CUE-INDEPENDENT EXTENDING INVERSE KINEMATICS FOR ROBUST POSE ESTIMATION IN 3D POINT CLOUDS

Nicolas H. Lehment, Moritz Kaiser, Dejan Arsić, Gerhard Rigoll

Institute for Human Machine Communication
Technische Universität München
Arcisstr. 21, 80634 München, Germany
{lehment, moritz.kaiser, arsic, rigoll @ tum.de}

ABSTRACT

While monocular gesture recognition slowly reaches maturity, the inclusion of 3D gestures remains a challenge. In order to enable robust and versatile depth-enabled gestures, a depth-image based tracking approach is developed. Using a model-based annealing particle filter approach, the pose of a single subject is retrieved and tracked over longer image and motion sequences. Other than many previous depth-image based systems, full body tracking is performed. The system is independent from specific camera types and is independent from color or texture cues. Pose space exploration in complex kinematic chains is enhanced by considering extending inverse kinematics. Exploiting the highly parallel nature of the 3D point based approach, the algorithm is partially implemented on a GPU, leading to near real time performance.

Index Terms— Pose Estimation, Pose Tracking, 3D, Annealing Particle Filter, Stereo, Inverse Kinematics

1. INTRODUCTION

Interaction in virtual reality environments is currently mostly performed by tracking markers located on the person's limbs and the subsequent 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. Therefore we suggest the use of a vision based pose estimation and tracking system. Such systems usually 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 approach. This model-based approach avoids most self-occlusion conflicts and ambiguities, without requiring pre-learned motions or dedicated detection of single body parts.

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 limited loss of 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.

Annealing Particle Filters (APF) have already been utilized in various 2D tracking applications, frequently with multiple camera angles ([2],[3]) and have been shown capable to handle depth information when using additional cues: Azad ([4]) and Bernier ([5]) used skin cues as anchor points for the wrists, Darby ([6]) relied on pre-learned motion patterns. 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 [7]. 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 method for exploring high-dimensional solution spaces and variations have been used in a number of tracking scenarios. Usually, an initial set of particles S_0, each representing a set of parameters, is matched against data from observations. Based on the quality of the matching a score w_0^i, also called weight, is assigned to each particle s_0^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 slightly mutating the parameters in the new set, the resulting set S_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 ([2]) and Gall ([8]). 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 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 (S) which have to be matched against the cloud of 3D data points (C).

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:

$$w_{\text{S2C}}{}' = \sum_{s \in S} e^{10 \, \min_c(\text{d}_{\text{eukl}}(p_s, p_c))}, \tag{1}$$

$$w_{\text{S2C}} = e^{1.0 - w_{\text{S2C}}{}'}. \tag{2}$$

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 (1) to use only edge points with

$$w_{\text{E2C}}{}' = \sum_{e \in S_e} e^{10 \, \min_c(\text{d}_{\text{eukl}}(p_e, p_c))}, \tag{3}$$

$$w_{\text{E2C}} = e^{1.0 - w_{\text{E2C}}{}'}. \tag{4}$$

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 C_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:

$$w_{\text{C2S}}^{(\text{i})}{}' = \sum_{c \in C_i} e^{10 \, \min_s(\text{d}_{\text{eukl}}(p_c, p_s))}, \tag{5}$$

$$w_{\text{C2S}}^{(\text{i})} = e^{1.0 - w_{\text{C2S}}^{(\text{i})}{}'}, \tag{6}$$

$$w_{\text{C2S}} = \prod_{\text{i}} w_{\text{C2S}}^{(\text{i})}. \tag{7}$$

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

$$w_{\text{score}} = w_{\text{C2S}} \times w_{\text{E2C}} \times w_{\text{S2C}} \tag{8}$$

In addition to the point cloud based weighting, other mechanisms influence the particle score as well: Self-collision between checks 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 $S_{\text{t, m}}$ at timestep t, annealing step m, consists of $S_{\text{t, m}} = [(\mathbf{s}_{\text{t, m}}^0, \mathbf{w}_{\text{t, m}}^0) \ldots (\mathbf{s}_{\text{t, m}}^N, \mathbf{w}_{\text{t, m}}^N)]$. The particle

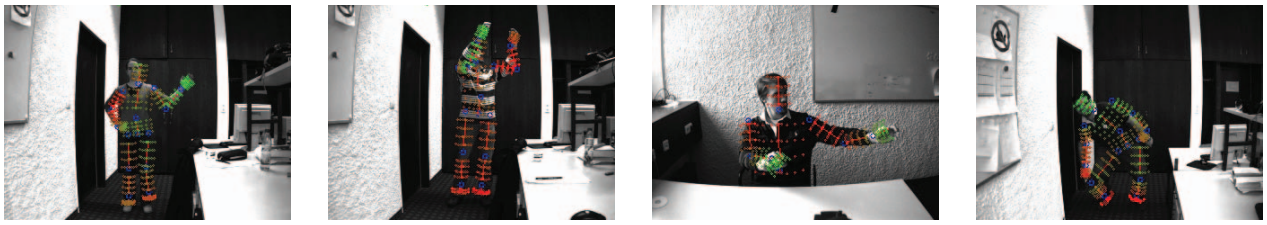

Fig. 1. Different poses with the matched body prototype overlaid, full body and upper body only

$\mathbf{s}^n_{\mathrm{t,\,m}}$ itself contains the encoded joint angles and basic transformations, while the weight, $\mathbf{w}^n_{\mathrm{t,\,m}}$, is computed from the fitting of the particle to the observed data. Resampling was performed by stochastic universal sampling, generating $S_{\mathrm{t+1,\,0}}$ from $S_{\mathrm{t,\,M}}$ with $\mathbf{R}_{\mathrm{scatter}}$ and $S_{\mathrm{t,\,m+1}}$ from $S_{\mathrm{t,\,m}}$ with $\mathbf{R}_{\mathrm{covariance}}$. Random scattering $\mathbf{R}_{\mathrm{scatter}}$ between timesteps is set to be quite large, while $\mathbf{R}_{\mathrm{covariance}}$ is set proportional to the covariance of the parameters in the particle set.

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. So 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 [4], we instead utilize the wrist position from the pose estimate. While 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 $\phi_{\mathrm{S,\,x}}$, $\phi_{\mathrm{S,\,y}}$, $\phi_{\mathrm{S,\,z}}$, ϕ_{Elbow} are used to build a new particle (see algorithm 3.3). Without anchor points such as separately tracked hands, the APF can get stuck in high elbow angles. To avoid this situation, we extend the arms slightly during inverse kinematics.

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 population. 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 ($\mathbf{R}_{\mathrm{scatter}}$) of the joint angles. This allows for a more thorough exploration of the configuration space even as the regular particle set is converging. The insertion of randomized particles also helps in recovering from failed tracking.

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 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 AND OUTLOOK

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 (300 particles, 8 annealing steps). However, GPU memory is not yet coalescent 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. Considering that we did not use any learned motion models or anchor points, these are quite remarkable results. The

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

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

sequences used can be obtained from the authors.
By simple modification of the body model, which is defined in a XML file, we were able to convert the full body tracker to an upper body tracker without any modification of the algorithm itself. This underscores the flexibility gained from abandoning pre-learned motion models in favor of a purely depth-centered approach, allowing for easy modifications, fast adaptions of existing modules and tracking of unknown motion sequences. 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 pre-learned motion models) is expected to become more pronounced.
For our future work, we expect to reach real time capability by moving further functions onto the GPU and optimizing memory usage. 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 usage scenarios.

Algorithm 2 Resampling between timesteps

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

Algorithm 3 Resampling between annealing steps

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

5. 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.

[2] Jonathan Deutscher and Ian Reid, "Articulated body motion capture by stochastic search," *Int. J. Comput. Vision*, vol. 61, no. 2, pp. 185–205, 2005.

[3] Michael Hofmann and Dariu 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*, Miami, FL, USA, June 2009, pp. 2214–2221.

[4] 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*, Daejeon, Dec. 2008, pp. 700–707.

[5] 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.

[6] J. Darby, B.H. Li, and N.P. Costen, "Human activity tracking from moving camera stereo data," in *BMVC08*, 2008.

[7] Z. Zivkovic, "Improved adaptive gaussian mixture model for background subtraction," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, Aug. 2004, vol. 2, pp. 28–31 Vol.2.

[8] Jürgen Gall, Jürgen Potthoff, Christoph Schnörr, Bodo Rosenhahn, and Hans-Peter Seidel, "Interacting and annealing particle filters: Mathematics and a recipe for applications," *Journal of Mathematical Imaging and Vision*, vol. 28, no. 1, pp. 1–18, 2007.