# 3D Human Body-Part Tracking and Action Classification Using a Hierarchical Body Model

Leonid Raskin
raskinl@cs.technion.ac.il
Michael Rudzsky
rudzsky@cs.technion.ac.il
Ehud Rivlin
rivlin@cs.technion.ac.il

Computer Science Department
Technion -Israel Institute of Technology
Haifa, Israel, 3200

**Abstract**

This paper presents a framework for hierarchical 3D articulated human body-part tracking and action classification. We introduce a Hierarchical Annealing Particle Filter (H-APF) algorithm, which applies nonlinear dimensionality reduction of the high dimensional data space to the low dimensional latent spaces combined with the dynamic motion model and the Hierarchical Human Body Model. The improved annealing approach is used for the propagation between different body models and sequential frames. The tracking algorithm generates trajectories in the latent spaces, which provide low dimensional representations of body poses, observed during the motion. These trajectories are used to classify human motions. The tracking and classification algorithms were checked on HumanEvaI, HumanEvaII, and other datasets, involving more complicated motion types and transitions and proved to be effective and robust. The comparison to other methods and the error calculations are provided.

## 1 Introduction

Human body pose tracking is a challenging task for several reasons. The large variety of poses and high dimensionality of the human model complicates the examination of the entire subject and makes it harder to detect each body part separately. However, the poses can be presented in a low dimensional space using the dimensionality reduction techniques. Such a reduction improves the tracker's robustness, ability to recover from temporary target loss, and the computational effectiveness. There exist several possible strategies for reducing the dimensionality. Firstly, it is possible to restrict the range of movement of the subject [12]. However, due to the restricting assumptions, the resulting trackers are not capable of tracking general human poses. Another approach is to learn low-dimensional latent variable models [20] using Isomap [17]. However, methods like Isomap and locally linear embedding (LLE) [13] do not provide a mapping between the latent space and the data space. Urtasun et al. [18] proposed to use a form of probabilistic dimensionality reduction by GPDM [19] to formulate the tracking as a nonlinear least-squares optimization problem. Andriluka et

BMVC 2009 doi:10.5244/C.23.12

al. [2] use Hierarchical Gaussian Process Latent Variable Model HGPLVM to model prior on possible articulations and temporal coherency within a *walking* cycle. Raskin et al. [11] introduced Gaussian Process Annealed Particle Filter (GPAPF), which uses GPDM to construct a latent space that describes poses and the tracking is performed in this latent space. Nevertheless, such a reduction usually allows tracking and detection only of the poses similar to those used for the learning process.

In this paper we introduce a Hierarchical Annealing Particle Filter (H-APF) tracker which exploits the Hierarchical Human Body Model in order to achieve accurate body part estimates. We apply a nonlinear dimensionality reduction using the Hierarchical Gaussian Process Latent Variable Model (HGPLVM) [6] and the Annealing Particle Filter (APF) [4] is used for the propagation between the sequential frames. A hierarchical model of the human body expresses conditional dependencies between the body parts and allows us to capture properties of separate parts. The human body model consists of two independent parts: one containing information about 3D location and orientation of the body and the other describing the articulation of the body. The articulation is represented as a hierarchy of body parts. The method uses previously observed poses from various motion types to generate mapping functions from the low-dimensional latent spaces to the data spaces that describe the poses. The tracking algorithm consists of two stages. First, the particles are generated in the latent space and are transformed to the data space using the learned mapping functions. Second, rotation and translation parameters are added to obtain valid poses. Finally, the likelihood function is calculated in order to evaluate how well these poses match the image. The resulting tracker estimates the locations in the latent spaces that represent poses with the highest likelihood.

During the last decade many different methods for behavior recognition and classification of human actions have been proposed. The popular methods include Hidden Markov Models (HMM), Finite State Automata (FSA), and context-free grammar (SCFG) etc. Sato et al. [14] presented a method to use extraction of human trajectory patterns that identify the interactions. Mori et al. [7] use hierarchies in the actions and Continuous HMM to recognize everyday gestures. S. Park et al. [10] proposed a method using a nearest neighbor classifier for the recognition of two-person interactions such as *hand-shaking, pointing, and standing hand-in-hand*. Hongeng et al. [5] proposed probabilistic finite state automata for recognition of a sequential occurrence of several scenarios. J. Park et al. [9] presented a recognition method that combines model-based tracking and deterministic finite state automata. Niebles and Li [8] use hierarchical representation of a human body model, that is used for the action categorization.

Our method performs the action classification in the latent spaces, produced by HGPLVM. A pose estimated on each frame corresponds to a coordinate in the latent space. Therefore, an action is represented by a curve in this latent space. The classification of the motion is based on the comparison of the sequences of latent coordinates that the tracker produces to the sequences that represent different actions (we denote such sequences as models). The modified Frèchet distance [1] is used in order to perform the comparison. This approach allows for the introduction of actions different from those used for the learning of the latent spaces by exploiting the model that represents it. We also show that the action classification, when performed in the latent space, is robust and has a high accuracy rate.

# 2 Hierarchical Annealing Particle Filter

## 2.1 Motivation: From GPAPF to H-APF

As mentioned many researchers use dimensionality reduction of the body pose in order to achieve robust and computationally effective trackers. Recently, Raskin et al. [11] suggested using dimensionality reduction of the human poses space. They introduced Gaussian Process Annealed Particle Filter (GPAPF). According to this method a latent space is generated by a nonlinear dimensionality reduction (using Gaussian Process Dynamic Model (GPDM)) of the space of previously observed poses from different motion types, such as *walking, running, punching and kicking*.

The drawback of the GPAPF algorithm is that a latent space describes only the poses that resemble those used in the learning process. However, if a person performs a new movement which differs from those already learned, then the new poses will be represented less accurately by the latent space. Therefore, after experimenting with the GPAPF algorithm we concluded that it is hardly possible to improve the quality of the results achieved with GPAPF. Specifically, when an action contains several motion types or previously unseen motion the error rate was high. The other concern is the ability of GPAPF to track the body parts during the transition between different motion types. Even if both motions were used in the learning to produce a common latent space, the error rate during the transition phase was relatively high.

## 2.2 Learning

The commonly-used human body model $\Gamma$ consists of two statistically independent subspaces $\Gamma = \{\Lambda, \Omega\}$. The first subspace $\Lambda \subseteq \mathbb{R}^6$ describes the body's 3D location: the rotation and the translation. The second one $\Omega \subseteq \mathbb{R}^{25}$ describes the actual pose, which is represented by the angles between different body parts (see. [3] for more details about the human body model). We define a Hierarchical Human Body Model such as that shown in Figure 1. We denote the number of the hierarchy layers as $H$.

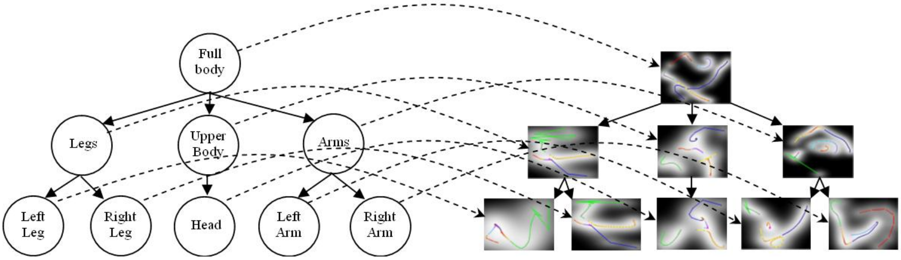


Figure 1: A hierarchical model of the human body (left) and the corresponding latent spaces (right). Each latent space corresponds to a set of body parts and is learned for five different motion types: hand waving(red), lifting an object (blue), kicking (green), sitting down (yellow), and punching (cyan).

Let us define $\Omega_{h,l} \subset \Omega$ as the $l$-th subspace in hierarchy level $h$. For instance, on the hierarchy depicted by Figure 1, the $\Theta_{3,2}$ stands for the subspace that describes the right leg. For each $\Omega_{h,l}$ the HGPLVM algorithm constructs a latent space $\Theta_{h,l}$ and the mapping function $\wp^{(h,l)}$ that maps this latent space to the partial data space $\Omega_{h,l}$.

$$\wp^{(h,l)} : \Theta_{h,l} \mapsto \Omega_{h,l} \tag{1}$$

Let us also define $\theta_{h,l} \in \Theta_{h,l}$ as the latent coordinate in the $l$-th latent space in the $h$-th hierarchy layer. $\omega_{h,l} \in \Omega_{h,l}$ is the partial data vector that corresponds to $\theta_{h,l}$. Applying Formula 1 we have $\omega_{h,l} = \wp^{(h,l)}\left(\theta_{h,l}\right)$.

An important property of a hierarchical model is that $\Omega_{h,l}$ is a subset of some $\Omega_{h-1,k}$ in the higher layer of the hierarchy. In other words, for any hierarchy level $h \in [2,H]$ and for any subspace $l \in [1,L_h]$ in this layer, there always exists a subspace $\hat{l}$ in the hierarchy level $h-1 \in [1,H-1]$, such that $\Omega_{h,l} \subset \Omega_{h-1,\hat{l}}$ (here $\hat{l}$ is the index of the parent node in the hierarchy tree).

Additional mapping functions that are learned by the HGPLVM are the mapping function between the latent spaces, that correspond to the subspaces, which are connected in the hierarchy tree $\phi^{(h,l)} : \Omega_{h,\hat{l}} \mapsto \Omega_{h+1,l}$.

Finally, $\lambda_{h,l,n} \in \Lambda$, $\omega_{h,l,n} \in \Omega_{h,l}$ and $\theta_{h,l,n} \in \Theta_{h,l}$ denote the location, full data space (full pose) vector and latent coordinates in hierarchy layer $h$ on the latent space $l$ on the frame $n$.

## 2.3 Tracking algorithm

In this section we present a Hierarchical Annealing Particle Filter (H-APF) for 3D body part tracking. A H-APF run is performed at frame $n$ using image-observations $y_n$. These observations can be a data from a single camera or, as shown in Section 3, from several cameras. In this section we follow the notations used in [4].

The model configuration on the frame $n$ and hierarchy layer $h$ on the latent space $l$ contains translation, rotation parameters, latent coordinates and the full data space vectors:

$$s^{(i)}_{h,l,n} = \{\lambda^{(i)}_{h,l,n}; \omega^{(i)}_{h,l,n}; \theta^{(i)}_{h,l,n}\} \tag{2}$$

The tracker state is represented by a set of weighted particles:

$$S^{\pi}_{h,l,n} = \{(s^{(0)}_{h,l,n}, \pi^{(0)}_{h,l,n}), \dots, (s^{(N)}_{h,l,n}, \pi^{(N)}_{h,l,n})\} \tag{3}$$

where $s^{(i)}_{h,l,n}$ stands for the model configuration and $\pi^{(i)}_{h,l,n}$ corresponds to a particle weight. The un-weighted set of particles is denoted by

$$S_{h,l,n} = \{s^{(0)}_{h,l,n}, \dots, s^{(N)}_{h,l,n}\} \tag{4}$$

The tracking algorithm consists of two stages. The first is the generation of new particles in the latent space. In the second stage a corresponding mapping function is applied that transforms latent coordinates to the data space. After the transformation, the translation and rotation parameters are added and the 31-dimensional vectors are constructed. These vectors represent a valid poses, which are projected to the cameras in order to estimate the likelihood.

Each H-APF run has the following steps:

**Step 1-Initialization.** For every frame the run is started at layer $h = 1$ and is initialized by a set of un-weighted particles $S_{1,1,n} = \{s^{(i)}_{1,1,n}\}_{i=1}^{N_p} = \left\{\lambda^{(i)}_{1,1,n}; \omega^{(i)}_{1,1,n}; \theta^{(i)}_{1,1,n}\right\}_{i=1}^{N_p}$.

**Step 2.** The weight of each particle is calculated:$\pi^{(i)}_{h,l,n} \propto w^m\left(y_n, s^{(i)}_{h,l,n}\right) = w^m\left(y_n, \lambda^{(i)}_{h,l,n}, \omega^{(i)}_{h,l,n}\right)$ where $w^m$ is the weighting function suggested by Deutscher and Reid [4]. The weights are normalized so that $\sum_{i=1}^{N_p} \pi^{(i)}_n = 1$.

**Step 3.** $N$ particles are drawn randomly with replacements and with a probability equal to their weight $\pi^{(i)}_{h,l,n}$. For every latent space $l$ in the hierarchy level $h+1$ the particle $s^{(j)}_{h+1,l,n}$ is produced using the $j^{th}$ chosen particle $s^{(j)}_{h,\hat{l},n}$ ($\hat{l}$ is the index of the parent node in the hierarchy tree): $\lambda^{(j)}_{h+1,l,n} = \lambda^{(j)}_{h,\hat{l},n} + B_{\lambda_{h+1}}$ and $\theta^{(j)}_{h+1,l,n} = \phi(\theta^{(j)}_{h,\hat{l},n}) + B_{\theta_{h,\hat{l}}}$, where $B_{\lambda_h}$ and $B_{\theta_{h,l}}$ are multivariate Gaussian random variables with covariances and $\Sigma_{\lambda_h}$ and $\Sigma_{\theta_{h,l}}$ correspondingly and mean 0. In order to construct a full pose vector $\omega^{(j)}_{h+1,l,n}$ is initialized with the $\omega^{(j)}_{h,\hat{l},n}$: $\omega^{(j)}_{h+1,l,n} = \omega^{(j)}_{h,\hat{l},n}$ and then updated at the coordinates defined by $\Omega_{h+1,l}$: $(\omega^{(j)}_{h+1,l,n})|_{\Omega_{h+1,l}} = \wp^{h+1,l}\left(\theta^{(j)}_{h+1,l,n}\right)$ (The notation $a|_B$ stands for the coordinates of vector $a \in A$ defined by the subspace $B \subseteq A$.) The idea is to use a pose that was estimated using the higher hierarchy layer, with small variations in the coordinates described by the $\Omega_{h+1,l}$ subspace. Finally, the new particle for the latent space $l$ in the hierarchy level $h+1$ is $s^{(j)}_{h+1,l,n} = \{\lambda^{(j)}_{h+1,l,n}; \omega^{(j)}_{h+1,l,n}; \theta^{(j)}_{h+1,l,n}\}$

**Step 4.** The sets $S_{h+1,l,n}$ have now been produced which can be used to initialize the layer *h+1*. The process is repeated until we arrive at the *H*-th layer.

**Step 5.** The $j^{th}$ chosen particle $s^{(j)}_{H,l,n}$ in every latent space $l$ in the lowest hierarchy level is used to produce $s^{(j)}_{1,1,n+1}$ un-weighted particle set for the next observation: $\lambda^{(j)}_{1,1,n+1} = \frac{1}{L_H}\sum_{l=1}^{L_H} \lambda^{(j)}_{H,l,n}$ and $\theta_{1,1,n+1} = \wp^{-1}_{1,1}\left(\omega^{(j)}\right)$, where $\omega^{(j)}$ is calculated using

$$for 1 \leq l \leq L_H \; do \quad \omega^{(j)}|_{\Omega_{H,l}} = \omega^{(j)}_{H,l,n}|_{\Omega_{H,l}} \tag{5}$$

Here $L_H$ denotes the number of subspaces in the last hierarchy layer $H$. Finally, $s^{(j)}_{1,1,n+1} = \{\lambda^{(j)}_{1,1,n+1}; \omega^{(j)}; \theta^{(j)}_{1,1,n+1}\}$.

**Step 6.** The final configuration can be calculated using the following method: $\lambda^{(opt)}_n = \frac{1}{L_H}\sum_{l=1}^{L_H}\sum_{j=1}^{N} \lambda^{(j)}_{H,l,n}\pi^{(j)}$ and $\omega^{(opt)}_n = \sum_{j=1}^{N} \omega^{(j)}\pi^{(j)}$, where $\pi^{(j)} \propto w^m\left(y_n, \lambda^{(opt)}_n, \omega^{(j)}\right)$ is the normalized weight of the selected particles so that $\sum_{i=1}^{N_p} \pi^{(i)} = 1$ and $\omega^{(j)}$ is calculated as in step 5.

### 2.4 Action Classification

The classification of the actions is based on the comparison of the predefined motion patterns and the sequences of the poses detected by the tracker during a performed motion. We use Frèchet distance [1] in order to determine the class of the motion, such as *walking*, *kicking*, *waving*. The Frèchet distance between two curves measures the similarity of the curves taking their direction into consideration. This method is quite tolerant of position errors. While in general it is hard to calculate the Frèchet distance, Alt et. al [1] have suggested an efficient algorithm to calculate it between two piecewise linear curves.

We define a polygonal curve $P^E$ as a continuous and piecewise-linear curve made of segments connecting vertices $E = \{v[0], ..., v[n]\}$. The curve can be parameterized with a parameter $\alpha \in [0, n]$, where $P^E(\alpha)$ refers to a given position on the curve, with $P^E(0)$ denotes $v[0]$ and $P^E(n)$ denotes $v[n]$. The distance between two curves $P^{E_1}$ and $P^{E_2}$ is defined as

$$F\left(P^{E_1}, P^{E_2}\right) = \min_{\alpha_{1,2}:[0,1]\to\left[0,n_{1,2}\right]} \left\{\max\left\{\|P^{E_1}(\alpha_1(t)) - P^{E_2}(\alpha_2(t))\|_2 : t \in [0,1]\right\}\right\} \tag{6}$$

where $\alpha_1(t)$ and $\alpha_2(t)$ represent sets of continuous and increasing functions with $\alpha_1(0) = 0$, $\alpha_1(1) = n_1$, $\alpha_2(0) = 0$, $\alpha_2(1) = n_2$.

Suppose there are $K$ different motion types. Each type $k$ is represented by a model $M_k$, which is a collection of sequences of the $l_k + 1$ latent coordinates on each latent space in every hierarchy layer. We denote a sequence that corresponds to the model $M_k$ on the $l$-th latent space in the hierarchy level $h$ as $M_{h,l,k} = \left\{\theta^{M_k}_{(h,l)}[0], ..., \theta^{M_k}_{(h,l)}[m_k]\right\}$. For a frame sequence $Y = \{y_0, ..., y_m\}$ the H-APF tracker generates a sequence of latent coordinates for each latent space. Such a sequence of the coordinates on the $l$-th latent space in the hierarchy level $h$ is denoted as $\Upsilon_{h,l} = \left\{\theta^{\Upsilon}_{(h,l)}[0], ..., \theta^{\Upsilon}_{(h,l)}[m]\right\}$. Now, using 6, we can compare the model and the sequence for each latent space. Finally, the cumulative distance is calculated:

$$d(Y, M_k) = \sum_{h=1}^{H} \sum_{l=1}^{L_h} F\left(\Upsilon_{h,l}, M_{h,l,k}\right) \tag{7}$$

where $H$ is the depth of hierarchy tree and $L_h$ is the number of the latent spaces in the layer $h$ of the hierarchy. The model with the smallest distance is chosen to represent the type of the action.

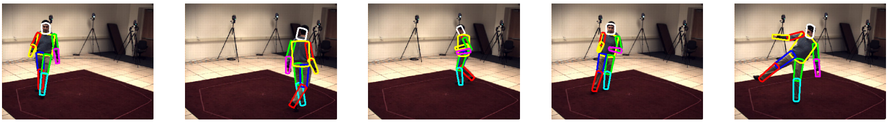
# 3 Results

### 3.1 Tracking

We tested H-APF tracker using the HumanEvaI and HumanEvaII datasets [15]. The sequences contain different activities, such as *walking*, *boxing*,and *jogging*, which were captured by several (four) synchronized and calibrated cameras. The sequences were captured using the MoCap system that provides the correct 3D locations of the body joints, such as shoulders and knees. This information is used for an evaluation of the results and a comparison to other tracking algorithms. The error is calculated, based on a comparison of the tracker's output to the ground truth, using the average distance in millimeters between 3-D joint locations [3].

The first sequence that we used contains a single person, walking in a circle. The video was captured at the frame rate of 60 fps and then down-sampled to 15 fps. We compared the results produced by APF (implemented by A. Balan for [3]), GPAPF and H-APF trackers. For APF and GPAPF algorithms we used five layers with 100 particles in each; for H-APF we used a hierarchy as on Figure 1 with 100 particles in each layer. Figure 3.a shows the error graphs, produced by APF (green), the GPAPF tracker (blue) and the H-APF tracker (red) trackers.

Next we trained HGPLVM with several different motion types. We used this latent space in order to track the body parts on the videos from the HumanEvaI and HumanEvaII datasets. Figure 2 shows the result of the tracking of the HumanEvaII(S2) dataset which combines three different behaviors: walking, jogging and balancing. Figure 3.b-d presents the errors for HumanEvaI(S1, walking1, frames 6-590)(top), HumanEvaII(S2, frames 1-1202)(middle) and HumanEvaII(S4, frames 2-1258)(bottom). Finally, Table 3.1 compares the average error for these sequences produced by APF, GPAPF and H-APF algorithms. We have also created a dataset with several different actions performed by different actors. Figure 4 shows the result of the of H-APF tracker on *running* (top), *kicking* (middle), and *object lifting* (bottom) sequences from dataset.



frame 50 frame 230 frame 640 frame 700 frame 800

Figure 2: Tracking results of H-APF tracker. Sample frames from the combo1 sequence from HumanEvaII(S2) dataset.

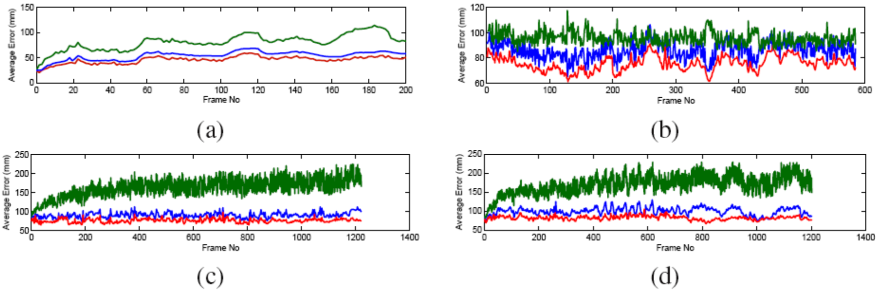


Figure 3: The errors produced by APF tracker (marked by green line), GPAPF tracker (marked by blue line) and H-APF tracker (marked by red line). The errors for (a) the walking sequence captured at 15 fps; (b) HumanEvaI(S1, walking1, frames 6-590); (c) HumanEvaII(S2, frames 1-1202)(middle) and (d) HumanEvaII(S4, frames 2-1258)(bottom).

| Dataset<br>Sequence | HumanEvaI<br>S1 | HumanEvaII<br>S2 | HumanEvaII<br>S4 |
|---|---|---|---|
| APF | 95.4 | 163.8 | 172.1 |
| GPAPF | 86.3 | 86.6 | 89.0 |
| H-APF | 75.4 | 75.2 | 81.8 |

Table 1: The average error for these sequences produced by APF, GPAPF and H-APF. The error measures the average distance in millimeters between 3-D joint locations.

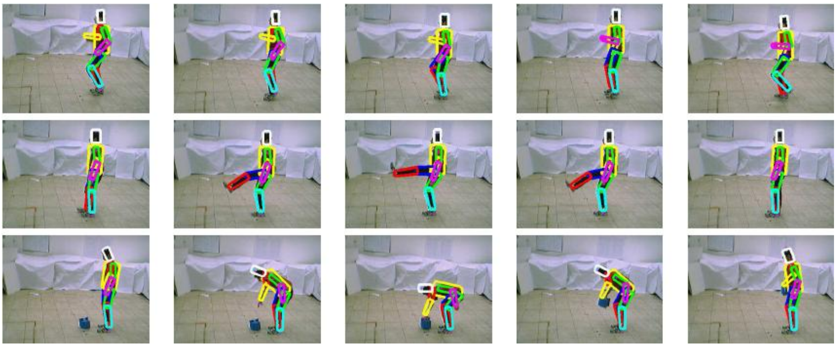


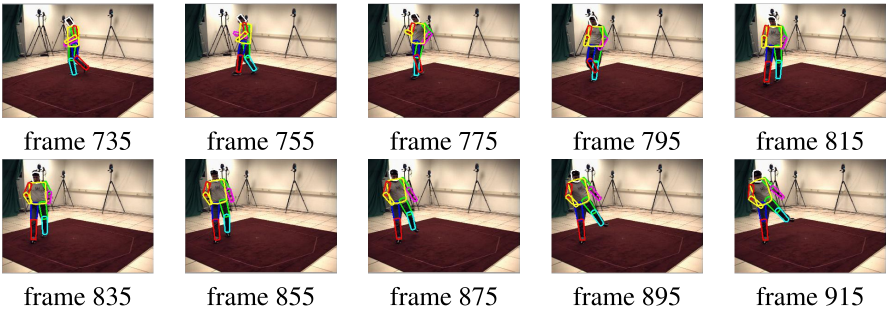
Figure 4: Tracking results of H-APF tracker. Sample frames from the *running* (top), *kicking* (middle), and *object lifting* (bottom) sequences from dataset created in our lab.

As we mentioned above, H-APF allows dealing with transitions between motions in a much more natural fashion. This is one of the reasons why we see the improvement in tracking results for the datasets involving several different motion types, such as HumanEvaII. Figure 5 shows the sample frames from HumanEvaII S1 dataset, which involves three different actions. The subject starts with *walking*, then continues with *jogging* and ends with *balancing*. The Figure shows the transition from *jogging* to *balancing*. As it is shown on Figure 3 there is no distinguishable peaks on the error graph during the transition, which implies that the tracker is capable of maintaining stable results during the motion type change.

## 3.2 Motion Classification

For the action classification testing we used the dataset, produced in our lab. Due to the ease of classification of HumanEvaI and HumanEvaII, we do not provide the results on these sets.

For the first experiment we used three different activities: (1) *lifting an object*, *kicking with* (2) *the left* and (3) *the right leg*. In the second experiment we used five different activities: (1) *hand waving*, (2) *lifting an object*, (3) *kicking*, (4) *sitting down*, and (5) *punching*. For each activity five different sequences were captured. A cross-validation procedure was applied: for each motion type one sequence was used in order to construct the latent space and define the model of the motion type and the rest were used for evaluation. Figure 6 shows the trajectories produced by the trackers for the *sitting down* action projected on different latent spaces from different hierarchy levels. The green line represents the correct

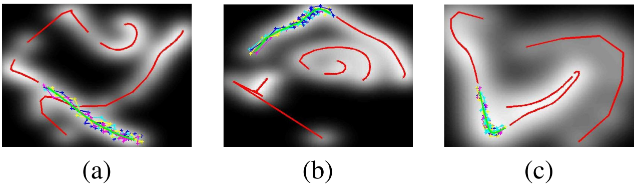


frame 735 frame 755 frame 775 frame 795 frame 815

frame 835 frame 855 frame 875 frame 895 frame 915

Figure 5: Sample frames showing the motion type transition (from *jogging* to *balancing*) in HumanEvaII(S4) dataset.

model (a model of *sitting down* action), the red lines represent the incorrect models (models of *waving, punching, kicking and picking an object* actions), and the colored lines represent the trajectories produced by the tracker. The crosses on the colored lines represent the actual latent locations, that were estimated by the H-APF tracker.



(a) (b) (c)

Figure 6: Tracking trajectories of *sitting down* action projected to different latent spaces: (a)hierarchy level 1, latent space 1; (b) hierarchy level 2, latent space 3; (c)hierarchy level 3, latent space 5. The green line represents the correct model (a model of *sitting down* action), the red lines represent the incorrect models (models of *waving, punching, kicking and picking an object* actions), and the colored lines represent the trajectories produced by the tracker. The crosses on the colored lines represent the actual latent locations, that were estimated by the H-APF tracker.

For the first set we were able to achieve perfect classification results. This is due to clear differences between the models of the different motions in the latent space. In the second experiment the models are less distinguishable, which makes the classification task harder. Table 2 shows the results of the classification for the actions from the second dataset. The lower classification rates of actions involving the hand gestures can be explained by the native similarity of the actions. The poor classification rates of sitting down and object lifting actions are due to the high self occlusions, which caused the tracker to produce less accurate results.

# 4 Conclusion and Future Work

In this paper we introduced an 3D human body part tracker that uses HGPLVM to improve the ability of the annealed particle filter to track the object in a high-dimensional space. The

Table 2: The accuracies of the classification, using the combined approach, for 5 different activities: *hand waving*, *object lifting*, *kicking*, *sitting down*, and *punching*. The rows represent the correct motion type; the columns represent the classification results.

| | Hand waving | Object lifting | Kicking | Sitting down | Punching |
|---|---|---|---|---|---|
| Hand waving | 17 | 0 | 0 | 0 | 3 |
| Object lifting | 0 | 18 | 0 | 2 | 0 |
| Kicking | 0 | 0 | 20 | 0 | 0 |
| Sitting down | 0 | 3 | 0 | 17 | 0 |
| Punching | 2 | 0 | 0 | 0 | 18 |

use of hierarchy allows for a better detection of body part positions and can thus perform more accurate tracking. We have also presented an algorithm for human motion classification using a hierarchy of low-dimensional latent spaces.

Currently the classification algorithm uses all the latent space in the hierarchy equally. However, some actions are defined only by a movement of certain body parts, and are completely independent of the articulations of the other parts. For instance, running or kicking are strictly defined by leg movements and are independent of the position of the head. Using this information may not only improve the ability to recognize the type of action but also to detect irregularities, such as walking with raised arms.

Another interesting problem is the construction of latent spaces for multiple actions. Typically one wants to have consistent smooth structure in the latent space for a given motion to ensure that Gaussian process can easily be used to track the motion. When a few actions are used for training the resulting latent spaces consist of smooth curves. However, using 5 or more different motion types usually do not sustain the property. We plan to address this in our future research.

# References

[1] H. Alt, C. Knauer, and C. Wenk. Matching polygonal curves with respect to the Frèchet distance. *Proc. 18th International Symposium on Theoretical Aspect of Computer Science (STACS)*, 2:63–74, 2001.

[2] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. *Proc. Computer Vision and Pattern Recognition (CVPR)*, 1:1–8, 2008.

[3] A. Balan, L. Sigal, and M. Black. A quantitative evaluation of video-based 3d person tracking. *IEEE Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS)*, pages 349–356, 2005.

[4] J. Deutscher and I. Reid. Articulated body motion capture by stochastic search. *International Journal of Computer Vision (IJCV)*, 61(2):185–205, 2004.

[5] S. Hongeng, F. Bremond, and R. Nevatia. Representation and optimal recognition of

human activities. *Proc. Computer Vision and Pattern Recognition (CVPR)*, 1:818–825, 2000.

[6] N. D. Lawrence and A. J. Moore. Hierarchical gaussian process latent variable models. *Proc. International Conference on Machine Learning (ICML)*, 2007.

[7] T. Mori, Y. Segawa, M. Shimosaka, and T. Sato. Hierarchical recognition of daily human actions based on continuous hidden markov models. *International Conference on Automatic Face and Gesture Recognition*, page 779-784, 2004.

[8] J. C. Niebles and F. F. Li. A hierarchical model of shape and appearance for human action classification. *Proc. Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007.

[9] J. Park, S. Park, and J. K. Aggrawal. Video retrieval of human interactions using model-based motion tracking and multi-layer finite state automata. *Lecture Notes in Computer Science (LNCS)*, 2728, 2003.

[10] S. Park and J.K. Aggrawal. Recognition of human interactions using multiple features in a grayscale images. *Proc. International Conference on Pattern Recognition (ICPR)*, 1:51–54, 2000.

[11] L. Raskin, M. Rudzsky, and E. Rivlin. Dimensionality reduction for articulated body tracking. *Proc. The True Vision Capture, Transmission and Display of 3D Video (3DTV)*, 2007.

[12] K. Rohr. Human movement analysis based on explicit motion models. *Motion-Based Recognition*, 8:171–198, 1997.

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

[14] K. Sato and J.K. Aggarwal. Recognizing two-person interactions in outdoor image sequences. *IEEE Workshop on Multi-Object Tracking*, 2001.

[15] L. Sigal and M. J. Black. Measure locally, reason globally: Occlusion-sensitive articulated pose estimation. *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2: 2041–2048, 2006.

[16] L. Sigal and M. J. Black. Synchronized video and motion capture dataset for evaluation of articulated human motion. *Technical Report CS-06-08, Dept. of Computer Science*, 2:2006.

[17] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, 2000.

[18] R. Urtasun, D. J. Fleet, and P. Fua. 3d people tracking with gaussian process dynamical models. *Proc. Computer Vision and Pattern Recognition (CVPR)*, 1:238–245, 2006.

[19] J. Wang, D. J. Fleet, and A. Hetzmann. Gaussian process dynamical models. *Information Processing Systems (NIPS)*, pages 1441–1448, 2005.

[20] Q. Wang, G. Xu, and H. Ai. Learning object intrinsic structure for robust visual tracking. *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2:227–233, 2003.