Automated Pose Estimation in 3D Point Clouds Applying Annealing Particle Filters and Inverse Kinematics on a GPU

Nicolas H. Lehment, Dejan Arsić, Moritz Kaiser, Gerhard Rigoll
Institute for Human Machine Communication
Technische Universität München
Arcisstr. 21, 80634 München, Germany
lehment, arsic, rigoll @ tum.de

Abstract

Current experiments with HCIs have shown a high demand for more natural interaction paradigms. Gestures are thereby considered the most important cue besides speech. In order to recognize gestures it is necessary to extract meaningful motion features from the body. Up to now mostly marker based tracking systems are used in virtual reality environments, since these were traditionally more reliable than purely image based detection methods. However, markers tend to be distracting and cumbersome. Following recent advances in processing power, it becomes possible to use a camera system in order to obtain a depth image of the test subject, match it to a pre-defined body model, and thus track the body parts over time. We will present a full-body system based on APF which enables full body tracking utilizing point clouds recorded with a 3D sensor. Further refinement is provided by a specially adapted inverse kinematics system. A GPU based implementation speeds up processing significantly and allows near real time performance.

1. Introduction

Over the last years, methods for interacting with computers have continuously moved towards more natural forms of communication. Following keyboard and mouse interfaces, recent trends point towards more accessible and intuitive interaction methods. Especially in the context of virtual reality (VR) and augmented reality (AR), natural methods of interaction with the virtual environment are desirable and more pleasant. Using motion patterns and gestures familiar from everyday use allows users to adapt quickly to new applications and lowers the learning curve for handling complex systems.

Currently such interactions are mostly achieved by tracking markers located on the person's limbs or hand-held controllers, followed by an analysis of the extracted trajectories. This procedure usually requires time intensive preparation and attaching active or inactive markers to the person tracked. In order to provide a more comfortable and realistic experience, it is desirable to avoid such markers by using a simple camera instead. Many of the HMI systems presented to date rely the on analysis of monocular images and are still limited by self-occlusions or ambiguities. While there are systems using multiple viewing angles to avoid such problems, these are usually unfit for everyday use outside of a laboratory. In order to avoid these drawbacks, we use depth images provided by a stereo camera to capture the tracked person. Analyzing the extracted point clouds, we fit a body prototype to the data using an annealing particle filter (APF) approach. This model-based approach avoids most self-occlusion conflicts and ambiguities, without requiring pre-learned motions or dedicated detection of single body parts.

Stochastic methods like the APF are usually more robust than deterministic methods ([9]) and have the additional benefit of effectively utilizing modern parallel hardware, either on a single computer or spread over several machines. Annealing particle filters have already been utilized in various pose tracking applications, frequently with multiple camera angles ([5],[6],[8]) and have been shown capable to handle depth information when using additional cues: Azad ([2]) and Bernier ([3]) used skin cues as anchor points for the wrists, Darby ([4]) relied on pre-learned motion patterns.

Whereas most previous works in the field of pose tracking relied on silhouettes, edges and specific anchor points like hands or heads, our approach aims to be independent from the source of the 3D point cloud. This practically means using only 3D data without any color or texture information, since a number of devices operate by time-of-flight (TOF) measurements, such as a photonic mixture device [1] or other non-visual methods. Although this leads to a loss of

978-1-4244-7028-0/10/$26.00 ©2010 IEEE

precision and jitter, it is sufficient for gesture recognition modules and allows for better interoperability with devices ranging from stereo-camera systems to TOF cameras. To compensate for the absence of a dedicated hand tracker, we developed a self-adapting inverse kinematics system for improved fitting of arms and legs. First trials were performed using depth data obtained from a stereo camera, as experience has shown a lack of detail with current PMD sensors.

We will demonstrate that it is possible to perform full-body APF based tracking using only depth-data in near real time by intelligently reducing the number of particles required and exploiting modern parallel computer hardware.

2. Acquisition of 3D Data

As the purpose of our approach was to create a versatile pose tracker for a number of different data sources, we aimed to become independent from specific products or systems. So while we used a proprietary system for image acquisition and the calculation of 3D data, all further processing steps are designed to be compatible with depth data from any source.

Stereo images were captured using a Point Grey Bumblebee XB3 camera. Before calculating the 3D data, the foreground was extracted using a gaussian mixture model as described in [10]. From the masked image, the tracked person's 3D cloud was then calculated with the Point Grey Triclops library. The data were then reformatted into an array of 3D points and passed on to the pose estimator.

3. Matching to 3D point cloud

3.1. Annealing Particle Filter

Particle filters are a powerful, stochastic method for exploring high-dimensional solution spaces and variations have been used in a number of tracking scenarios. Compared to deterministic methods, which had been the mainstay of tracking research until recently, they show superior robustness and precision while enabling full use of modern parallel hardware. Based on Monte-Carlo methods, particle filters rely on random sampling to approximate solution space PDFs without requiring exact knowledge of the underlying functions. This is especially advantageous in tracking scenarios, where there is often only limited knowledge of the observed scene. Usually, an initial set of particles X_t, each representing a set of parameters, is matched against data $\mathbf{Z}_t$ from observations. The particle set can be formalized as follows:

$$\mathbf{x} = [\mathbf{T}_{\text{base}}, \mathbf{R}_{\text{base}}, \theta_1, \ldots, \theta_{DoF}] \tag{1}$$

$$\mathbf{X}_t = [\mathbf{x}^{(0)}, \ldots, \mathbf{x}^{(X)}] \tag{2}$$

$$\mathbf{X}_t^{\pi} = [(\mathbf{x}^{(0)}, \pi^{(0)}), \ldots, (\mathbf{x}^{(X)}, \pi^{(X)})] \tag{3}$$

Based on the quality of the matching a score $w(\mathbf{x}^{(i)}, \mathbf{Z}) = \pi^{(i)}$, also called weight, is assigned to each particle $x^{(i)}$. The particles are then resampled based on their weight and yield a new particle set which can be used to estimate a solution. After mutating the parameters in the new set, the resulting set X_{t+1} can then be used for the next observation.

The annealing particle filter concept is a variation on common particle filters described in great detail by Deutscher ([5]) and Gall ([7]). The basic idea is to replace the single weighting step of the particle filter by several gradual steps (the simulated annealing), thereby achieving a better exploration of the configuration space and avoiding local maxima of the weighting function. This approach, as all particle filters, allows for an easy parallelization and has the added advantage of requiring fewer particles.

3.2. Designing The Weighting Function

While the conceptual framework of the annealing particle filter can be applied to a number of different problems, the specific weighting function for the individual particles needs to be tailored closely to the application. Since we are working solely with a cloud of 3D data points signifying the detected surface of the tracked person, we aim to minimize the difference between the detected cloud and the mesh of a cylinder-based stickman model. The full-body model used consists of 15 cylinders, with a mesh of 4 x 5 points, projected onto the visible side, and currently allows for 30 degrees of freedom. An exemplary stickman is therefore represented by a total of 300 3D points ($\mathbf{M}$) which have to be matched against the cloud of 3D data points ($\mathbf{Z}$).

To achieve best matching, we use three separate weighting criteria: Matching of skeleton points against the point cloud, matching cylinder edges against the point cloud and finally reverse-matching the point cloud against the skeleton points. In the following we will briefly explain the reasoning and method behind the weighting process. Matching the 3D cylinder points against the 3D point cloud aims at minimizing the euclidean distance between each cylinder point and the closest cloud point. The first weight is therefore computed as:

$$w_{\text{M2Z}}{}' = \sum_{\mathbf{m} \in \mathbf{M}} e^{10 \, \min_c (\text{d}_{\text{eukl}}(\mathbf{m}, \mathbf{z}))}, \tag{4}$$

$$w_{\text{M2Z}} = e^{1.0 \, - \, w_{\text{M2Z}}{}'}. \tag{5}$$

Since the visible edges of the cylinders are especially sensitive to disalignment with the cloud point, we can use these for a more precise fitting. We therefore modify the skeleton-to-cloud scoring (4) to use only edge points $\mathbf{M}_e$ which leads to

$$w_{\text{E2Z}}{}' = \sum_{\mathbf{m}_e \in \mathbf{M}_e} e^{10\ \min_c(\text{d}_{\text{eukl}}(\mathbf{m}_e,\mathbf{z}))}, \tag{6}$$

$$w_{\text{E2Z}} = e^{1.0 - w_{\text{E2Z}}{}'}. \tag{7}$$

While these two weights give a good indication of how well a skeleton fits inside the point cloud, we also want to make sure that the point cloud is totally filled by the skeleton. Due to the size of the cloud, we segment it into smaller subsections for improved fitting of smaller regions $\mathbf{Z}_i$. The segmentation is achieved by k-means clustering of the point cloud. Outlying regions with only a few points, like hands and feet, therefore tend to have their own clusters, giving them equal weight to larger, more central regions like the torso. Thus, problems arising from the unequal distribution of data points over the body are largely avoided. To ensure that all parts of the point cloud are close to some part of the skeleton, we use the following relationship:

$$w^{(\text{i})}_{\text{Z2M}}{}' = \sum_{\mathbf{z} \in \mathbf{Z}_i} e^{10\ \min_s(\text{d}_{\text{eukl}}(\mathbf{z},\mathbf{m}))}, \tag{8}$$

$$w^{(\text{i})}_{\text{Z2M}} = e^{1.0 - w^{(\text{i})}_{\text{Z2M}}{}'}, \tag{9}$$

$$w_{\text{Z2M}} = \prod_{\text{i}} w^{(\text{i})}_{\text{Z2M}}. \tag{10}$$

Now we can multiply these weights to get the final score for a single particle:

$$w_{\text{score}} = w_{\text{Z2M}} \times w_{\text{E2Z}} \times w_{\text{M2Z}} \tag{11}$$

The different characteristics of the individual weighting functions are illustrated by a simplified 2 DoF example in fig. 1. The simplified example consists of an arm-torso model, in which the arm is spread slightly away from the torso, with the lower arm parallel to the body at 10 cm distance. It becomes obvious that the w_{Z2M} function leads to the best fitting, avoiding ambiguities arising from misplacement of limbs close to the torso (discernible as vertical ridges in the two other functions).

In addition to the point cloud based weighting, other mechanisms influence the particle score as well: Self-collision checks between limbs and joint limit checks are included in order to avoid illegal poses and can impose severe penalties on unwanted configurations. However, a full description of these would exceed the focus of this paper. Using the final score to judge the quality of a given pose particle, we can now use the framework of an APF to estimate the best pose fitting the observations.

3.3. Resampling of Particles

The weighted particle set $\mathbf{X}^{\pi}_{t,k}$ at timestep t, annealing step k, consists of $\mathbf{X}^{\pi}_{t,k} = [(\mathbf{x}^{(0)}, \pi^{(0)}), \dots, (\mathbf{x}^{(X)}, \pi^{(X)})]$. The particle $\mathbf{x}^{(i)}_{\text{t, k}}$ itself contains the encoded joint angles and basic transformations, while the weight, $\mathbf{w}^{(i)}_{\text{t, k}}$, is computed from the fitting of the particle to the observed data. Resampling was performed by stochastic universal sampling, generating $\mathbf{X}_{\text{t, 0}}$ from $\mathbf{X}_{\text{t-1, K}}$ with a wide random scattering over the legal range of the parameter

$$b^{(i)}_{\text{IF}} = \frac{1}{12}(x^{(i)}_{\text{max}} - x^{(i)}_{\text{min}})^2 \qquad \forall x^{(i)} \in \mathbf{x} \tag{12}$$

$$\mathbf{B} = \text{Diag}(\mathbf{b}_{\text{IF}}) \tag{13}$$

$$\mathbf{x}^{(n)}_{t,0} = \mathbf{x}^{(n)}_{t-1,K} + c_{\text{IF}}\mathbf{R}(\mathbf{0},\mathbf{B}) \qquad \forall \mathbf{x}^{(n)}_{t-1,K} \in \mathbf{X}^{\pi}_{t-1,K} \tag{14}$$

and $\mathbf{X}_{\text{t,k}}$ from $\mathbf{X}_{\text{t,k-1}}$ with a random scattering set proportional to the covariance of the parameters in the particle set.

$$\mathbf{B} = \text{Cov}(\mathbf{X}) \tag{15}$$

$$\mathbf{x}^{(n)}_{t,k} = \mathbf{x}^{(n)}_{t,k-1} + c_{\text{step}}\mathbf{R}(\mathbf{0},\mathbf{B}) \qquad \forall \mathbf{x}^{(n)}_{t,k-1} \in \mathbf{X}^{\pi}_{t,k-1} \tag{16}$$

The different resampling and scattering schemes are summarized in the algorithms 2 and 3.

Inverse Kinematics: The regular APF approach is well suited for fitting complex functions with independent variables. However, the human body consists of a number of kinematic chains, where the position of the wrist depends on four different joint angles. So in order to improve the matching of an arm with a low number of particles, it may not be sufficient to randomly mutate the angles. Instead, we use a simplified inverted kinematic chain to inject a number of modified particles in the resampling step, exploring alternative poses of the arm.

While such approaches usually rely on individual tracking of the hands as anchor points for the inverse kinematics as in [2], we instead utilize the wrist position from the last pose estimate. Although this is less accurate, it eliminates the need for a dedicated hand tracking and consequently speeds up processing. To generate the modified particles, we start with the last 'optimal' estimate. The elbow is rotated out of its current 'optimal' position and the resulting shoulder and elbow angles $\phi_{\text{S, x}}$, $\phi_{\text{S, y}}$, $\phi_{\text{S, z}}$, ϕ_{Elbow} are used to build a new particle (see algorithm 1.Without fixed and reliable anchor points such as separately tracked hands, the APF tends to curl up the arms close to the body. We counteract this effect by artificially extending the arms during the calculation of the inverse kinematics, as shown in fig. 2.

Static Particles: Nine basic arm poses were selected (arms forward, to the left/right and hanging down in all combinations) and are combined with particles from the current pop-

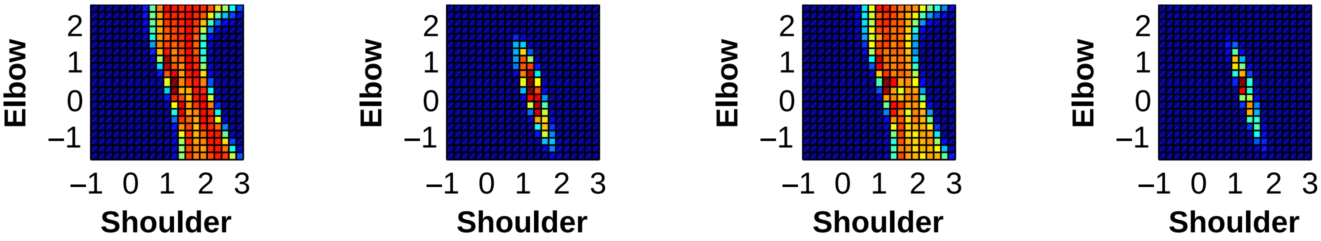

Figure 1. Components of the final weighting mechanism, from left to right: w_{M2Z}, w_{Z2M}, w_{E2Z}, $\pi_{t,m}^{(n)}$. All results normalized. Simplified arm-torso model with 2 DoF in Shoulder and Elbow.

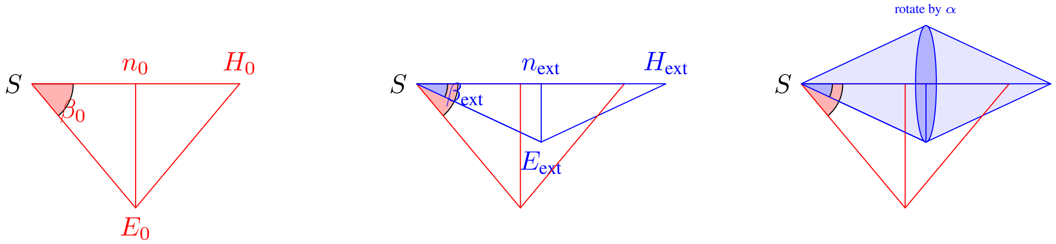

Figure 2. Artificially extending the elbow during calculation of the inverse kinematics.

ulation. The static particles are required to help initialization and recover the arms after a number of common tracking failures.

Randomizing: To counteract premature convergence on local optima, we insert about 10% of randomized particles. These are particles drawn by stochastic universal sampling from the existing population and then scattered by large random mutation ($\mathbf{R}_{\text{scatter}}$) of the joint angles. This allows for a more thorough exploration of the configuration space even as the regular particle set is converging.

3.4. Utilizing GPU and Multicore Processing

A typical skeleton, calculated from a single particle, consists of 300 points. Using 300 particles and assuming a 3D cloud of a thousand points, 90×10^6 point-to-point distances have to be calculated. While this would be prohibitive on a regular CPU, modern GPUs are perfectly fitted for just such a task. By exploiting the simple basic structure of the euclidean distance calculation and the weighting functions, we were able to run large parts of the weighting process on a regular NVIDIA Geforce GTX 275 graphics card. To this end, the euclidean distance calculation $d(\mathbf{m}, \mathbf{z})$, the search for the minimal distance $\min(d(\ldots))$ and the final weight calculation were implemented in CUDA as kernel functions.

To further improve performance, parts of the processing pipeline were parallelized into threads running on different cores of a multi-core CPU. While the APF is calculating the pose for a point cloud, the image processing is already preparing the point cloud for the next time step. This leads to an improved utilization of CPU und GPU resources.

4. Performance Evaluation

The algorithm has been tested on a 2.66 GHz Intel Core2 Quad CPU with 3 GB RAM and NVIDIA GTX 275 graphics card, achieving 2.5 fps (250 particles, 8 annealing steps). However, GPU memory management is not yet optimized and a number of functions, like pose calculation, k-means clustering and resampling, are still to be moved from the CPU to the GPU. We therefore expect further significant increases in processing speed, up to real time capability.

For testing, a set of 80 different movement sequences was analysed with four iterations each, giving 320 sequences. Four different persons performed a number of gestures and poses of varying complexity, ranging from simple waving to complexer poses like ducking, bowing or dragging virtual objects. At 300 particles and 8 annealing steps, we observed 80.6% successful tracks. Of these, 31.0% suffered brief lapses and were recovered successfully. By introducing 10% particles with inverse kinematics, the tracking error on hands and elbows was decreased by 12.74% compared to an otherwise identical tracking algorithm with no inverse kinematics. Some exemplary poses are shown in fig. 3 with an overlaid body model . However, the tracking experiences a fair amount of jitter. This is in part due to the missing anchor point tracking, but also caused by the rather basic cylinder based body model used.

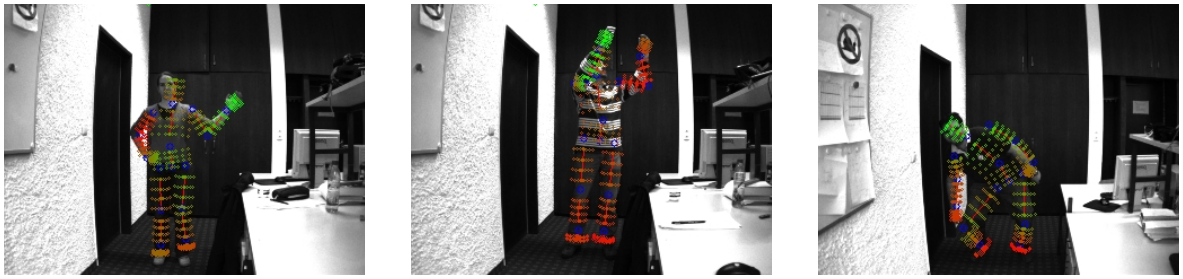

Figure 3. Different poses with the matched body prototype overlaid, full body and upper body only

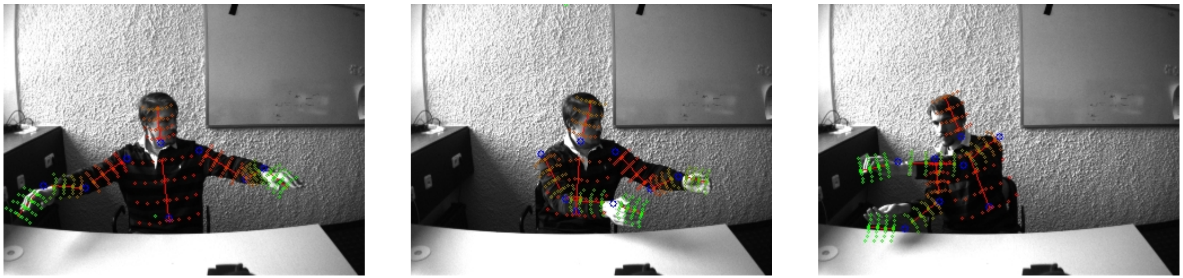

Figure 5. Examples of a quick adaption from full-body tracking to upper-body tracking.

To evaluate the effectiveness of the extending inverse kinematics injection, the relative error for several injection percentages was computed from manually labeled reference points and normalized to a range from 0.0 to 1.0 to gain the relative improvement. As shown in fig. 4, the relative limb error drops significantly with increased injection of modified particles. While one may suspect that this might also be due to more stable particles being artificially introduced into the particle set, the lack of improvement on the rest of the body shows that this is most likely not the case. If the introduction of particles derived from the optimal pose were to affect the tracking of the body in general, this would result in a significant drop for all other body parts as well. The fact that the rest of the body tracking remains virtually unaffected by the insertion of modified particles (the mean error varies only by $\Delta e_{Other} = 0.0016$ meters) shows that the arm tracking is indeed improved by the inverse kinematics and not some other effect.

We were able to convert the full body tracker to an upper body tracker without any modification of the algorithm itself simply by providing a modified body model (fig. 5). This underscores the flexibility gained from abandoning pre-learned motion models in favor of a purely depth-centered approach. However, this flexibility comes at the price of decreased stability and robustness. With the introduction of affordable and more precise TOF cameras as an alternative to stereo-based vision systems the conflict between flexibility (using only depth data) and precision (using additional cues such as skin-tracking) is expected to become more pronounced.

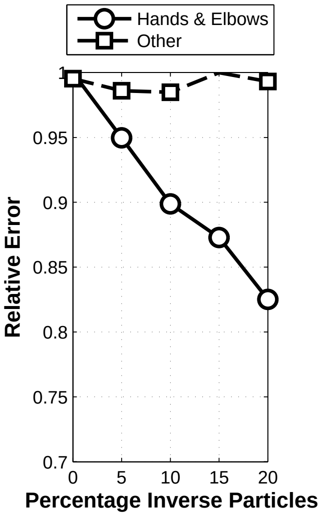

Figure 4. Relative error based on percentage of particles modified by inverse kinematics (0% - 20%), with 1.0 being the maximum error for each group. Mean error ranges in meter: $\Delta e_{Arms} = 0.0219$, $\Delta e_{Other} = 0.0016$.

5. Conclusion and Outlook

In this paper we have shown a purely depth data based approach for tracking full body movements in frontal view by APF. By reducing particles, exploiting parallelism and intelligently exploring the pose space we achieved near real time performance on off-the-shelf hardware. Furthermore an approach to inverse kinematics independent from typical anchor points such as hands was demonstrated.

For our future work, we expect to reach real time capability by moving further functions onto the GPU and optimizing memory management. With further refinement of the body prototypes and improvements and accelerations on the occlusion detection we expect significantly increased tracking precision. Ultimately we hope to build a versatile pose tracking module fit for fast re-adaption to different data sources and application scenarios.

Algorithm 1 Calculation of shoulder angles after elbow extension and rotation by α, all vectors in shoulder reference system

```
$\mathbf{a} = \mathbf{T}_{\mathrm{S}}^{\mathrm{E}_0}$
$\mathbf{b} = \mathbf{T}_{\mathrm{S}}^{\mathrm{H}_0} - \mathbf{T}_{\mathrm{S}}^{\mathrm{E}_0}$
$\mathbf{c} = \mathbf{T}_{\mathrm{S}}^{\mathrm{H}_0}$
$\beta = \arccos\left(\left(|\mathbf{a}|^2 + |\mathbf{c}|^2 - |\mathbf{b}|^2\right) / (2\ |\mathbf{a}|\ |\mathbf{c}|)\right)$
$\mathbf{m} = |\mathbf{a}| \cos(\beta) \frac{\mathbf{c}}{|\mathbf{c}|}$
$\mathbf{n}_0 = \mathbf{a} - \mathbf{m}$
$\beta_{\mathrm{ext}} = 0.6\,\beta$
$\mathbf{m}_{\mathrm{ext}} = |\mathbf{a}| \cos(\beta_{\mathrm{ext}}) \frac{\mathbf{c}}{|\mathbf{c}|}$
$\mathbf{n}_{\mathrm{ext}} = |\mathbf{a}| \sin(\beta_{\mathrm{ext}}) \frac{\mathbf{n}_0}{|\mathbf{n}_0|}$
$\mathbf{a}_{\mathrm{ext}} = \mathbf{n}_{\mathrm{ext}} + \mathbf{m}_{\mathrm{ext}}$
$\mathbf{c}_{\mathrm{ext}} = \sqrt{|\mathbf{b}|^2 - |\mathbf{n}_{\mathrm{ext}}|^2} \frac{\mathbf{c}}{|\mathbf{c}|}$
$\mathbf{b}_{\mathrm{ext}} = \mathbf{c}_{\mathrm{ext}} + \mathbf{a}_{\mathrm{ext}}$
$\gamma_{\mathrm{ext}} = \arccos\left(\left(|\mathbf{a}|^2 + |\mathbf{b}|^2 - |\mathbf{c}_{\mathrm{ext}}|^2\right) / (2\ |\mathbf{a}|\ |\mathbf{b}|)\right)$
$\phi_{\mathrm{Elbow}} = \gamma_{\mathrm{ext}} - \pi$
$\mathbf{q}_0 = \mathrm{QuaternionFromAxisAngle}(\mathbf{c}_{\mathrm{ext}},\ \alpha)$
$\mathbf{n} = \mathbf{q}_0\, \mathbf{n}_{\mathrm{ext}}\, \mathbf{q}_0^{-1}$
$\mathbf{h}_1 = (0, 0, 1)^{\mathrm{T}}$
$\mathbf{h}_2 = \frac{\mathbf{a}_{\mathrm{ext}}}{|\mathbf{a}_{\mathrm{ext}}|}$
$\delta_1 = \mathrm{VecOnVecRoundAxis}(\mathbf{h}_1,\ \mathbf{h}_2,\ \mathbf{h}_1 \times \mathbf{h}_2)$
$\mathbf{q}_1 = \mathrm{QuaternionFromAxisAngle}(\mathbf{h}_1 \times \mathbf{h}_2,\ \delta_1)$
$\mathbf{q}_2 = \mathrm{QuaternionFromEuler}(\phi_{\mathrm{Elbow}},\ 0,\ 0)$
$\mathbf{q}_3 = \mathbf{q}_1\, \mathbf{q}_2$
$\mathbf{h}_3 = \mathbf{q}_3\, \mathbf{h}_1\, \mathbf{q}_3^{-1}$
$\delta_2 = \mathrm{VecOnVecRoundAxis}(\mathbf{h}_3,\ \mathbf{b},\ \mathbf{a}_{\mathrm{ext}})$
$\mathbf{q}_4 = \mathrm{QuaternionFromEuler}(0,\ 0,\ \delta_2)$
$\mathbf{q}_5 = \mathbf{q}_1\, \mathbf{q}_4$
$(\phi_{\mathrm{S,\,x}},\ \phi_{\mathrm{S,\,y}},\ \phi_{\mathrm{S,\,z}}) = \mathrm{QuaternionToEuler}(\mathbf{q}_5)$
```

Algorithm 2 Resampling between timesteps

```
$S_{t+1,\mathrm{norm}} = \mathrm{SUS}(S_{t,M}) + \mathbf{R}_{\mathrm{scatter}}$
$S_{t+1,\mathrm{inverse}} = \mathrm{InverseKinematics}(s_{t,M}^{\mathrm{optimal}}) + 0.1\,\mathbf{R}_{\mathrm{scatter}}$
$S_{t+1,\mathrm{static}} = \mathrm{StaticPoses}(S_{t,M}) + 0.1\,\mathbf{R}_{\mathrm{scatter}}$
$S_{t+1,0} = [S_{t+1,\mathrm{norm}}, S_{t+1,\mathrm{inverse}}, S_{t+1,\mathrm{static}}]$
```

Algorithm 3 Resampling between annealing steps

```
$S_{t+1,\mathrm{norm}} = \mathrm{SUS}(S_{t,M}) + \mathbf{R}_{\mathrm{covariance}}$
$S_{t+1,\mathrm{crossover}} = \mathrm{Crossover}(S_{t,M}) + \mathbf{R}_{\mathrm{covariance}}$
$S_{t+1,\mathrm{random}} = \mathrm{SUS}(S_{t,M}) + \mathbf{R}_{\mathrm{scatter}}$
$S_{t+1,0} = [S_{t+1,\mathrm{norm}}, S_{t+1,\mathrm{crossover}}, S_{t+1,\mathrm{random}}]$
```

References

[1] D. Arsić, B. Hörnler, B. Schuller, and G. Rigoll. Resolving partial occlusions in crowded environments utilizing range data and video cameras. In *Proceedings 16th IEEE International Conference on Digital Signal Processing, Special Session "Fusion of Heterogeneous Data for Robust Estimation and Classification" , DSP2009, Santorini, Greece*, July 2009. 1

[2] P. Azad, T. Asfour, and R. Dillmann. Robust real-time stereo-based markerless human motion capture. In *Humanoid Robots, 2008. Humanoids 2008. 8th IEEE-RAS International Conference on*, pages 700–707, Daejeon, Dec. 2008. 1, 3

[3] O. Bernier, P. Cheungmonchan, and A. Bouguet. Fast nonparametric belief propagation for real-time stereo articulated body tracking. *Computer Vision and Image Understanding*, July 2008. 1

[4] J. Darby, B. Li, and N. Costen. Human activity tracking from moving camera stereo data. In *BMVC08*, 2008. 1

[5] J. Deutscher and I. Reid. Articulated body motion capture by stochastic search. *Int. J. Comput. Vision*, 61(2):185–205, 2005. 1, 2

[6] M. Fontmarty, F. Lerasle, and P. Danes. Data fusion within a modified annealed particle filter dedicated to human motion capture. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 3391–3396, San Diego, CA, Oct./Nov. 2007. 1

[7] J. Gall, J. Potthoff, C. Schnörr, B. Rosenhahn, and H.-P. Seidel. Interacting and annealing particle filters: Mathematics and a recipe for applications. *Journal of Mathematical Imaging and Vision*, 28(1):1–18, 2007. 2

[8] M. Hofmann and D. M. Gavrila. Multi-view 3d human pose estimation combining single-frame recovery, temporal integration and model adaptation. In *Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on*, pages 2214–2221, Miami, FL, USA, June 2009. 1

[9] T. B. Moeslund, A. Hilton, and V. Krger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(2-3):90 – 126, 2006. Special Issue on Modeling People: Vision-based understanding of a person's shape, appearance, movement and behaviour. 1

[10] Z. Zivkovic. Improved adaptive gaussian mixture model for background subtraction. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 2, pages 28–31 Vol.2, Aug. 2004. 2