3D HUMAN TRACKING WITH GAUSSIAN PROCESS ANNEALED PARTICLE FILTER

Leonid Raskin, Ehud Rivlin, Michael Rudzsky
Computer Science Department, Technion Israel Institute of Technology, Technion City, Haifa, Israel
raskinl@cs.technion.ac.il, ehudr@cs.technion.ac.il, rudzsky@cs.technion.ac.il

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

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. Finally, the most challenging difficulty that has to be solved in order to achieve satisfactory results of pose understanding is the ambiguity caused by body.

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

There are two main approaches for body pose estimation. The first on is based one a single frame, (Perona et al. (2000)) or appearance (D. A. Forsyth et al. (2001), J. Malik et al. (2002)). 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 (Ramanan (2003)), as well as multiple views (Deutscher et al. (2004)).

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. An alternative is to apply a strong model of dynamics (Mikolajczyk et al. (2004)).

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.

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

3.1 Particle filter

The particle filter algorithm was developed for tracking objects, using the Bayesian inference framework. Let us denote $x_n$ as the hidden state vector and $y_n$ be the measurement in time $n$. The algorithm builds an approximation of a maximum posterior estimate of the filtering distribution: $p(x_n | y_{1:n})$, where $y_{1:n} \triangleq (y_1, \ldots, y_n)$ 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

$$\pi_n^{(i)} \propto p(y_n | x_n^{(i)})$$

Using Bayes’ rule, the filtering distribution can be calculated using two steps:

**Prediction step:**

$$p(x_n | y_{1:n-1}) =$$

$$= \int p(x_n | x_{n-1}) p(x_{n-1} | y_{1:n-1}) dx_{n-1} \tag{1}$$

**Filtering step:**

$$p(x_n | y_{1:n}) \propto p(y_n | x_n) p(x_n | y_{1:n-1}) \tag{2}$$

Therefore, starting with a weighted set of samples $\{x_0^{(i)}, \pi_0^{(i)}\}_{i=1}^{N_p}$, the new sample set $\{x_n^{(i)}, \pi_n^{(i)}\}_{i=1}^{N_p}$ is generated according to the distribution, that may depend on the previous set $\{x_{n-1}^{(i)}, \pi_{n-1}^{(i)}\}_{i=1}^{N_p}$ and the new measurements $y_n$:

$$x_n^{(i)} \sim q(x_n^{(i)} | x_{n-1}^{(i)}, y_n), i = 1, \ldots, N_p.$$ The new weights are calculated using the following formula:

$$\pi_n^{(i)} = k \pi_n^{(i)} p(y_n | x_n^{(i)}) p(x_n^{(i)} | x_{n-1}^{(i)}) \tag{3}$$

$$k = \left( \sum_{i=1}^{N_p} \pi_n^{(i)} \right)^{-1} \left( \sum_{i=1}^{N_p} p(y_n | x_n^{(i)}) p(x_n^{(i)} | x_{n-1}^{(i)}, y_n) \right)$$

The main problem is that the distribution $p(y_n | x_n)$ may be very peaky and far from being convex. This usually happens for the high dimensional problems, like body part tracking. In this case a large number of samples have to be taken in order to find the global maxima, instead of choosing a local one. The other problem that arises is that the approximation of the $p(x_n | y_{1:n})$ for high dimensional spaces is a very computationally inefficient and hard task. Often a weighting function $w_n^{(i)}(y_n, x)$ can be constructed according to the likelihood function as it is in the condensation algorithm of Isard and Blake (1998), such that it provides a good approximation of the $p(y_n | x_n)$, but is also relatively easy to calculate. Therefore, the problem becomes to find configuration $x_k$ that maximizes the weighting function $w_n^{(i)}(y_n, x)$.

3.2 Annealed Particle Filter
The main idea is to use a set of weighting functions instead of using a single one. While a single weight function may contain several local maxima, the weight function should be as smooth as possible. A series of \( \{ w_n(y_n, x) \}_{n=0}^{M} \) is used, where \( w_{n+1}(y_n, x) \) differs only slightly from \( w_n(y_n, x) \) and represents a smoothed version of it. The samples should be drawn from the \( w_0(y_n, x) \) function, which might be peaky, and therefore a large number of particles needed to be used in order to find the global maxima. Therefore, \( w_M(y_n, x) \) is designed to be a very smoothed version of \( w_0(y_n, x) \). The usual method to achieve this is by using \( w_n(y_n, x) = w_0(y_n, x)^{b_n} \), where \( 1 = \beta_0 > ... > \beta_M \)

and \( w_0(y_n, x) \) is equal to the original weighting function. 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 \( \{ x_{n,m}^{(i)}, \pi_{n,m}^{(i)} \}_{i=1}^{N} \). The annealed particle filter algorithm:

Initialization: \( \{ x_{n,M}^{(i)}, 1/N \}_{i=1}^{N} \)

For each annealed layer \( m \) do:

1. Calculate the weights:
   \[
   \pi_{n,m}^{(i)} = k \frac{w_0(y_n, x_n)|p(x_{n,m}^{(i)}|x_{n-1}^{(i)})}{q(x_{n,m}^{(i)}|x_{n-1,m}^{(i)}, y_n)}
   \]

   \[
   k = \frac{\sum_{i=1}^{N} w_0(y_n, x_n)|p(x_{n,m}^{(i)}|x_{n-1}^{(i)})}{q(x_{n,m}^{(i)}|x_{n-1,m}^{(i)}, y_n)}^{-1}
   \]  

2. Draw \( N \) particles from the weighted set \( \{ x_{n,m}^{(i)}, \pi_{n,m}^{(i)} \}_{i=1}^{N} \) with replacement and with distribution \( p(x = x_{n,m}^{(i)}) = \pi_{n,m}^{(i)} \).

3. \( x_{n,m-1} = q(x_{n,m-1}^{(i)}|x_{n,m}^{(i)}, y_n) = x_{n,m} + N_{m} \) where \( N_{m} \) is a Gaussian noise \( N_{m} \sim N(0, P_{n,m}) \). If \( m \neq 0 \) go to 1.

4. The optimal configuration can be calculated using the following formula:

\[
x_{n} = \sum_{i=1}^{N} x_{n,0}^{(i)} \pi_{n,0}^{(i)}. \]

The unweighted particle set for the next observation is produced using \( x_{n+1,M} = x_{n,0} + N_{0} \).

4 GAUSSIAN FIELDS

The Gaussian Process Dynamical Model (GPDM) represents a mapping from the latent space to the data: \( y = f(x) \), where \( x \in \mathbb{R}^d \) denotes a vector in a \( d \)-dimensional latent space and \( y \in \mathbb{R}^D \) 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:

\[
x_i = \sum_j a_j \phi_i(x_i) + n_{x,t}
\]

\[
y_i = \sum_j b_j \psi_j(x_i) + n_{y,t}
\]

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 \( \phi_i \) and \( \psi_j \) 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:

\[
p(Y|X, \beta) = \frac{|W|^{N}}{(2\pi)^{NP/2}|K_y|^{1/2}} e^{-\frac{1}{2} tr(K_yWYy^T)}
\]

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

\[
(K_y)_{i,j} = \beta_i \exp(-\frac{\beta_i}{2} \| x_i - x_j \|^2) + \frac{\delta_{k y}}{\beta_i}
\]

\( W \) is a scaling diagonal matrix. It is used to account for the different variances in different data elements. The hyper parameter \( \beta_i \) represents the scale of the
output function. $\beta_2$ represents the inverse of the RBF's and $\beta_3^{-1}$ represents the variance of $n_{ij}^e$.

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 $A$, it can be shown (see Wang et al. (2005)) that

$$p(X|\overline{\alpha}) = \frac{p(x_i)}{(2\pi)^{\frac{n_{ij}}{2}}|K|^\frac{1}{2}} e^{-\frac{1}{2}e(K^\top x_{i}-x_{i})}$$  \hspace{1cm} (8)

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

$$(K_{i})_{i,j} = \alpha_i \exp(-\frac{\alpha_i}{2}\|x_i-x_j\|^2)$$  

\hspace{1cm} + $\alpha_i x_i^T x_j + \frac{\delta_{i,j}}{\alpha_i}$  \hspace{1cm} (9)

Following Wang et al. (2005)

$$p(X,\overline{\alpha},\overline{\beta}|Y) \propto p(Y|X,\overline{\beta})p(X|\overline{\alpha})p(\overline{\alpha})p(\overline{\beta})$$  \hspace{1cm} (10)

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

$$\Lambda = \frac{d}{2}\ln|K| + \frac{1}{2}tr(K^{-1}X_{out}X_{out}^T) + \sum_{i} \ln \alpha_i$$

\hspace{1cm}  -$N\ln|W| + \frac{D}{2}\ln|K| + \frac{1}{2}tr(K^{-1}YW^2X^T) + \sum_{j} \ln \beta_j$  \hspace{1cm} (11)

5  GPA PF 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 limb lengths and $\Gamma$ are the angles between limb 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 (see Fig. 1). 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 $\Gamma$ 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 matching.

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 $N_{cv}$ hypothesized edge are drawn. A squared probability function is calculated for these samples:

$$P^e(X,Z) = \frac{1}{N_{cv}} N_{cv} \sum_{i=1}^{N_{cv}} (1 - p_i^e(\Gamma_i, Z_i))^2$$  \hspace{1cm} (12)

where $N_{cv}$ is a number of camera views, $Z_i$ is the $i$-th image. The $p_i^e(\Gamma_i, Z_i)$ 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 $p_i^e(\Gamma_i, Z_i)$ 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. The
higher this probability is, the less information this image plane carries about the particular body part, and thus it should have lower impact on the weighting function. The whole process can be also explained by the next formula. 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 follows:

$$w(\Gamma_i, Z_j) = \sum_{k=1}^{N_e} p_k^{fg}(\Gamma_i, Z_j) / \sum_{l=1}^{N_e} \sum_{j=1}^{N_{bp}} p_l^{fg}(\Gamma_i, Z_j)$$

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

$$\tilde{p}(\Gamma_{bp}, Z_{cv}) = \sum_{k=1}^{N_e} (1 - p_k^{bg}(\Gamma_{bp}, Z_{cv}))^2$$

The second feature is the silhouette obtained by subtracting the background from the image. Background segmentation has separated the subject from the background and the obtained results can be seen in Fig. 1. 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:

$$P^\text{bg}(\Gamma, Z) = \frac{1}{N_{cv}} \frac{1}{N_e} \sum_{i=1}^{N_e} \sum_{j=1}^{N_{bp}} (1 - p_i^{fg}(\Gamma, Z_j))^2$$

where $N_{bp}$ is the number of different body parts in the model, $p_i^{fg}(\Gamma, Z_j)$ is the value is the foreground pixel values at the sample points, as it is shown in the Fig. 1.

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:

$$P_k(\Gamma_{bp}, Z_{cv}) = \frac{P_k^{ref}(\Gamma_{bp}, Z_{cv})P_k^{hyp}(\Gamma_{bp}, Z_{cv})}{\sqrt{\sum_{j=1}^{N_{cv}} (1 - P_j(\Gamma_j, Z_i))}}$$

where $P_j^{ref}(\Gamma, Z_i)$ is the value of bin $j$ in the reference histogram, and the $P_j^{hyp}(\Gamma, Z_i)$ 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:

$$w(\Gamma, Z) = \exp(-P(\Gamma, Z) + P^\text{bg}(\Gamma, Z) + P_k(\Gamma, Z))$$

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. 2). 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.

![Figure 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.](image)

The particles are drawn in the latent space. In order to calculate the weighting function we apply the Z transform 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. 2, 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. 2 the lighter pixels represent lower variance.

![Frame 137](image)

![Frame 141](image)

Figure 3: 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. 3).

Another 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 disadvantage of the tracker is that the latent space is not continuous (see Fig. 2). Two sequential poses may appear not too close in the latent space; therefore there is a minimal number of particles that should be drawn in order to be able to perform the tracking.

The other 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.

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. 4 and 5). The error was calculated, based on comparison of the tracker's output and the result of the MoCap system. In 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 can be seen, the GPAPF tracker produces more accurate estimation of the body location. Same results were achieved for 15 fps. In Fig. 5 one can see the actual pose estimation for this sequence. The estimation is projected to the first and second cameras. The first 2 rows show the results of the GPAPF tracker. The 3rd and 4th rows show the results of the annealed particle filter.

The second sequence was captured in our lab. On that sequence we have filmed similar behaviour, produced by a different actor. The frame rate was 15 fps. However, the learning was done on the first sequence data. The GPAPF tracker was able to track the person and produced results similar to the ones, which were produced for the original sequence.

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. We have also presented a method to approximate the possibility of self occlusion and we have suggested a way to adjust the weighed function for such cases, in order to be able to produce more accurate evaluation of a pose.

The main problem is that the training process is done for a specific action. The ability of the tracker to use this data to track an activity that is completely different from the one learned, has not been shown yet. In the future we plan to change the approach in order to be able to track different activities, using same learned data.

The other challenging task is to track two or more people simultaneously. The main problem here 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 one camera that is not the case for the crowded scenes.

REFERENCES


