Tracking 3D Shapes in Noisy Point Clouds with Random Hypersurface Models

Florian Faion, Marcus Baum, and Uwe D. Hanebeck
Intelligent Sensor-Actuator-Systems Laboratory (ISAS),
Institute for Anthropomatics,
Karlsruhe Institute of Technology (KIT), Germany.
florian.faion@kit.edu, marcus.baum@kit.edu, uwe.hanebeck@ieee.org

***Abstract*—Depth sensors such as the Microsoft Kinect$^{\text{TM}}$ depth sensor provide three dimensional point clouds of an observed scene. In this paper, we employ Random Hypersurface Models (RHMs), which is a modeling technique for extended object tracking, to point cloud fusion in order to track a shape approximation of an underlying object. We present a novel variant of RHMs to model shapes in 3D space. Based on this novel model, we develop a specialized algorithm to track persons by approximating their shapes as cylinders. For evaluation, we utilize a Kinect network and simulations based on a stochastic sensor model.**

I. Introduction

State of the art depth-sensors such as range cameras (time-of-flight, structured light, stereo) or laser rangefinder obtain three-dimensional point cloud data of a given real-world scene. Recently, multi-sensor setups [1], [2], [3], [4] have received increasing attention.

Point clouds have become highly relevant for many real world applications, such as surveillance, target tracking, 3D reconstruction, telepresence, and free viewpoint television [5].

Due to the large data amount of raw data, a depth sensor network gathers over time, the fusion of noisy point clouds has become necessary. One important application of point cloud fusion is shape fitting, where only a few parameters are used to represent the relevant information of an entity.

A problem similar to point cloud fusion can be found in target tracking [6], [7], where the dimensions of the target are large compared to the sensor noise. In this case, the number of measurements received from the target surface can vary significantly. Recently, several approaches have been suggested to estimate an approximation of the shape of such an extended target [8], [9], [10]. Extended object tracking methods are best suited for scenarios with high measurement noise and only a few available measurements.

Due to the limited sensor resolution and field of view of range cameras, a similar situation frequently occurs when dealing with point clouds. However, in this case standard methods for point cloud fusion are not applicable or yield poor results.

In order to overcome this issue, this paper explores the idea to apply extended object tracking methods to point cloud fusion.

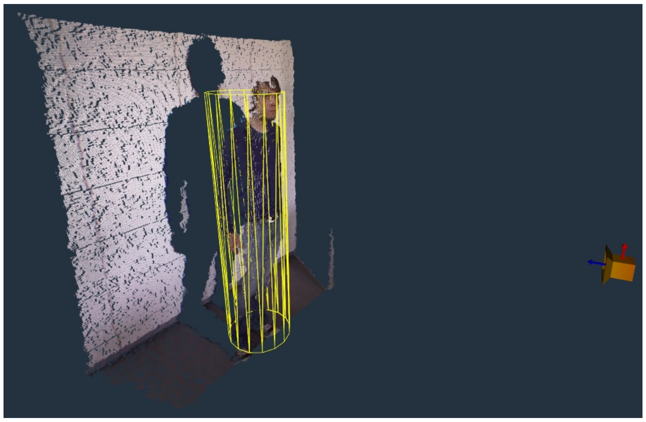


Fig. 1: Cylindrical approximation of noisy point cloud measurements.

A. Contribution

The objective of this paper is to apply an extended object tracking method based on Random Hypersurface Models (RHMs) to point cloud fusion. In particular, tracking people based on Kinect sensors is considered (Figure 1). For this purpose, we introduce an algorithm to track people by approximating their shapes as a cylinder, an approach popular, for example, in the field of surveillance.

This algorithm is based on a novel interpretation of RHMs. A main property of this interpretation is that the dimensionality of the RHM can be larger than the base shape. We give a general derivation of the new RHM and derive fusion formulas for a cylindrical approximation of the underlying shape. In addition, to the best of our knowledge there are currently no recursive Bayesian cylinder fitting algorithms available in literature for this kind of problem.

B. Related Work

Treated as simultaneous estimation of position and shape, point cloud fusion results in an errors-in-variables problem [11]. For example, Random Sampling Consensus (RANSAC) was utilized to estimate the parameters of cylinders [4] and other simple geometric shapes [12]. A tracking approach for fitting linked cylindrical approximations [13] to a human body, is based on iterative methods such as Iterative Closest

Point (ICP). From the viewpoint of point cloud reduction, relevant approaches are, for example, discrete grids [14] or downsampling [15], [16].

In the context of target tracking, spatial distributions [9], [10] are used for modeling an extended target. RHMs [17] are a further method for extended object tracking, which allow for modeling complex extended objects [18]. A special case of RHMs, also discussed in this paper, is circle fitting [19].

C. Overview

The remainder of this paper is structured as follows: First, the problem is described in detail and arising challenges are outlined in Section II. In Section III, we explain the concepts behind the new interpretation of RHMs. Based on this idea, we derive the formulas for a recursive Bayesian fusion algorithm in Section IV. In Section V, we present the results of the extensive evaluation. This paper concludes with a summary of the authors' insights and proposals for future work in Section VI.

II. Problem Formulation

In this work, the problem of recursively tracking a complex shape by fusing point cloud measurements is considered. The underlying extended object is reduced to an approximation represented by a parameter vector $\underline{\boldsymbol{x}}_k = \left[\underline{p}_k^{\mathrm{T}}, \underline{c}_k^{\mathrm{T}}\right]^{\mathrm{T}}$, where $\underline{p}_k$ describes the shape and $\underline{c}_k$ the position of the center. At each time step $k+1$, the currently estimated parameter vector $\underline{\boldsymbol{x}}_k$ is updated with n_{k+1} noisy point measurements $\hat{\mathcal{Y}}_{k+1} = \{\hat{\underline{y}}^i_{k+1} \,|\, i = 1, \ldots, n_{k+1}\}$ from the extended object. In the following, time indices are dropped for clarity.

Definition 1 (Shape Function) *Let $\mathcal{S}$ be a shape centered on the origin, defined by the shape parameter $\underline{p}$. Let $\underline{y}$ be an arbitrary point in space. Then, the shape function $g(\underline{p}, \underline{y})$ is defined as the signed distance between $\underline{y}$ and the shape boundary. For translated shapes, where $\underline{c}$ denotes the new center, the shape function can be rearranged into $g(\underline{p}, \underline{y} - \underline{c})$.*

Particularly, for each point $\underline{y}^m$ located on the shape boundary,

$$g(\underline{p}, \underline{y}^m - \underline{c}) = 0 \tag{1}$$

holds.

Example 1 (Spherical Shape) *Let $\mathcal{S}_{sph}$ be a spherical shape with radius r, i.e., $\underline{p} = r$ and with center $\underline{c}$. The corresponding representation of $\mathcal{S}_{sph}$ is given by the parameter vector $\underline{x} = \left[r, \underline{c}^{\mathrm{T}}\right]^{\mathrm{T}}$. The spherical shape function $g_{sph}(\underline{p}, \underline{y} - \underline{c})$ is*

$$g_{sph}(\underline{p}, \underline{y} - \underline{c}) = ||\underline{y} - \underline{c}|| - r\,.^1 \tag{2}$$

Updating the current estimate $\underline{\boldsymbol{x}}_k$ of the object representation with a new point cloud measurement $\mathcal{Y}_{k+1}$ raises several challenges. First, $\underline{\boldsymbol{x}}_k$ may have evolved (in shape and position) from k to $k+1$. In consequence, a prediction step is required. Second, each measurement $\hat{\underline{y}} \in \hat{\mathcal{Y}}$ has to be associated to its generating source $\underline{y}^m \in \mathcal{Y}^m$. This is not possible in general due to measurement noise. Third, partial observations, for example those hindered by occlusion, may cause a different number of measurements at each time step. This makes it necessary to allow a partial update for $\underline{\boldsymbol{x}}_k$. Finally, each single measured point $\hat{\underline{y}}$ is supposed to be distorted by additive noise

$$\hat{\underline{y}} = \underline{y}^m + \underline{\boldsymbol{v}}\,. \tag{3}$$

The distortion $\underline{\boldsymbol{v}} \sim \mathcal{N}(\underline{0}, \mathbf{C}_{\hat{y}})$ has to be determined by the sensor model of the given measurement source $\underline{y}^m$. Within this work, we assume the measured points to be stochastically independent.

III. Theoretical Background

Random Hypersurface Models (RHM) are suitable for modeling complex extended objects. Measurement sources are assumed to be elements of a hypersurface, given by scaling a shape boundary. The base idea of this work is to reinterpret RHMs as the extrusion scaling of a base shape instead, while leaving the boundary parameters fixed.

The formal definition of RHMs [18] is, slightly rewritten, as follows.

Definition 2 (RHM with boundary scaling) *Let a shape $\mathcal{S}$ be given by the shape parameter $\underline{p}$ and the center $\underline{c}$. Then, the measurement source $\underline{y}^m$ can be assumed as being generated by the boundary scaled shape $\overline{\mathcal{S}}$, which is the scaling of $\mathcal{S}$ by a scalar s drawn from the one-dimensional random variable $\boldsymbol{s}$. This can be described as*

$$\underline{y}^m \in \mathcal{S}(s) + \underline{c}\,. \tag{4}$$

This definition illustrates that scaling originally was interpreted in the sense of adjusting the size of the boundary, while leaving the center untouched. In Figure 2, an example RHM with boundary scaling using a complex star-convex shape is depicted. It should be noted that measurement sources may be located both inside the object and on its boundary.

Remark 1 (Dimension of RHM with Boundary Scaling) *When modeling an arbitrary shape as RHM with boundary scaling, the dimension of the shape $\mathcal{S}$ is equal to the dimension of the shape $\overline{\mathcal{S}}$ defined by the RHM.*

Our novel interpretation of RHMs is as follows.

Definition 3 (RHM with Extrusion Scaling) *Let a shape $\mathcal{S}$ be given by the shape parameter $\underline{p}$ and the center $\underline{c}$. Then, the measurement source $\underline{y}^m$ can be assumed as being generated by the extrusion shape $\overline{\mathcal{S}}$, which is the extrusion of $\mathcal{S}$ along a curve determined by $c(s)$, where s is drawn from the one-dimensional random variable $\boldsymbol{s}$. The resulting shape can be described as*

$$\underline{y}^m \in \mathcal{S} + \underline{c}(s)\,. \tag{5}$$

[1] $g(\underline{p}, \underline{y} - \underline{c})$ can allow negative values, since the algorithm minimizes the absolute value

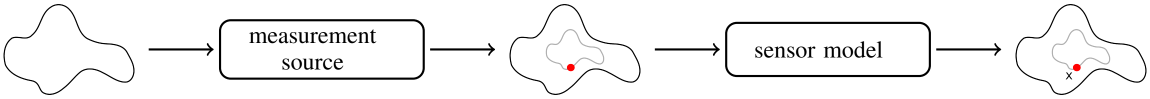


Fig. 2: Random Hypersurface Model with boundary scaling.

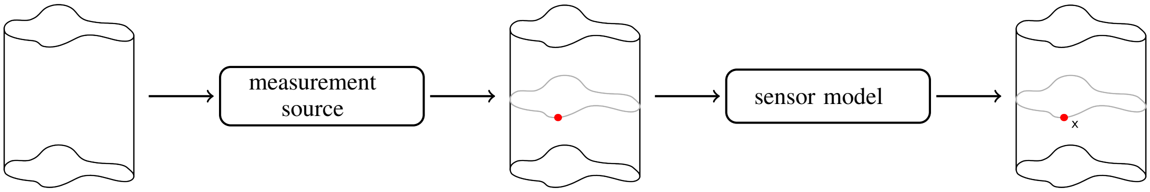


Fig. 3: Random Hypersurface Model with extrusion scaling.

Using this definition, scaling is interpreted in the sense of varying the shape's center position, while leaving the boundaries untouched. In Figure 3, the same complex star-convex shape as in Figure 2 is now scaled by extrusion. In this case, measurement sources are only located on the boundary of the object.

Example 2 (Circle Fitting) *Fitting a circle to noisy points [19] can be understood as a special case of both RHM interpretations, where $\underline{y}^m \in \mathcal{S} + \underline{c}$, i.e., the boundary and the center are fixed and thus, do not depend on a scaling parameter.*

Remark 2 (Dimension of RHM with Extrusion Scaling) *When modeling an arbitrary shape as RHM with extrusion scaling, the dimension of the shape $\mathcal{S}$ may be smaller than the shape $\overline{\mathcal{S}}$ defined by the RHM.*

Remark 3 (RHM with Extrusion and Boundary Scaling) *The combination of both RHM types is given by $\underline{y}^m \in \mathcal{S}(s) + \underline{c}(s)$, where s simultaneously affects the boundary and the center of the shape. For example a cone could be modeled by a combined RHM, scaling the radius of a circle along the long axis.*

IV. Recursive Tracking Algorithm

In this section, the recursive Bayesian estimator for RHMs with extrusion scaling is explained. For the sake of simplicity, specific formulas for a circle shape extruded along a linear axis are given. The resulting shape clearly is a cylinder.

A. Bayesian Estimator

A Bayesian state estimator recursively estimates the parameter vector $\underline{\boldsymbol{x}}_k$ of the shape approximation, where $\underline{\boldsymbol{x}}_k$ is modeled as a random vector with probability distribution $f^e(\underline{x}_k)$. In this work, all $f(\underline{x})$ are assumed to be Gaussian, i.e., $\underline{\boldsymbol{x}} \sim \mathcal{N}(\hat{\underline{x}}, \mathbf{C}_{\hat{\underline{x}}})$. Bayesian estimation includes a time update, predicting the estimated state at a future time step $k+1$, and a measurement update, correcting a predicted state using a sensor measurement $\underline{\boldsymbol{y}}_{k+1}$[2]. The prediction step is defined by the Chapman-Kolmogorov equation

$$f^p(\underline{x}_{k+1}) = \int f(\underline{x}_{k+1}|\underline{x}_k) \cdot f^e(\underline{x}_k)\, \mathrm{d}\underline{x}_k \,, \qquad (6)$$

where $f(\underline{x}_{k+1}|\underline{x}_k)$ is the conditional density, representing the assumed system behavior and $f^p(\underline{x}_{k+1})$ is the predicted probability distribution for the next time step. For the measurement update, Bayes' rule is applied

$$f^e(\underline{x}_{k+1}) = c_k \cdot f^L(\underline{y}_{k+1}|\underline{x}_{k+1}) \cdot f^p(\underline{x}_{k+1}) \,, \qquad (7)$$

with $f^L(\underline{y}_{k+1}|\underline{x}_{k+1})$ being the Likelihood function and c_k a normalization constant. To clarify the notation used in this paper, one recursion step of the algorithm is shown in Figure 4.

```
**Input:** $\underline{\boldsymbol{x}}_k^e \sim \mathcal{N}(\hat{\underline{x}}_k^e, \mathbf{C}_{\hat{\underline{x}}_k^e})$, $\mathcal{Y}_{k+1} = \{\hat{\underline{y}}_{k+1}^i \,|\, i = 1, \ldots, n_k\}$
  1: $\underline{\boldsymbol{x}}_{k+1}^p$ = predict $\underline{\boldsymbol{x}}_k^e$
  2: $\underline{\boldsymbol{x}}_{k+1}^e$ = update $\underline{\boldsymbol{x}}_{k+1}^p$ with $\mathcal{Y}_{k+1}$
**Output:** $\underline{\boldsymbol{x}}_{k+1}^e$
```

Fig. 4: A recursion step of the Bayesian state estimator.

B. State Representation

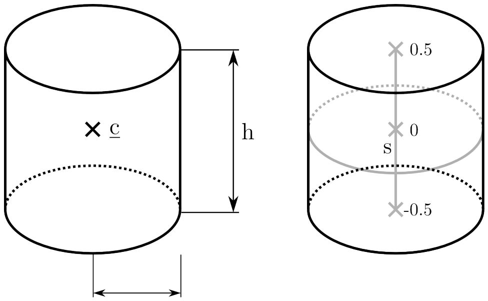
According to Definition 1, the parameter vector $\underline{\boldsymbol{x}}_k$ consists of the shape parameters $\underline{\boldsymbol{p}}_k$ and the translated shape center $\underline{\boldsymbol{c}}_k$. In this work, the underlying shape that generates the point cloud measurements is assumed to be approximately a cylinder. A cylinder centered on the origin can be represented by its radius $\boldsymbol{r}_k$ and height $\boldsymbol{h}_k$, resulting in $\underline{\boldsymbol{p}}_k = [\boldsymbol{r}_k, \boldsymbol{h}_k]^{\mathrm{T}}$. The translated center is given by $\underline{\boldsymbol{c}}_k = [\boldsymbol{c}_{k,1}, \boldsymbol{c}_{k,2}, \boldsymbol{c}_{k,3}]^{\mathrm{T}}$. Then, the parameter vector is specified by

$$\underline{\boldsymbol{x}}_k = \begin{bmatrix} \boldsymbol{r}_k \\ \boldsymbol{h}_k \\ \underline{\boldsymbol{c}}_k \end{bmatrix} \,. \qquad (8)$$

Figure 5a shows an annotated cylinder. As mentioned in the previous section, the parameter vector is modeled as a

[2] Within this paper, $\underline{\boldsymbol{x}}^p$ denotes the predicted and $\underline{\boldsymbol{x}}^e$ the updated parameter vector.

Gaussian random vector with $\underline{\boldsymbol{x}}_k^e \sim \mathcal{N}(\hat{\underline{x}}_k^e, \mathbf{C}_{\hat{\underline{x}}_k^e})$. A proper initialization avoids negative values for radius and height.



(a) Cylinder state representation.

(b) Cylinder RHM by extruding a circle.

Fig. 5: Cylinder RHM.

C. Time Update

The prediction of the cylindrical shape to a future time step (see Figure 4, Line 1) models the behavior by means of a system model

$$\underline{\boldsymbol{x}}_{k+1} = a_k(\underline{\boldsymbol{x}}_k, \underline{u}_k, \underline{\boldsymbol{d}}_k)\,, \tag{9}$$

where a_k denotes the system function, $\underline{u}_k$ is the system input, and $\underline{\boldsymbol{d}}_k$ is the system noise. In this work, we assume a_k to be a constant velocity model.

D. RHM for Cylindrical Shapes

A cylindrical shape approximation modeled by an RHM with extrusion scaling is essentially a circle shape $\mathcal{S}_{circ}$ of fixed size, whose center $\underline{c}$ is translated along the long axis of the cylinder by the parameter s. According to (5), the RHM is then given by

$$\mathcal{S}_{circ} + \underline{c}(s)\,. \tag{10}$$

The shape of a circle $\mathcal{S}_{circ}$ can be described in a similar way to the sphere in Example 1 by the shape function

$$g_{cir}(\underline{p}, \underline{y} - \underline{c}) = \begin{bmatrix} \left\| \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} - \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \right\|^2 - r^2 \\ y_3 - c_3 \end{bmatrix}\,. \tag{11}$$

Note that g_{cir} describes a two-dimensional circle in three-dimensional space, i.e., in the xy-plane. Translating the center of this circle shape by means of the scaling parameter can be formalized by

$$\underline{c}(s) = \begin{bmatrix} c_1 \\ c_2 \\ c_3 + s \cdot h \end{bmatrix}\,, \tag{12}$$

where $s = -0.5$ translates the circle to the bottom, and $s = 0.5$ to the top of the long axis (Figure 5b). The resulting shape function of the cylinder can be determined by substituting $\underline{c}$ by $\underline{c}(s)$ in (11):

$$g_{cyl}(\underline{p}, \underline{y} - \underline{c}(s)) = \begin{bmatrix} \left\| \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} - \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \right\|^2 - r^2 \\ y_3 - (c_3 + s \cdot h) \end{bmatrix}\,. \tag{13}$$

E. Scaling Parameter

According to Definition 3, a parameter s drawn from the one-dimensional random variable $\boldsymbol{s}$ defines a given translated circle. Measurement sources $\mathcal{Y}^m$ are assumed to be located on these hypersurfaces. In this work, a uniform distribution for the scaling parameter is assumed, i.e., $\boldsymbol{s} \sim \mathcal{U}(-0.5, 0.5)$. A more sophisticated method would incorporate the actual sensor fields of view and the predicted parameter vector $\underline{\boldsymbol{x}}_{k+1}^p$ for calculating an adapted random variable $\boldsymbol{s}_{k+1}$ at each time step.

F. Measurement Update

Before determining a simultaneous measurement update of $\underline{\boldsymbol{x}}_{k+1}^p$ with all points $\hat{\mathcal{Y}}_{k+1}$, first a single update with one measurement $\hat{\underline{y}} \in \hat{\mathcal{Y}}$ is discussed. Plugging $\hat{\underline{y}}$ given by (3) into the shape function $g_{cyl}(\underline{p}, \underline{y} - \underline{c}(\underline{\boldsymbol{s}}))$ results in

$$\begin{aligned} g_{cyl}(\underline{p}, \hat{\underline{y}} - \underline{c}(\underline{\boldsymbol{s}})) &= g_{cyl}(\underline{p}, (\underline{y}^m + \underline{\boldsymbol{v}}) - \underline{c}(\underline{\boldsymbol{s}})) \qquad (14)\\ &= \begin{bmatrix} \left\| \begin{bmatrix} y_1^m + \boldsymbol{v_1} \\ y_2^m + \boldsymbol{v_2} \end{bmatrix} - \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \right\|^2 - r^2 \\ y_3^m + \boldsymbol{v_3} - (c_3 + \boldsymbol{s} \cdot h) \end{bmatrix} \qquad (15)\\ &= \begin{bmatrix} \left\| \begin{bmatrix} y_1^m \\ y_2^m \end{bmatrix} - \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \right\|^2 - r^2 + \boldsymbol{w} \\ y_3^m - (c_3 + \boldsymbol{s} \cdot h) + \boldsymbol{v_3} \end{bmatrix} \qquad (16)\\ &= \underbrace{g_{cyl}(\underline{p}, \underline{y}^m - \underline{c})}_{\overset{(1)}{=}0} + \begin{bmatrix} \boldsymbol{w} \\ \boldsymbol{v}_3 \end{bmatrix}\,, \qquad (17) \end{aligned}$$

where $\boldsymbol{w} = 2y_1^m\boldsymbol{v_1} + 2y_2^m\boldsymbol{v_2} - 2c_1\boldsymbol{v_1} - 2c_2\boldsymbol{v_2} + \boldsymbol{v_1}^2 + \boldsymbol{v_2}^2$. It follows from (1) that $g(\underline{p}, \underline{y}^m - \underline{c}) = 0$. Therefore,

$$\underline{0} = g_{cyl}(\underline{p}, \hat{\underline{y}} - \underline{c}(\underline{\boldsymbol{s}})) - \begin{bmatrix} \boldsymbol{w} \\ \boldsymbol{v}_3 \end{bmatrix} \tag{18}$$

holds, which leads to the explicit measurement equation

$$\underline{0} = h(\underline{\boldsymbol{x}}, \underline{\boldsymbol{v}})\,. \tag{19}$$

This equation maps the random vectors $\underline{\boldsymbol{x}}$ and $\underline{\boldsymbol{v}}$ to the pseudo measurement $\underline{0}$. However, (19) also depends on the unknown measurement source $\underline{y}^m$. In the implementation, $\underline{y}^m$ is approximated by an appropriate point estimate $\underline{y}^m \approx \tilde{\underline{y}}^m$. A simple way to obtain a point estimate is to assume $\tilde{\underline{y}}_{k+1}^m$ at a certain time step $k+1$ to be the closest point on the cylinder, given by the predicted state $\underline{x}_{k+1}^p$.

Due to the nonlinearity of the measurement equation and the non-additive Gaussian noise, standard Kalman Filter techniques [20] cannot be applied. Instead, we suggest to use a standard quadratic estimator can be used to update $\underline{\boldsymbol{x}}$ with $\hat{\underline{y}}$ (Figure 4, Line 2). In [21], it was seen that (13) can be modified in order to improve the result estimated by an Unscented Kalman Filter [22].

G. Simultaneous Update With All Measurements

So far, the measurement equation is designed to update the estimate $\underline{\boldsymbol{x}}$ with exactly one measurement $\hat{\underline{y}} \in \hat{\mathcal{Y}}$. Of course, by applying this technique, updates can be performed sequentially for each measurement. A different, more elegant approach is to simultaneously update $\underline{\boldsymbol{x}}$ with all measurements $\underline{\hat{y}}$. This is achieved by aggregating all measurements $\mathcal{Y} = \{\underline{\hat{y}}^i \,|\, i = 1, \ldots, n\}$ into a single measurement vector

$$\underline{\hat{Y}} = \left[\underline{\hat{y}}^{1\,\mathrm{T}}, \ldots, \underline{\hat{y}}^{n\,\mathrm{T}}\right]^{\mathrm{T}} . \qquad (20)$$

The resulting measurement equation can be derived from (18)

$$\begin{bmatrix} \underline{0} \\ \vdots \\ \underline{0} \end{bmatrix} = \begin{bmatrix} g_{cyl}(\underline{p}, \underline{\hat{y}}^1 - \underline{c}(\boldsymbol{s}^1)) - \begin{bmatrix} \boldsymbol{w}^1 \\ \boldsymbol{v}_3^1 \end{bmatrix} \\ \vdots \\ g_{cyl}(\underline{p}, \underline{\hat{y}}^n - \underline{c}(\boldsymbol{s}^n)) - \begin{bmatrix} \boldsymbol{w}^n \\ \boldsymbol{v}_3^n \end{bmatrix} \end{bmatrix} . \qquad (21)$$

V. Evaluation

For the evaluation of the developed algorithm we considered a person tracking scenario. The point cloud measurements of the human body were assumed to stem approximately from a cylindrical shape (see Figure 1). The state parameters of the cylindrical approximation estimated by our algorithm were compared to those calculated by a cylinder fitting algorithm [23]. However, this algorithm does not allow the height to be estimated, and because of this, an approximation of the distance between the least and the greatest z-value of a measured point cloud $\hat{\mathcal{Y}}$ were used.

A. Sensor Model

For the evaluation, we used a Kinect sensor model [24] to generate noisy observations $\underline{\hat{y}} \in \hat{\mathcal{Y}}$ for a given measurement source $\underline{y}^m \in \mathcal{Y}^m$, according to (3). This model assumes $\underline{\boldsymbol{v}}$ to be a Gaussian random vector with Kinect-specific uncertainties. The corresponding covariance ellipses for a selected subset of 3D points $\underline{y}^m$ are depicted in Figure 6. The Kinect depth sensor is essentially a camera that provides a depth image. The field of view is colored in transparent gray. Points that are closer to the camera center have a smaller uncertainty.

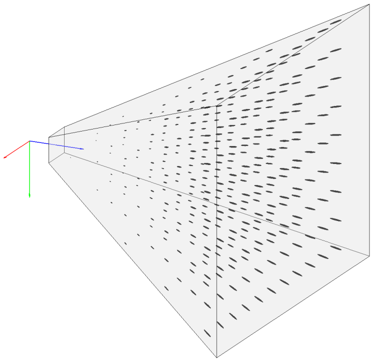


Fig. 6: Illustration of the Kinect sensor model and the uncertainty of measured points [24].

B. Experiments

Real and synthetic data experiments were performed, but this paper focuses on synthetic data in order to compare the results with a known ground truth. A multi-Kinect network was simulated in order to generate synthetic point cloud measurements. This network consisted of four sensors that measured noisy points from a virtual cylinder according to the sensor model described in the previous section. Height and radius of the ground truth cylinder were set to $h = 2$m and $r = 0.5$m. The cylinder was then moved along a certain track within the viewable region of the network. This scenario is illustrated in Figure 7. At each time step, n random measurement were drawn from sources on the ground truth cylinder, obeying visibility constraints. Then, noisy observations were generated from these measurement sources. Three experiments with different number of measurements were performed, i.e., $n = 5$, $n = 20$, and $n = 50$. In all cases the states were initialized with $\underline{\boldsymbol{x}}_0 \sim \mathcal{N}(\underline{\hat{x}}_0, \mathbf{C}_{\underline{\hat{x}}_0})$, where $\underline{\hat{x}}_0 = [0, 0, 0, 1, 1]^{\mathrm{T}}$ and $\mathbf{C}_{\underline{\hat{x}}_0} = \mathbf{I}$.

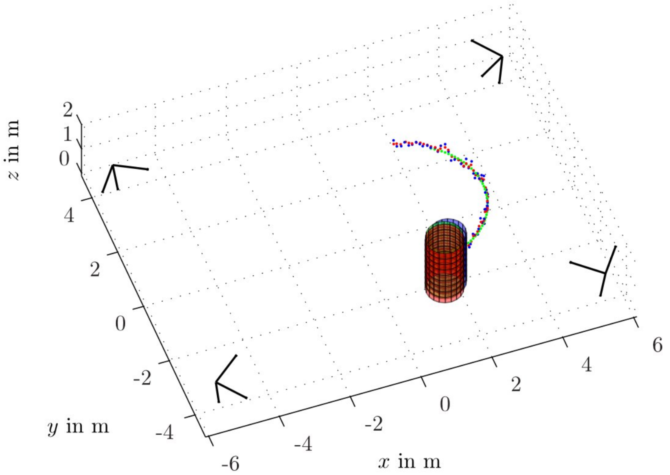
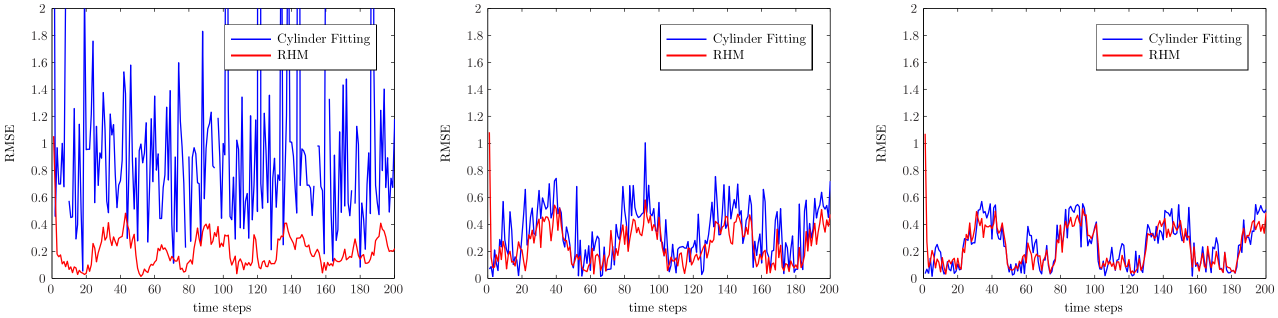


Fig. 7: Illustration of the simulated Kinect network. The ground truth cylinder and its trace are colored *green*, the RHM estimation *red* and the cylinder fitting method *blue*.

Results: The Root Mean Square Error (RMSE) was collected from 200 successive time steps for the RHM approach and cylinder fitting. As the results show (Figure 8), the RHM approach clearly outperforms the cylinder fitting algorithm. Especially when there is only a small number of measurements available, the RHM has a significant advantage. The cylinder fitting algorithm requires at least five measurements a time step. However, the RHM approach would also work with only one measurement. Note that the variation of the RMSE results from the underlying assumption of a constant velocity model.

VI. Conclusion and Outlook

In this paper, we applied Random Hypersurface Models to point cloud fusion. Originally, RHMs were designed as a method for extended object tracking. For 3D point cloud fusion, a novel interpretation of RHMs was introduced. A



(a) $n = 5$ measurements per time step.

(b) $n = 20$ measurements per time step.

(c) $n = 50$ measurements per time step.

Fig. 8: Result of the simulated data experiment. The Root Mean Square Error (RMSE) of the estimated state to the ground truth is drawn over 200 time steps.

main property of this interpretation is that the dimensionality of the RHM can be larger than those of the base shape. This extrusion of the base shape constitutes a significant extension to RHMs and allows for modeling a wide variety of shape approximations. Specific formulas for a cylindrical approximation were derived and evaluated. As a particular application, tracking people based on a Kinect sensor network was considered. Especially when only a small number of measurements is available, the new RHM approach clearly outperforms iterative cylinder fitting algorithms.

A. *Future Work*

An obvious extension would be the additional estimation of orientation parameter. Furthermore, scaling could be reinterpreted again, for scaling orientation. In consequence, flexible tubes could be estimated. Another important improvement would be modeling dependencies of the point cloud measurements. Finally, given a set of RHMs, an interactive multiple model approach (IMM) may be applied for classifying the best fitting shape.

VII. Acknowledgements

We would like to thank Jannik Steinbring and Antonio Zea for their support.

References

[1] a. Maimone and H. Fuchs, “Encumbrance-free telepresence system with real-time 3D capture and display using commodity depth cameras,” *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, pp. 137–146, Oct. 2011.

[2] A. Scholz and K. Berger, “Multiple Kinect Studies,” Tech. Rep., 2011.

[3] A. Wilson and H. Benko, “Combining multiple depth cameras and projectors for interactions on, above and between surfaces,” in *Proceedings of the 23nd annual ACM symposium on User interface software and technology*. ACM, 2010, pp. 273–282.

[4] R. B. Rusu and S. Cousins, “3D is here: Point Cloud Library (PCL),” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[5] M. Tanimoto, M. Tehrani, and T. Fujii, “Free-viewpoint TV,” *Signal Processing*, no. January, pp. 67–76, 2011.

[6] Y. Bar-Shalom, T. Kirubarajan, and X.-R. Li, *Estimation with Applications to Tracking and Navigation*. New York, NY, USA: John Wiley & Sons, Inc., 2002.

[7] R. Mahler, “PHD Filters for Nonstandard Targets, I: Extended Targets,” in *Proceedings of the 12th International Conference on Information Fusion (Fusion 2009)*, Seattle, Washington, 2009, pp. 915–921.

[8] M. Feldmann, D. Franken, and W. Koch, “Tracking of Extended Objects and Group Targets using Random Matrices,” *IEEE Transactions on Signal Processing*, vol. PP, no. 99, p. 1, 2010.

[9] K. Gilholm, S. J. Godsill, S. Maskell, and D. Salmond, “Poisson Models for Extended Target and Group Tracking,” in *SPIE: Signal and Data Processing of Small Targets*, 2005.

[10] K. Gilholm and D. Salmond, “Spatial Distribution Model for Tracking Extended Objects,” *IEEE Proceedings on Radar, Sonar and Navigation*, vol. 152, no. 5, pp. 364–371, 2005.

[11] T. Soderstrom, “Errors-in-variables methods in system identification,” *Automatica*, vol. 43, no. 6, pp. 939–958, Jun. 2007.

[12] R. Schnabel, R. Wahl, and R. Klein, “Efficient RANSAC for Point-Cloud Shape Detection,” *Computer Graphics Forum*, vol. 26, no. 2, pp. 214–226, Jun. 2007.

[13] S. Knoop, S. Vacek, and R. Dillmann, “Sensor fusion for 3D human body tracking with an articulated 3D body model,” in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, 2006, pp. 1686–1691.

[14] S. Izadi, D. Kim, O. Hilliges, and D. Molyneaux, “KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera,” *Proceedings of the 24th*, pp. 559–568, 2011. [Online]. Available: http://dl.acm.org/citation.cfm?id=2047270

[15] Y. Liu, Q. Dai, and W. Xu, “A point-cloud-based multiview stereo algorithm for free-viewpoint video.” *IEEE transactions on visualization and computer graphics*, vol. 16, no. 3, pp. 407–18, 2010.

[16] H. Eberhardt, V. Klumpp, and U. D. Hanebeck, “Optimal Dirac Approximation by Exploiting Independencies,” in *Proceedings of the 2010 American Control Conference (ACC 2010)*, Baltimore, Maryland, 2010.

[17] M. Baum and U. D. Hanebeck, “Random Hypersurface Models for extended object tracking,” *2009 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, pp. 178–183, Dec. 2009.

[18] ——, “Shape Tracking of Extended Objects and Group Targets with Star-Convex RHMs,” in *Proceedings of the 14th International Conference on Information Fusion (Fusion 2011)*, Chicago, Illinois, USA, 2011.

[19] M. Baum, V. Klumpp, and U. D. Hanebeck, “A Novel Bayesian Method for Fitting a Circle to Noisy Points,” in *Proceedings of the 13th International Conference on Information Fusion (Fusion 2010)*, Edinburgh, United Kingdom, 2010.

[20] R. Kalman, “A new approach to linear filtering and prediction problems,” *Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[21] M. Baum, F. Faion, and U. D. Hanebeck, “Modeling the Target Extent as Multiplicative Noise,” in *Fusion 2012*, 2012, p. (to appear).

[22] S. J. Julier and J. K. Uhlmann, “Unscented Filtering and Nonlinear Estimation,” *Computer Engineering*, vol. 92, no. 3, 2004.

[23] I. Smith, “MATLAB LEAST SQUARES GEOMETRIC ELEMENT SOFTWARE,” 2004. [Online]. Available: http://www.eurometros.org/

[24] F. Faion, S. Friedberger, A. Zea, and U. D. Hanebeck, “Intelligent Sensor-Scheduling for Multi-Kinect-Tracking,” in *International Conference on Intelligent Robots and Systems*, 2012, p. (submitted).