Tracking in Action Space

Dennis L. Herzog and Volker Krüger

Copenhagen Institute of Technology, Aalborg University,
Lautrupvang 2, 2750 Ballerup, Denmark
{deh,vok}@cvmi.aau.dk

Abstract. The recognition of human actions such as pointing at objects ("*Give me that...*") is difficult because they ought to be recognized independent of scene parameters such as viewing direction. Furthermore, the parameters of the action, such as pointing direction, are important pieces of information. One common way to achieve recognition is by using 3D human body tracking followed by action recognition based on the captured tracking data. General 3D body tracking is, however, still a difficult problem. In this paper, we are looking at human body tracking for action recognition from a context-driven perspective. Instead of the space of human body poses, we consider the space of possible actions of a given context and argue that 3D body tracking reduces to action tracking in the parameter space in which the actions live. This reduces the high-dimensional problem to a low-dimensional one. In our approach, we use parametric hidden Markov models to represent parametric movements; particle filtering is used to track in the space of action parameters. Our approach is content with monocular video data and we demonstrate its effectiveness on synthetic and on real image sequences. In the experiments we focus on human arm movements.

1 Introduction

Human communicative actions such as pointing ("*Give me this*") or object grasping are typical examples of human actions in a human-to-human communication problem [1,2]. These actions are usually context-dependent and their parametrization defines an important piece of their information [3,1]. To capture these communicative actions is challenging because a) the capturing should be independent of scene parameters such as viewing direction or viewing distance and b) one needs complex action models that allow to recover *what* action is performed and *which* parameters it has. The observation that the parameters of an action carry important information about the *meaning* of the action was already earlier pointed out by Wilson and Bobick in [3] using the example "*The Fish was this big*".

One strategy for recognizing such actions [3,4] is to first track the human movements using, e.g., a 3D body tracker and to then in a second step feed these tracks into an action recognition engine, such as, e.g., HMMs [5,6] or even parametric HMMs (PHMMs) [3]. Considering the first ingredient of the above outlined strategy, it was pointed out recently again [7] that 3D tracking and pose

K.N. Kutulakos (Ed.): ECCV 2010 Workshops, Part I, LNCS 6553, pp. 100–113, 2012.

estimation, especially from monocular views, is non-trivial. Common approaches are model-based generative ones [8,9,10], that compare a synthesized candidate pose with the image data.

In this paper, we would like to argue that such a 2 step approach as the above is un-necessary complication. Human actions are usually goal-directed and are performed within a certain context (*eating, cooking* etc.). Furthermore, actions are often performed on objects [11,12] which leads to the observation that the objects can prime the actions performed on them (e.g. reaching, pointing, grasping, pushing-forward) [13,14]. Thus, we would like to suggest to look at 3D human body tracking from an object and context-driven perspective: Instead of asking "*What is the set of joint angles that make a human model fit to the observation*" we ask "*What action causes a pose that fits to the observation*". By replacing in a particle filter approach the propagation model for joint angles [8] with a propagation model for human actions we become able to re-formulate the 3D tracking problem instead as a problem of recognizing the action itself (incl. its parameters). In other words, instead of having to estimate the high-dimensional parameter vector of the human body model, we sample the action and its parameters in the low-dimensional *action space*. By using a generative model for the action representation we can deduce the appropriate 3D body pose from the sampled action parameters and compare it to the input data. We call this approach *Tracking in Action Space*. In our work we use parametric hidden Markov models (PHMMs) [3] as the generative action model. While classical HMMs can recognize essentially only a specific trajectory or movement, PHMMs are able to model different parametrizations of a particular movement. For example, while a common HMMs would be able to recognize only one specific pointing action, PHMMs are able to recognize pointing action into different directions [3]. Furthermore, (P)HMMs are generative models which means that for recognizing an observation they compare it with a model of the expected observation. In the experiments in this paper, our action space spans over the human arm actions *pointing, reaching, pushing*, the corresponding 2D coordinates of *where* to point at or reach to, plus a timing parameter. One might argue that such an approach cannot be used for general 3D body tracking because the action space will always be too limited. However, following the arguments in [15,16,17,18,19] that human actions are composed out of motor primitives similarly to human speech being composed out of phonemes, we believe that the limited action space considered here can be generalized to the space spanned by the action primitives. Stochastic action grammars could be used as in [20,17,19] to model more complex actions. Furthermore, [19] explains how a language for human actions can be generated based on grounded concepts, kinetology, morphology and syntax. For estimating the action and action parameters during tracking we have used classical Bayesian propagation over time which, as we will discuss below, provides an excellent embedding for tracking in action space, including the use of primitives and action grammars. Our key contribution are:

– introduction of *Tracking in Action space* by posing the 3D human pose estimation problem as one of action and action parameter recognition,

- good recovery of 3D pose and action recognition plus action parameters
- is content with monocular video data
- potentials to run in real-time, our implementation runs with just edge features
- concise formulation in particle filtering framework
- concept of approaching action recognition as a context and object dependent problem.

The paper is structured as follows: In Section 2 we will discuss the *Tracking in Action Space* in detail. Section 2.1 explains the use of parametric HMMs (PHMMs) for modeling the action space and Section 2.2 explains details on using the PHMMs for tracking in action space. In Section 2.3, we will discuss the calculation of the observation likelihood for a given image. In Section 3, we provide experimental results for synthetic and for real video data, and we conclude with final comments in Section 4.

Related Work
As it can be seen in the survey [21], early deterministic methods, as gradient based methods, have been overcome by stochastic methods due to problems as depth disambiguations, occlusions, etc. The methods rage from the basic particle filtering, as described in [22], to, e.g., belief propagation [23,7]. Efficient techniques for particle filtering [8,10,24] in combination with (simple) motion models [9] to constrain the particle propagation or the state space [25] are investigated since the number of required particles generally scale exponentially with the degree of freedom. Novel frameworks use for example multistage approaches ([7] considers the stages: coarse tracking of people, body part detection and 2D joint location estimation, and 3D pose inference) or implement various constrains ([23] considers the constrains concerning self-occlusion, kinematic, and appearance and uses belief propagation to infer the pose within a graphical model). Contrary to the simple motion models, which roughly approximate the state space used by certain motions, for example as a linear subspace [25], advanced approaches, as for example locally linear embedding allow to learn [26] the intrinsic low dimensional manifold, or aim at the learning of nonlinear dynamic system (NLDS) on motion data, as it is approached through the Gaussian process model developed in [27] in a more efficient way. Interestingly for our context, the learning of NLDS requires a vast amount of training data [27], whereas *"classic"* HMMs for example can be easily trained but are limited in their expressiveness for complex motions. In our work, we use the *parametric extension* to HMMs and are interested in both the pose estimation and the uncovering the action parameters.

2 Tracking in Action Space

In this section we want to discuss our approach for *Tracking in Action Space* in detail.

We are starting our discussion looking at the classical Bayesian propagation over time:

$$p_t(\boldsymbol{\omega}_t) \propto \int P(\boldsymbol{I}_t|\boldsymbol{\omega}_t) P(\boldsymbol{\omega}_t|\boldsymbol{\omega}_{t-1}) p_{t-1}(\boldsymbol{\omega}_{t-1}) d\boldsymbol{\omega}_{t-1}, \quad (1)$$

where $\boldsymbol{I}_t$ is the current observation, $p_t(\boldsymbol{\omega}_t)$ the probability density function (pdf) for the random variable (RV) $\boldsymbol{\omega}_t$ at time t, $P(\boldsymbol{\omega}_t|\boldsymbol{\omega}_{t-1})$ the propagation density, and $P(\boldsymbol{I}_t|\boldsymbol{\omega}_t)$ the likelihood measurement of $\boldsymbol{I}_t$, given $\boldsymbol{\omega}_t$. If applied for 3D human body tracking, the RV $\boldsymbol{\omega}$ usually specifies the set of joint angles for some human body model (see, e.g., [8]). The propagation density is used to constrain the RV $\boldsymbol{\omega}_t$ to the most likely pose values at each time step t [24,9,8]. In order to compute the likelihood $P(\boldsymbol{I}_t|\boldsymbol{\omega}_t)$, a human body model is generated using the pose values from $\boldsymbol{\omega}_t$ and is then compared with the input image $\boldsymbol{I}_t$.

For *tracking in action space* we use the RV $\boldsymbol{\omega}$ to control a generative action model with $\boldsymbol{\omega} = (a, \boldsymbol{\theta}, k)$. Here, the parameter a identifies which action it is, $\boldsymbol{\theta}$ is a vector specifying the parameters of action a, and k is a timing parameter which specifies the timing within the action model. In our work, we use parametric hidden Markov models (PHMMs) [3] which we will discuss in detail, below. We train the PHMM for each action a on joint location data captured from human performances of the action a. Using one PHMM for each action, the parameter a refers then to the a-th PHMM, the parameter vector $\boldsymbol{\theta}$ specifies the parameters for the PHMM, e.g., *where* we point to or grasp at, and the parameter k is discrete and specifies the PHMM state. Then, the likelihood $P(\boldsymbol{I}_t|\boldsymbol{\omega}_t) = P(\boldsymbol{I}_t|(a, \boldsymbol{\theta}, k)_t)$ is computed by first using the a-th PHMM to *generate* the joint location of the 3D human body pose for parameters $\boldsymbol{\theta}$ and HMM-state k. We can do this because the PHMM is trained on joint location data for the action a, as discussed above. In the second step, these joint angles of the 3D body pose are translated into the corresponding 3D body model which is then projected onto the image plane and compared to the input image $\boldsymbol{I}_t$. For computing the observation likelihood, we also make use of the standard deviations of the observation densities of the PHMM. This second step is in principle the same as the likelihood measurement in Eq. (1). The propagation density $P(\boldsymbol{\omega}_t|\boldsymbol{\omega}_{t-1})$ can be considerably simplified. Assuming that a human finishes one action primitive before starting a new one, the action identifier a is constant until an action (primitive) is finished. If we have an action grammar model for complex human actions as in [19,17] then the corresponding action grammar can be used to control the progression of a. Likewise, the action parameters $\boldsymbol{\theta}$ are approximately constant until the action primitive is completed. The timing parameter k changes according to the transition matrix of the HMM.

In the following, we will discuss some details of PHMMs (Section 2.1), how we use the PHMMs in our tracking scheme (Section 2.2) and how we compute the observation likelihood (Section 2.3).

2.1 Parametric Hidden Markov Models

In this section we give a short introduction to parametric hidden Markov models (PHMMs) [3], which are an extension of the hidden Markov model (HMM)

[28] through additional parameters. The additional variables allow to model a systematic variation within a class of modeled sequences. For example, for a pointing, reaching or pushing action, the variation is given by the target location (e.g., a location pointed at).

A hidden Markov model is a generative model. It is a finite state machine extended in a probabilistic manner. For an continuous (P)HMM $\boldsymbol{\lambda} = (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$, the vector $\boldsymbol{\pi} = (\pi_i)$ and the transition matrix $\boldsymbol{A} = (a_{ij})$ define the prior state distribution of the initial states i and the transition probability between hidden states. In continuous HMMs, the output of each hidden state is described by a density function $b_i(\boldsymbol{x})$, which is in our context a multivariate Gaussian density $b_i(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$. The HMM parameters can be estimated through the Baum-Welch algorithm [28] for a set of training sequences.

An output sequence $\boldsymbol{x} = \boldsymbol{x}_1 \dots \boldsymbol{x}_T$ can be drawn from the model by generating step-by-step a state sequence $\boldsymbol{Q} = q_1 \dots q_T$ with respect to the initial probabilities π_i and the transition probabilities a_{ij} and drawing for each state q_t the output $\boldsymbol{x}_t$ from the corresponding observation distribution $b_{q_t}(\boldsymbol{x})$. Generally, there is no unique correspondence between an output sequence $\boldsymbol{X}$ and a state sequence $\boldsymbol{Q}$ as different hidden state sequences can generate the same output sequence $\boldsymbol{X}$. Since we are interested in the temporal behavior and correspondence between parts of the sequence and the state, we use a left-right model [28] to model the trajectories of different actions. A left-right model allows only transitions from each state to itself or to a state with a greater state index.

The movement trajectories we are considering generally underlie a systematic variation, e.g., the *pointed at* position. A general HMM can handle this only as noise or with a great number of states. A parametric HMM (PHMM) [3] models the systematic variation as a variation of the means of the observation distributions $b_i^{\boldsymbol{\theta}}(\boldsymbol{x})$, where $b_i^{\boldsymbol{\theta}}(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_i^{\boldsymbol{\theta}}, \boldsymbol{\Sigma}_i)$. The means are functions $\boldsymbol{\mu}_i^{\boldsymbol{\theta}} = \boldsymbol{f}_i(\boldsymbol{\theta})$ that are approximated for each state separately in the training process.

In [3] a linear model and a more general nonlinear model is used to model $\boldsymbol{f}_i(\boldsymbol{\theta})$. In the linear case, each function $\boldsymbol{f}_i(\boldsymbol{\theta})$ is of the form $\boldsymbol{\mu}_i = \bar{\boldsymbol{\mu}}_i + \boldsymbol{W}_i \boldsymbol{\theta}$, where the matrix $\boldsymbol{W}_i$ describes the linear variation. In the more general nonlinear case, a neural network is used for each state i that is trained to approximate a more general nonlinear dependency. For both models, the training procedures are supervised by providing the parametrization $\boldsymbol{\theta}$ for each training sequence. For training in the linear case, an extension of the Baum-Welch approach is used. For the non-linear case, a generalized EM (GEM) procedure was developed. We will denote a PHMM with parameter $\boldsymbol{\theta}$ by $\boldsymbol{\lambda}^{\boldsymbol{\theta}}$.

2.2 Action Tracking: PHMM-Based

In this section we want to discuss the details of using PHMMs for modeling the actions for the action tracking. In our problem scenario we have a set $\mathcal{A} = \{1, \dots, M\}$ of actions, where we have for each action $a \in \mathcal{A}$ a trained PHMM $\boldsymbol{\lambda}_a^{\boldsymbol{\theta}}$. They define each action through the corresponding sequences of human joint settings. On these sequences of joint settings, the PHMMs are trained. E.g. the PHMM for the pointing action is trained on joint location sequence

coming from different pointing actions, including pointing actions into different directions. We consider left-right PHMMs with a single multi-variate Gaussian as the observation density $b_i(\boldsymbol{x}) = b^{\boldsymbol{\theta}}_{a,i}(\boldsymbol{x})$ for each state i of $\boldsymbol{\lambda}^{\boldsymbol{\theta}}_a$ with a rather small covariance.

Our human action model has the following parameters: the value a identifies the PHMM $\boldsymbol{\lambda}_a$. The value k specifies the hidden state, i.e., the progress of the action a. The parametrization $\boldsymbol{\theta}$ (e.g., a pointing location) of the PHMM $\boldsymbol{\lambda}^{\boldsymbol{\theta}}_a$ is used as the parameter for the observation functions $b^{\boldsymbol{\theta}}_{a,i}$. Hence, we have defined our random space over the tracking parameter $\boldsymbol{\omega}_t = (a, \boldsymbol{\theta}, k)_t$.

In order to *generate* an action using a PHMM one simple way is to sample the PHMM: the first state $k_{t=0}$ is drawn according to the initial distribution $\boldsymbol{\pi}_a$. At each time step the state change is governed by the probability mass function $P(k_t|k_{t-1})$ specified by the transition matrix $\boldsymbol{A}_a$. The actual synthesis is then done by sampling from the observation density $b^{\boldsymbol{\theta}}_{a,k_t}(\boldsymbol{x})$, parametrized with $\boldsymbol{\theta}$. In principle the likelihood for an observation $\boldsymbol{I}_t$ and for a given $\boldsymbol{\omega}_t = (a, \boldsymbol{\theta}, k)$ can be computed simply by $P(\boldsymbol{I}_t|\boldsymbol{\omega}_t) = b^{\boldsymbol{\theta}}_{a,k}(\boldsymbol{I}_t)$ if the observation space is the same as the one $b^{\boldsymbol{\theta}}_{a,k}$ is defined on. In our case, $P(\boldsymbol{x}|\boldsymbol{\omega}) = b^{\boldsymbol{\theta}}_{a,k}(\boldsymbol{x})$ defines the distribution of joint locations of 3D body poses which generates a corresponding 3D human body model (see Figure 2, left) which is then matched against the input image $\boldsymbol{I}_t$:

$$P(\boldsymbol{I}_t|\boldsymbol{\omega}_t) = \int_x P(\boldsymbol{I}_t|\boldsymbol{x})P(\boldsymbol{x}|\boldsymbol{\omega}_t)dx. \tag{2}$$

Finally, the propagation density $P(\boldsymbol{\omega}_t|\boldsymbol{\omega}_{t-1})$ is given as follows: k is propagated as mentioned above using $\boldsymbol{A}^{\boldsymbol{\theta}}_a$, and $\boldsymbol{\theta}$ is changed using Brownian motion. The variable a is initially drawn from an even distribution and is then kept in this work constant for each particle. We use a particle-filter approach [22] to estimate $\boldsymbol{\omega} = (a, \boldsymbol{\theta}, k)$. It is worth having a close look at this approach: The entropy of the density $p_t(\boldsymbol{\omega}_t)$ of Eq. 1 reflects the uncertainty of the so far detected parameters. Furthermore, by marginalizing over $\boldsymbol{\theta}$ and k, we can compute the posterior probability of each action a (see Figure 5). And by marginalizing over a and k, we can compute the action parameters, $\boldsymbol{\theta}$, respectively. This is displayed in Figure 1. The figure shows the progression of the RV $\boldsymbol{\omega}$ over time for a pointing action. The red and green lines show the most likely pointing coordinates u and v (for $\boldsymbol{\theta} = (u, v)$). The dotted lines show their corresponding uncertainties. The horizontal thin lines mark the corresponding correct values for u and v. As one can see, the uncertainty decreases with time, and after ≈ 60 frames, the correct parameters are recovered. This is about the time when the arm is fully stretched. In the next section, we will discuss how the observation likelihood $P(\boldsymbol{I}_t|\boldsymbol{\omega}_t)$ is computed.

2.3 Observation Model

We use an articulated model of the human body, see Figure 2 (left), where the kinematic structure is similar to the one used in [8]. Based on this model, we compute the observation function $P(\boldsymbol{I}|\boldsymbol{x})$ for an arm pose drawn from $P(\boldsymbol{x}|\boldsymbol{\omega})$

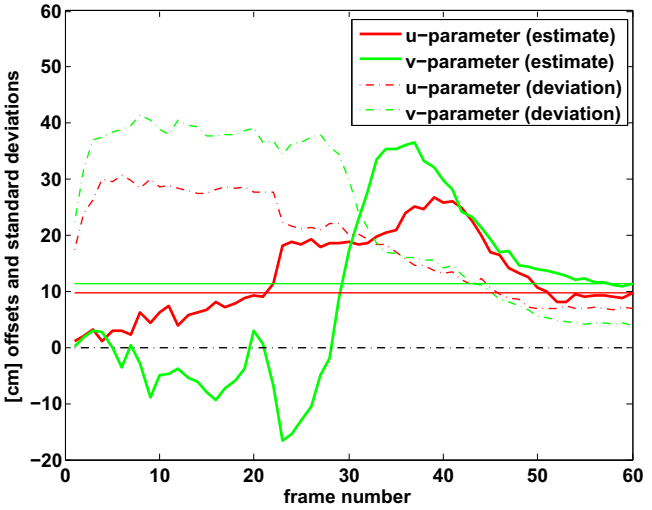

Fig. 1. The figure shows the progression of the current estimate of the action parameters over time. The estimate parameters u, v are computed as mean of all particles $\boldsymbol{\omega} = (u, v, k)$. The action parameters u, v correspond to the x-, and y-offsets to the center of the active table-top region (x corresponds to the horizontal, y to the deeps direction in Figure 3). The dotted lines show the standard deviation of the particles.

for a particle $\boldsymbol{\omega}_i$. Here, the arm pose is defined through $\boldsymbol{x} = (\boldsymbol{p}, \boldsymbol{q}, \boldsymbol{r})$, where $\boldsymbol{p}$, $\boldsymbol{q}$, and $\boldsymbol{r}$ are the positions of shoulder, elbow, and finger-tip in $\mathbb{R}^3$. The mapping to the model's kinematic is purely trigonometric. However, since the vector $\boldsymbol{x}$ is drawn from a Gaussian which is an observation density of an HMM, the lengths of the upper arm $|\boldsymbol{p} - \boldsymbol{q}|$ and forearm $|\boldsymbol{q} - \boldsymbol{r}|$ are not preserved. Generally, we set the finger-tip and shoulder positions of the model as given through $\boldsymbol{r}$ and $\boldsymbol{p}$. The elbow position $\boldsymbol{q}$ is then corrected with respect to the model's arm lengths through refining $\boldsymbol{q}$ to the nearest possible point on the plane defined through $\boldsymbol{p}$, $\boldsymbol{q}$, and $\boldsymbol{r}$. The rather unlikely case that $|\boldsymbol{p} - \boldsymbol{r}|$ is greater than the overall arm length is mapped on an arm pose, where the stretched arm points from the shoulder position $\boldsymbol{p}$ in the direction of $\boldsymbol{r}$. The computation of the observation function is based on the edge information of the arm silhouette, therefore, the contour $\mathcal{C}$ of the model is extracted from the rendered view for a pose $\boldsymbol{x}$. We defined the observation function similar to the method described in [8] on a smoothed edge image (see Figure 2, right), where the pixel values are interpreted as distances to the nearest edge. The edge distance image is calculated as follows. We calculate a normalized gradient image of the observed image $\boldsymbol{I}$, gray values above some threshold are set to 1. The image is finally smoothed with a Gaussian mask, and normalized. This edge image, denoted by G, can be interpreted as a distance to edge image. The value of $1 - G(\boldsymbol{c})$ of a pixel $\boldsymbol{c}$ can then be interpreted as distance values between 0 and 1, where the value 1 corresponds to a pixel with no edge in the vicinity. This distance interpretation is in some sense similar to the edge detection along normals as used in [22], but faster to evaluate.

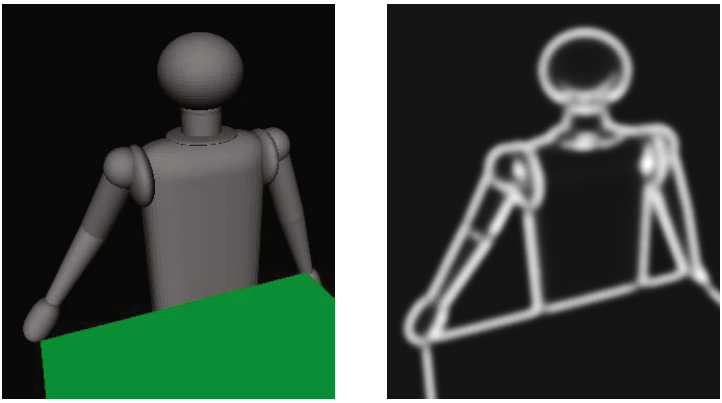

Fig. 2. *Left:* We use an articulated human model, where the skeletal structure is modeled by cones and super-quadratics. *Right:* The edge image (here of the model itself) is a smoothed gradient image, serving as a distance to edges image.

The observation function is computed by

$$P(\boldsymbol{I}|\boldsymbol{x}) = \exp -\frac{1}{2\gamma^2}\frac{1}{|\mathcal{C}|}\sum_{p\in\mathcal{C}}(1-G(p))^2, \tag{3}$$

where $\mathcal{C}$ is the model's contour and G is the smoothed edge image. A value of $\gamma = 0.15$ turned out to be reasonable in the experiments. An extension to multiple camera views is straight forward:

$$P(\boldsymbol{I}|\boldsymbol{x}) = \exp -\frac{1}{2\gamma^2}\sum_i \frac{1}{|\mathcal{C}_i|}\sum_{p\in\mathcal{C}_i}(1-G_i(p))^2, \tag{4}$$

where $\mathcal{C}_i$ and G_i are the corresponding contour sets and edge images of each view i.

3 Experiments

We have evaluated our approach on synthetic data and on monocular video data. For the testing with real data we captured performances of reaching and pointing movements simultaneously with a single video camera and Vicon motion capture system, the video camera was synchronized with the Vicon system. Our experiments were carried out on the monocular video data while the Vicon system provides us with ground truth. In this paper, we focus our attention on human arm pointing, reaching and pushing actions in particular (see tracked sequence, Figure 3). We call our scenario a *table-top scenario* where the actions are meant be performed on objects on a table. For each action, we used a linear PHMM $\boldsymbol{\lambda}^{(u,v)}$ trained on 20 demonstration of each action recorded with the Vicon system. The pointing and reaching 2D locations (u, v) at table-top cover a table-top

Fig. 3. *Pose Estimation through Tracking in Action Space.* In the rows 2–5, a whole pointing action (approaching and withdrawing motion) is shown with the recovered arm pose superimposed. The sequence has about 110 frames, of which every $\approx 6th$ frame is shown. The recovered pose corresponds to the sample/particle which explains the observation in the single monocular view best. The measurement is based only on the edge information in the gradient images (see top row). The estimated action parameters are indicated through the dot on the table: the color of the dot reflects the uncertainty of the indicated location (given through the entropy): a red dot indicates a large uncertainty, a green dot a low one. The last two rows show 10 completed pointing performances where one can see the stretched arm and the recovered pointing position (red dot).

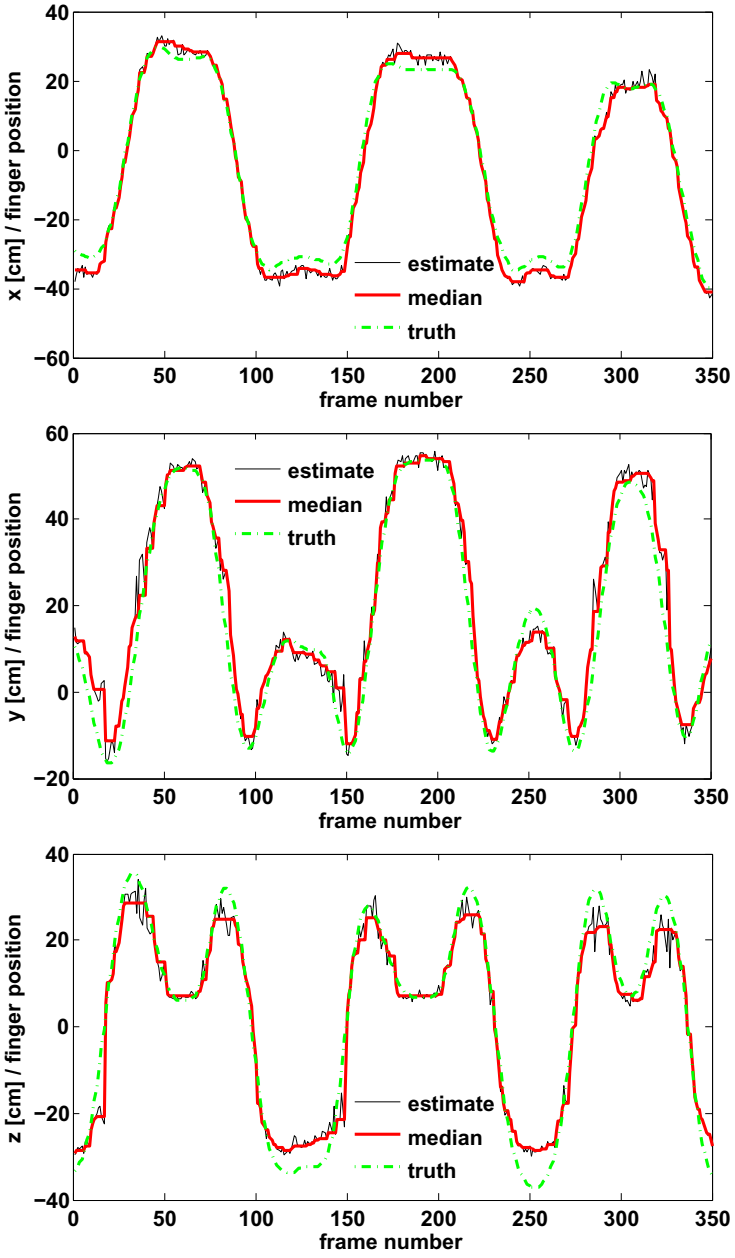

Fig. 4. The three plots compare the estimated finger position (red) to the true position (green) over three pointing actions. The position value (red) is the median filtered value of the pose estimates through action tracking (black), the true position is given through marker-based tracking. Here, the x-, y-, and z-position belong to the horizontal, depth, and height directions in Figure 3.

region of 30cm×80cm; these positions correspond with the parametrization of the corresponding PHMM, and were during the training procedure given by the finger tip location at the time of maximal arm extension. For training we used actions directed approximately to the four outer corners of the table top, with 5 repetitions each. The testing was done using continuous video data of 40 different random locations on the table top. Ground truth was available through the vicon system, the markers on the human were used by the vicon system, but were **not** used for our tracking experiments. We used a large number of HMM states which assured a good predictive model (resolution in time) and small state covariances of the observation densities. The used PHMM is a forward model which allows self-transitions and transitions to the subsequent three states. In order to allow the tracking of repeated performances of the actions, we have defined a very simple action grammar that models a loop. For the particle based tracking in action space we use 400 particles ($\boldsymbol{\omega}_i = (a; u, v; k)$). The propagation over time is done as described in Section 2.2. We decrease the diffusion of the u and v during the propagation step in dependence of the HMM state number k by $\sigma(k) = 0.4 \cdot \exp\{-2 \log_e(1/4) \cdot k/N\}$, where N number of states of the HMM. Our argument for the cooling down of the diffusion is that for the first frames the visual evidence for the right u and v is very weak, but as visual evidence increases with time, we inversely reduce the standard deviation.

The sampling and normalization of the image observation are performed as described in [22]; as discussed the observation function is based on the evaluation of the edge information in the monocular video data and the human body model for a particle $\boldsymbol{\omega}_i$.

The images in Figure 3 show that the arm pose is (visually) very accurately estimated. The following three complicating factors emphasize the capabilities of our *tracking in action space* approach: 1) all information is gathered from a *single monocular* view, 2) we use only a *single feature type* (edge information), and 3) the edge images (especially the first sequence part in Figure 3 due to the chair) have a lot of clutter, so that the silhouette of the arm is difficult to segment accurately. Besides the pose estimation, one can see in Figure 3 that the estimation of the action parameters (corresponding to the position indicated by the small colored dot on the table) converges to the true parameters of the action when the arm approaches the table-top.

The quality of the pose estimation over a sequence of several pointing actions is shown in Figure 4. Here, we compare the positions of the shoulder, elbow, and finger estimated through action tracking to the ground truth positions recorded with the marker-based Vicon system. The route-mean-square error of the three joint positions over the three pointing actions which are plotted in Figure 4 is 3.3cm, whereas the component-wise average error is just 2.4cm. It is interesting to note that this errors correlate with the natural variance of the human movements as recorded with the vicon system. This gives a mean error for the recovered table locations of 1.3cm.

Despite the good results above, the recognition rate between reaching and grasping was very low. This was due to the fact that these two actions have

Table 1. The table shows the errors in *cm* between the recovered parameters and the ground truth at the specified frames (completed pointing action). Frame numbers here are the same as in Figure 4.

frame	Error X	Error Y	Error Z
61	-2.53	-0.53	0.72
195	-3.77	-0.48	-0.36
301	-0.05	-1.70	1.62

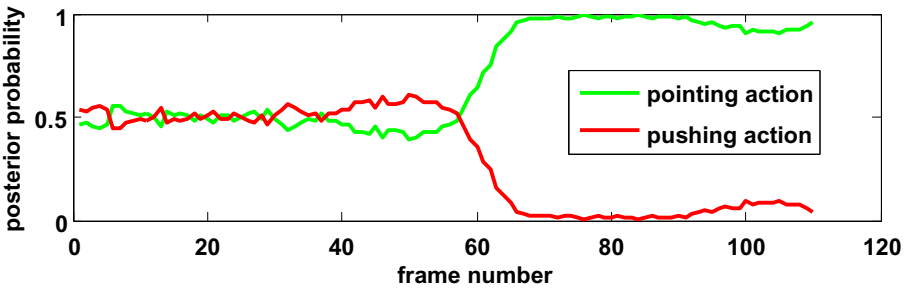

Fig. 5. The plot shows the posterior probability of the two actions *point* and *push* over time

the same general arm trajectory but differ only in the hand movement. On the other hand, testing on videos showed pointing and pushing actions in random order with at least 40 pointing and 40 pushing actions, the recognition rate was with $\approx 98\%$ as high as expected. Figure 5 shows the posterior probability for the two actions over time for a test video showing the pointing action: The action label a of a particle $\boldsymbol{\omega} = (a; \boldsymbol{\theta}; k)$ identifies the pointing or pushing action. By marginalizing $\boldsymbol{\omega}$ over $\boldsymbol{\theta} = (u, v)$ and k we compute the likelihood of a. The actions are very similar in the beginning. This is also visible in the plot: after 60 frames, the pushing action starts to differ from the observed pointing action and the posterior probability of the pushing action converges.

For the particle filter, we use only 400 particles, the edge features are fast to compute and on a standard workstation with non-threaded code we require presently 3.9s per frame. Ongoing work is to port our approach to CUDA for faster processing on a GPU.

4 Conclusions

We presented a novel concept of *tracking in action space* which combines the aspects of recognition, tracking and prediction based on parametric and time dependent action models. One can argue that this approach is too limited because it is not possible to model all different possible actions each with a PHMM. As our response, the starting point for our approach was (1) the observations that most actions are object and context dependent which means that a) object affordances and b) the scenario and the scene state greatly reduce the set of possible

actions, and (2) that according to neuroscientific evidence actions are composed using action primitives and grammars. Thus, even though the number of possible actions at any time is indeed large, only a small number of actions actually *can* appear, with a certain likelihood. Furthermore, all these possibly appearing actions do not need to be modeled each with a PHMM. Instead, it is sufficient to identify the building blocks of these action, i.e., the action primitives, to model only those with PHMMs and to then compose the complete actions out of these action primitive PHMMs. In the experiments, we have focused on arm actions as these were the ones needed in our human-robot communication scenario. But we believe that our approach should scale well to more body parts and more complex actions. In our future work we are going to consider different actions and the use of stochastic grammars in order to allow proper concatenation of actions as, e.g., reach for an object, move the object, withdraw arm etc. Extension to, e.g., dual arm actions in combination with upper body tracking is also ongoing work.

Acknowledgments. This work was partially funded by PACO-PLUS (IST-FP6-IP-027657).

References

1. Asfour, T., Welke, K., Ude, A., Azad, P., Dillmann, R.: Perceiving Objects and Movemetns to Generate Actions on a Humanoid Robot. In: Kragic, D., Kyrki, V. (eds.) Unifying Perspectives in Computational and Robot Vision. LNEE, vol. 8, pp. 41–55. Springer, Heidelberg (2008)
2. Krüger, V., Kragic, D., Ude, A., Geib, C.: The meaning of action: A review on action recognition and mapping. Advanced Robotics 21, 1473–1501 (2007)
3. Wilson, A.D., Bobick, A.F.: Parametric hidden markov models for gesture recognition. PAMI 21, 884–900 (1999)
4. Ren, H., Xu, G., Kee, S.: Subject-independent Natural Action Recognition. In: International Conference on Automatic Face and Gesture Recognition, Seoul, Korea, May 17-19 (2004)
5. Lv, F., Nevatia, R.: Recognition and Segmentation of 3-D Human Action Using HMM and Multi-class AdaBoost. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006, Part IV. LNCS, vol. 3954, pp. 359–372. Springer, Heidelberg (2006)
6. Xiang, T., Gong, S.: Beyond Tracking: Modelling Action and Understanding Behavior. International Journal of Computer Vision 67, 21–51 (2006)
7. Lee, M., Nevatia, R.: Human pose tracking in monocular sequences using multilevel structured models. PAMI 31, 27–38 (2009)
8. Deutscher, J., Blake, A., Reid, I.: Articulated body motion capture by annealed particle filtering. In: CVPR, vol. 2, pp. 126–133 (2000)
9. Sidenbladh, H., Black, M.J., Sigal, L.: Implicit Probabilistic Models of Human Motion for Synthesis and Tracking. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002, Part I. LNCS, vol. 2350, pp. 784–800. Springer, Heidelberg (2002)
10. Sminchisescu, C., Triggs, B.: Covarinace Scaled Sampling for Monocular 3D Body Tracking. In: CVPR, Kauai Marriott, Hawaii (2001)

11. Gupta, A., Davis, L.: Objects in action: An approach for combining action understanding and object perception. In: CVPR (2007)
12. Kjellström, H., Romero, J., Martínez, D., Kragić, D.: Simultaneous Visual Recognition of Manipulation Actions and Manipulated Objects. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 336–349. Springer, Heidelberg (2008)
13. Helbig, H.B., Graf, M., Kiefer, M.: The role of action representation in visual object. Experimental Brain Research 174, 221–228 (2006)
14. Bub, D., Masson, M.: Gestural knowledge evoked by objects as part of conceptual representations. Aphasiology 20, 1112–1124 (2006)
15. Rizzolatti, G., Fogassi, L., Gallese, V.: Neurophysiological Mechanisms Underlying the Understanding and Imitation of Action. Nature Reviews 2, 661–670 (2001)
16. Rizzolatti, G., Fogassi, L., Gallese, V.: Parietal Cortex: from Sight to Action. Current Opinion in Neurobiology 7, 562–567 (1997)
17. Guerra-Filho, G., Aloimonos, Y.: A sensory-motor language for human activity understanding. HUMANOIDS (2006)
18. Jenkins, O., Mataric, M.: Deriving Action and Behavior Primitives from Human Motion Data. In: International Conference on Intelligent Robots and Systems, Lausanne, Switzerland, September 30-October 4, pp. 2551–2556 (2002)
19. Guerra-Filho, G., Aloimonos, Y.: A language for human action. Computer 40, 42–51 (2007)
20. Ivanov, Y., Bobick, A.: Recognition of Visual Activities and Interactions by Stochastic Parsing. PAMI 22, 852–872 (2000)
21. Moeslund, T., Hilton, A., Krueger, V.: A survey of advances in vision-based human motion capture and analysis. Computer Vision and Image Understanding 104, 90–127 (2006)
22. Isard, M., Blake, A.: Condensation – conditional density propagation for visual tracking. International Journal of Computer Vision 29, 5–28 (1998)
23. Gupta, A., Mittal, A., Davis, L.S.: Constraint integration for efficient multiview pose estimation with self-occlusions. PAMI 30, 493–506 (2008)
24. Gall, J., Patthoff, J., Schnoerr, C., Rosenhahn, B., Seidel, H.P.: Interacting and annealing particle filters: Mathematics and recipe for applications. Jounral of Mathematical Imaging and Vision 28, 1–18 (2007)
25. Urtasun, R., Fua, P.: 3D Human Body Tracking Using Deterministic Temporal Motion Models. In: Pajdla, T., Matas, J. (eds.) ECCV 2004, Part III. LNCS, vol. 3023, pp. 92–106. Springer, Heidelberg (2004)
26. Elgammal, A., Lee, C.S.: Inferring 3D body pose from silhouettes using activity manifold learning. In: CVPR (2004)
27. Wang, J.M., Fleet, D.J., Hertzmann, A.: Correction to "gaussian process dynamical models for human motion". PAMI 30, 1118 (2008)
28. Rabiner, L.R., Juang, B.H.: An introduction to hidden Markov models. IEEE ASSP Magazine, 4–15 (1986)