Using Segmented 3D Point Clouds for Accurate Likelihood Approximation in Human Pose Tracking

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

Abstract

The observation likelihood approximation is a central problem in stochastic human pose tracking. In this paper, we present a new approach to quantify the correspondence between hypothetical and observed human poses in depth images. Our approach is based on segmented point clouds, enabling accurate approximations even under self-occlusion and in the absence of color or texture cues. The segmentation step extracts small regions of high saliency such as hands or arms and ensures that the information contained in these regions is not marginalized by larger, less salient regions such as the chest. The proposed approximation function is evaluated on both synthetic and real camera data. In addition, we compare our approximation function against the corresponding function used by a state-of-the-art pose tracker.

1. INTRODUCTION

Human pose tracking is a highly complex task which has undergone rapid development during the last decade. Applications range from entertainment systems [12] to professional HCI applications, e.g. in sterile environments.

In recent years, the two classical approaches of monocular and multi-camera pose tracking have been joined by depth image based pose tracking (as summarized in [13]). The majority of depth data based pose tracking systems employ stochastic methods. The observation likelihood function lies at the heart of these stochastic methods, providing a measure of confidence that a given pose hypothesis is supported by the observed data. Typically, an observation likelihood function is derived from edge or feature matching between a deformable body model and the current observation ([9], [5]). Azad et al. [2] use edge cues in combination with a separate hand and head tracker, Bernier et al. [3] consider a combination of 3D contour points and a separate hand tracker, Darby et al. [4] work only with the 3D contour points while Fontmarty et al. [6] use edge points and a number of other cues.

Most methods considered so far use a stereo camera setup. The growing availability of affordable and precise depth sensing cameras, such as Microsoft's Kinect system [12] or Asus' Xtion camera [1], have sparked an increased interest in the use of pure depth data for pose estimation. Most recent approaches to pose tracking, such as presented by Zhu et al. [17], Ganapathi et al. [8] and Shotton et al.[15], work directly on the depth data obtained from the sensor supported by key-point detection.

The various methods mentioned above perform their likelihood approximation based on depth and color cues. However, the recent release of tools like the point cloud library (PCL, [14]) enables evaluation of pose hypotheses directly in 3D space. This significantly reduces the need for reprojecting complex mesh models onto the image plane (as done in [8] and [17]). Additionally a point cloud based approximation can easily be extended to process data coming from several depth sensors. We therefore designed an approximation function to compute a segmented 3D point cloud for every pose hypothesis. This hypothetical point cloud can then be used to generate a likelihood score given the current sensor data. The proposed likelihood approximation can be used as a weighting function in the context of any human pose tracking system using stochastic approaches.

To the best of our knowledge, this paper is the first systematic evaluation of a likelihood function approximation for purely spatial observation data in a human pose tracking context. Previous work on likelihood function design ([7], [11]) does not appear directly applicable to data obtained from depth sensing cameras and similar devices.

The rest of the paper is structured as follows: In Section 2.1 we will clarify the mathematical context. Our body model is described in detail in Section 2.2. Section 2.3 contains a description of the proposed approximation function and the evaluation method. The results are presented and discussed in Section 3. Finally, Section 4 summarizes our results and gives a brief outlook.

2. LIKELIHOOD APPROXIMATION AND EXPERIMENTAL SETUP

2.1. Mathematical Background

Stochastic approaches to pose tracking attempt to find the best pose hypothesis $\mathbf{x}_{\text{opt}}$ given the previous states $\mathcal{X}_t$ and a current observation $\mathbf{Z}_t$ by maximizing the posterior

$$\mathbf{x}_{\text{opt}} = \text{argmax}(p(\mathbf{x}_t|\mathbf{Z}_t, \mathcal{X}_t)) \tag{1}$$

In the context of particle filters [9] and graphical models [3], the posterior is often simplified to the form of

$$p(\mathbf{x}_t|\mathbf{Z}_t, \mathcal{X}_t) = k\, p(\mathbf{x}_t|\mathcal{X}_t)\, p(\mathbf{Z}_t|\mathbf{x}_t) \tag{2}$$

In this simplified form, k is a normalizing constant, $p(\mathbf{x}_t|\mathcal{X}_t)$ represents a motion model and $p(\mathbf{Z}_t|\mathbf{x}_t)$ the observation likelihood. The simplified posterior assumes independence between motion dynamics and observation likelihood while using an uniform distribution of the priors $p(\mathbf{x}_t)$, $p(\mathcal{X}_t)$ and $p(\mathbf{Z}_t)$. Since a full observation model is usually not available, an approximation $w(\mathbf{x}_t, \mathbf{Z}_t) \propto p(\mathbf{Z}_t|\mathbf{x}_t)$ is used instead. The general process flow of the likelihood approximation is illustrated in Figure 1. In the following sections, we shall propose a novel formulation of such an approximation and evaluate its performance using both synthetic and real world data sets.

2.2. Deformable Body Model

The observation likelihood can be determined by comparing model points on a deformable body model with the actual observation. We use a 22 degrees of freedom ellipsoid upper body model (a dense versions is shown in Figure 2), consisting of nine basic body elements with 32 model points each. Since ellipsoids are simply the 3-dimensional equivalent of an ellipse, they are very easy to generate and manipulate. The surface of each ellipsoid is populated by model points. The ellipsoids and their respective model points are transformed and rotated according to the parameters encoded in a specific pose $\mathbf{x}$ using a skeletal hierarchy with the pelvis as the highest node.

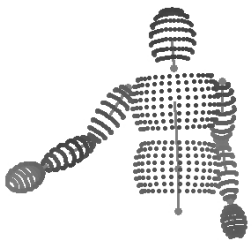

Figure 2. A dense version of the ellipsoid body model.

Once the ellipsoid body elements are arranged according to a pose $\mathbf{x}$, we calculate visibility and collisions for the hypothetical point cloud. Each single model point $\mathbf{s} \in \mathbf{S}(\mathbf{x})$ (where $\mathbf{S}(\mathbf{x})$ signifies a single sample cloud computed from pose $\mathbf{x}$) is checked against each ellipsoid body element $e \in E$ for collision or occlusion:

$$\mathbf{q} = -\mathbf{s} \tag{3}$$

$$\mathbf{q}_{\text{s}} = \frac{(\mathbf{r}_e^{-1}\mathbf{q}\mathbf{r}_e)^{\text{T}}}{\|\mathbf{q}\|}\mathbf{g}_e \tag{4}$$

$$\mathbf{s}_{\text{s}} = (\mathbf{r}_e^{-1}(\mathbf{s} - \mathbf{T}_e)\mathbf{r}_e)^{\text{T}}\mathbf{g}_e \tag{5}$$

In the preceding equations, the model point $\mathbf{s}$ is given in the camera coordinate system. The normalized direction from the camera to the sample point in the reference system of the body element is given as $\mathbf{q}_{\text{s}}$. The vector $\mathbf{s}_{\text{s}}$ denotes the line from the sample point to the center of the ellipsoid body element. The quaternion $\mathbf{r}_e$ denotes the rotation of the specific body element, whereas $\mathbf{T}_e$ is the translation from the camera to the limb ellipsoid center. The scaling vector $\mathbf{g}_e$ transforms the ellipsoid into a unity sphere for distance and intersection calculation. Therefore, the problem is reduced to an intersection check between a line and a sphere. We can now determine occlusion and collision for the model point:

$$d = (\mathbf{q}_{\text{s}}^{\text{T}}\mathbf{s}_{\text{s}})^2 - \mathbf{s}_{\text{s}}^{\text{T}}\mathbf{s}_{\text{s}} - 1 \tag{6}$$

$$f_{occl} = \begin{cases} 1, & \text{if } d > 0 \cap -\mathbf{q}_{\text{s}}^{\text{T}}\mathbf{s}_{\text{s}} + \sqrt{d} > 0 \\ 0, & \text{else} \end{cases} \tag{7}$$

$$f_{coll} = \begin{cases} 1, & \text{if } \sqrt{\mathbf{s}_{\text{s}}^{\text{T}}\mathbf{s}_{\text{s}}} < 1 \cap e \notin E_{\text{Neighbors}} \\ 0, & \text{else} \end{cases} \tag{8}$$

The flags f_{occl} and f_{coll} signal occlusion and collisions respectively. By performing both checks using the same basic principles, significant reductions in computational effort can be achieved.

In order to constrain the quaternion joint rotation, we use a simplified version of the approach described in [16]. For each joint, the allowed limb reach cone and twist is predefined during model generation. The rotational quaternions can then be efficiently tested and adjusted during run-time. Another consideration would be the scattering of quaternion encoded joint rotations during resampling. While outside the scope of this paper, a simple interpolation operation (NLERP or SLERP) with a randomly generated quaternion and a scattering coefficient should suffice. Thus, costly conversions to Euler angles could be eliminated completely.

At the end of this process, we arrive at a point cloud $\mathbf{S}(\mathbf{x})$ of the hypothetical pose which can then be used to calculate the likelihood score.

2.3. Likelihood Approximation

The approximation function presented here computes the observation likelihood by finding nearest-neighbor pairs between the non-occluded 3D model points $\mathbf{S}(\mathbf{x})$ computed

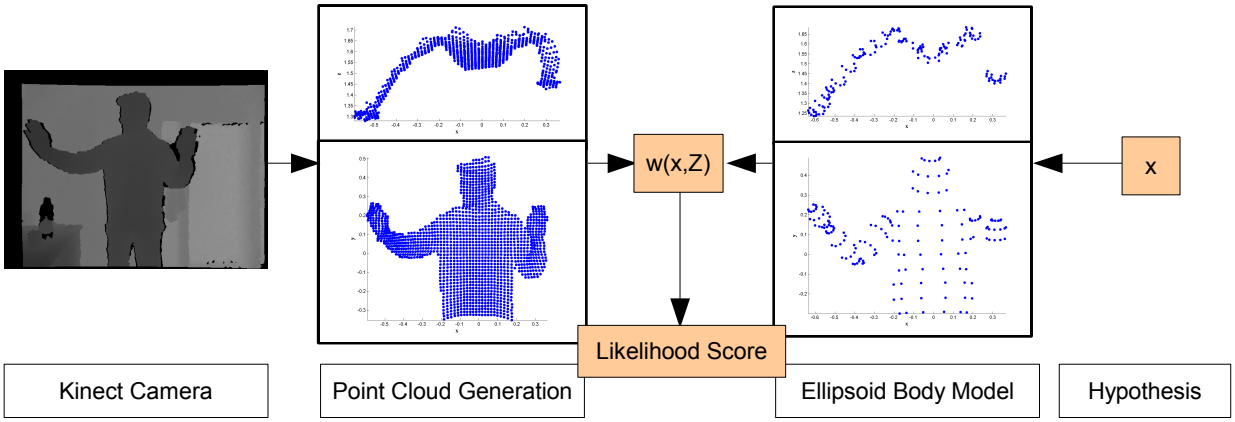

Figure 1. The general flow of the likelihood approximation. Using depth data coming from a Kinect camera, the point cloud of the user is extracted. For an arbitrary pose hypothesis $\mathbf{x}$, a point cloud is computed using the approach proposed in Section 2.2. An approximation function then computes a likelihood score as described in Section 2.3.

for a pose $\mathbf{x}$ and the 3D data points $\mathbf{R}(\mathbf{Z})$ extracted from the observed image $\mathbf{Z}$. When the observed pose and the sampled pose are very similar, we can expect points of $\mathbf{S}(\mathbf{x})$ to be very close to the points of $\mathbf{R}(\mathbf{Z})$.

At first glance it may seem sufficient to simply find the closest model point to each data point and consider the mean distance. However, this approach would ignore all regions of model points which have no nearby data point (indicating a badly fitting hypothesis). On the other hand, we would encounter the same problem if we only were to consider the closest data point to each model point (this would effectively ignore data). Therefore, we have to find both the closest data point to each model point and vice versa in order to build our likelihood approximation:

$$w_1(\mathbf{S}(\mathbf{x}),\mathbf{R}(\mathbf{Z})) = \prod_{e=1}^{E} \frac{1}{N_e} \sum_{\mathbf{s}\in\mathbf{S}_e(\mathbf{x})} \exp(-c_1 \min_{\mathbf{r}} \|\mathbf{s}-\mathbf{r}\|) \tag{9}$$

$$W_1(\mathbf{S}(\mathbf{x}),\mathbf{R}(\mathbf{Z})) = \exp(c_2(1.0 - w_1(\mathbf{S}(\mathbf{x}),\mathbf{R}(\mathbf{Z})))) \tag{10}$$

$$w_2(\mathbf{S}(\mathbf{x}),\mathbf{R}(\mathbf{Z})) = \prod_{k=1}^{K} \frac{1}{N_k} \sum_{\mathbf{r}\in\mathbf{R}_k(\mathbf{Z})} \exp(-c_1 \min_{\mathbf{s}} \|\mathbf{s}-\mathbf{r}\|) \tag{11}$$

$$W_2(\mathbf{S}(\mathbf{x}),\mathbf{R}(\mathbf{Z})) = \exp(c_2(1.0 - w_2(\mathbf{S}(\mathbf{x}),\mathbf{R}(\mathbf{Z})))) \tag{12}$$

$$w(\mathbf{x},\mathbf{Z}) = W_1(\mathbf{S}(\mathbf{x}),\mathbf{R}(\mathbf{Z})) \times W_2(\mathbf{S}(\mathbf{x}),\mathbf{R}(\mathbf{Z})) \tag{13}$$

To increase the impact of local discrepancies, segmentation is used in (9) and (11). High minimum distances in small but significant regions, such as hands, thus lead to low likelihood values for the whole pose. The segmentation of the model points for $w_1(\mathbf{S}(\mathbf{x}),\mathbf{R}(\mathbf{Z}))$ is performed either by the $E = 9$ body elements or by k-means clustering. Both methods will be evaluated in Section 3.1. Grouping by body element, segments with less than 3 visible points are assigned a generic partial score of 1. The observed data points used in $w_2(\mathbf{S}(\mathbf{x}),\mathbf{R}(\mathbf{Z}))$ are meanwhile segmented by k-means clustering. Ten data point clusters were found to yield best results under most circumstances. The constants c_1 and c_2 are used for shaping the steepness of the approximated likelihood function (typically $c_1 = 20$, $c_2 = 2$). The variables N_k and N_e denote the number of visible points per data or model cluster.

In order to penalize excessive self-collisions between limbs, we use the number N_C (gained by counting all f_{coll} flags from Equation 8):

$$w(\mathbf{x},\mathbf{Z})_{\text{Final}} = \begin{cases} w(\mathbf{x},\mathbf{Z}) & N_C < N_{\text{Threshold}} \\ \dfrac{w(\mathbf{x},\mathbf{Z})}{N_C - N_{\text{Threshold}}} & N_C \geq N_{\text{Threshold}} \end{cases} \tag{14}$$

The intuition for improving accuracy by segmentation is given in Figure 3. Shown is a simplified example with only two body parts and a score of 1 for correctly placed model points. The torso on the left fits the observation perfectly, generating a score of 1 for each model point. The arm on the right is posed incorrectly (correct pose shown in grey), thus gaining a score of 0 for all incorrectly placed model points. Using no segmentation, the score would be $\frac{1}{N}\sum_{\mathbf{s}\in\mathbf{S}(\mathbf{x})} w_s = \frac{25}{29} = 0.862$ with w_s representing the simplified model point score. While the pose actually does not fit the observation very well, it would still get a high observational likelihood. Using segmentation by body element, as in in (9) we arrive at $\prod_{e=1}^{E} \frac{1}{N_e} \sum_{\mathbf{s}\in\mathbf{S}_e(\mathbf{x})} w_s = 0.2$. The segmented approximation thus reduces the marginalization of more salient regions by larger, less salient regions.

It is important to note that this approximation $p(\mathbf{Z}_t|\mathbf{x}_t) \propto w(\mathbf{x},\mathbf{Z})_{\text{Final}}$ works without color or texture

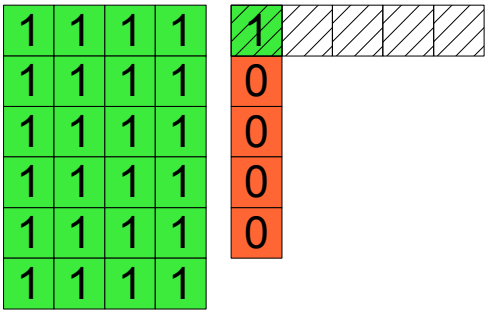

Figure 3. Simplified illustration of a segmented likelihood approximation, using one torso element and one arm. The torso is fitted correctly, while the arm hypthesis does not fit the real, outstretched arm pose (shown in grey). Detailed discussion in Section 3.1.

cues, making this approach suitable for all scenarios where only pure depth data is available.

2.4. Experimental Setup & Performance Evaluation

Both synthetic and real-world point clouds[1] of exemplary upper body poses were used for evaluation. A data set consists of one reference pose and 800 sample poses, corresponding to one observation and 800 hypotheses. For the 20 synthetic test sets, reference poses were generated using a densely meshed upper body model with light noise. Each of the synthetic reference clouds consists of an average of 500 data points with ideal joint and limb positions stored as ground truths. The 15 real-world point clouds were recorded both with a PointGrey BumblebeeXB3 and a Microsoft Kinect from depth-data only and stored with a hand-labeled ground truth. For each reference pose, the 800 sample point clouds are generated using the regular, lower definition body model as described in section 2.2. Starting from the original reference pose $\mathbf{x}_{\text{ref}}$, uniform noise of varying amplitude (from 0.05 rad to 0.5 rad) is added to the joint variables, generating varied sample poses $\mathbf{X}_{\text{Sample}}$. Joint and limb positions are then stored together with the corresponding point clouds.

Using the approximation $p(\mathbf{Z}_t|\mathbf{x}_t) \propto w(\mathbf{x},\mathbf{Z})_{\text{Final}}$ in Equation 14 and the average limb position error $\mathbf{e}_{\text{Sample}}$ for each sample, we can establish the correspondence between estimated likelihood and limb position error. We gain the error $\mathbf{e}_{\text{Sample}}$ for a single sample pose by calculating the mean euclidean distance between reference and sample limb positions ($\mathbf{T}_{\text{e, Sample}}$ and $\mathbf{T}_{\text{e, ref}}$):

$$\mathbf{e}_{\text{Sample}} = \frac{1}{N_e}\sum_{e=1}^{E}\sqrt{(\mathbf{T}_{\text{e, ref}} - \mathbf{T}_{\text{e, Sample}})(\mathbf{T}_{\text{e, ref}} - \mathbf{T}_{\text{e, Sample}})^{\text{T}}} \tag{15}$$

[1]The data sets used are available from the author upon request

Configuration Parameters		
Configuration	Variable	Value
All	c_1	20
	c_2	2
	Camera Depth Noise	$\pm$2.5 cm
	Camera Raster Size	2.5 cm x 2.5 cm
	Test Poses (synthetic)	20
	Test Poses (real)	15
	Samples per Test Pose	800
	$N_{\text{Threshold}}$	5 (if active)
(1)	Data Clusters	1
	Model Clusters	1
(2)	Data Clusters	10 (k-means)
	Model Clusters	1
(3)	Data Clusters	1 (k-means)
	Model Clusters	9 (by element)
(4) & (5)	Data Clusters	10 (k-means)
	Model Clusters	9 (by element)
(6) & (7)	Data Clusters	10 (k-means)
	Model Clusters	5 (k-means)

Table 1. Settings for the different configurations, numbering of configurations as shown in Figure 5. Algorithm parameters were selected for best performance in repeated experiments.

The mean likelihood is then computed for bins of samples with similar $\mathbf{e}_{\text{Sample}}$. We can thus plot the mean likelihood over given mean position errors. A normalization is not strictly necessary, but helps in visualizing the effect of various settings. The influence of segmentation and collision detection on observation likelihood function has been shown by using varying parameters (summarized in Table 1).

To facilitate comparison with other current approaches, the observation likelihood approximation used by a state-of-the-art tracker [8] was implemented in a Matlab script and tested against our Kinect dataset. The results were normalized to the range $(0, 1)$ and are given as a reference curve (R) in Figure 5. The required sample depth images were generated using Autodesk 3ds Max 2011 and 3DVIA Virtools 5.0 for posing and rendering. The sample poses and original depth data were identical to the ones used by our proposed method.

For a closer examination of occlusion and ambiguity

handling in a 2D joint space a third, synthetic data set was created. Here, the derived samples were varied only in one shoulder angle and the elbow to enable detailed analysis of occlusion and ambiguity handling in a constrained case.

It should be noted that the function is evaluated outside of a tracker framework. We used this testbench approach in order to focus solely on the approximation of the likelihood function while excluding other effects originating in the tracker structure. We therefore do not use motion sequences in our analysis, but focus on the observational likelihood approximation quality for different, non-consecutive poses.

3. RESULTS AND DISCUSSION

3.1. Evaluation of Different Clustering Strategies

In Figure 5 the normalized average weights over position error are given for a number of different approximation configurations. It is obvious that a simple distance measure, as shown in curve (1), does not give sufficient precision or robustness. This is mostly due to the negligible influence of errors in smaller regions, such as hands, on the overall score.

Curve (2) shows the effect of clustering the data points by k-means previous to the computation of $W_2(\mathbf{S}(\mathbf{x}), \mathbf{R}(\mathbf{Z}))$. The decline of likelihood over rising error is more pronounced than in configuration (1) but still shows a lack of robustness especially for larger position errors. Similarly, curve (3) illustrates the effect of clustering the model points by body element, such as a hand or a forearm. The improvement in accuracy is not as pronounced as for configuration (2), though still significant.

Configuration (4) combines clustering of the model points by body element and clustering of data by k-means clustering. The decline of likelihood is even more pronounced than in configuration (2) and (3), suggesting higher precision. While the strong gradient indicates a high standard deviation, one can expect the higher precision (synthetic: 13.9% lower weight at $0.1\,m$ error) to offset the slightly lower robustness. Configuration (5) is essentially a variation of configuration (4), including the collision penalty in the calculation of the likelihood. Since in this test scenario only few collisions are possible, the effect is not readily visible.

The curve (6) shows a similar configuration as curve (4) with k-means clustering used instead of the partitioning by body element. 5 clusters were found to yield best results. Despite a similar increase of precision (synthetic: 10.5% lower weight at $0.1\,m$ error compared to configuration (2)), this method has the disadvantage of requiring k-means clustering for every single pose sample. As clustering by body elements shows a better precision and lower computational effort, it appears preferable to k-means clustering. For the sake of completeness, curve (7) shows the influence of additional collision penalties.

Considering Figure 4, we can see that the real-world data essentially follows the performance shown in the synthetic scenarios. It should be noted that the weighting does not drop towards zero as quickly as in the synthetic dataset. This is mostly due to differences between the body model and the real, observed person. Due to loose clothing on the real person and the more slender shape of the ellipsoid body model the observed point cloud usually has a larger volume. As the limbs of the body model can assume slightly differing poses within the observed point cloud, larger deviations from the reference pose become possible, adding up to higher average errors. This may be remedied by performing an automatic adaption of the ellipsoid model within an initialization stage (e.g. using a pre-known initialization pose).

Comparing the two camera systems directly, no significant differences in accuracy or convergence appear. It is however interesting to note that the Kinect camera system yields a slightly smoother approximation than the BumblebeeXB3. Since both depth images were sampled to the same resolution, this might indicate the lower depth noise level of depth images obtained with a structured lighting approach.

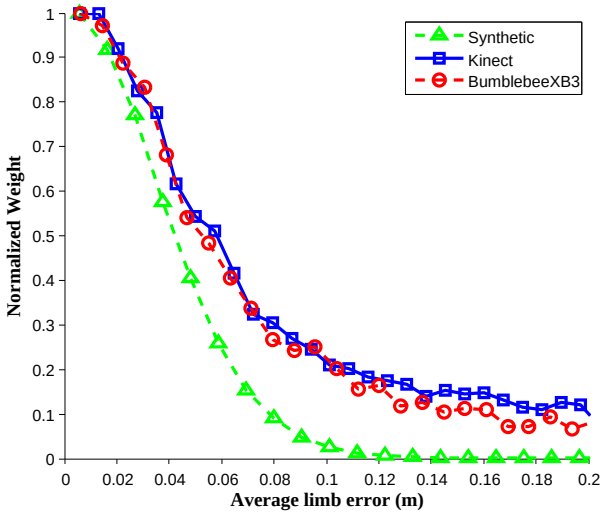

Figure 4. Comparison of average error for synthetic and real camera data, using configuration (5).

3.2. Comparison to a State-Of-The-Art Technique

The direct comparison between the results of our approach and the approximation used in a state-of-the-art tracker [8] shows the benefits of segmentation: As shown in Figure 5, the reference approach (R) is more robust and precise than the unsegmented approach shown by (1). Once partial scores (2) or clustered evidence (3) are introduced,

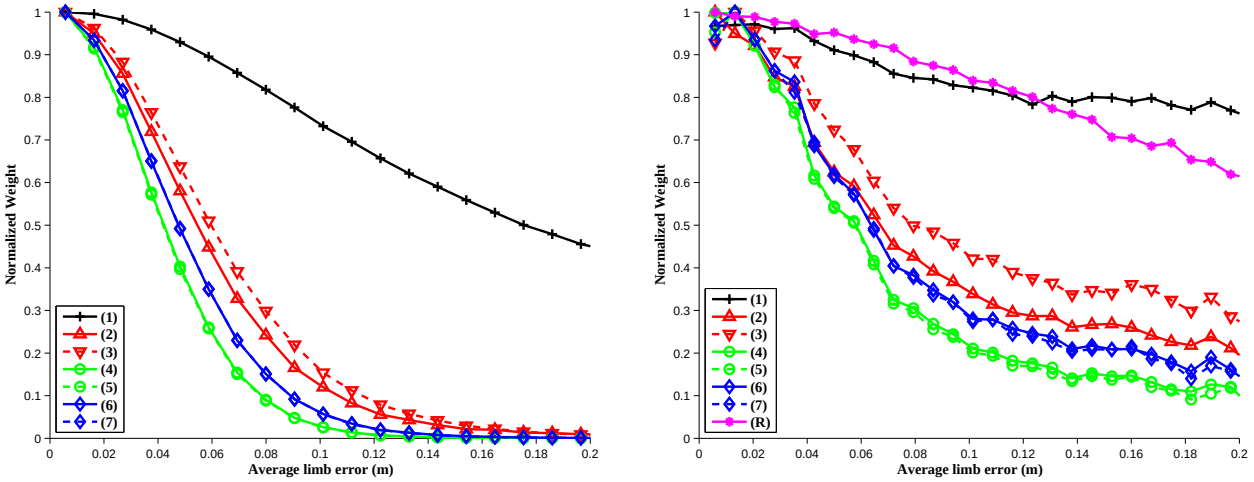

Figure 5. Average Error for various likelihood approximations over position error for depth data from a synthetic source (left) and Kinect (right). The following configurations were used (summarized in table 1): (1) No clustering or collision check at all, (2) k-means clustering of data points, (3) clustering of model points by body element, (4) clustering both model points and data points, (5) same as (4) with collision penalities, (6) clustering of model points by k-means, (7) same as (6) with collision penalties. (R) shows the results obtained from the reference approach (taken from [8]).

our approximation yields more precise results. This is to be expected, as the reference likelihood approximation does not differentiate between large areas of high uniformity, like the chest, and smaller areas which are more crucial to a precise fitting, e.g. the arms. However, it should be noted that the reference approach allows for higher framerates on current hardware. As that algorithm is based on standard graphics operations, modern graphics hardware can maximize its performance easily.

3.3. Collision Detection in Ambiguity Handling

The effect of collision detection becomes apparent studying Figure 6. The left plot (without collision penalties) shows a ridge around joint angles placing the right upper arm and hand near the left arm. In the right plot, the collision detection enacts a penalty for any pose placing the right arm within the volume of the left arm (see Equation 14), resulting in a distinct trough for angles placing the right arm in an illegal position. Although the trough does not totally eliminate the ridge and a small local maximum, it generates a more distinct gradient towards the global maximum.

3.4. Clustering and Self-Occlusions

In Figure 7 only one shoulder angle and the elbow angle of the nearly obscured right arm are varied. No model point clustering was performed for the left plot. Since any pose bringing many unoccluded model points close to the clustered data points leads to higher average likelihood scores in Equation 9, there is a high ridge for shoulder angles bending the arm forwards, towards the visible data points. This resolution would obviously be wrong, as there are in fact no observations indicating the arm being visible. However, as proximity of model points to observed points grants higher similarity scores, the correct occluded pose gets lower scores than the incorrect poses placing the arm closer to observed points. We can conclude that without clustering of the model points by body elements, the averaging of the likelihood score in Equation 9 leads to a diminuished influence of evidence proximity to model points.

Using configurations (4) and (5), partial scores for occluded body parts are set to 1 (see Section 2.3). Thus, scores for poses placing the arm behind the body are not negatively influenced. Simultaneously, the gain from placing a few model points closer to the observed data points is offset by an overall lower partial likelihood score for single body elements. This effect is visualized by the large range of likely angles for the single shoulder and elbow joints, which though undetermined do not contradict the actual observation. The model point clustering thereby ensures the correct handling of occlusions by flagging not-visible body parts as weighting-neutral. In the absence of further data all poses not contradicting the observation are assumed as equally likely. This is founded on the separation between observation likelihoods and the motion model as specified in Section 2.1.

3.5. Performance using CUDA

To evaluate the potential for real-time tracking applications, the proposed observation likelihood function has been implemented in CUDA and used as a weighting function within an APF based upper body pose tracker (derived from [10]). It was found that the implementation evaluated 300 sample poses with 288 points each against an average of 550 observed points in less than 25 ms. The machine was

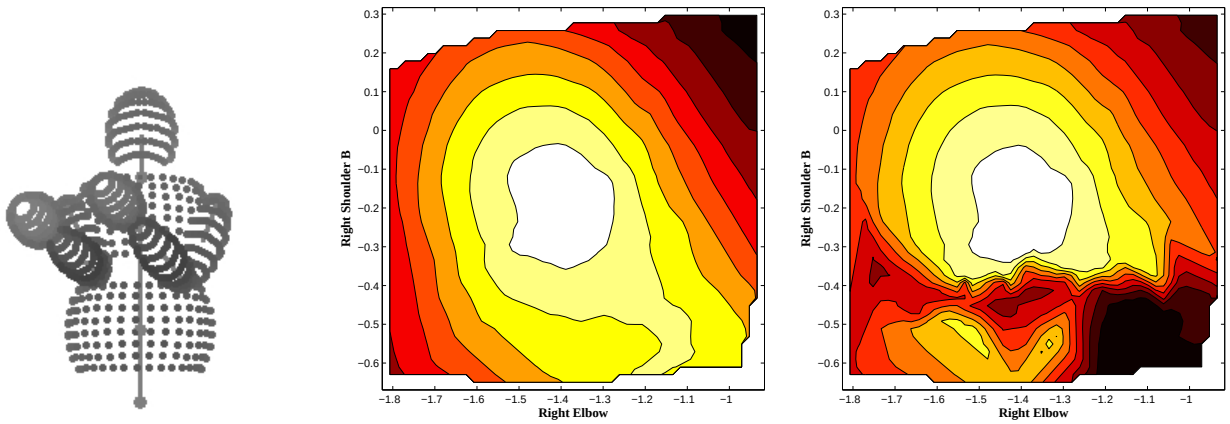

Figure 6. (left) reference pose with two arms in close proximity, (middle) likelihood over joint angles without collision penalty, (right) likelihood over joint angles with collision penalty. Brighter areas indicate higher likelihood.

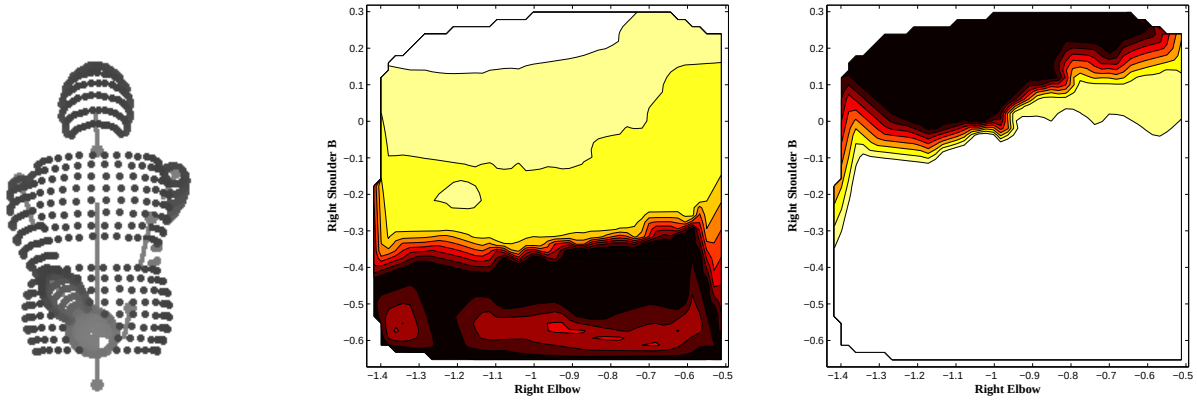

Figure 7. (left) reference pose with the articulated right arm nearly obscured, (middle) likelihood considering only minimal point distances with clustered data points, (right) likelihood after clustering of both reference and model points. Brighter areas indicate higher likelihood.

a 2.66 GHz Intel Core2Quad CPU with 4 GB RAM and a NVIDIA Geforce GTX 275 graphics card. It should be noted that this implementation used brute force to find nearest neighbors. More efficient implementations using octrees or kd-trees should enable even faster evaluation.
The tracker itself performed well, tracking hand and arm movements of varying complexity, as shown in Figure 8. Although a detailed evaluation of the tracker performance would exceed the scope of this paper (i.e. observational likelihood approximation), preliminary results appear quite promising and indicate significant improvements over the previously used weighting function.

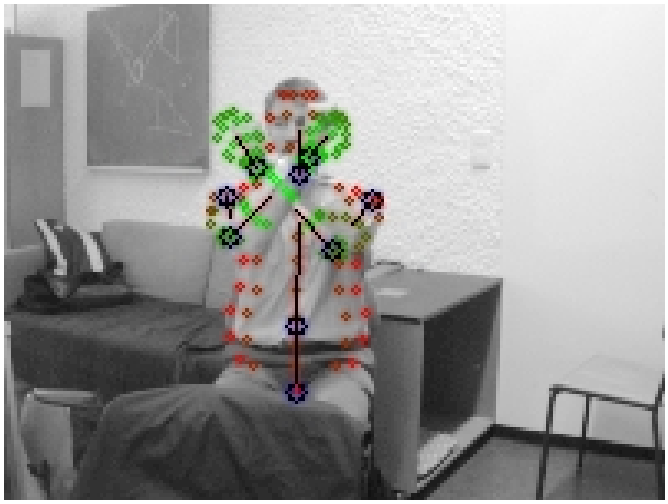

Figure 8. Scene from upper body tracking: Both arms crossed in front of face, color coding shows point distance from camera. No further body part detection is used, the tracking relies solely on the proposed likelihood approximation.

4. CONCLUSION

The observation likelihood approximation proposed and evaluated in this paper has shown promising performance under test-bench conditions for both synthetic and real-world datasets. The focus on using solely 3D data as evidence allows for likelihood estimation on a wide range of sensors, while the exclusion of color cues makes the system lighting-independent. Our proposed approach compared fa-

vorably against the approximation function used by a state-of-the-art system with regards to accuracy, if not speed. It is also interesting to note the large impact segmentation and clustering had on approximation accuracy (see Section 3.1). By giving smaller, more salient regions equal weight as larger, less salient regions, approximation accuracy can be boosted and marginalization effects are reduced.

The ellipsoid body model allows for fast sample generation. The seamless integration of the occlusion handling by model point clustering produces good results even for nearly obscured limbs, preventing misplacements and constraining regions of likelihood to comply with available observations. The collision penalties have been shown to detect and effectively suppress impossible poses, although a smoother likelihood gradient in the border state-space would be desirable. Further attention should be given to methods for automatically adjusting the ellipsoid model to the current user in order to improve precision. In addition, a more efficient computation of nearest-neighbor point pairs is expected to yield vastly reduced computational overhead.

The proposed observation likelihood approximation is adaptable to nearly all current stochastic tracking approaches, presuming the availability of depth information. With only slight modifications, the same method may be adapted to calibrated multi-camera scenarios.

References

[1] ASUSTeK Computer Inc. Wavi Xtion. 1

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

[3] O. Bernier, P. Cheungmonchan, and A. Bouguet. Fast nonparametric belief propagation for real-time stereo articulated body tracking. *Computer Vision and Image Understanding*, 113(1):29–47, July 2008. 1, 2

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

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

[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, IEEE/RSJ International Conference on*, pages 3391–3396, San Diego, CA, October 2007. 1

[7] M. Fontmarty, F. Lerasle, and P. Danes. Likelihood tuning for particle filter in visual tracking. In *Image Processing, 16th IEEE International Conference on*, pages 4101 –4104, November 2009. 1

[8] V. Ganapathi, C. Plagemann, D. Koller, and S. Thrun. Real time motion capture using a single time-of-flight camera. In *Computer Vision and Pattern Recognition, 23rd IEEE Conference on*, pages 755 –762, May 2010. 1, 4, 5, 6

[9] M. Isard and A. Blake. Condensation - conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, August 1998. 1, 2

[10] N. H. Lehment, D. Arsić, and G. Rigoll. Cue-independent extending inverse kinematics for robust pose estimation in 3d point clouds. In *Image Processing , IEEE International Conference on*, Hong Kong, China, September 2010. 6

[11] J. Lichtenauer, M. Reinders, and E. Hendriks. Influence of the observation likelihood function on particle filtering performance in tracking applications. In *Automatic Face and Gesture Recognition, 6th IEEE International Conference on*, pages 767 – 772, May 2004. 1

[12] Microsoft Corp. Redmond WA. Kinect for Xbox360. 1

[13] R. Poppe. Vision-based human motion analysis: An overview. *Computer Vision and Image Understanding*, 108(1-2):4 – 18, November 2007. 1

[14] R. B. Rusu and S. Cousins. 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011. 1

[15] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from a single depth image. In *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, Colorado Springs, USA, June 2011. IEEE. 1

[16] J. Wilhelms and A. V. Gelder. Efficient spherical joint limits with reach cones. Technical report, 2001. 2

[17] Y. Zhu and K. Fujimura. Bayesian 3D human body pose tracking from depth image sequences. In *Computer Vision, 9th Asian Conference on*, pages 267–278, September 2009. 1