Tracking of the Articulated Upper Body on Multi-View Stereo Image Sequences

Julius Ziegler
julius.ziegler@stud.uni-karlsruhe.de

Kai Nickel
nickel@ira.uka.de

Rainer Stiefelhagen
stiefel@ira.uka.de

Interactive Systems Laboratories
Universität Karlsruhe (TH)
Germany

Abstract

We propose a novel method for tracking an articulated model in a 3D-point cloud. The tracking problem is formulated as the registration of two point sets, one of them parameterised by the model's state vector and the other acquired from a 3D-sensor system. Finding the correct parameter vector is posed as a linear estimation problem, which is solved by means of a scaled unscented Kalman filter. Our method draws on concepts from the widely used iterative closest point registration algorithm (ICP), basing the measurement model on point correspondences established between the synthesised model point cloud and the measured 3D-data. We apply the algorithm to kinematically track a model of the human upper body on a point cloud obtained through stereo image processing from one or more stereo cameras. We determine torso position and orientation as well as joint angles of shoulders and elbows. The algorithm has been successfully tested on thousands of frames of real image data. Challenging sequences of several minutes length where tracked correctly. Complete processing time remains below one second per frame.

1. Introduction

Image based, marker-less articulated body tracking has become an active field of research in the area of computer vision within the last decade. Detailed body tracking is useful since it facilitates automatic analysis of human motion. It can be used to provide features to a classification backend for gesture recognition, making novel methods of human-computer interaction possible. Other applications not requiring classification include motion capture, for example to directly teach motion to human-like robots.

In the following, we will give a short overview of seminal contributions to the topic of articulated body tracking and comment on the novelty of our approach compared to them.

Some approaches [4, 11] to articulated tracking utilise a particle filter (condensation filter) for tracking. This probabilistic framework employs Monte Carlo approximation to track the state vectors probability density. The main problem in its application to articular body tracking is the high number of degrees of freedom, since the required number of Monte Carlo samples (*particles*) rises exponentially in the dimensionality of the state space.

Deutscher *et al*. [4] refine the particle filtering scheme by introducing additional resampling steps, evading local minima of the state probability in a manner similar to simulated annealing. With this modification, they can track full body motion with as little as 100 particles. The particles are weighted by rendering synthetic silhouette and edge images from every particle. These are compared with silhouette and edges extracted from the image sequence.

Lee *et al*. [11] effectively reduce the dimensionality of the state probability distribution by analytically incorporating the results of feature detectors (head, hands, body main axis) into the particle representation. The weighting function is based on silhouette extraction.

Approaches exploiting 3D-data for articulated tracking are relatively rare. Jojic *et al*. [7] present an algorithm that tracks articulated motion in disparity maps by modelling body parts as 3D-point clusters and tracking their respective second order stochastic moments.

Kehl *et al*. [10] propose an algorithm which processes 3D-data obtained from volumetric reconstruction of at least four camera views. They state articulated tracking as the minimisation of a scalar objective function. This is accomplished using a gradient descent method with local step size adaption. The objective function is not minimised with respect to all degrees of freedom at once, but for the torso and the single extremities consecutively.

Mikic *et al*. [13] exploit volumetric reconstruction to obtain 3D data as well. The algoithm identifies single body parts and filters their position and orientation using an extended Kalman filter (EKF) to assert that kinematic constraints are met.

Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)
0-7695-2597-0/06 $20.00 © 2006 IEEE

Urtasun and Fua [16] formulate full body tracking as a minimisation problem as well. Here, 3D-data is obtained by stereo image processing. The focus is on the application of strong temporal motion models to guide the minimisation process. This currently restricts the algorithm to tracking of some distinctive classes of motion, like walking and running.

Demirdjian and Darrell [3] apply the iterative closest point algorithm to fit models of single limbs to a point cloud acquired from stereo image sequences. In this phase, limbs are treated as uncoupled rigid objects. Kinematic constraints resulting from the human body are impressed on them subsequently.

Our approach is based on a kinematic model, so that kinematic constraints of the human body are met implicitly. All degrees of freedom are tracked at once, as opposed to the hierarchical scheme used in [10]. We believe this to be advantageous, since all body parts mutually constrain each other in every phase of the algorithm. We also found out, that fitting the torso position independently of the limbs is difficult, as the torso on its own has too few features and relatively indistinct principal moments. In contrary to [16], we do not constrain the range of feasible motions by strong motion models.

We obtain the 3D-data used for tracking from disparity images, which are obtained by stereo image processing. Disparity images are insusceptible to lighting changes and shadows. They allow for straight forward, robust foreground segmentation. In contrast, volumetric reconstruction as applied in [10] heavily relies on conventional foreground segmentation techniques.

Like particle filtering approaches, our algorithm is formulated within a probabilistic, Bayesian framework. More concretely, the tracking problem is posed as a linear estimation problem and solved by an unscented Kalman filter. Consequently, the state probability density is not represented by a large set of Monte Carlo samples, but by the compact unscented representation. The unscented representation captures mean and variance of the state probability density by maintaining a small set of deterministically chosen samples. The unscented Kalman filter avoids the necessity to compute the Jacobian matrices of the model functions, so it allows us to use a complex measurement model without complicating filter design. The employed measurement model is inspired by the iterative closest point (ICP) algorithm for point cloud registration. It relies on point correspondences that are established between a model surface and the measured 3D-data based on spatial proximity. ICP in its original form is tailored to the registration of rigid objects. By integrating the ICP algorithm with an unscented Kalman filter, we yield a novel registration algorithm capable of tracking articulated structures.

This article is structured as follows: Section 2 addresses established concepts this contribution builds upon. Concretely, the ICP algorithm and the unscented Kalman filter are presented. Section 3 describes the preprocessing steps used to obtain the point cloud that serves as input to our algorithm. These include stereo processing and segmentation. Section 4 describes the core aspect of our algorithm, namely the formulation of registration as a linear estimation problem. Section 5 presents experimental results, including an evaluation in terms of tracking precision. We conclude the article with a short summary in section 6.

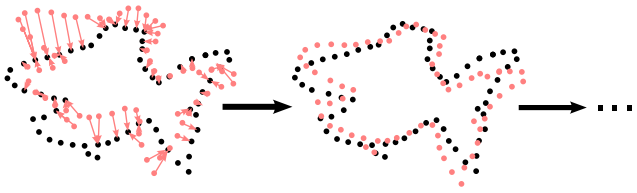


Figure 1. Illustration of ICP registration algorithm.

2. Theoretical Background

2.1. Iterative Closest Point (ICP)

The *iterative closest* (also: *corresponding*) *point* (*ICP*) algorithm is a widely applied method for the registration of two datasets. For the following description, we will assume that the datasets are Euclidian point sets.

Let X denote the fixed model point set and P the adjustable data point set that is to be registered with it. The algorithm returns a rigid body transformation θ that aligns P to an optimal fit with X. The algorithm works as follows (*cf.* [2]):

Repeat the following steps until termination:

1. Initialisation:
   $k = 0, \theta_0 = id, P_0 = P$

2. Repeat steps a, b, c and d until termination:

   (a) For every point in P_k compute the closest point in X, yielding the set of correspondents Y_k.

   (b) Compute the local registration θ_k that minimises the summed square distances between the points in P_k and their correspondents in Y_k.

   (c) Apply the registration: $P_{k+1} = \theta_k(P_0)$.

   (d) If the change of the summed square distances drops below the threshold τ that has been chosen according to the desired precision, terminate with result θ_k.

See figure 1 for an illustration.

Several closed form solutions exist for the calculation of local registration in step b of the algorithm. The most common ones are either based on singular value decomposition [1] or on a representation in quaternion space [6]. Both are

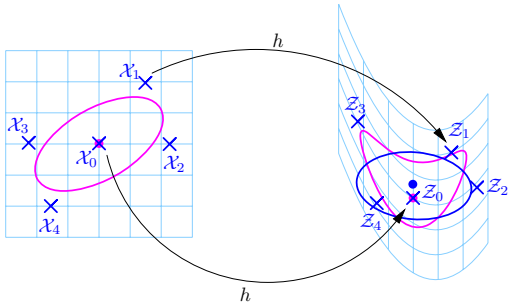


Figure 2. Unscented Transformation of $\hat{\mathbf{x}}$ and $\mathbf{P}$ through a nonlinear function h. Weighted averaging of the transformed σ-points yields an approximation of the transformed probability distribution.

restricted to the case of registering rigid bodies and not applicable to articulated structures. Therefore, we restated the local registration as a linear estimation problem, so we can use an unscented Kalman filter to solve it.

2.2. Unscented Kalman Filter (UKF)

The Kalman filter [12] is a tool to estimate a state vector that can only be observed through indirect measurements which are subject to noise. The functional dependency between a state vector $\mathbf{x}$ and the measurement vector $\mathbf{z}$ is modelled by the function h: $\mathbf{z} = h(\mathbf{x})$. All deviations from the expected behaviour are summarised and described in terms of their covariance matrix $\mathbf{R}$. The Kalman filter allows to incorporate knowledge of system dynamics into the estimation in form of the functional system model f. f models the state transition from one discrete time instant k to the next: $\mathbf{x}_{k+1} = f(\mathbf{x}_k)$. Errors of this model are again summarised by their covariance matrix, termed $\mathbf{Q}$.

In the following, we will use $\hat{\mathbf{x}}[i|j]$ to denote the state vector's estimated mean at time instant i, given all measurements up to time instant j. The notation $\mathbf{P}[i|j]$ for the covariance matrix of the state estimate is to be interpreted accordingly.

The general Kalman filtering framework consist of the iteration of the following steps. k is the discrete time index and is incremented with every iteration:

- **Prediction:** Extrapolate the a priori estimate $\hat{\mathbf{x}}[k+1|k]$ and its covariance matrix $\mathbf{P}[k+1|k]$ from the optimal state estimate $\hat{\mathbf{x}}[k|k]$ and its covariance matrix $\mathbf{P}[k|k]$, by propagating it through the functional system model f. The covariance matrix $\mathbf{Q}$ of the system noise will be incorporated into $\mathbf{P}[k+1|k]$ as an additive component. $\hat{\mathbf{x}}[k+1|k]$ and $\mathbf{P}[k+1|k]$ are subsequently propagated through the measurement model h, yielding a prediction for the expected measurement, $\hat{\mathbf{z}}[k+1|k]$, and its covariance $\mathbf{P}_{\mathbf{zz}}[k+1|k]$. The deviation of the actual measurement $\mathbf{z}[k+1]$ from the expected measurement, $\nu[k+1] = \mathbf{z}[k+1] - \hat{\mathbf{z}}[k+1|k]$, is called the *measurement residual*. The variance of this quantity $\mathbf{P}_{\nu\nu}$ is obtained by adding the covariance matrix $\mathbf{R}$ of the measurement noise: $\mathbf{P}_{\nu\nu}[k+1] = \mathbf{P}_{\mathbf{zz}}[k+1|k] + \mathbf{R}$.

- **Update:** By incorporating the measurement $\mathbf{z}[k+1]$, the extrapolated value $\hat{\mathbf{x}}[k+1|k]$ is corrected to yield $\hat{\mathbf{x}}[k+1|k+1]$. The error covariance of the corrected estimate, $\mathbf{P}[k+1|k+1]$, is minimised. The state update equations are:

$$\hat{\mathbf{x}}[k+1|k+1] = \hat{\mathbf{x}}[k+1|k] + \mathbf{i}[k+1] \quad (1)$$
$$\mathbf{P}[k+1|k+1] = \mathbf{P}[k+1|k] - \mathbf{I}[k+1] \quad (2)$$

with the two correctional terms

$$\mathbf{i}[k+1] = \mathbf{K}[k+1]\nu[k+1] \quad (3)$$
$$\mathbf{I}[k+1] = \mathbf{K}[k+1]\mathbf{P}_{\nu\nu}[k+1|k]\mathbf{K}[k+1]^{\mathrm{T}} \quad (4)$$

$\mathbf{K}[k+1]$ is called the *Kalman-Gain*. It is calculated according to the following equation:

$$\mathbf{K}[k+1] = \mathbf{P}_{xz}[k+1|k]\mathbf{P}_{\nu\nu}^{-1}[k+1|k] \quad (5)$$

$\mathbf{P}_{xz}[k+1|k]$ is the cross covariance matrix describing the linear dependence of the random variables $\mathbf{x}$ and $\mathbf{z}$.

As we can see, filtering requires the propagation of mean $\hat{\mathbf{x}}$ and variance $\mathbf{P}$ of the state estimate through the functional models f and subsequently h. With the latter propagation (through the measurement model h), we additionally need to calculate the covariance between the original and the transformed random variable (cross covariance). The simple Kalman filter allows linear models only, so that the propagation can be expressed in terms of matrix multiplications. The same is true for the widely used extended Kalman filter (EKF), since it linearises the models around the mean of the state estimate by building their Jacobian matrices.

The unscented Kalman filter (UKF) as proposed by Julier and Uhlman [8] exploits the unscented transformation to propagate mean and covariance through arbitrary functions. The Unscented representation of $\mathbf{P}$ is the set of column vectors from a positive and a negative scaled matrix square root:

$$\{\sigma_1, ..., \sigma_n\}_{\underleftarrow{\text{column vectors}}} \pm \sqrt{(n+\kappa)\mathbf{P}}. \quad (6)$$

n is the dimensionality of the state space, κ a scaling factor to be described later. The obtained set of vectors σ_i has zero mean, so adding $\hat{\mathbf{x}}$ to each of the σ_i results in a set $\{\mathcal{X}_1, ..., \mathcal{X}_{2n}\}$with mean $\hat{\mathbf{x}}$. $\hat{\mathbf{x}}$ itself is inserted into the set to yield the set of the $2n+1$ so called σ-points $\{\mathcal{X}_0, ..., \mathcal{X}_{2n}\}$ with

$$\mathcal{X}_0 = \hat{\mathbf{x}} \quad (7)$$
$$\mathcal{X}_i = \sigma_i + \hat{\mathbf{x}}. \quad (8)$$

The σ-points completely capture mean and covariance of the original distribution. By individually transforming them through the model function with

$$\mathcal{Z}_i = h(\mathcal{X}_i), \tag{9}$$

one yields the set of transformed σ-points, $\{\mathcal{Z}_0, ..., \mathcal{Z}_{2n}\}$, which approximates the transformed distribution. Through weighted averaging, mean $\hat{\mathbf{z}}$ and variance $\mathbf{P_{zz}}$ of the transformed distribution can be reconstructed. See figure 2 for a two dimensional example. Averaging is done as follows:

$$\hat{\mathbf{z}} = \frac{1}{n+\kappa}\{\kappa\mathcal{Z}_0 + \frac{1}{2}\sum_{i=1}^{2n}\mathcal{Z}_i\}, \tag{10}$$

$$\mathbf{P_{zz}} = \frac{1}{n+\kappa}\{\kappa[\mathcal{Z}_0 - \hat{\mathbf{z}}][\mathcal{Z}_0 - \hat{\mathbf{z}}]^\top + \frac{1}{2}\sum_{i=1}^{2n}[\mathcal{Z}_i - \hat{\mathbf{z}}][\mathcal{Z}_i - \hat{\mathbf{z}}]^\top\}. \tag{11}$$

The cross covariance matrix is obtained by averaging the direct products of the original and the transformed points:

$$\mathbf{P_{xz}} = \frac{1}{n+\kappa}\{\kappa[\mathcal{X}_0 - \hat{\mathbf{x}}][\mathcal{Z}_0 - \hat{\mathbf{z}}]^\top + \frac{1}{2}\sum_{i=1}^{2n}[\mathcal{X}_i - \hat{\mathbf{x}}][\mathcal{Z}_i - \hat{\mathbf{z}}]^\top\}. \tag{12}$$

The parameter κ allows to arbitrarily "scale" the distribution of the σ-points around the mean. Bringing the σ-points close to the mean is desirable in order to avoid sampling non-local effects of the respective model function. This is especially true for trigonometric functions as used here to describe articulated body kinematics. In [9], Julier and Uhlman refine the concept of scaling, additionally guaranteeing positive semi definiteness for the transformed covariance matrix. Their method has been adopted for this work.

3. Preprocessing and Segmentation

The input to our tracking algorithm is a 3D point cloud representing the appearance of the human body. It is obtained from the raw stereo data through several preprocessing and segmentation steps, to remove points belonging to the background.

The first preprocessing step is disparity computation. It is accomplished using a commercially available stereo system.[1] Usually, disparity computation yields many invalid values in low-textured or very dark image regions. These regions can be identified by an edge detection filter. Figure 3(b) shows an example of a disparity map, with invalidated pixels displayed in black.

Disparity maps allow extracting foreground regions in a very straight forward way. Each pixel of an acquired disparity image is compared to the corresponding pixel in a background model. If the current disparity is greater than that in the model, the pixel must belong to the foreground. The background model can be averaged over the past frames with an α-process to allow slow adaption of changes in the scenery. The segmentation method used here is similar to that proposed by Eveland *et al.* [5]. Disparity maps are insensitive to lighting changes, and so is the implemented foreground segmentation. Figure 3(c) shows an example.

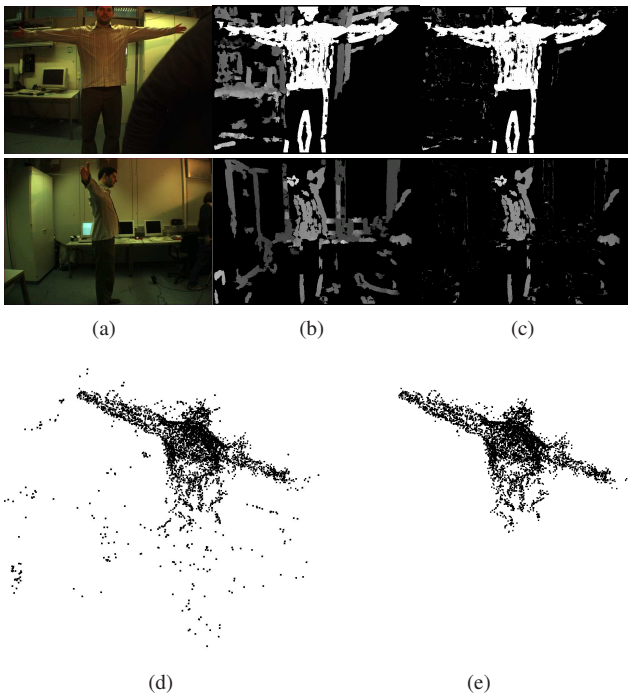
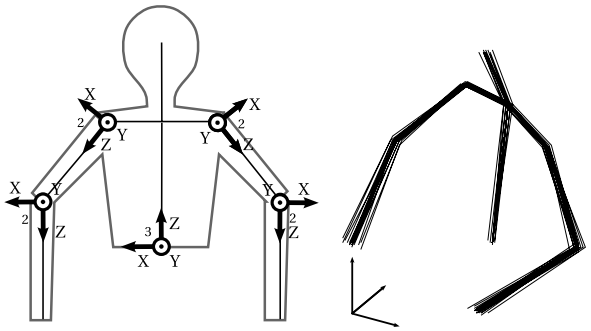


Figure 3. Preprocessing of stereo images. (a) Original images. (b) Disparity maps. (c) Result of foreground segmentation. (d) Joint point cloud in a global coordinate system. (e) Point cloud after application of noise filter.

The 3D-position $\mathbf{x}_{cam} = \begin{pmatrix} x & y & z \end{pmatrix}$ of the associated point can be reconstructed from the image coordinates of a pixel and its disparity. The obtained position $\mathbf{x}_{cam}$ is relative to a coordinate system originating in the principal point of the respective camera. Multiplication with a camera specific homogenous transformation matrix $\mathbf{T}_{cam}$ yields a 3D point in a common world coordinate system, allowing the precise fusion of the 3D data of all cameras. The $\mathbf{T}_{cam}$ matrices are determined through a conventional camera calibration procedure using a checkerboard pattern. Our algorithm will process the fused point cloud, so tracking over multiple cameras is implemented implicitly. Figure 3(d) shows the result of fusing pre-segmented 3D-data from two stereo cameras. It reveals a characteristic weakness in the foreground segmentation which results in artefacts having the form of small, isolated groups of points. These are removed by a subsequent filtering stage, yielding the final point cloud to be used for tracking (Figure 3(e)).

[1] Triclops system [15], developed by Point Grey.



(a) (b)

Figure 4. (a): Kinematic model of the upper body. Each of the indicated joints is represented by a rotation matrix, which is parameterised by the depicted number of degrees of freedom. Together with the translation vector describing the position of the torso in space, we yield a total of 14 degrees of freedom. (b) Unscented representation of model state and its uncertainty. Here, it has been depicted by rendering a stick figure for each of the 29 σ-poses.

Depending on number and setup of cameras, the obtained point cloud representation of the body may be incomplete due to occlusions. This fact is accounted for in the measurement model, see section 4.2.

4. Tracking of Articulated Objects in 3D Point Clouds

Kalman filtering requires the description of both, system dynamics and the input/output-relation of the measurement system, as functional models. Since we use the unscented variant of the Kalman filter, we neither need to supply the models in a linear form, nor do we need to compute their Jacobian matrices as we would have to with the commonly used extended Kalman filter. This simplifies filter design considerably and allows us to use a complex measurement model.

4.1. System Model

The state vector $\mathbf{x}$ comprises the 14 degrees of freedom depicted in figure 4(a). The state vector's evolution is merely modelled as a random walk, without incorporating higher order dynamics. This accounts for the fact that the frame rate of the video system used is relatively low ($\sim$10 Hz) compared to the frequency of human motion, so that modelling of velocities or even accelerations could easily lead to errors due to under sampling. Consequently, the model function is the identity: $f(\mathbf{x}) = \mathbf{x}$. Hence, a change in the state from frame to frame is actually modelled as error and comprised in the system covariance matrix $\mathbf{Q}$. Figure 4(b) shows the unscented representation of the model state consisting of 29 σ-points. We will also speak of *σ-poses*, since the σ-points are sampled from the model's pose space.

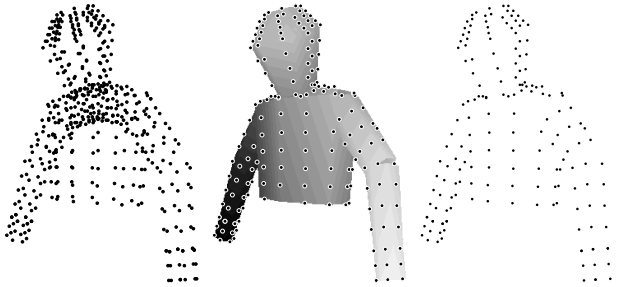


Figure 5. Removing occluded model points with the z-buffer-algorithm. Each point is compared against the z-buffer and deleted if its z-value is greater than the correspondig value in the z-buffer.

4.2. Measurement Model

For generating the measurement vector and evaluating the measurement model, we have implemented a procedure to render a point cloud depicting the upper body taking up the pose equivalent to a specific state $\mathbf{x}$.

As we want the generated point cloud to resemble as good as possible the point cloud as it would be output from the stereo system we only render points visible in at least one of the cameras. Removing occluded points is crucial for adapting the system to different camera setups. Thus, adding and removing new views becomes possible easily. The problem of occlusion is handled with the z-buffer algorithm. This algorithm is commonly used in computer graphics and fast implementations are ubiquitous and assisted by standard graphics hardware. It works by maintaining a depth map (*z-buffer*) while rendering a volumetric representation of the model from 3D-primitives (polygons, in our case). [17] elaborates further on the z-buffer algorithm. Figure 5 shows how points are tested for occlusion by comparing their projections against the z-buffer. We term the point cloud generated for a state $\mathbf{x}$ as $\mathcal{G}(\mathbf{x})$.

For generating the measurement vector, we render the point cloud $P_{\hat{\mathbf{z}}} = \mathcal{G}(\mathbf{x}[k+1|k])$ from the predicted state. Then we establish correspondences between $P_{\hat{\mathbf{z}}}$ and the point cloud acquired from the stereo system, yielding the point set $P_{\mathbf{z}}$ (see figure 6(a)).

We generate the measurement vector $\mathbf{z}$ by projecting every element of $P_{\mathbf{z}}$ onto the line connecting it with its corresponding point in $P_{\hat{\mathbf{z}}}$. Every element of $P_{\mathbf{z}}$ yields one scalar component of $\mathbf{z}$. Equivalently, we generate the predicted measurement $\hat{\mathbf{z}} = h(\hat{\mathbf{x}}[k+1|k])$ from the point set $P_{\hat{\mathbf{z}}}$. As it can be seen from figure 6(b), the difference of $\mathbf{z}$ and $\hat{\mathbf{z}}$, called the *measurement residual* or *innovation* ν within the Kalman filtering framework, comprises the distances between corresponding points.

Since we use the Unscented transformation to propagate $\hat{\mathbf{x}}[k+1|k]$ and $\mathbf{P}[k+1|k]$ through the measurement model, we additionally need to evaluate h at the σ-points distributed around $\hat{\mathbf{x}}[k+1|k]$. This is done analoguely to

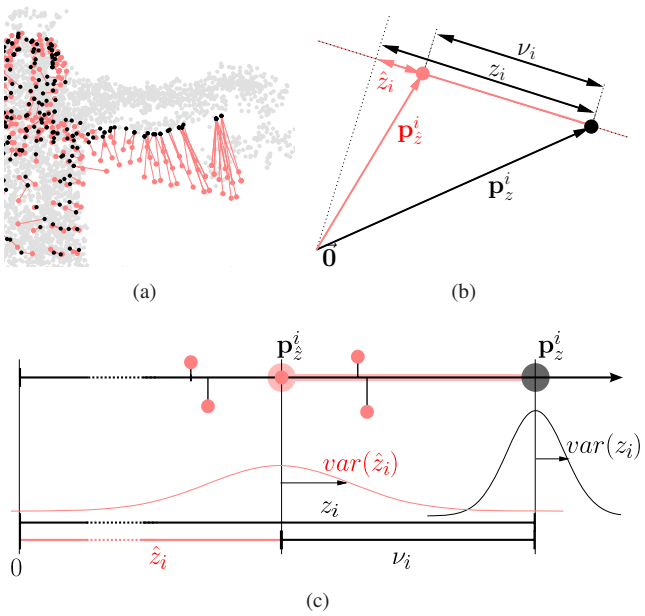


Figure 6. Generation of measurement vector $\mathbf{z}$ and predicted measurement $\hat{\mathbf{z}}$. (a) shows the predicted point cloud $P_{\hat{\mathbf{z}}}$ in red, the point cloud obtained from stereo processing in light grey and the points corresponding to the prediction, $P_{\mathbf{z}}$, in black. (b) shows the generation of one scalar component of $\mathbf{z}$ and $\hat{\mathbf{z}}$, respectively. (c) schematically shows the propagation of the state uncertainty through this measurement model by means of the unscented transformation, see the text for further explanation (the time index $[k+1|k]$ has been omitted here).

the calculation of $\hat{\mathbf{z}}$. For the sake of computational cost, correspondences are not recalculated for the σ-points, but are based on the predicted state $\hat{\mathbf{x}}[k+1|k]$ only. The same applies for the occlusion test.

Figure 6(c) illustrates, how the state's unscented representation is used to predict the probability distribution of the measurement vector. The axis $\overline{\mathbf{p}^i_{\hat{z}}\mathbf{p}^i_z}$ from figure 6(b) has been oriented towards the abscissa. The small red points were obtained by sampling the same model point from all σ-poses representing the predicted state and its variance (compare figure 4(b)). Only five out of these 29 points are actually depicted. They are projected onto $\overline{\mathbf{p}^i_{\hat{z}}\mathbf{p}^i_z}$, and subsequently, the projections are averaged according to equations 10 and 11 to yield mean $\hat{\mathbf{z}}[k+1|k]$ and variance $\mathbf{P_{zz}}[k+1|k]$ of the respective component of the measurement prediction (both depicted in red). The variance of $\mathbf{z}$ (depicted in black) is the measurement noise $\mathbf{R}$. Note that figure 6(c) shows the transformation for one component of the measurement vector only. Actually, the transformation is done for all model points at once, yielding the multidimensional vector $\hat{\mathbf{z}}[k+1|k]$ and the covariance matrix $\mathbf{P_{zz}}[k+1|k]$. ν_i in figure 6(c) is the component of the measurement residual generated from the correspondence $(\mathbf{p}^i_{\hat{z}}\ \mathbf{p}^i_z)$. As described above, it contains the distance of the corresponding points.

Figure 7 concretises the meaning of cross covariance

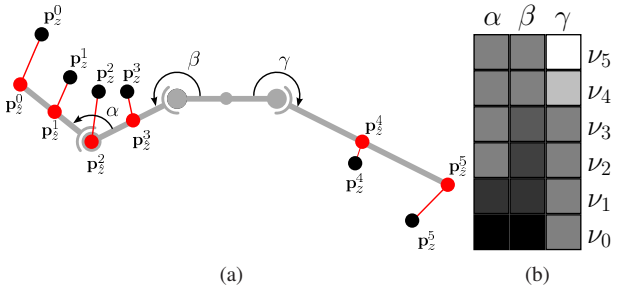


Figure 7. Explanation of cross covariance considering a simple articular model with the three dimensional state vector $\mathbf{x} = (\alpha\, \beta\, \gamma)^{\mathrm{T}}$. The cross covariance is a measure for the linear dependence of state vector and measurement residual ν. Given the correspondences $(\mathbf{p}^i_{\hat{z}}\ \mathbf{p}^i_z)$ depicted left, the resulting cross covariance $\mathbf{P_{xz}}$ is qualitatively displayed on the right. White means strong negative correlation, black means strong positive correlation. Medium grey fields indicate no correlation.

within the algorithm. The cross covariance matrix $\mathbf{P_{xz}}$ is created from the σ-poses $\{\mathcal{X}_0, ..., \mathcal{X}_{28}\}$ and their transformed equivalents $\{\mathcal{Z}_0, ..., \mathcal{Z}_{28}\}$ with equation 12.

Finally, we calculated all quantities ($\mathbf{P_{zz}}$, $\mathbf{P_{xz}}$ and ν) to perform the state update according to equations 1 - 5. We use multiple iterations of update steps, as we expect the correspondences to improve with every iteration. This proceeding has been inherited from the ICP algorithm.

Instead of sampling the model points deterministically like depicted in figure 5, we resample model points stochastically at every iteration. This prove to advance tracking performance, especially if few model points where used. This finding conforms with [10].

Tracking with the described method is susceptible to situations where body parts touch or come very close to each other. Point correspondences are based on spatial proximity, so wrong correspondences can easily occur in these situations. As a remedy, we detect these problematic configurations and artifically push the respective limbs apart by applying simple inverse kinematics. This proceeding additionally asserts that geometric constraints of the human body are met, and overlap of body parts is avoided.

5. Experimental Results

The tracking system has been tested on four image sequences with four different persons. Each of the sequences spans several minutes during which the subjects performed unconstrained, fully articulated motions while walking around, taking turns and bending over. The total number of frames collected was 6029. Recordings where done with four synchronised stereo cameras facing each other in a cross over setup, with a distance to the opposite camera of approximately 5 m. All sequences where tracked successfully over the full length, without divergence. For the results shown here, 64 vertices where sampled from the

torso position [cm]	7.0
torso longitudinal [°]	8.8
torso transverse [°]	12.8
upper arm [°]	22.7
lower arm [°]	25.7

Table 1. Mean tracking errors averaged over all 6029 frames.

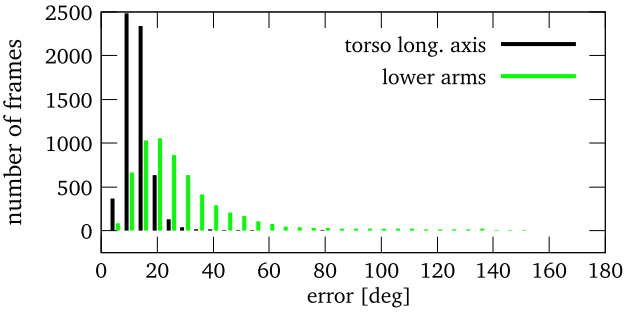


Figure 8. Distribution of tracking errors for torso longitudinal axis and lower arms.

body model. Full processing time for one frame remained below one second.

Tracking precision was evaluated by comparision against a ground truth that was obtained by manual labeling. Distinctive points of the body (hands, ellbows, shoulders, head centroid and two points on the hips) where labeled in every frame on all camera views. The kinematic upper body model was fitted to these labels in order to obtain a reference trajectory. The reference trajectory was used to calculate the mean error of torso position, torso longitudinal axis (hip to head), torso transverse axis (shoulder to shoulder) and angular errors of upper and lower arms. See table 1 for the results. Figure 8 shows the distribution of tracking errors as a histogram for the lower arm and one of the torso axes.

Figure 9 shows some short scenes taken from the full length sequences. Note upper right scene, in which the person bends over to tie his shoe strings. The subject is temporarily almost completely outside of the front camera's field of view. Though tracking errors get high during this scene, the tracker recovers nicely.

6. Summary and Conclusions

We proposed a novel method for fast tracking of articulated structures within 3D sensor data. It builds upon concepts from the ICP rigid body registration algorithm, but integrates an unscented Kalman filter in order to allow tracking of articulated structures. The algorithm was applied to track the human upper body kinematically in 3D data that was acquired from four stereo cameras. Tracking of an upper body model with 14 degrees of freedom was achieved with a frame rate of less than a second. The correct track was maintained over challenging, several minute long sequences. Tracking precision was evaluated by comparison against a ground truth trajectory obtained by manual labeling.

For the future, we plan to fuse the tracking system with a feature tracker [14] that is capable of extracting hand and head positions from the stereo image sequences. It can be integrated seamlessly into the existing stochastic framework by formulating an additional functional measurement model. The feasibility of this approach has already been proven by incorporating hand and head positions obtained from manual labeling into the pose estimate. The results are very promising, surpassing the precision achieved here even with a one camera setup.

Acknowledgements

This work has been funded by the German Research Foundation (DFG) as part of the Sonderforschungsbereich 588 "Humanoide Roboter", and by the European Union under contract no. FP6-IST-506909 (Project CHIL).

References

[1] Huang T.S. Blostein S.D. Arun, K.S. Least-squares fitting of two 3d point sets. *Transactions on Pattern Analysis and Machine Intelligence*, 9(5):698–700, 1987.

[2] P. Besl and N. McKay. A method for registration of 3-d shapes. *Transactions on Pattern Analysis and Machine Intelligence*, 14(2), 1992.

[3] D. Demirdjian and T. Darrell. 3-d articulated pose tracking for untethered diectic reference. In *Proceedings of the 4th IEEE International Conference on Multimodal Interfaces*, page 267, 2002.

[4] J. Deutscher, A. Blake, and I. Reid. Articulated body motion capture by annealed particle filtering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2126–2133, 2000.

[5] C. Eveland, K. Konolige, and R. C. Bolles. Background modeling for segmentation of video-rate stereo sequences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 266–271, 1998.

[6] Berthold K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629, 1987.

[7] Nebojsa Jojic, Matthew Turk, and Thomas S. Huang. Tracking articulated objects in dense disparity maps. In *International Conference on Computer Vision*, pages 123–130, 1999.

[8] S. Julier and J. Uhlmann. A new extension of the kalman filter to nonlinear systems. In *SPIE AeroSense Symposium*, 1997.

[9] S. J. Julier and J. K. Uhlmann. The Scaled Unscented Transformation. In *Proceedings of the IEEE American Control Conference*, pages 4555–4559. IEEE, 2002.

[10] R. Kehl, M. Bray, and L.J. Van Gool. Full body tracking from multiple views using stochastic sampling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 129–136, 2005.

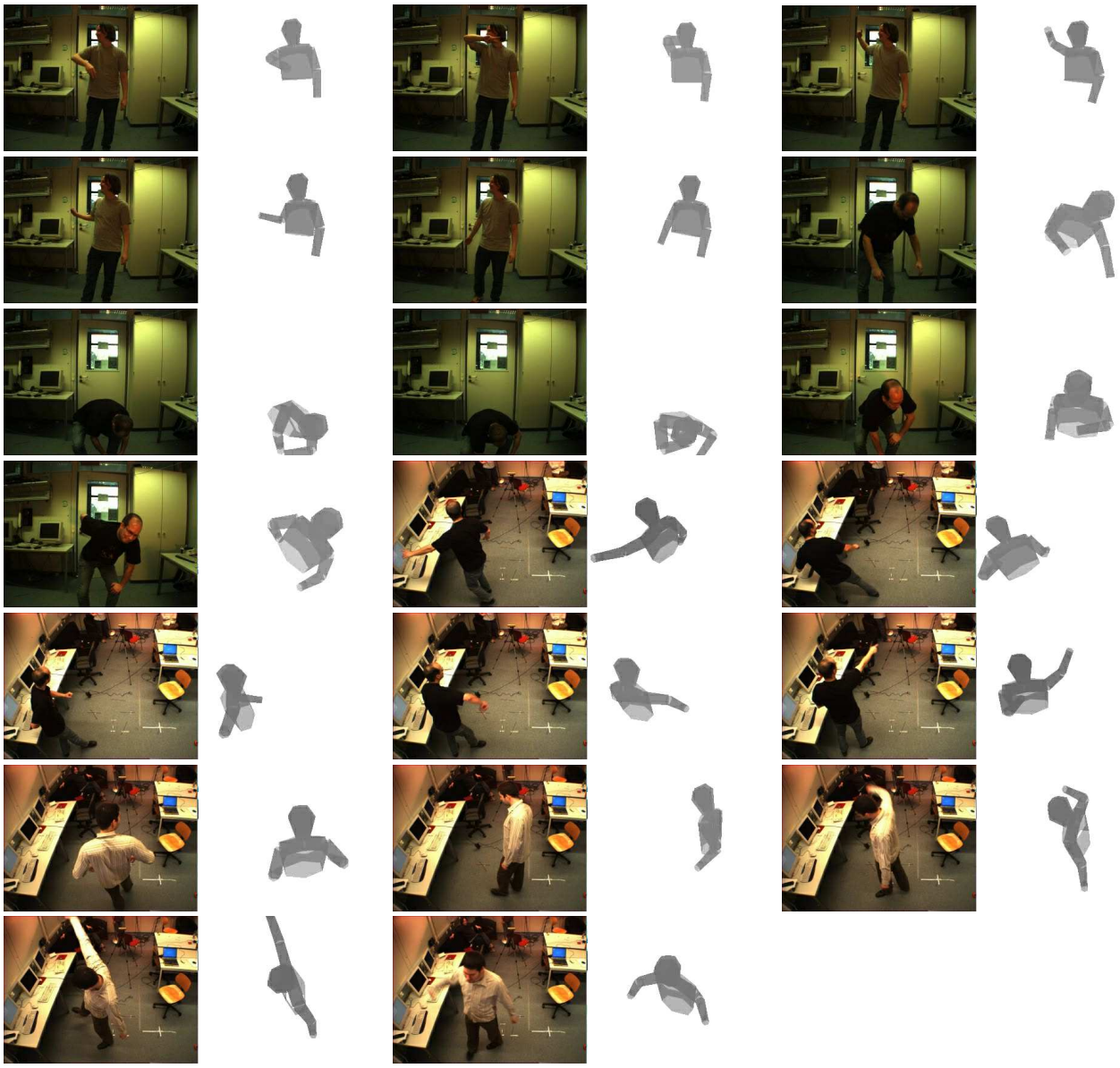


Figure 9. Sample frames from the evaluation sequences with visualisation of tracking result.

[11] Mun Wai Lee, I. Cohen, and Soon Ki Jung. Particle filter with analytical inference for human body tracking. In *Workshop on Motion and Video Computing*, page 159, 2002.

[12] Peter S. Maybeck. *Stochastic models, estimation, and control*, volume 141 of *Mathematics in Science and Engineering*. 1979.

[13] Ivana Mikic, Mohan M. Trivedi, Edward Hunter, and Pamela C. Cosman. Articulated body posture estimation from multi-camera voxel data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 455–462, 2001.

[14] K. Nickel, E. Seemann, and R. Stiefelhagen. 3d-tracking of heads and hands for pointing gesture recognition in a human-robot interaction scenario. In *Proceedings of the International Conference on Face and Gesture Recognition*, 2004.

[15] Point Grey Research Inc. *Triclops Stereo Vision System Manual Version 3.1*, 2003.

[16] Raquel Urtasun and Pascal Fua. 3d human body tracking using deterministic temporal motion models. In *European Conference on Computer Vision*, 2004.

[17] Mason Woo, Jackie Neider, Tom Davis, and Dave Shreiner. *OpenGL 1.2 Programming Guide: The Official Guide to Learning OpenGL*. Addison-Wesley, 1999.