GPAPF: A Combined Approach for 3D Body Part Tracking

Leonid Raskin, Michael Rudzsky, Ehud Rivlin

Technion - IIT, Computer Science, Kyrjat Technion,
32000 Haifa, Israel
{raskinl,rudzsky,ehudr}@cs.technion.ac.il

Abstract. In this paper we present a combined approach for body part tracking in 3D using multiple cameras, called GPAPF. This approach combines annealed particle filtering, which has been shown as effective tracker for body parts, with Gaussian Process Dynamical Model, which is used in order to reduce the dimensionality of the problem. That reduction improves the tracker's performance and increases the tracker's stability and ability to recover from the loosing the target. We also compare GPAPF tracker with the annealed particle filter and show that our tracker has a better performance even for low frame rate sequences.

Key words: Tracking, Annealing particle filter, GPDM, Latent space

1 Introduction

Human body pose estimation and tracking is a challenging task for several reasons. First, the large dimensionality of the human 3D model complicates the examination of the entire subject and makes it harder to detect each body part separately. Secondly, the significantly different appearance of different people that stems from various clothing styles and illumination variations, adds to the already great variety of person images. 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), [8,15], and an annealed particle filter proposed by Deutscher and Reid [4].
The annealed particle filter has a good performance when applied on videos with a high frame rate (60 fps, as reported by Balan et al. [6]), 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.

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

2 Related Works

There are two main approaches for body pose estimation. The first one is based one a single frame [3, 5]. The second approach is to approximate body pose based on a sequence of frames. A variety of methods, have been developed for tracking people from single views [11], as well as multiple views [4].

One of the common approaches for tracking is using a Particle Filtering. Particle Filtering uses multiple predictions, obtained by drawing samples based on location prior and then propagating them using the dynamic model, which is refined by comparing them with the image data [6, 3]. The prior is typically quite diffused , 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 [13]. Annealed particle filter [4] 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. [12]. The assumption is that the subject is performing a specific action. Agarwal et al. [1] 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. [15] uses a form of probabilistic dimensionality reduction by Gaussian Process Dynamical Model (GPDM) [8,16] formulate the tracking as a nonlinear least-squares optimization problem.

Our approach is similar in spirit to the one proposed by Urtasun et al. [14], 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 to project the particles to the images in order to evaluate how well the particle fits the images.

The article is organized as follows. In Section 3 and Section 4 we give a 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 Annealed Particle Filter

The particle filter algorithm was developed for tracking objects, using the Bayesian inference framework. Let us denote $\mathbf{x}_n$ as the hidden state vector and $\mathbf{y}_n$ be the measurement in time n. The algorithm builds an approximation of maximum a posterior estimate of the filtering distribution: $p(\mathbf{x}_n \mid \mathbf{y}_{1:n})$, where. This distribution is represented by a set of pairs $\left\{\mathbf{x}_n^{(i)}, \pi_n^{(i)}\right\}_{i=1}^{N_p}$, where $\pi_n^{(i)} \propto p\left(\mathbf{y}_n \mid \mathbf{x}_n^{(i)}\right)$. The main

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

problem is that the distribution $p(\mathbf{y}_n \mid \mathbf{x}_n)$ may be very picky. Often a weighting function $w_n^{(i)}(\mathbf{y}_n, \mathbf{x})$ can be constructed in a way that it provides a good approximation of the $p(\mathbf{y}_n \mid \mathbf{x}_n)$, but is also easy to calculate [6]. Therefore, the problem becomes to find configuration $\mathbf{x}_k$ that maximizes the weighted function $w_n^{(i)}(\mathbf{y}_n, \mathbf{x})$.

The main idea in the annealed particle filter is to use a set of weighting functions instead of using a single one. The weight function should be as smoothed as possible. A series of $\{w_n(\mathbf{y}_n, \mathbf{x})\}_{n=0}^{M}$ is used, where $w_{n+1}(\mathbf{y}_n, \mathbf{x})$ represents a smoothed version of $w_n(\mathbf{y}_n, \mathbf{x})$. The usual method to achieve this is by using $w_n(\mathbf{y}_n, \mathbf{x}) = w_0(\mathbf{y}_n, \mathbf{x})^{\beta_n}$, where $\beta_0 > \ldots > \beta_M$ and $w_0(\mathbf{y}_n, \mathbf{x})$ is equal to the original weighting function. Therefore, in 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 $\{\mathbf{x}_{n,m}^{(i)}, \pi_{n,m}^{(i)}\}_{i=1}^{N_p}$.

The annealed particle filter algorithms: for each annealed layer m do:

1. Calculate the weights:

$$\pi_{n,m}^{(i)} = k \frac{w^m(\mathbf{y}_n, \mathbf{x}_n^{(i)}) \, p(\mathbf{x}_n^{(i)} \mid \mathbf{x}_{n-1}^{(i)})}{q(\mathbf{x}_n^{(i)} \mid \mathbf{x}_{n-1}^{(i)}, \mathbf{y}_n)} \qquad (1)$$

$$k = \left(\sum_{i=1}^{N_p} \frac{w^m(\mathbf{y}_n, \mathbf{x}_n^{(i)}) \, p(\mathbf{x}_n^{(i)} \mid \mathbf{x}_{n-1}^{(i)})}{q(\mathbf{x}_n^{(i)} \mid \mathbf{x}_{n-1}^{(i)}, \mathbf{y}_n)} \right)^{-1}$$

2. Draw N particles from the weighted set $\{\mathbf{x}_{n,m}^{(i)}, \pi_{n,m}^{(i)}\}_{i=1}^{N}$ with replacement and with distribution $p(\mathbf{x} = \mathbf{x}_{n,m}^{(i)}) = \pi_{n,m}^{(i)}$.
3. $\mathbf{x}_{n,m-1}^{(i)} = \mathbf{x}_{n,m}^{(i)} + \mathbf{N}_m$ where $\mathbf{N}_m$ is a Gaussian noise $\mathbf{N}_m \sim N(0, P_m)$.

The optimal configuration can be calculated using the following formula:

$$\mathbf{x}_n = \sum_{i=1}^{N} \mathbf{x}_{n,0}^{(i)} \pi_{n,0}^{(i)} \qquad (2)$$

The unweighted set for the next observation is produced using $\mathbf{x}_{n+1,M}^{(i)} = \mathbf{x}_{n,0}^{(i)} + \mathbf{N}_0$.

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

4 Gaussian Fields

The Gaussian Process Dynamical Model (GPDM) represents a mapping from the latent space to the data: $\mathbf{y} = f(\mathbf{x})$, 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. The model that is used to derive the GPDM is a mapping with first-order Markov Dynamics:

$$x_n = \sum_i a_i \phi_i(x_{n-1}) + n_{x,n} \qquad (3)$$
$$y_n = \sum_j b_j \psi_j(x_n) + n_{y,n}$$

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

$$p(Y \mid X, \overline{\beta}) = \frac{|W|^N}{\sqrt{(2\pi)^{ND} |K_y|^D}} e^{-\frac{1}{2} tr\left(K_y^{-1} Y W^2 Y^T\right)} \qquad (4)$$

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

$$(K_y)_{i,j} = \beta_1 \exp\left(-\frac{\beta_2}{2} \|x_i - x_j\|^2\right) + \frac{\delta_{x_i, x_j}}{\beta_3} \qquad (5)$$

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

$$p(X \mid \overline{\alpha}) = \frac{p(x_1)}{\sqrt{(2\pi)^{(N-1)d} |K_x|^d}} e^{-\frac{1}{2} tr\left(K_x^{-1} X_{out} X_{out}^T\right)} \qquad (6)$$

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:

$$\left(K_x\right)_{i,j} = \alpha_1 \exp\left(-\frac{\alpha_2}{2}\left\|x_i - x_j\right\|^2\right) + \alpha_2 x_i^T x_j + \frac{\delta_{x_i,x_j}}{\alpha_4} \tag{7}$$

Following Wang et al. (2005)

$$p\left(X,\bar{\alpha},\bar{\beta}|Y\right) \propto p\left(Y|X,\bar{\beta}\right)p\left(X|\bar{\alpha}\right)p\left(\bar{\alpha}\right)p\left(\bar{\beta}\right) \tag{8}$$

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

$$\begin{aligned}\Lambda = &\frac{d}{2}\ln\left|K_x\right| + \frac{1}{2}tr\left(K_X^{-1}X_{out}X_{out}^T\right) + \sum_i \ln\alpha_i - N\ln\left|W\right| \\ &+ \frac{D}{2}\ln\left|K_y\right| + \frac{1}{2}tr\left(K_Y^{-1}YW^2X^T\right) + \sum_j \ln\beta_j\end{aligned} \tag{9}$$

5 GPAPF Filtering

5.1 The Model

In our work we use a model similar to the one proposed by Deutscher et al [4]. The body model is defined by a pair $\mathrm{M}=\{\mathrm{L},\Gamma\}$, 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 (see [4] for more details).

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 $\mathrm{w}\left(\Gamma,\mathrm{Z}\right)$, where Γ is the model's configuration (i.e. angles) and Z stands for the captures images. Our function is based on a function suggested by Deutscher [4]. We have experimented with 3 different features: edges silhouette and foreground histogram matching.

The first feature is the edges. As Deutscher proposes this feature provides a good outline for visible parts, such as arms and legs. We calculate the edges maps, in which each pixel is assigned a value dependent on its proximity to an edge. Each part is projected on the image plane and samples of the $\mathrm{N_e}$ hypothesized edges are drawn. A squared probability function is calculated for these samples:

$$P^{e}(\Gamma,Z) = \frac{1}{N_e}\frac{1}{N_{cv}}\sum_{i=1}^{N_{cv}}\sum_{j=1}^{N_e}\left(1-p_j^{e}(\Gamma,Z_i)\right)^2 \quad (10)$$

where N_{cv} is a number of cameras views, Z_i is the i-th image . The $p_j^{e}(\Gamma,Z_i)$ are the edges maps.

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:

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

where N_{bp} is the number of different body parts in the model, $p_j^{fg}(\Gamma,Z_i)$ is the value is the foreground pixel map values at the sample points (see Fig. 1).

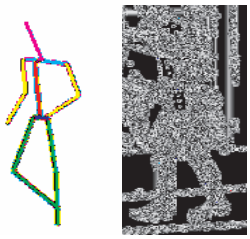


Fig. 1: The 3D body model (left) and the samples drawn for the weighting function calculation (right). On the right image the blue samples are used to evaluate the edge matching, the cyan points are used to calculate the foreground matching, the rectangles with the edges on the red points are used to calculate the part-based body histogram.

The third feature is the part-based histogram. The reference histogram is calculated for each body part. On each frame a histogram is calculated for a hypothesized body location and is compared to the referenced one using the Bhattacharya metrics:

$$P^{h}(\Gamma,Z) = \frac{1}{N_{cv}}\sum_{i=1}^{N_{cv}}\sum_{j=1}^{N_{bp}}\sum_{k=1}^{N_{bins}}\sqrt{p_k^{ref}(\Gamma_j,Z_i)\,p_k^{hyp}(\Gamma_j,Z_i)} \quad (12)$$

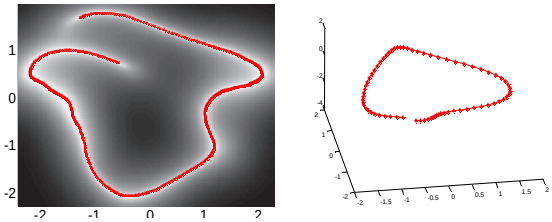
where $p_j^{orig}(X,Z_i)$ is the value of bin j on the reference histogram, and the $p_j^{hyp}(X,Z_i)$ is the value of the same bin on the current frame using hypothesized body part location. The features are combined together using the following formula:

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

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

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. Balan [2] show that the annealed particle filter can track body parts with ~125 particles using 60 fps video input. However, using a lower frame rate (15 fps) causes the tracker to produce bad results and eventually to loose the target. The other problem is that once the body pose was wrongly estimated, it becomes highly unlikely that the pose will be estimated correctly in the following frames.



Fig, 2: 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.

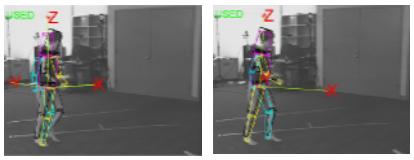
In order to solve these problems we learn the common poses before the tracking. This is done offline. We use a set of poses in order to create a latent space with a low dimensionality. Using that latent space the tracker generates poses that are natural, and therefore the tracking is more successful. We decompose our state to 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 (25 DoF). Then we applying GPDM to reduce the dimensionality of the second part. This way we construct a latent space (Fig. 2), which has a significantly lower dimensionality (usually 2D or 3D DoF). Unlike Urtasun et al. [14,15], who's latent state variables include translation and rotation information, our latent space includes solely pose information and is therefore rotation and translation invariant. This important property can be later used in order to analyze the poses.

The generated particles are drawn in the latent space. After the generation of the particles we apply the mapping function from the latent space to the data space. We add the rotation and translation information and generate regular poses, for which we can apply the weighted function in order to calculate how well the projection of the poses fits the images. The main difficulty in this approach is that the latent space is not uniformly distributed. Meaning that for each vector in the latent space there exists a probability that this vector represents the correct pose. In order to solve that problem we estimate the variance for each particle. This variance is used for generation the new ones (see Fig. 2).

This approach has several advantages. One of them is that the tracker is more robust and is capable of recovering after poor estimations. The reason for this is that particles generated in the latent space are representing valid poses more authentically. Moreover, the latent space has a very low dimensionality, and thus can be covered with relatively small number of particles. In this case the coverage means that most of

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

the possible poses will be generated. 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 (see Fig. 3).



Frame 137 **Frame 141**

Fig. 3: Losing and finding the tracked target despite the mis-tracking on the previous frame.

6 Results

We have tested out algorithm on the sequences, provided by L.Sigal, which is available at his site and sequences provided for the Workshop on Evaluation of Articulated Human Motion and Pose Estimation. The sequences contain different activities, such as walking, boxing etc. which were captured by 7 cameras. The sequences were captured using MoCap system that provides the correct 3D locations of the body parts for evaluation of the results and comparison to other tracking algorithms.

The first sequence that we have used was walking. The video was captured at the frame rate 60 fps. We have tested the annealed particle filter tracker, implemented by A. Balan [2] , and compared the results with the ones produced by GPAPF tracker (see Fig. 4 and 5) using the evaluation method suggested by A. Balan [2]. The error was calculated, based on comparison of the tracker's output and the result of the MoCap system. The quantitive comparison between two trackers is shown on the Fig. 4. 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 it stems from the graph, the estimation error of the GPAPF tracker is lower then the error of the annealed particle filter. Similar results were achieved for 15 fps. On Fig. 5 one can see the actual pose estimation for this sequence. The estimation is projected to two cameras. The first 2 rows show the results of the GPAPF tracker. The last two rows show the results of the annealed particle filter.

In order to show the ability of the tracker to track different sequences we have filmed several films in our lab. On these sequences we have filmed similar behaviors to those that we have learned. The learning was done on the sequence provided by L. Sigal and the tracking was done on the new sequence. The tracker was able to successfully track the body parts on the new films.

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

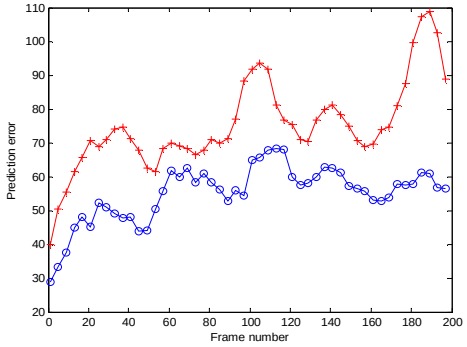
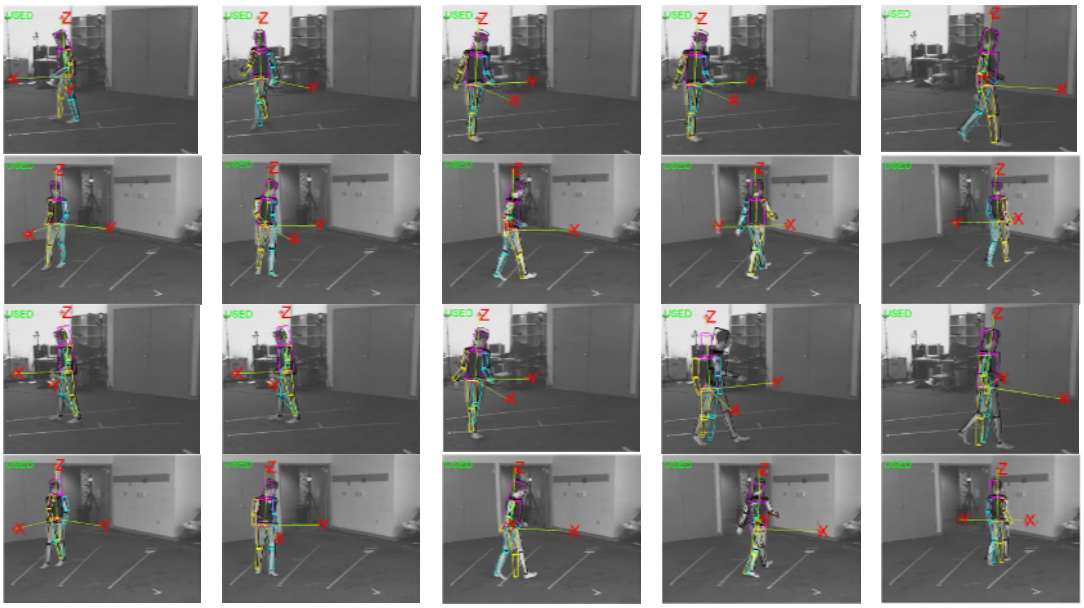


Fig. 4: The errors of the annealed tracker (red crosses) and GPAPF tracker (blue circles) for a walking sequence captured at 30 fps.



Frame 37 **Frame 73** **Frame 117** **Frame 153** **Frame 197**

Fig. 5: Tracking results of annealed particle filter tracker and GPAPF tracker. Sample frames from the walking sequence. First row: GPAPF tracker, first camera. Second row: GPAPF tracker, second camera. Third row: annealed particle filter tracker, first camera. Forth row: annealed particle filter tracker, second camera.

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

7 Conclusion and Future Work

We have presented an approach that uses GPDM in order to reduce the dimensionality and in this way to improve the ability of the annealed particle filter tracker to track the object even in a high dimensional space. We have also shown that using GPDM can increase the ability to recover from temporal target loss.

The challenging task is to track several people simultaneously. The main problem is that in this case there is high possibility of occlusion. Furthermore, while for a single person each body part can be seen from at least by one camera that is not the case for the crowded scenes.

References

1. Agarwal, A. and Triggs B. 2004. Learning to track 3D human motion from silhouettes. In *Proc. ICML* , pp. 9–16.
2. Balan, A., Sigal, L. and Black., M. 2005 A quantitative evaluation of video-based 3D person tracking. IEEE Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS), pp. 349–356.
3. Bregler, C., and Malik, J. 1998 Tracking people with twists and exponential maps. In *CVPR*, pp. 8–15.
4. Deutscher, J and Reid, I. 2004.Articulated body motion capture by stochastic search. *International Journal of Computer Vision*, 61(2):185-205.
5. Ioffe S. and Forsyth. D. A. 2001. Human tracking with mixtures of trees In *Proc. ICCV*.
6. Isard, M. and Blake, A. 1998 CONDENSATION - conditional density propagation for visual tracking, Int. J. Computer Vision, 29, 1, 5-28.
7. Isard, M. MacCormick, J. 2001 Bramble: A Bayesian multiple blob tracker. In *ICCV*, pages 34–41.
8. Lawrence, N. D. 2004 Gaussian process latent variable models for visualization of high dimensional data. *NIPS* 16, pp. 329-336.
9. Mori G., and Malik J. 2002. Estimating human body configurations using shape context matching. In *Proc. ECCV*,
10. Ramanan, D., and Forsyth, D. A., 2003 Automatic Annotation of Everyday Movements. *Neural Info. Proc. Systems* (NIPS), Vancouver, Canada.
11. Rohr, K. 1997. Human movement analysis based on explicit motion models. In *Motion-Based Recognition*. Kluwer Academic Publishers, Dordrecht Boston, ch. 8, pp. 171–198.
12. Sigal, L., Bhatia, S., Roth, S., Black M. and Isard, M. 2004 Tracking loose-limbed people. In *CVPR*, vol. 1, pp. 421–428.
13. Sidenbladh, H., Black, M. and Fleet, D. 2000 Stochastic tracking of 3d human figures using 2d image motion. In *Proc. ECCV*.
14. Urtasun, R., Fleet, D.J., Hertzmann, A., Fua, P. 2005 Priors for people tracking from small training sets. In *Proc. ICCV*, Beijing v1, pp. 403-410.
15. Urtasun, R., Fleet, D. J., Fua, P. 2006. 3D People Tracking with Gaussian Process Dynamical Models. .In *Proc. CVPR'06* , v.1, pp. 238-245.
16. Wang, J.,Fleet, D. J., and Hetzmann, A. 2005. Gaussian process dynamical models.In *Proc.NIPS*. Vancouver, Canada: pp. 1441-1448.

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8