based on a function suggested by Deutscher et al (2004) with some changes made to it. We have experimented with 3 different features: edges, silhouette and foreground histogram. The first feature is the edges. As Deutscher proposes this feature is the most important one, and provides a good outline for visible parts, such as arms and legs. The other important property of this feature is that it is invariant to the colour and lighting condition. The edges maps, in which each pixel is assigned a value dependent on its proximity to an edge, are calculated for each image plane. Each part is projected on the image plane and samples of the N_e hypothesized edge are drawn. A squared probability function is calculated for these samples:

$$P^e(\Gamma,Z)=\frac{1}{N_e}\frac{1}{N_{cv}}\sum_{i=1}^{N_{cv}}\sum_{j=1}^{N_e}\left(1-p_j^e(\Gamma,Z_i)\right)^2 \quad (8)$$

where N_{cv} is a number of camera views, Z_i is the i-th image . The $p_j^e(\Gamma,Z_i)$ are the edge maps. However, the problem that occurs using this feature is that the occluded body parts will produce no edges. Even the visible parts, such as the arms, may not produce the edges, because of the colour similarity between the part and the body. This will cause $p_j^e(\Gamma,Z_i)$ to be close to zero and thus will increase the weighting function. Therefore, a good pose which may match the visual context may results in a high value of weighting function and may be omitted. In order to overcome this problem we calculate a weight for each image plane. For each sample point on the edge we estimate the probability of the point being covered by another body part. Let N_i be the number of hypothesized edges that are drawn for the part i. The total number of drawn sample points can be calculated using $N_e=\sum_{i=1}^{N_{bp}}N_i$. The weight of the part for the j-th image plane can be calculated as following:

$$w(\Gamma_i,Z_j)=\sum_{k=1}^{N_i}p_k^{fg}(\Gamma_i,Z_j)\Big/\sum_{j=1}^{N_{cv}}\sum_{k=1}^{N_i}p_k^{fg}(\Gamma_i,Z_j) \quad (9)$$

where Γ_i is the model configuration for part i, Z_j is the j-th image plane and $p_k^{fg}\left(\Gamma_i, Z_j\right)$ is the probability that the k-th sample is covered by another body part. The weighting function therefore has the following form:

$$\tilde{P}\left(\Gamma_{bp}, Z_{cv}\right) = \sum_{k=1}^{N_i}\left(1-p_k^e\left(\Gamma_{bp}, Z_{cv}\right)\right)^2$$
$$P^e\left(\Gamma, Z\right) = 1/N_e \; 1/N_{cv} \sum_{i=1}^{N_{cv}} \sum_{j=1}^{N_{bp}} w(\Gamma_j, Z_i)\tilde{P}\left(\Gamma_j, Z_i\right) \quad (10)$$

The second feature is the silhouette obtained by subtracting the background from the image. The foreground pixel map is calculated for each image plane with background pixels set to 0 and foreground set to 1 and SSD is computed:

$$P^{fg}\left(\Gamma, Z\right) = 1/N_e \; 1/N_{cv} \sum_{i=1}^{N_{cv}} \sum_{j=1}^{N_e}\left(1-p_j^{fg}\left(\Gamma, Z_i\right)\right)^2 \quad (11)$$

where N_{bp} is the number of different body parts in the model, $p_j^{fg}\left(\Gamma, Z_i\right)$ is the value is the foreground pixel map values at the sample points. The third feature is the foreground histogram. The reference histogram is calculated for each body part. It can be a grey level histogram or three separated histograms for colour images. Then, on each frame a normalized histogram is calculated for a hypothesized body part location and is compared to the referenced one and the Bhattacharya distance is computed:

$$P_k\left(\Gamma_{bp}, Z_{cv}\right) = \sqrt{p_k^{ref}\left(\Gamma_{bp}, Z_{cv}\right) p_k^{hyp}\left(\Gamma_{bp}, Z_{cv}\right)}$$
$$P^h\left(\Gamma, Z\right) = \sum_{i=1}^{N_{cv}} \sum_{j=1}^{N_{bp}}\left(1-\sum_{k=1}^{bins} P_k\left(\Gamma_{bp}, Z_{cv}\right)\right) \quad (12)$$

where $p_j^{orig}\left(\Gamma, Z_i\right)$ is the value of bin j in the reference histogram, and the $p_j^{hyp}\left(\Gamma, Z_i\right)$ is the value of the same bin on the current frame using the hypothesized body part location. The main drawback of that feature is that it is sensitive to changes in the light condition and the texture of the tracked object. Therefore, the reference histogram has to be updated, using the weighted average from the recent history.

In order to calculate the weighting function the features are combined together using the following formula:

$$w(\Gamma, Z) = \exp(-(P^e(\Gamma, Z) + P^{fg}(\Gamma, Z) + P^h(\Gamma, Z))) \quad (13)$$

As was stated above, the purpose of the tracker is to minimize the weighting function.

5.3 GPAPF Learning

The drawback in the particle filter tracker is that a high dimensionality of the state space causes an exponential increase in the number of particles that are needed to be generated in order to preserve the same density of particles. In our case, the dimension of the data is 29. In their work Balan et al. (2005) show that the annealed particle filter can track body parts with ~125 particles using 60 fps video input. However, using a significantly lower frame rate (15 fps) causes the tracker to produce bad results and eventually to lose the target.

The other problem is that once a target is lost (i.e. the body pose was wrongly estimated, which can happen for the fast and not smooth movements) it becomes highly unlikely that the next pose will be estimated correctly in the following frames. In order to reduce the dimension of the space we have used Gaussian Process Annealed Particle Filter (GPAPF). We use a set of poses in order to create a latent space with a low dimensionality. The poses are taken from different sequences, such as walking, running, punching and kicking. We divide our state into two independent parts. The first part contains the global 3D body rotation and translation, which is independent of the actual pose. The second part contains only information regarding the pose (26 DoF). We use GPDM to reduce the dimensionality of the second part. This way we construct a latent space (Fig. 1). This space has a significantly lower dimensionality (for example 2 or 3 DoF). Unlike Urtasun et al. (2006), whose latent state variables include translation and rotation information, our latent space includes solely pose information and is therefore rotation and translation invariant. The particles are drawn in the latent space. In order to calculate the weighting function we transform the data from the latent space to the data space and then calculate the weighting function as explained above. However, the latent space is not capable of producing all the poses; therefore we apply an additional iteration of the annealed tracker only for the data space in order to make fine adjustments. This final iteration is performed using a low covariance matrix, which is nearly invariant for all frames rates.

The main difficulty in this approach is that the latent space is not uniformly distributed. Therefore we are using a dynamic model, as proposed by Wang et al. (2005), in order to achieve smoothed transitions between sequential poses in the latent space. However, as it is shown in Fig. 1, there are still some irregularities and discontinuities. Moreover, while in a regular space the change in the angles is independent on the actual angle value, in a latent space this is not the case. Each pose has a certain

probability to occur and thus the probability to be drawn as a hypothesis should be dependent on it. For each particle we can calculate an estimate of the variance that can be used for generation the new ones. In the left part of Fig. 1 the lighter pixels represent lower variance.

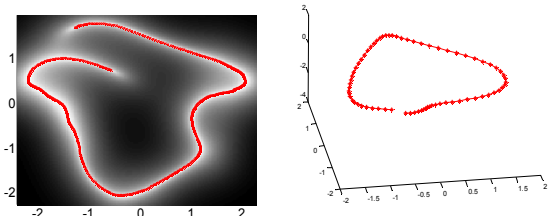
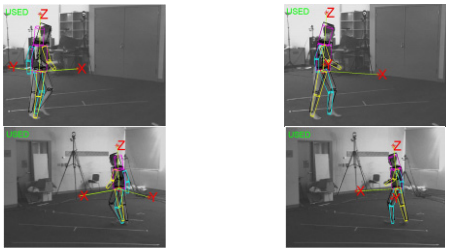


Figure 1: The latent space that is learned from different poses during the walking sequence. Left: the 2D space. Right: the 3D space. On the left image: the brighter pixels correspond to more precise mapping.



Frame 137 **Frame 141**

Figure 2: Losing and finding the tracked target despite the mis-tracking on the previous frame. Top: camera 1, Bottom: camera 4.

Another advantage of this method is that the tracker is capable of recovering after several frames, from poor estimations. The reason for this is that particles generated in the latent space are representing valid poses more authentically. Furthermore because of its low dimensionality the latent space can be covered with a relatively small number of particles. Therefore, most of possible poses will be tested with emphasis on the pose that is close to the one that was retrieved in the previous frame. So if the pose was estimated correctly the tracker will be able to choose the most suitable one from the tested poses. However, if the pose on the previous frame was miscalculated the tracker will still consider the poses that are quite different. As these poses are expected to get higher value of the weighting function the next layers of the annealed will generate many particles using these different poses. In this way the pose is likely to be estimated correctly, despite the mis-tracking on the previous frame (see Fig. 2).
An additional advantage of our approach is that the generated poses are, in most cases, natural. Poses generated by the Condensation algorithm or by annealed particle filtering, where the large variance in the data space, cause a generation of unnatural poses. The poses that are produced by the latent space that correspond to points with low variance are usually natural and therefore the effective number of the particles is higher, which enables more accurate tracking. The drawback of this approach is that it requires more calculation than the regular annealed particle filter. The additional calculations are the result of transformation from the latent space into the data space.

6 RESULTS

We have tested out algorithm on the sequences provided by L.Sigal, which are available at his site: http://www.cs.brown.edu/~ls/Software/index.html. The sequences contain different activities, such as walking, boxing etc. which were captured by 7 cameras, however we have used only 4 inputs in our evaluation. The sequences were captured using the MoCap system, that provides the correct 3D locations of the body parts for evaluation of the results and comparison to other tracking algorithms.

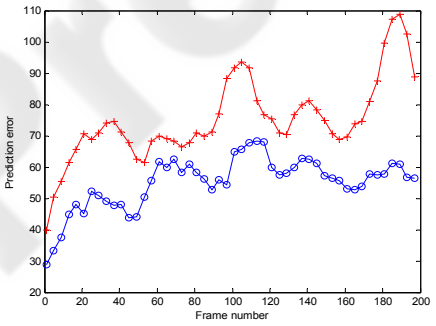
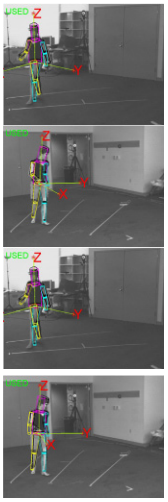


Figure 3: The errors of the annealed tracker (red crosses) and GPAPF tracker (blue circles) for a walking sequence captured at 30 fps.

The first sequence that we have used was a walk on a circle. The video was captured at frame rate 120 fps. We have tested the annealed particle filter based body tracker, implemented by A. Balan (Balan et al. (2005)) , and compared the results with the ones produced by the GPAPF tracker (see Fig. 3 and 4). The error was calculated, based on comparison of the tracker's output and the result of the MoCap system. In Fig. 3 we can see the error graphs, produced by GPAPF tracker (blue circles) and by the annealed particle filter (red crosses) for the walking sequence taken at 30 fps. As can be seen, the GPAPF tracker produces more accurate estimation of the body location. Same results were achieved for 15 fps. In Fig. 4 one can see the actual



Frame 73

son between annea
APF tracker, first
camera. Forth row

s sequence. The
d second camera
of the GPAPF tra
esults of the ann
ence was captur
ave filmed simil
actor. The fram
done on the fi
acker was able
results similar
or the original se

ON AND FU

approach that us
ensionality and i
f the annealed
ect even in a high
shown that using
recover from te
so presented a
ility of self occl
o adjust the weig
r to be able to p
a pose.
is that the traini
ion. The ability
an activity that
learned, has not
ging task is to
ously. The main

Ramanan, D., and Forsyth, D. A., 2003 Automatic Annotation of Everyday Movements. *Neural Info. Proc. Systems* (NIPS), Vancouver, Canada.

Sigal, L., Bhatia, S., Roth, S., Black M. and Isard, M. 2004 Tracking loose-limbed people. In *CVPR*, vol. 1, pp. 421–428.

Sidenbladh, H., Black, M. and Fleet, D. 2000 Stochastic tracking of 3d human figures using 2d image motion. In *Proc. ECCV*.

Song, Y., Feng, X. and Perona P. 2000. Towards detection of human motion. In *Proc. CVPR*, pp 810–17.

Urtasun, R., Fleet, D. .J., Hertzmann, and A., Fua, P. 2005 Priors for people tracking from small training sets. In *Proc. ICCV*, Beijing v1, pp. 403-410.

Urtasun, R., Fleet, D. J., and Fua, P. 2006. 3D People Tracking with Gaussian Process Dynamical Models. .In *Proc. CVPR'06* , v.1, pp. 238-245.

Wang, J.,Fleet, D. J., and Hetzmann, A. 2005. Gaussian process dynamical models.In Proc.NIPS. Vancouver, Canada: pp. 1441-1448.

Raskin, L. Rivlin, E., and Rudzsky M. 2007. 3D Human Tracking with Gaussian Process Annealed Particle Filter. Technical.Report. CS-2007-01