

Recursive Bayesian Pose and Shape Estimation of 3D Objects Using Transformed Plane Curves

Florian Faion, Antonio Zea, Jannik Steinbring, Marcus Baum, and Uwe D. Hanebeck

Intelligent Sensor-Actuator-Systems Laboratory (ISAS)

Institute for Anthropomatics and Robotics

Karlsruhe Institute of Technology (KIT), Germany

firstname.lastname@kit.edu

Abstract—We consider the task of recursively estimating the pose and shape parameters of 3D objects based on noisy point cloud measurements from their surface. We focus on objects whose surface can be constructed by transforming a plane curve, such as a cylinder that is constructed by extruding a circle. However, designing estimators for such objects is challenging, as the straightforward distance-minimizing approach cannot observe all parameters, and additionally is subject to bias in the presence of noise. In this article, we first discuss these issues and then develop probabilistic models for cylinder, torus, cone, and an extruded curve by adapting related approaches including Random Hypersurface Models, partial likelihood, and symmetric shape models. In experiments with simulated data, we show that these models yield unbiased estimators for all parameters even in the presence of high noise.

I. INTRODUCTION

The major task when designing a tracking algorithm for objects in 3D boils down to defining a probabilistic model that takes a set of pose and shape parameters and then relates them to the potentially sparse, noisy point measurements. In literature, there are roughly two lines of research to this task. On the one side, *non-parametric* models can represent detailed shapes by occupancy grids, point clouds, or polygon meshes. However, varying the shape complexity requires non-trivial refinement and coarsening operations. On the other side, *parametric* models allow for specifying simple shapes such as ellipsoids or bounding boxes using a small number of parameters. In this article, we follow the parametric approach and propose to construct the surface of more general 3D objects by transforming plane curves (see Fig. 1). There are similar approaches in the context of surface reconstruction [1], [2] and extended object tracking [3], [4]. However, developing a probabilistic model that relates the curve- and its transformation-parameters to measurements in the presence of high noise is challenging and may cause issues such as bias or unobservable parameters.

Our main contribution is a thorough discussion of the decision process for a suitable probabilistic model and compatible estimator. Specifically, we develop recursive Bayesian estimators for the objects in Fig. 1, whose shapes are obtained by translating, rotating, and scaling closed curves. For these estimators, we incorporate insights and methods from our previous research [3–10]. Note that these estimators can

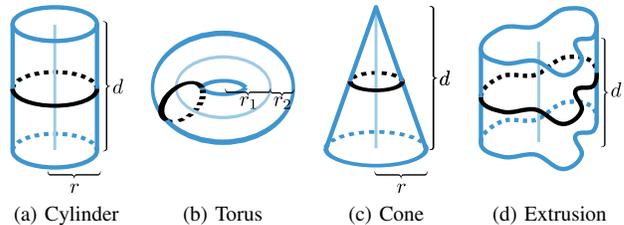


Figure 1: All of these blue objects can be constructed by translating, rotating, and scaling the black plane curves.

be easily extended to full Bayesian tracking algorithms by including motion models [11].

II. MODELING OBJECTS AS TRANSFORMED PLANE CURVE

In this section, we show how to use plane curves to construct the surface of an object. A *plane curve* is one that can be embedded into a 2D plane, such as the circle in Fig. 1(a-c). All involved parameters that are required to specify the pose and shape of the surface are aggregated in a state vector \underline{x} . Technically, modeling a surface then refers to the task of (i) specifying a function $\phi_x : \mathbb{R} \rightarrow \mathbb{R}^2$ that lets us iterate through all points in the curve, and (ii) embedding it into 3D space by applying a transformation $\Phi_x : \mathbb{R}^2 \rightarrow \mathbb{R}^3$. Specifically, we consider transformations in the form of

$$\Phi_x(u, s) = \mathbf{R}_u \begin{bmatrix} f_u \cdot \phi_x(s) \\ 0 \end{bmatrix} + \underline{c}_u. \quad (1)$$

In this formula, the variable $s \in S \subseteq \mathbb{R}$ iterates through the curve and $u \in U \subseteq \mathbb{R}$ controls the transformation.

Using these parameters, the plane curve can be translated and rotated using the vector $\underline{c}_u \in \mathbb{R}^3$ and the 3×3 rotation matrix \mathbf{R}_u , respectively. The scalar $f_u \in \mathbb{R}$ allows for scaling operations for a given curve, and by using different curve functions $\phi_x(s)$, the shape itself can be modified. Note that for (1), we considered the object in its local coordinate system. Arbitrary object poses can be implemented by applying a rigid transformation $\mathbf{R}_x \Phi_x(u, s) + \underline{c}_x$ to the surface function. In the following, we derive specific instances of (1) for the shapes in Fig. 1.

A. Translation (Cylinder)

Let us start with modeling the mantle of the cylinder in Fig. 1a, which can be characterized by the state vector $\underline{x} =$

$[r, d]^T$ where r is the radius and d is the length of the extrusion. For the object coordinate system, let the bottom of the cylinder be centered on the origin, axis aligned, and let y_3 be the axis of extrusion. For the circular plane curve (black), we can use the function $\phi_x(s) = r \cdot [\cos(s), \sin(s)]^T$ with $s \in [0, 2\pi]$. The translation can be modeled as a linear shift according to $\underline{c}_u = [0, 0, u \cdot d]^T$ with $u \in [0, 1]$. The remaining transformation mechanisms for rotation and scaling are not needed and set to their respective identities $\mathbf{R}_u = \mathbf{I}$ and $f_u = 1$. Plugging all into the general formula (1) yields

$$\Phi_x(u, s) = \begin{bmatrix} r \cdot \cos(s) \\ r \cdot \sin(s) \\ u \cdot d \end{bmatrix}, \quad (2)$$

where each u refers to a circle in a plane that lies parallel to the y_1y_2 -plane and is shifted along the y_3 -axis.

B. Translation and Rotation (Torus)

By allowing translation and rotation, we can model more complex surfaces such as the torus in Fig. 1b. Its surface can be specified by the state vector $\underline{x} = [r_1, r_2]^T$ where r_1 is the central radius and r_2 is the lateral radius. For the object coordinate system, let the center of mass lie in the origin and the central circle lie in the y_1y_2 -plane. For the circular plane curves (black), we can again use the function $\phi_x(s) = r_2 \cdot [\cos(s), \sin(s)]^T$ with $s \in [0, 2\pi]$. The translation can be modeled as a circular function $\underline{c}_u = r_1 \cdot [\cos(u), \sin(u), 0]^T$ with $u \in [0, 2\pi]$. Then, the rotation matrix \mathbf{R}_u can be obtained using the tangent vector to the central circle at \underline{c}_u . Finally, by setting the scaling to $f_u = 1$, we arrive at

$$\Phi_x(u, s) = \begin{bmatrix} (r_1 + r_2 \cdot \cos(s)) \cdot \cos(u) \\ (r_1 + r_2 \cdot \cos(s)) \cdot \sin(u) \\ r_2 \cdot \sin(s) \end{bmatrix}. \quad (3)$$

C. Translation and Scaling (Cone)

The cone in Fig. 1c can be constructed in a similar way as the cylinder, except from an additional scaling term that controls the circle radius. The state vector is given by $\underline{x} = [r, d]^T$, where r is the radius at the bottom and d is the height of the cone. Then, we can modify the cylinder model from (2) by replacing the formerly constant scaling factor by $f_u = 1 - u$ which yields the cone model

$$\Phi_x(u, s) = \begin{bmatrix} (1 - u) \cdot r \cdot \cos(s) \\ (1 - u) \cdot r \cdot \sin(s) \\ u \cdot d \end{bmatrix}, \quad (4)$$

where again $s \in [0, 2\pi]$ and $u \in [0, 1]$.

D. Translation of Star-convex Closed Curve (Extruded Curve)

Instead of using circular plane curves, we could also use more general curves. In [6], it was proposed to use polar functions for star-convex curves in \mathbb{R}^2 . Star-convex means that there is a point within the shape, where each line segment to

any point on the boundary remains in the shape. This allows for a convenient representation by means of a polar function $r(s)$ that returns the radius for a given angle $s \in [0, 2\pi]$. Specifically, $r(s)$ can be implemented by means of a Fourier series

$$r(s) = \frac{a_0}{2} + \sum_{m=1}^M a_m \cos(m \cdot s) + b_m \sin(m \cdot s) \quad (5)$$

that is controlled by a list of $2M + 1$ coefficients a, b .

Using this polar function, we can iterate through all points on the curve by $\phi_x(s) = r(s) \cdot [\cos(s), \sin(s)]^T$. For the special case of $M = 0$, the corresponding function $r(s) = \frac{a_0}{2}$ specifies the constant radius of a circle.

For a cylinder based on star-convex curves, the state vector is given by $\underline{x} = [d, a_0, a_1, b_1, \dots, a_M, b_M]^T$, where d is the height. Adapting the cylinder formula from (2), we obtain

$$\Phi_x(u, s) = \begin{bmatrix} r(s) \cdot \cos(s) \\ r(s) \cdot \sin(s) \\ u \cdot d \end{bmatrix}. \quad (6)$$

Next, we discuss the task of estimating the state parameters \underline{x} for the considered objects based on noisy measurements of the modeled surfaces.

III. RECURSIVE BAYESIAN ESTIMATION

In Bayesian estimation, the relationship between object parameters \underline{x} and measurements $\underline{y}_1, \dots, \underline{y}_n$ is usually expressed in terms of a *likelihood* $p(\underline{y}_1, \dots, \underline{y}_n | \underline{x})$. This likelihood can then be used to update a prior distribution $p(\underline{x})$ on the state parameters by applying Bayes' rule

$$p(\underline{x} | \underline{y}_1, \dots, \underline{y}_n) \propto p(\underline{y}_1, \dots, \underline{y}_n | \underline{x}) \cdot p(\underline{x}). \quad (7)$$

This allows for developing a recursive Bayesian estimator [12] by using the posterior distribution as the prior for the next update. In addition, by incorporating *prediction steps* between the *measurement update steps*, the recursive estimator can be extended to a tracking algorithm.

In the following, we assume the sensor noise to be additive Gaussian $\mathcal{N}(\underline{0}, \mathbf{C}_{v,i})$ and mutually independent among the measurements \underline{y}_i . This allows us to factorize the likelihood $p(\underline{y}_1, \dots, \underline{y}_n | \underline{x}) = p(\underline{y}_1 | \underline{x}) \cdot \dots \cdot p(\underline{y}_n | \underline{x})$ and to consider each measurement individually.

IV. PROBABILISTIC MODEL

In this section, we derive a likelihood $p(\underline{y} | \underline{x})$ for shapes, modeled according to (1). In doing so, we discuss the association problem of measurements to the surface and its existing solutions.

A. Likelihood Prototype

Let us start with developing a likelihood prototype for (1). For this purpose, we have to model for a given state vector \underline{x} , how likely its corresponding surface $\Phi_x(u, s)$ with $u \in \mathcal{U}$ and $s \in \mathcal{S}$ will *produce* a measurement $\underline{y} \in \mathbb{R}^3$. As we assume

additive Gaussian sensor noise, the generative measurement model can be written as

$$\underline{y} = \Phi_x(u, s) + \underline{v}, \quad (8)$$

where $\underline{v} \sim \mathcal{N}(\underline{0}, \mathbf{C}_v)$ is the noise variable. This generative model can be rewritten to the likelihood

$$\begin{aligned} p(\underline{y}|\underline{x}) &= \int_U \int_S p(\underline{y}|\underline{x}, s, u) \cdot p(s, u|\underline{x}) \, ds \, du \\ &= \int_U \int_S \underbrace{\mathcal{N}(\underline{y}; \Phi_x(u, s), \mathbf{C}_v)}_{\text{sensor model}} \cdot \underbrace{p(s, u|\underline{x})}_{\text{source model}} \, ds \, du \end{aligned} \quad (9)$$

by assuming u and s are probabilistically known. The *sensor model* then specifies the expected Gaussian sensor noise with covariance matrix \mathbf{C}_v when measuring a specific $\Phi_x(u, s)$, and the *source model* specifies how likely it is that this source is measured at all. However, when designing the source model $p(s, u|\underline{x})$, we are faced with the so-called *association problem* [7], as it is generally not known from which point in the surface a measurement originates.

B. The Association Problem and its Solutions

a) *Spatial Distribution Model (SDM)*: An intuitive solution to the association problem is to make the assumption that measurements will originate uniformly from the entire surface [13], or from the part that is visible to the sensor [14]. In doing so, $p(s, u|\underline{x})$ is modeled as a uniform distribution in (9) and yields unbiased estimators [13] even in the presence of high noise (given the true distribution is uniform as well). Note that the popular Random Matrices approach [15] for ellipses is an instance of SDM. However, SDMs are not suitable for the shapes in this article as, due to the involved integrals, they are computational expensive when it comes to more complex shapes. In addition, in situations where $p(s, u|\underline{x})$ is not known and modeled incorrectly, the resulting estimator will find biased estimates [7].

b) *Greedy Association Model (GAM)*: A less expensive, yet naïve solution to the association problem refers to the assumption that each measurement \underline{y} originates from its most likely (or closest) source $\Phi_x(\hat{u}, \hat{s})$ on the surface, which simplifies the likelihood prototype [10] to

$$p(\underline{y}|\underline{x}) = \mathcal{N}(\underline{y}; \Phi_x(\hat{u}, \hat{s}), \mathbf{C}_v). \quad (10)$$

The resulting estimator then will find parameters that minimize a distance measure between the measurements and the object surface. Approaches using (10) include *orthogonal least squares* and *geometric fitting*. However, these approaches are known to be biased in the presence of noise [16] and unable to observe specific extent parameters of objects, such as the length of a line segment [13].

c) *Partial Information Model (PIM)*: The third elemental solution to the association problem refers to the idea of first dividing the likelihood into “how well” measurements fit to the object and “where” on the object they are related to, and then only use the “how well” part for the measurement update.

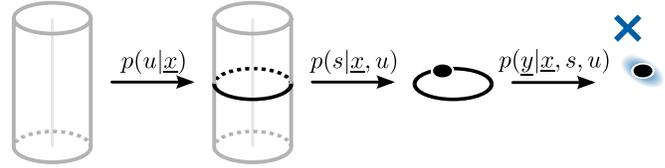


Figure 2: Sketch of an RHM for a cylinder.

This approach can be seen as an instance of *partial likelihood* [7] and yields

$$p(\underline{y}|\underline{x}) = \mathcal{N}(l_y; \mathbb{E}\{l_y\}, \text{Var}\{l_y\}) \quad (11)$$

where (for isotropic noise) the signed Euclidean distance $l_y = \pm \|\underline{y} - \Phi_x(\hat{u}, \hat{s})\|$ naturally implements the “how well” component. This partial likelihood yields an unbiased estimator according to the criterion from [17] and is a systematic way to justify the bias correction techniques in [16] and [10]. However, analogously to the GAM, it is still not capable of estimating specific extent parameters.

d) *Random Hypersurface Model (RHM)*: In [4], [6], we proposed to combine elemental models by first factorizing the likelihood prototype from (9) according to

$$p(\underline{y}|\underline{x}) = \int_U \int_S p(\underline{y}|\underline{x}, s, u) \cdot p(s|\underline{x}, u) \, ds \cdot p(u|\underline{x}) \, du, \quad (12)$$

and then using an SDM for the u variable and an instance of GAM or PIM for the s variable. The idea is visually explained in Fig. 2 for the cylinder from (2). Each transformed curve u is assigned a probability $p(u|\underline{x})$ that it will be measured by the sensor, right in the fashion of an SDM. Then, within each curve, the distribution $p(s|\underline{x}, u)$ that refers to the originating source in the curve is ignored (by using a GAM or PIM).

As a result, we can combine the likelihood for SDM and PIM by replacing the inner integral over s in (12) by (11)

$$p(\underline{y}|\underline{x}) = \int_U \mathcal{N}(l_y; \mathbb{E}\{l_y\}, \text{diag}(\text{Var}\{l_y\})) \cdot p(u|\underline{x}) \, du. \quad (13)$$

Note that we replaced the scalar distance l_y by a vector-valued expression $l_y \in \mathbb{R}^2$, as it is not straightforward to define a signed distance between a 3D measurement and a 1D curve in 3D. Instead, l_y is composed of the signed Euclidean distance of the measurement to the plane that contains the curve, and the signed Euclidean distance of the measurement’s projection onto the plane to the curve. In the following, we will refer to (13) as “RHM-PIM”.

V. PROPOSED APPROACH

In this section, we discuss the suitability of the probabilistic models for a given object and identify the “simplest” that still works. Essentially, there are two major issues that may occur when designing an estimator.

Issue 1 (The estimator cannot find a parameter at all)

In order to enable the estimator to find the value of a parameter, the likelihood must produce different values as the parameter

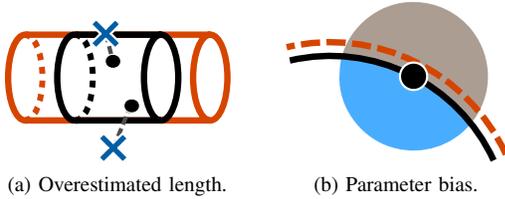


Figure 3: Issues that may occur when designing the estimator.

changes. However, when ignoring the distributions for u , s , and using distances instead (GAM and PIM), the likelihood could become invariant to parameter changes. This issue is visually explained for a cylinder in Fig. 3a. The closest distance of measurements to the object surface is identical for the true (black) and overestimated (red) height parameter. That is, when varying a parameter does not effect the closest distances, this parameter is likely to be unobservable. This issue can be either resolved by regularization terms or by incorporating knowledge about the distribution of measurement sources [5], [8], [9], [13] in the fashion of SDMs and RHM.

Issue 2 (The estimator finds a biased parameter)

It is well-known that when using a GAM, estimates of curvature parameters are likely to be biased in the presence of noise [16]. This issue is illustrated in Fig. 3b, where the black and red curves respectively mark the true boundary and its biased estimate. According to the sensor noise (filled circle), measurements of the black source are more likely to occur on the concave side of the boundary. However, the GAM-estimator will find a biased boundary that, “roughly speaking”, balances the probability mass on both sides [10]. That is, bias will occur when the surface has a significant curvature within the magnitude of the noise. In these situations, we can use an SDM or a PIM [7], [10], [16] that do not suffer from this issue.

We can now design probabilistic models for the considered objects that will not cause these issues in the estimator. In doing so, we also propose compatible recursive Bayesian estimators, including the Unscented Kalman Filter (UKF) [18], the Smart Sampling Kalman Filter (S2KF) [19], and the Progressive Gaussian Filter (PGF) [20]. We want to emphasize that estimation based on the proposed models is not restricted to these filters. They are rather a recommendation to the required complexity of the filter. While UKF and S2KF both are “Sigma Point Kalman Filters” (the latter with a variable number of deterministic samples), the PGF is a sophisticated nonlinear estimator that uses a particle-filter like measurement update.

A. Proposed Cylinder Estimator

A cylinder is potentially subject to both issues, i.e., an unobservable length and a biased radius. In consequence, we propose to use an RHM-PIM according to (13), where the known distribution for u enables the length estimation and the PIM component prevents bias. If measurements originate uniformly from the cylinder mantle, $p(u|\underline{x})$ can be modeled as $\mathcal{U}(0, 1)$ with $E\{u\} = \frac{1}{2}$ and $\text{Var}\{u\} = \frac{1}{12}$. In this form [4], the RHM-PIM can be used with a nonlinear estimator,

such as the PGF [5]. However, in [8], [9] we showed how to exploit the reflectional symmetry of the cylinder in order to use a simpler S2KF. Here, we use an instance of this estimator with $20n$ deterministically drawn samples [21].

B. Proposed Torus Estimator

In contrast to the cylinder, the torus is not affected by Issue 1, as there is no length parameter and varying the radii will always have an effect on the closest distances of points to the surface. However, as both radii are subject to Issue 2, we propose to use a PIM according to (11). Then, the torus parameters can be estimated with a standard UKF.

C. Proposed Cone Estimator

Similar to the cylinder, the cone is subject to both issues, i.e., an unobservable height and a biased radius. In consequence, we again propose to use an RHM-PIM according to (13), where the known distribution for u enables the height estimation and the PIM component prevents bias. The probability $p(u|\underline{x})$ of each scaled circle is proportional to its perimeter $2\pi \cdot (1 - u) \cdot r$ and scales linearly with u . As such, the variable u can be modeled as a triangle distribution between 0 and 1 with $E\{u\} = \frac{1}{3}$ and $\text{Var}\{u\} = \frac{1}{18}$. Then, an S2KF can be used for estimation.

D. Proposed Extruded Curve Estimator

Being essentially a cylinder, the extruded curve is subject to both issues, as well. Again, we propose to use an RHM-PIM according to (13) with $p(u|\underline{x})$ as $\mathcal{U}(0, 1)$. For deriving the closest Euclidean distances between measurement and curves, the polar Fourier curve can be approximated by a polygon. Due to the higher complexity in the shape, we propose to use a PGF instead of the S2KF.

VI. EVALUATION

For evaluation, we conducted two experiments with low noise and high noise. In both experiments, we compared the proposed approach to traditional approaches.

A. Traditional Approaches

For a static object, the “Traditional Batch” approach would be processing all measurements using an instance of GAM (10) together with a maximum likelihood estimator. A reference implementation for cylinder, torus, and cone is publicly available in the LSGE toolbox [22]. For a potentially dynamic object, the “Traditional Recursive” approach would be using an instance of GAM (10) together with a recursive Bayesian estimator such as a particle filter [12]. For the ease of a lower computational complexity, we use a similar PGF implementation [21] instead.

B. Experiments

For ground truth, we modeled all objects at position $[0.1, 0.4, 0.2]$ with an orientation $[0, 0.5, 0]$ in *axis-angle* representation. The shape parameters were set to $[r, d] = [1, 2]$ (cylinder), $[r, d] = [1.5, 4]$ (cone), $[r_1, r_2] = [1, 0.5]$ (torus), $[a_0, a_1, b_1, d] = [4, 0, 1, 6]$ (extruded curve with 2-axial symmetry [9]). Then, we simulated 1250 measurements of each object by uniformly drawing measurement sources from the surface

	Trad. Batch	Trad. Recursive		Proposed Recursive	
		Model	Filter	Model	Filter
Cylinder	LSGE	GAM	PGF	RHM-PIM	S2KF
Torus	LSGE	GAM	PGF	PIM	UKF
Cone	LSGE	GAM	PGF	RHM-PIM	S2KF
Extr. Curve	-	GAM	PGF	RHM-PIM	PGF

Table I: Overview of the implemented estimators.

	Low noise			High noise		
	T. B.	T. R.	P. R.	T. B.	T. R.	P. R.
Cylinder r	5.9	6.3	3.7	52	53	12
Torus r_1, r_2	8.0	8.3	6.1	86	90	31
Cone top	63	70	201	594	616	330
Extr. Curve a_0, a_1, b_1	-	62	45	-	425	223

Table II: RMSE of selected parameters $\times 10^{-3}$ units.

and then adding Gaussian sensor noise with $\mathbf{C}_v = 0.01 \cdot \mathbf{I}$ (low noise) and $\mathbf{C}_v = 0.1 \cdot \mathbf{I}$ (high noise).

For estimation, we implemented the approaches in Table I. We initialized all parameters with Gaussian random values, drawn from the ground truth using variances of $5 \cdot 10^{-2}$. The Fourier coefficients for the extruded curve were initialized as a circular curve with $a_0 = 5$ and $a_1 = b_1 = 0$. In the recursive approaches, we incorporated a *random walk model* [12] into the prediction step, which inflates the state covariance matrix with process noise in order to prevent local minimums. Specifically, we chose a logarithmically decreasing diagonal covariance in the magnitude from 10^{-2} to 10^{-12} . Measurement updates were performed in 250 steps with packages of 5 measurements. The following results are obtained from 100 Monte-Carlo runs.

C. Results

The average estimation result after processing all measurements is illustrated in Fig. 4. It can be seen that all estimators can find the object orientation very accurately. This accuracy also applies to the position estimates, except from a random linear shift along the length axis (cylinder, cone, extr. curve) in the traditional approaches, which is due to the missing capability of estimating the length (Issue 1).

From the second and fourth column in Fig. 4, it can be seen that the proposed approach finds accurate parameters for all objects, even in the presence of high noise. In these situations, the estimated curvature parameters of the traditional approaches are biased (Issue 2), as Table II numerically confirms. This result can be explained by the PIM-component that accounts for the bias in the proposed approach. However, the PIM-component is relatively sophisticated and might perform worse than a simpler model in some situations. An example can be found when considering the top of the cone, which is estimated more accurately by the traditional approaches for low noise.

In sum, traditional GAM-based approaches are suitable when the problematic length parameters are known in advance, and when the surface has a negligible curvature with respect to the magnitude of the noise. Otherwise, the experiments show that RHM and PIM components can effectively compensate for Issue 1 and Issue 2, respectively. Note that in situations

with low noise and the task of estimating length parameters, it might be also reasonable to combine an RHM with a GAM instead of a PIM.

VII. CONCLUSION

In this article, we developed tracking algorithms for 3D objects that can be constructed by plane curves. We discussed two issues that may occur when designing the estimator, i.e., unobservable and biased parameters. As main contribution, we showed how to avoid both issues when designing probabilistic models for a cylinder, torus, cone, as well as an extruded curve. For this purpose, we incorporated ideas from related approaches including Random Hypersurface Models, partial likelihood, and symmetric shape models.

In the evaluation, we showed that the proposed tracking algorithms are capable of finding unbiased estimates for all modeled parameters of all considered objects. Compared to a state-of-the-art fitting approach, we could reduce the RMSE in the curvature parameters by 44%-77% in the presence of high noise.

REFERENCES

- [1] A. Rivers, F. Durand, and T. Igarashi, "3D Modeling with Silhouettes," *ACM Transactions on Graphics*, vol. 29, no. 4, Jul. 2010.
- [2] C. Wu, S. Agarwal, B. Curless, and S. M. Seitz, "Schematic Surface Reconstruction," in *The 25th Conference on Computer Vision and Pattern Recognition (CVPR)*, Providence, Rhode Island, USA, 2012, pp. 1498–1505.
- [3] A. Zea, F. Faion, and U. D. Hanebeck, "Tracking Extended Objects using Extrusion Random Hypersurface Models," in *9th Workshop: Sensor Data Fusion (SDF)*, Bonn, Germany, 2014.
- [4] F. Faion, M. Baum, and U. D. Hanebeck, "Tracking 3D Shapes in Noisy Point Clouds with Random Hypersurface Models," in *Proceedings of the 15th International Conference on Information Fusion (FUSION)*, Singapore, 2012, pp. 2230–2235.
- [5] M. Baum, F. Faion, and U. D. Hanebeck, "Modeling the Target Extent with Multiplicative Noise," in *Proceedings of the 15th International Conference on Information Fusion (FUSION)*, Singapore, 2012, pp. 2406–2412.
- [6] M. Baum and U. D. Hanebeck, "Shape Tracking of Extended Objects and Group Targets with Star-Convex RHMs," in *Proceedings of the 14th International Conference on Information Fusion (FUSION)*, Chicago, Illinois, USA, 2011, pp. 1–8.
- [7] F. Faion, A. Zea, M. Baum, and U. D. Hanebeck, "Partial Likelihood for Unbiased Extended Object Tracking," in *Proceedings of the 18th International Conference on Information Fusion (Fusion 2015)*, Washington D. C., USA, 2015.
- [8] —, "Bayesian Estimation of Line Segments," in *Proceedings of the IEEE ISIF Workshop on Sensor Data Fusion: Trends, Solutions, Applications (SDF 2014)*, Bonn, Germany, 2014.
- [9] —, "Symmetries in Bayesian Extended Object Tracking," *ISIF Journal of Advances in Information Fusion (JAIF)*, vol. 10, no. 1, pp. 13–30, 2015. [Online]. Available: <http://isif.org/journal>
- [10] F. Faion, A. Zea, and U. D. Hanebeck, "Reducing Bias in Bayesian Shape Estimation," in *IEEE 17th International Conference on Information Fusion (FUSION)*. Salamanca, Spain: IEEE, 2014, pp. 1–8.
- [11] X. Rong Li and V. Jilkov, "Survey of Maneuvering Target Tracking. Part I: Dynamic Models," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1333–1364, Oct. 2003.
- [12] Z. Chen, "Bayesian filtering: From Kalman filters to particle filters, and beyond," *Statistics*, 2003.
- [13] M. Werman and D. Keren, "A Bayesian Method for Fitting Parametric and Nonparametric Models to Noisy Data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 5, pp. 528–534, 2001.
- [14] N. Petrov and L. Mihaylova, "A Novel Sequential Monte Carlo Approach for Extended Object Tracking Based on Border Parameterisation," in *Proceedings of the 14th International Conference on Information Fusion (FUSION)*, Chicago, Illinois, USA, 2011, pp. 306–313.

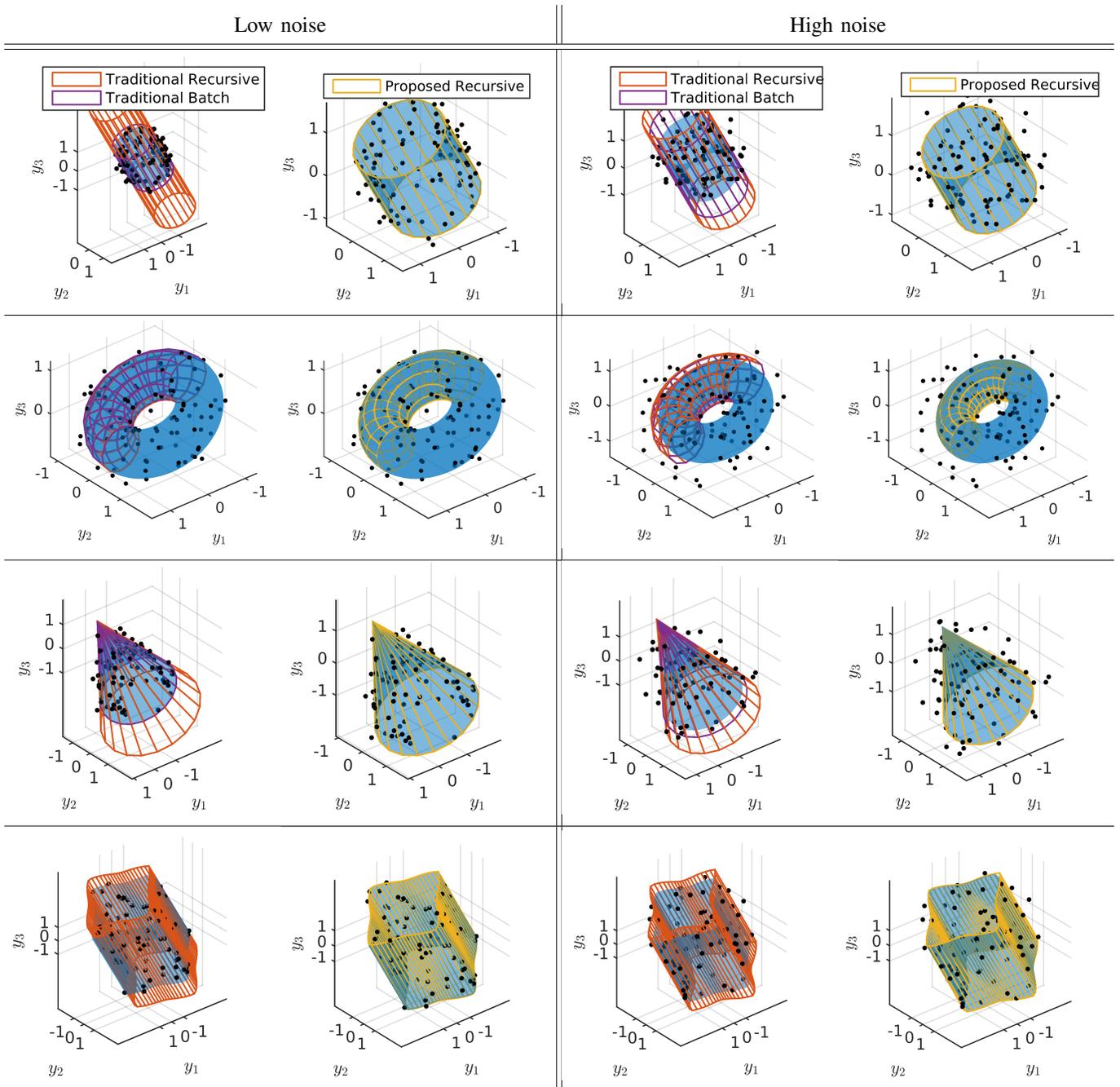


Figure 4: Average estimates for 100 Monte Carlo runs. Exemplary measurements are drawn to illustrate the noise level.

- [15] M. Feldmann, D. Franken, and W. Koch, "Tracking of Extended Objects and Group Targets Using Random Matrices," *IEEE Transactions on Signal Processing*, vol. 59, no. 4, pp. 1409–1420, Apr. 2011.
- [16] T. Okatani and K. Deguchi, "On Bias Correction for Geometric Parameter Estimation in Computer Vision," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 959–966, 2009.
- [17] V. Godambe and M. Thompson, "Estimating Equations in the Presence of a Nuisance Parameter," *The Annals of Statistics*, vol. 2, no. 3, pp. 568–571, 1974.
- [18] S. J. Julier and J. K. Uhlmann, "Unscented Filtering and Nonlinear Estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, Mar. 2004.
- [19] J. Steinbring and U. D. Hanebeck, "S2KF: The Smart Sampling Kalman Filter," in *Proceedings of the 16th International Conference on Information Fusion (FUSION)*, Istanbul, Turkey, 2013, pp. 2089–2096.
- [20] J. Steinbring and U. D. Hanebeck, "Progressive Gaussian Filtering Using Explicit Likelihoods," in *Proceedings of the 17th International Conference on Information Fusion (Fusion 2014)*, Salamanca, Spain, 2014.
- [21] J. Steinbring, "Nonlinear Estimation Toolbox," 2015. [Online]. Available: <https://bitbucket.org/nonlinearestimation/toolbox>
- [22] I. Smith, "LSGE: The Least Squares Geometric Elements Library," Teddington, Middlesex, UK, 2004. [Online]. Available: <http://www.eurometros.org/metros/packages/lsgel/>