Kernel Particle Filter for Real-Time 3D Body Tracking in Monocular Color Images

Joachim Schmidt and Jannik Fritsch
Applied Computer Science
Faculty of Technology, Bielefeld University
33594 Bielefeld, Germany
{jschmidt, jannik}@techfak.uni-bielefeld.de

Bogdan Kwolek
Rzeszów University of Technology
W. Pola 2
35-959 Rzeszów, Poland
bkwolek@prz.rzeszow.pl

Abstract

This paper presents the application of a kernel particle filter for 3D body tracking in a video stream acquired from a single uncalibrated camera. Using intensity-based and color-based cues as well as an articulated 3D body model with shape represented by cylinders, a real-time body tracking in monocular cluttered image sequences has been realized. The algorithm runs at 7.5 Hz on a laptop computer and tracks the upper body of a human with two arms. First experimental results show that the proposed approach has good tracking as well as recovering capabilities despite using a small number of particles. The approach is intended for use on a mobile robot to improve human robot interaction.

1 Introduction

A large research area within computer vision is concerned with tracking humans. A rather coarse model of the human body is usually adopted in surveillance applications or for recognizing large-scale activities. Obviously, the recognition of small-scale activities that are related to movements of individual body parts requires a finer human body model. For example, the ability of robot companions to recognize hand gestures is of crucial importance for human-robot-interaction as it allows humans to interact with the robot in a more natural way.

Our goal is to realize human-robot interaction using a mobile robot with limited computational power. This limitation and the need to track in real-time the body configurations of humans with arbitrary clothing rules out approaches using stereo cameras. Such approaches require image disparities to successfully carry out the computationally expensive depth calculation (e.g., [12]). A different approach is taken in model-based tracking methods performing a probabilistic integration of image cues (see, e.g., [16, 18, 4, 17, 13, 9]). However, extracting 3D body configurations from 2D image data is connected with complicated modeling as well as feature extraction difficulties. Matching a complex self-occluding model to a cluttered scene is an inherently difficult task and depending on the number of image cues and their discrimination capabilities it is only feasible with multiple cameras [2].

In our approach, we focus on estimating in real-time the pose of the upper body on the basis of a 3D model of the human body and monocular uncalibrated video. Motions of body segments in depth, towards or away from the camera, cannot be tracked precisely with a monocular camera, but as long as tracking is not lost, a very rough estimate of the body configuration is still available. Different from other approaches we use a larger number of cues. Their combination produces results that are less dependent on the background, enabling tracking from onboard a mobile robot. A skin color model is used for detecting a person's hands and face. For the detection of the limbs we utilize edge, ridge, and color cues.

Tracking the 3D model over time is realized using a kernel particle filter which avoids the need for a huge number of particles to represent probability distributions in high dimensional state space. On the basis of mean shift based mode-seeking the dominant mode is determined and then used to select the particles for the next time step. A linear motion model is used for propagating a fraction of the particles while uniform random noise is used for the other part. Special emphasis is placed on the scalability of the framework to enable the real-time estimation of a 3D body configuration. First experimental results demonstrate the suitability of our approach to support human-robot interaction using a mobile robot equipped with a standard laptop.

The paper is organized as follows: Section 2 gives an overview of related work and Section 3 gives a system overview. The 3D model and the image processing are described in Section 4. The next section describes kernel

In Proc. FG2006, pages 567 – 572, Southampton, UK, 2006

based particle filtering. The configuration of the overall framework to enable real-time tracking and the results obtained are the topic of Section 6. The paper concludes with a summary in Section 7.

2 Related Work

Tracking of a human in 3D with limited computational resources on a mobile robot was already described in 1996 by Kortenkamp et al. [10]. This approach used depth information from a stereo camera to track the 3D body model. In more recent approaches, skin color is often used as an additional cue to get the 3D hand position and its pointing direction (e.g., [12]).

As the depth calculation from stereo images causes high computational costs and relies on the presence of image disparities, the use of monocular cameras as basis for estimating the 3D body configuration is a promising alternative. Often multiple cameras are used in order to cope with body self-occlusions (see, e.g., [4, 17, 14, 13, 9]). While some of these approaches use a body model and aim to find matching features in the image [4, 9] other approaches are taking an opposite direction and construct in a probabilistic way a body model out of body parts detected in the image [17, 13]. However, all approaches are computationally intensive prohibiting their use for human-machine interaction.

The use of multiple cameras in a task consisting in tracking the human body from a mobile robot is technically difficult. Only few authors have addressed the problem of 3D full-body tracking using a single uncalibrated camera [16, 15, 18]. One such approach for tracking a detailed 3D human body model was proposed by Sidenbladh [16, 15]. It is based on a variety of gray-level image cues and a particle filter for tracking the human motions. To cope with the huge search space, motion priors are used to predict the 3D body configuration prohibiting the tracking of unconstrained motions. Following Sidenbladh's work, Sminchisescu used a more precise modeling of the 3D body model and a complex parameter space exploration [18], but the computational time required prohibits its use for real-time tracking. To cope with a large parameter space, kernel-based Bayesian filtering has been proposed to track objects or isolated body parts in the 2D image space (e.g., [7, 3]).

3 System Overview

An overview of our algorithm for matching 3D object features of a generic human model and 2D image features extracted from input images is depicted in Fig. 1. The algorithm has been inspired by Sidenbladh's work [15] and uses similar edge and ridge cues. In addition to these cues, we developed color-based cues to support the detection of the face, hand, and clothing. Different from Sidenbladh's approach [15], we calculate weighted limb-specific probabilities and use a kernel particle filter for probabilistic tracking.

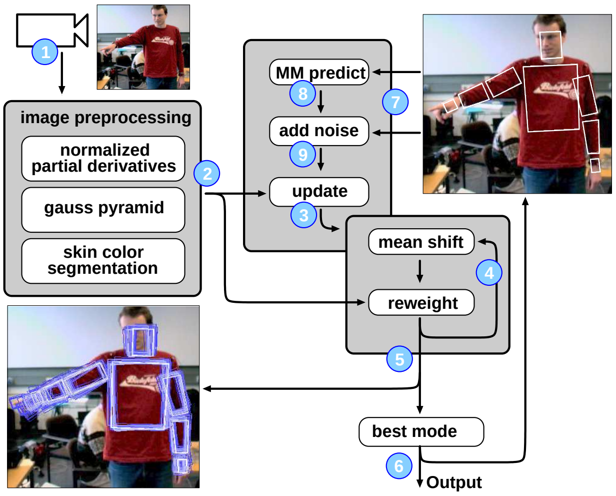

Figure 1. Outline of the algorithm.

One iteration of our algorithm can be described as follows: Input images are acquired with an uncalibrated monocular color camera (1) and preprocessed. The results (2) are used for matching configurations of the body model with the current image. The outcome (3) is a probability distribution which is further explored in multiple iterations (4) of the mean shift algorithm. That leads to the identification of different modes (5) of the underlying probability distribution from which a single mode (6) representing the most likely human body pose is selected as output for subsequent recognition algorithms. This mode is also used as input (7) for the next time step of the particle filter. Particles generated from this mode are then - partly after applying a motion model (8) - disturbed (9) to give an estimation of body configurations for the next time step. Finally, the next picture is acquired (1), preprocessed (2), and the propagated configurations are evaluated (3) on the new image.

4 Modeling the Appearance of Humans

We use an articulated 3D body model consisting of cylinders with ellipsoid cross sections. This representation generates the best results when the cylinder is observed by the camera from the side. The kinematic structure is completed by defining individual joint angle limits which model the physical constraints of the human body. Our model of the upper body with two arms has 14 degrees of freedom, whereas a coarse model of the whole body needs at least 34 degrees of freedom.

The 3D body model is back-projected into the image plane using a pinhole-camera model. This yields an approximate representation of the 3D body model consisting of a 2D polygon in the image plane for each limb. The estimation of a single pose is done by combining likelihoods for each limb $l = 1, \ldots, L$ of the body model using up to four image cues $c \in \{E, R, C, S\}$, where E stands for the edge cue, R is the ridge cue, C is the mean color cue and S denotes the skin color cue. The image processing is briefly discussed in the following:

1) Edge Cue: The edge cue [15] uses the first partial derivatives that are sensitive to strong changes in contrast. For recognizing human body parts the presence of edges is most important, not their magnitude. Therefore all images with partial derivatives have been scaled with a nonlinear normalization function. This function has been used to smooth low magnitude edges stemming from textured backgrounds and to emphasize stronger ones. The edge cue provides an accurate match for the position of a limb by comparing the angle of the edge gradient $[\partial_x(\mathbf{z}),\ \partial_y(\mathbf{z})]^T$ with the estimated limb angle α, which has been obtained from the 3D model. This is done for $m = 1, \ldots, M_E$ feature points $\mathbf{z}^{(m)}$ positioned equally spaced on the limb boundaries, where $\mathbf{z}^{(m)} = [x, y]^T$ denotes the location of one pixel in the image plane. The response of such an edge filter is:

$$f_E^{(l)}(\mathbf{z}^{(m)}) = \partial_y(\mathbf{z}^{(m)})c(\alpha) - \partial_x(\mathbf{z}^{(m)})s(\alpha), \qquad (1)$$

where $s(\alpha)$ stands for $\sin(\alpha)$ and $c(\alpha)$ stands for $\cos(\alpha)$. The filter response for the whole limb is calculated by averaging over the M_E feature points:

$$\bar{f}_E^{(l)} = \frac{1}{M_E} \sum_{m=1}^{M_E} f_E^{(l)}(\mathbf{z}^{(m)}). \qquad (2)$$

2) Ridge Cue: The ridge cue [15] is utilized to find elongated structures of a specified thickness in the image. As it depends on the size of the limbs in the image, it will only provide appropriate results if the observed limb is in a particular distance to the camera. Consequently, the correct resolution level μ in the Gaussian image pyramid is selected based on the current distance of the limb to the camera. The cue suppresses point-like edge features by searching for edges parallel to the expected limb angle α and missing edges in perpendicular direction. This is achieved by evaluating the normalized second partial derivatives $[\partial_{xx}^{(\mu)}(\mathbf{z}), \partial_{xy}^{(\mu)}(\mathbf{z}), \partial_{yy}^{(\mu)}(\mathbf{z})]^T$ at $m = 1, \ldots, M_R$ feature points $\mathbf{z}^{(m)}$ equally distributed on the main limb axis (superscript m is omitted):

$$\begin{aligned} f_R^{(l)}(\mathbf{z}, \alpha) = \\ \left| s^2(\alpha)\, \partial_{xx}^{(\mu)}(\mathbf{z}) + c^2(\alpha)\, \partial_{yy}^{(\mu)}(\mathbf{z}) - 2sc(\alpha)\, \partial_{xy}^{(\mu)}(\mathbf{z}) \right| \quad - \\ \left| c^2(\alpha)\, \partial_{xx}^{(\mu)}(\mathbf{z}) + s^2(\alpha)\, \partial_{yy}^{(\mu)}(\mathbf{z}) + 2sc(\alpha)\, \partial_{xy}^{(\mu)}(\mathbf{z}) \right| \qquad (3) \end{aligned}$$

where $sc(\alpha)$ stands for $\sin(\alpha)\cos(\alpha)$. The filter response for the whole limb l is computed by averaging over all M_R feature points:

$$\bar{f}_R^{(l)} = \frac{1}{M_R} \sum_{m=1}^{M_R} f_R^{(l)}(\mathbf{z}^{(m)}, \alpha). \qquad (4)$$

The ridge cue gives a coarser estimate of the limb position than the edge cue, but produces less false maxima.

3) Mean Color Cue: The mean color cue models the appearance of limbs using an adapted color model for each limb. The algorithm uses B_l polygons on each limb l. The mean color value is calculated trough averaging over the color values of the M_C pixels in the polygon b positioned in the back-projected limb. The number of polygons B_l and their positions are chosen on the basis of the limb type. To calculate the filter response, the mean color $C_t(\mathbf{z}^{(b,l)})$ of each polygon b at position $\mathbf{z}^{(b,l)}$ is compared to the adapted mean color $\bar{C}_{t-1}^{(b,l)}$ of this polygon on limb l using the L2 norm in the utilized RGB color space:

$$f_C^{(b,l)} = \sqrt{\left(C_t(\mathbf{z}^{(b,l)})\ -\ \bar{C}_{t-1}^{(b,l)}\right)^2}. \qquad (5)$$

The filter response for the limb l is calculated from:

$$\bar{f}_C^{(l)} = \frac{1}{B_l} \sum_{b=1}^{B_l} f_C^{(b,l)}. \qquad (6)$$

To deal with varying illumination conditions we adapt the current mean color values according to the mean color values $\hat{C}_{t-1}(\mathbf{z}^{(b,l)})$ extracted on the basis of the back-projection of the best mode in the last time-step $t-1$:

$$\bar{C}_{t-1}^{(b,l)} = \beta \cdot \hat{C}_{t-1}(\mathbf{z}^{(b,l)}) + (1-\beta) \cdot \bar{C}_{t-2}^{(b,l)} \qquad (7)$$

where β is an adaptation factor. This cue reliably finds the coarse limb position as color is a very discriminative cue.

4) Skin Color Cue: The skin color cue uses a skin color segmentation in rg color space providing a binary segmentation image [5]. The skin color model allows to find the position of the hands and the head. Similar to the mean color cue, all M_S pixels $\mathbf{z}^{(m)}$ in a polygon on the limb are analyzed. The filter response for the limb l is calculated using the ratio of pixels being classified as skin or non-skin:

$$\bar{f}_S^{(l)} = \frac{1}{M_S} \sum_{m=1}^{M_S} \psi(\mathbf{z}^{(m)}) \qquad (8)$$

where $\psi(\mathbf{z}^{(m)}) = 1$ if the pixel belongs to the skin class and $\psi(\mathbf{z}^{(m)}) = 0$ if not. In varying illumination conditions the skin color model can be adapted over time.

The edge, ridge, mean color, and skin color cues generate a separate filter response for each limb. The filter responses are converted into likelihoods using the following Gaussian weighting function:

$$p(c,l) = \exp\left(-\frac{(\bar{f}_c^{(l)})^2}{2\,\sigma_c^2}\right) \tag{9}$$

where the standard deviations σ_c are derived from the variability of the responses of each utilized cue c. To account for the variations in the number of cues per limb N_l the cue likelihoods are scaled according to a balancing factor. Assuming that the cues and limbs are independent the overall likelihood for the pose is calculated as:

$$p(\mathbf{y}_t \mid \mathbf{x}_t) = \prod_{c\in\{E,R,C,S\}} \prod_{l=1}^{L} p(c,l)^{\frac{1}{N_l}} \tag{10}$$

This observation model describes the likelihood that a body state $\mathbf{x}_t$ causes the observation $\mathbf{y}_t$.

5 Probabilistic tracking

The idea of using the mean shift mode seeking within a particle filter is relatively new and until now it has mainly been utilized in experiments consisting of tracking objects or isolated body parts in the 2D image space (e.g., [7, 3]). In this section we briefly introduce the kernel particle filter and demonstrate how it is applied for tracking in the high-dimensional space associated with 3D human modeling.

5.1 Standard Particle Filtering

The particle filter (PF) is a probabilistic framework to propagate the conditional density to the next step [1]. Given in the d-dimensional space $\mathbb{R}^d$ a particle set $\mathbf{S}_{t-1} = \{\mathbf{s}_{t-1}^{(n)}\}_{n=1}^{N}$ and the associated weights $\{w_{t-1}^{(n)}\}_{n=1}^{N}$, which are approximately distributed according to $p(\mathbf{x}_{t-1} \mid \mathbf{Y}_{t-1})$, where $\mathbf{Y}_{t-1} = \{\mathbf{y}_0, ..., \mathbf{y}_{t-1}\}$ is the history of observations up to time $t-1$, the PF operates through predicting new particles over time. To give a new particle representation $\{\mathbf{s}_t^{(n)}, w_t^{(n)}\}_{n=1}^{N}$ of the posterior density $p(\mathbf{x}_t \mid \mathbf{Y}_t)$ the weights of the particles are set to $w_t^{(n)} = p(\mathbf{y}_t \mid \mathbf{s}_t^{(n)})$ with w normalized to $\sum_{n=1}^{N} w_t^{(n)} = 1$. In the specific variant of sequential importance sampling known also as CONDENSATION [8], a re-sampling step using the new weights is applied to avoid degeneration of the particle based representation. In high-dimensional search space, even if a particle escapes out of a local minimum, the probability of hitting the low-weight surroundings is much larger than that of hitting a region with high weights. This is due to the huge increase of volume with radius that results in the need for a large number of particles which in turn increases computation time.

5.2 Kernel Particle Filtering Using Mean Shift

In order to perform tracking with a small number of particles, an iterative mode-seeking in the form of the mean-shift algorithm is applied to shift the particles to high weight areas. For this purpose, the true density distribution is estimated through placing a kernel function on each sample. The estimate of the posterior distribution $p(\mathbf{x}_t \mid \mathbf{Y}_t)$ with kernel K can be formulated as follows:

$$\hat{p}(\mathbf{x}_t \mid \mathbf{Y}_t) = \sum_{n=1}^{N} K_h(\mathbf{x}_t - \mathbf{s}_t^{(n)}) w_t^{(n)} \tag{11}$$

where $K_h(\mathbf{x}_t - \mathbf{s}_t^{(n)}) = \frac{1}{Nh^d} K(\frac{\mathbf{x}_t - \mathbf{s}_t^{(n)}}{h})$, and h is the kernel bandwidth. For a radially symmetric kernel we have $K(\mathbf{x}_t - \mathbf{s}_t^{(n)}) = ck(\|\mathbf{x}_t - \mathbf{s}_t\|)$, where c is a normalization constant which makes the integral $K(\mathbf{x}_t - \mathbf{s}_t^{(n)})$ to one, and $k(r) = k(\|\mathbf{x}_t - \mathbf{s}_t\|)$ is called the profile of the kernel K. In our application we use the Epanechnikov kernel:

$$K_E(\mathbf{x}) = \begin{cases} \frac{1}{2} c_d^{-1}(d+2)(1 - \|\mathbf{x}\|^2) & 0 \le \|\mathbf{x}\| \le 1 \\ 0 & \|\mathbf{x}\| > 1 \end{cases} \tag{12}$$

Given a particle set $\mathbf{S}_t$ and the associated weights $\{w_t^{(n)}\}_{n=1}^{N}$, the particle mean is determined by

$$m(\mathbf{s}_t^{(n)}) = \frac{\sum_{i=1}^{N} H_h(\mathbf{s}_t^{(n)} - \mathbf{s}_t^{(i)}) w_t^{(i)} \mathbf{s}_t^{(i)}}{\sum_{i=1}^{N} H_h(\mathbf{s}_t^{(n)} - \mathbf{s}_t^{(i)}) w_t^{(i)}} \tag{13}$$

where $h(r) = -k'(r)$ is in turn a profile of kernel H_h. It can be shown that the mean shift vector $m(\mathbf{x}) - \mathbf{x}$ always points toward steepest ascent direction of the density function [3]. Following the shifting of particles using the mean shift vector, the particle weights $w_t^{(n)}$ are recomputed. Additionally, a reweighting is performed to guarantee that each mean shift iteration follows the correct posterior gradient [3].

The choice of the kernel bandwidth h is of crucial importance in kernel based density estimation and is usually scaled down at each mean shift iteration in order to concentrate on the most dominant modes. In our implementation, the initial bandwidth h_0 is scaled at every iteration i according to $h = 0.8^i\, h_0$ where the value 0.8 has been determined empirically, similar to [3].

The mode-seeking continues until a maximum number of iterations has been reached or until the Euclidean distance between the corresponding modes in the last two iterations is below an empirically determined threshold. Following mode-seeking, the most dominant mode is obtained by a weighted averaging over all particles in a window centered at the peak of the posterior. This mode serves as estimate of the current body pose, see also Fig. 1. The back-projection of this pose into the image plane is utilized as reference model for updating the mean color using Eq. 7.

For propagating the particles from the dominant mode to the next time step we combine two different strategies: A linear motion model is used for propagating a fraction of all particles, the remaining particles are subject to random propagation. For each of the two strategies the cumulated probabilities of the particles in the posterior are calculated to decide on the ratio of particles for the next time step. A minimum percentage of random propagation is enforced to guarantee the ability to recover from tracking failures. The velocity in the linear motion model is estimated on the basis of the location of the best mode in time t and its corresponding location in time $t-1$. Particles propagated with the motion model are subject to uniform noise with variance $0.25\,D$ while uniform noise with variance D is used for the rest of the particles. The values of the joint angle variances D have been determined experimentally based on the camera view and the application domain. If the propagation results in an invalid body configuration, the propagation step is repeated until a valid body pose is obtained. For the mean shift iterations, the initial bandwidth is chosen as $h_0 = D$, the window for determining the most dominant mode is set to $0.25\,D$, which has been determined empirically.

6 Results

For the intended application of recognizing pointing gestures with a camera mounted on a mobile robot we track only the upper body and the two arms. In our current body model the hands are fixed to the lower arms and the head is fixed to the torso leading to a model with 14 degrees of freedom. This reduction of the model complexity is acceptable for tracking a human that is oriented roughly towards the camera and interacts with a mobile robot.

As cues we use $c_{torso} = \{E, C\}$, $c_{upperarm} = \{E, R, C\}$, $c_{lowerarm} = \{E, R\}$ and $c_{hand} = c_{head} = \{S\}$ with $M_E = 20$ and $M_R = 30$. To calculate the mean color cue we use $B_{arm} = 3$ and $B_{torso} = 4$ polygons. We set $D_{arm} = [22°, 20°, 35°, 25°]$ for the three shoulder joint angles and the elbow and D_{torso} between $1°$-$3°$ for rotation and $1cm$-$5cm$ for translation. These parameters have been found to be an acceptable tradeoff between detection capabilities and computational load.

Currently the body model has to be initialized manually before starting the tracking. No occlusion test is performed to speed up processing. Matching difficulties can arise if the hands occlude the head or point in the direction of the camera. In such situations tracking may temporarily fail. However, the recovering capabilities of the particle filter allow the algorithm to continue the broken tracking when the occlusion ends.

The algorithm was tested on a Sony Vaio VGN-S5HP (Pentium M 740, 1.73 GHz) with an SONY DFW VL500 FireWire camera acquiring images of size 320×240. A configuration with 150 particles and 2 mean shift iterations provides an estimate of the 3D hand position at a framerate of 7.5 Hz. Figure 2 shows an example 3D hand trajectory of a human performing pointing towards an object of interest. The rather coarse estimate of the 3D hand position due to the small image resolution is sufficient for tracking a pointing gesture referencing an object in the scene.

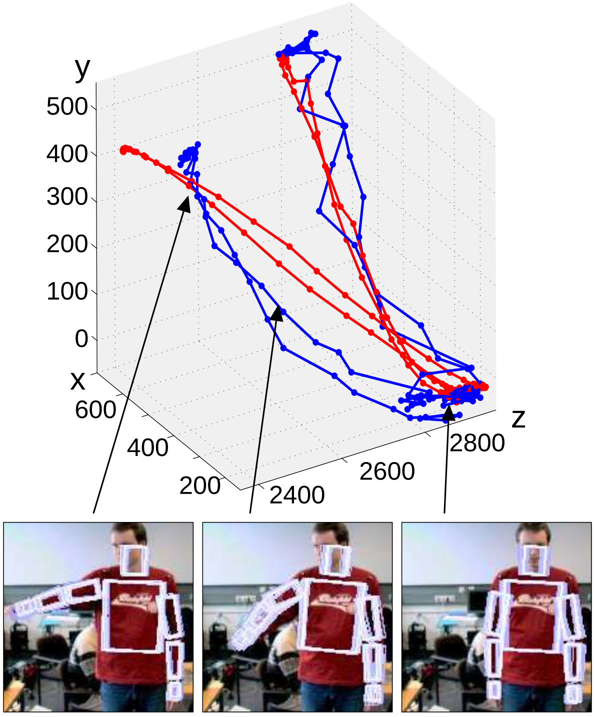

Figure 2. 3D hand trajectory (blue), ground truth data (red) and corresponding images

With the configuration described above, we carried out experiments using an image sequence recorded at 7.5 Hz with 330 frames of size 320×240. A human user slowly performed a total of six pointing gestures with the right hand: three to the side and three to an object in front of the human at roughly $45°$. Ground truth was obtained from a marker-based Lukotronic AS 200 motion capturing system [11] by placing active markers on the shoulders, the elbows, and on each wrist ensuring good visibility during pointing. The marker-based tracking data was manually aligned to the image sequence and interpolated to account for the different recording frequencies. For each image the 'correct' body pose represented by the markers was used as the reference to calculate the reconstruction error during monocular tracking. For the evaluation we used the ground truth wrist position of the right arm. This is compared to the position of the wrist joint connected to the lower arm limb of the body model, as our focus is on tracking pointing gestures and not on the exact arm configuration.

After manually fitting the body model to the camera image monocular tracking is started. The reconstruction error for one frame is calculated as Euclidean difference between the marker-based wrist position and the end of the lower arm. The overall tracking error is the cumulated error normalized by the number of frames. We evaluated the tracking quality of the probabilistic approach by averaging over 5 runs for each configuration. The mean and variance listed in Table 1 demonstrate that the tracking quality improves with the number of particles used. For all configurations tracking of lateral movements is possible and a rough estimate of the hand position is available. For pointing in an angular direction, tracking works well only if at least 250 particles are used. In order to enable real-time tracking at 7.5 Hz, we used only 2 iterations to maximize the number of particles covering the 14 DOF parameter space. However, with 150 particles tracking non-lateral movements turned out to be not robust. Ongoing work aims at improving the algorithm to increase the number of particles.

		Side pointing		Angular pointing	
		Mean	Var.	Mean	Var.
# Part.	# Iter.	[mm]	[mm]	[mm]	[mm]
1000	3	77.5	42.1	124.2	46.5
500	3	75.9	46.7	141.4	60.8
250	3	94.1	55.7	156.6	77.1
150	2	116.0	70.6	253.8	216.0

Table 1. Mean and variance of tracking error.

7 Summary

We presented a probabilistic framework which utilizes a variety of image cues to track a human in 3D in a videostream acquired from a single uncalibrated camera. Robust tracking is achieved by the use of color cues that are combined with ridge and edge cues. For evaluation a monocular image sequence depicting pointing gestures has been recorded together with marker-based ground truth. The results demonstrate that the kernel particle filter allows tracking of human motions in cluttered environments as long as no large self-occlusions occur.

Using 150 particles and only 2 iterations, the algorithm runs at 7.5 Hz and can be used for coarse tracking of a gesturing human. Consequently, with a standard camera and a laptop computer mounted onboard a mobile robot, our approach can be used for gesture recognition to improve human-robot interaction. We currently aim at improving the robustness and speed to use this approach for resolving multi-modal object references [6] given by a human interacting with a mobile robot through speech and gesture.

References

[1] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking. *IEEE Trans. on Signal Processing*, 50(2):174–188, 2002.

[2] A. O. Balan, L. Sigal, and M. J. Black. A quantitative evaluation of video-based 3D person tracking. In *Int. Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, 2005.

[3] C. Chang and R. Ansari. Kernel particle filter for visual tracking. *Signal Processing Letters*, 12(3):242–245, 2005.

[4] J. Deutscher and I. Reid. Articulated body motion capture by stochastic search. *Int. J. Comput. Vision*, 61(2):185–205, 2005.

[5] J. Fritsch, S. Lang, M. Kleinehagenbrock, G. A. Fink, and G. Sagerer. Improving adaptive skin color segmentation by incorporating results from face detection. In *Int. Workshop on Robot and Human Interactive Communication (RO-MAN)*, pages 337–343, 2002.

[6] A. Haasch, N. Hofemann, J. Fritsch, and G. Sagerer. A multi-modal object attention system for a mobile robot. In *Int. Conf. on Intelligent Robots and Systems*, pages 1499–1504, August 2005.

[7] B. Han, Y. Zhu, D. Comaniciu, and L. Davis. Kernel-based bayesian filtering for object tracking. In *Int. Conf. on Computer Vision and Patt. Recognition*, pages 227–234, 2005.

[8] M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. In *Europ. Conf. on Computer Vision*, pages 343–356, 1996.

[9] R. Kehl, M. Bray, and L. V. Gool. Markerless full body tracking by integrating multiple cues. In *ICCV Workshop on Modeling People and Human Interaction*, 2005.

[10] D. Kortenkamp, E. Huber, and R. P. Bonasso. Recognizing and interpreting gestures on a mobile robot. In *Nat. Conf. on Artificial Intelligence*, pages 915–921, 1996.

[11] Lukotronic. AS 200. `http://www.lukotronic.com`.

[12] K. Nickel, E. Seemann, and R. Stiefelhagen. 3D-Tracking of Heads and Hands for Pointing Gesture Recognition in a Human-Robot Interaction Scenario. In *Int. Conf. on Face and Gesture Recognition*, 2004.

[13] D. Ramanan and D. A. Forsyth. Finding and tracking people from the bottom up. In *Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 467–474, 2003.

[14] R. Rosales, M. Siddiqui, J. Alon, and S. Sclaroff. Estimating 3D body pose using uncalibrated cameras. In *Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 821–827, 2001.

[15] H. Sidenbladh. *Probabilistic Tracking and Reconstruction of 3D Human Motion in Monocular Video Sequences*. PhD thesis, KTH Sweden, 2001.

[16] H. Sidenbladh, M. Black, and D. Fleet. Stochastic tracking of 3D human figures using 2D image motion. In *Europ. Conf. on Computer Vision*, pages 702–718, 2000.

[17] L. Sigal, S. Bhatia, S. Roth, M. J. Black, and M. Isard. Tracking loose-limbed people. In *Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 421–428, 2004.

[18] C. Sminchisescu and B. Triggs. Mapping minima and transitions of visual models. *Int. J. of Computer Vision*, 61(1), 2005.