Shape Tracking Using Fourier-Chebyshev Double Series for 3D Distance Measurements

Tim Baur and Johannes Reuter
Institute of System Dynamics (ISD)
University of Applied Sciences Konstanz (HTWG)
Konstanz, Germany
{tbaur,jreuter}@htwg-konstanz.de

Antonio Zea and Uwe D. Hanebeck
Intelligent Sensor-Actuator-Systems Laboratory (ISAS)
Karlsruhe Institute of Technology (KIT)
Karlsruhe, Germany
{antonio.zea,uwe.hanebeck}@kit.edu

***Abstract*—In the past years, algorithms for 3D shape tracking using radial functions in spherical coordinates represented with different methods have been proposed. However, we have seen that mainly measurements from the lateral surface of the target can be expected in a lot of dynamic scenarios and only few measurements from the top and bottom parts leading to an error-prone shape estimate in the top and bottom regions when using a representation in spherical coordinates. We, therefore, propose to represent the shape of the target using a radial function in cylindrical coordinates, as these only represent regions of the lateral surface, and no information from the top or bottom parts is needed. In this paper, we use a Fourier-Chebyshev double series for 3D shape representation since a mixture of Fourier and Chebyshev series is a suitable basis for expanding a radial function in cylindrical coordinates. We investigate the method in a simulated and real-world maritime scenario with a CAD model of the target boat as a reference. We have found that shape representation in cylindrical coordinates has decisive advantages compared to a shape representation in spherical coordinates and should preferably be used if no prior knowledge of the measurement distribution on the surface of the target is available.**

***Index Terms*—3D shape tracking, Fourier-Chebyshev double series, LiDAR.**

I. Introduction

With the high resolution of nowadays sensors such as LiDAR, RADAR, or depth cameras, a vast amount of measurements per time step are generated from a target. Processing these measurements in a filtering algorithm leads to the field of extended object tracking (EOT) [1], [2] where the extent or shape is estimated concurrently to the kinematic state of the target (see Fig. 1). Tracking extended objects can have a huge effect on a lot of applications such as autonomous systems since the surrounding can be described in more detail. Furthermore, modules based on environment perception such as planning and control can operate with higher precision. The motion commands in autonomous applications, for example, can only be executed if the extent parameters of the dynamic objects in the environment are known. A lot of these applications require high-precision information in 3D space and modern sensors inherently provide those. A filtering algorithm and especially the measurement model predicting the measurement sources must therefore also operate in 3D space. Designing a flexible, robust, efficient, and suitable measurement model in 3D space is, however, a challenging task.

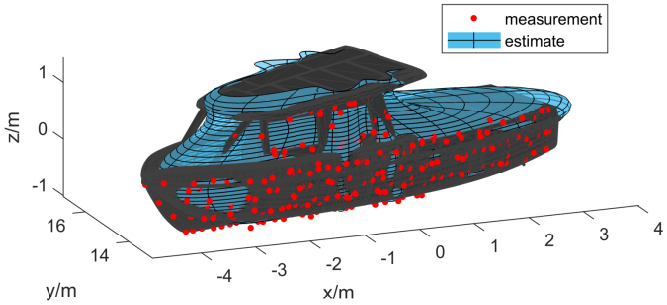

Fig. 1: Result of "Solgenia" shape tracking with artificial measurements at a specific time step. CAD as black surface.

In the past, effort has been made to handle 3D measurements in various measurement model approaches. In [3], a modified Random Hypersurface Model (RHM) [4] has been designed to track objects using a cylindrical shape. In [5], this approach has been extended to a scaled extrusion RHM for a circular base shape, and for a few more shapes in [6]. In [7], a human body model constructed by different cylinders for every body part is presented. Another model can be found in [8], where non-uniform rational B-splines (NURBS) surfaces are used for a flexible shape representation. All these models have in common that a predefined shape must be chosen as prior knowledge before the tracking filter can be initialized. If the target, however, does not fit this shape, poor tracking results can be the consequence. A solution to this problem can be either to choose an appropriate shape on-line or to track the shape itself. In shape tracking, modeling the shape as star-convex radial function in spherical coordinates has been a common approach in the past years. In doing so, a sufficient representation of the shape function has to be chosen and the shape coefficients have to be estimated. In [9]–[12], spherical double Fourier series (DFS), Gaussian processes (GP), and spherical harmonics (SH) have been used for shape representation. When using spherical DFS or SH, the continuous shape function $f : \mathbb{R}^2 \to \mathbb{R}$, $[\theta, \varphi]^T \mapsto r$ is expanded in different bases and the shape can be tracked by estimating the coefficients of the expansion. For the GP model,

the shape function is discretized and the function distribution is modeled as $f(\gamma) \sim \mathcal{GP}\left(\mu(\gamma), k(\gamma, \gamma')\right)$ with mean function $\mu(\gamma)$, kernel function $k(\gamma, \gamma')$, and $\gamma = [\theta, \varphi]^T$.

Shape tracking in spherical coordinates, however, raises a problem: if mainly measurements from the lateral surface of the target are gathered, information from the top and bottom parts of the target is missing and estimates can be error-prone in these areas. This can naturally be solved if the radial function for shape representation is defined in cylindrical instead of spherical coordinates. In [13], cylindrical DFS (CDFS) as a solution to a boundary value problem of the Laplace equation are used for expanding a cylindrical radial function. In this study, we found that a shape representation in cylindrical coordinates should preferably be used if mainly measurements from the lateral surface of the target can be expected. Using DFS, an expansion for a radial function in cylindrical coordinates can be obtained if the function is 0 at the top and bottom, and piecewise continuous, i.e. closed and therefore periodic in the height parameter. In practice, however, most shapes are not periodic. A desirable shape expansion, therefore, comprises a periodic basis in the angular and a non-periodic basis in the height parameter.

In this paper, we propose to use a Fourier-Chebyshev double series (FCDS) for shape representation in cylindrical coordinates. Chebyshev series [14] yield a basis for expanding non-periodic functions in the interval $[-1, 1]$, while Fourier series are well known for expanding periodic functions in the interval $[0, 2\pi]$. The contributions of the paper are the following:

- We propose a measurement model for 3D shape tracking using FCDS.
- We investigate an efficient extended Kalman filter (EKF) implementation [15] for inference.
- We present a comprehensive study using simulated and real-world LiDAR data in a maritime environment for comparing our approach to shape tracking in spherical coordinates and our previous approach [13].

The research vessel "Solgenia" of the HTWG Konstanz is utilized as target in both, the real-world and the simulated investigation. In the simulated investigation, a CAD model is used to generate simulated LiDAR measurements (see Fig. 1).

II. Shape Representation

A. Chebyshev Polynomials

Chebyshev Polynomials $T_n(u)$ of the first kind [14] are defined by

$$T_n(u) = \cos(n\theta), \quad n \in \mathbb{N}_0,\ \theta \in [0, \pi], \tag{1}$$

with $u = \cos(\theta)$. This relation forms a series of orthogonal polynomials in the interval $u \in [-1, 1]$. The polynomials can either be calculated using the recursion

$$T_n(u) = 2uT_{n-1}(u) - T_{n-2}(u) \tag{2}$$

with the initial conditions $T_0(u) = 1$ and $T_1(u) = u$, or with the explicit expression

$$T_n(u) = \cos(n \arccos(u)). \tag{3}$$

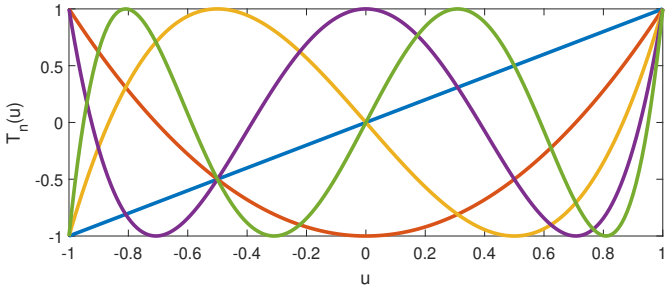

Fig. 2: Chebyshev Polynomials up to order $n = 5$.

The explicit derivative subject to u yields

$$T_n'(u) = \frac{n \sin(n \arccos(u))}{\sqrt{1 - u^2}}. \tag{4}$$

The first five polynomials and their derivatives that are needed in the derivation of the EKF are explicitly given by

$$T_1(u) = u \qquad T_1'(u) = 1 \tag{5}$$
$$T_2(u) = 2u^2 - 1 \qquad T_2'(u) = 4u \tag{6}$$
$$T_3(u) = 4u^3 - 3u \qquad T_3'(u) = 12u^2 - 3 \tag{7}$$
$$T_4(u) = 8u^4 - 8u^2 + 1 \qquad T_4'(u) = 32u^3 - 16u \tag{8}$$
$$T_5(u) = 16u^5 - 20u^3 + 5u \qquad T_5'(u) = 80u^4 - 60u^2 + 5 \tag{9}$$

and depicted in Fig. 2. Using these polynomials, any non-periodic and piecewise smooth and continuous function $g(u)$ in the interval $u \in [-1, 1]$ can be expanded as

$$g(u) \approx \sum_{n=0}^{n_u} a_n T_n(u) \tag{10}$$

using appropriate coefficients a_n.

B. FCDS Radial Function

In our model, we aim to represent the shape of the target using a radial function in cylindrical coordinates. An illustration of such a radial function can be found in [13]. The shape is meant to be non-periodic in the height parameter $u \in [-1, 1]$ and periodic in the angular parameter $\theta \in [0, 2\pi]$. In [13], the shape function was periodic in both parameters. With an estimated height of the target, the height parameter u can easily be calculated. By combining Fourier series and Chebyshev polynomials, the radial function can be expressed as

$$f(\theta, u) = \frac{a_{00}}{4} + \frac{1}{2}\sum_{n=1}^{n_u} a_{n0} T_n(u) + \frac{1}{2}\sum_{m=1}^{n_\theta} \Psi_{0m}(\theta) + \sum_{n=1}^{n_u}\sum_{m=1}^{n_\theta} T_n(u)\Psi_{nm}(\theta), \tag{11}$$

where $\Psi_{nm}(\theta)$ is the Fourier series

$$\Psi_{nm}(\theta) = a_{nm}\cos(m\theta) + b_{nm}\sin(m\theta). \tag{12}$$

This representation of the shape requires $1 + n_u + 2n_\theta + 2n_u n_\theta$ coefficients. If the shape should involve a vertical plane of

symmetry aligned to the orientation and the direction of motion of the target, the Fourier series can be forced to be even by truncating the sinusoidal components as

$$\Psi_{nm}(\theta) = a_{nm}\cos(m\theta). \tag{13}$$

In doing so, the number of coefficients for shape representation reduces to $1 + n_u + n_\theta + n_u n_\theta$. The expansion containing a vertical plane of symmetry is of special interest since only the side of the target facing the sensor is measured in a lot of applications. Then, the unseen backside can be assumed as being symmetric to the front side which is appropriate for a lot of targets. Additionally, this representation requires considerably fewer coefficients. In the rest of the paper, we will therefore only use a shape representation involving a vertical plane of symmetry for our investigation.

Modeling the shape using a radial function in cylindrical coordinates instead of spherical coordinates has decisive advantages. First of all, it can naturally handle the problem that mainly measurements from the lateral surface are gathered in applications such as autonomous systems. Furthermore, the representation is more flexible. A radial function in spherical coordinates can only represent star-convex shapes whereas radial functions in cylindrical coordinates have to be star-convex only for a slice at a specific height. This makes our approach, to the best of our knowledge, the most flexible shape tracking procedure in 3D space presented so far.

III. Measurement Model

A. System State Space

In this paper, the system state $\underline{x}_k$ comprises the position $\underline{m}_k$, the orientation ϕ_k in the xy-plane, the kinematic state components $\underline{w}_k$ containing velocity and turn rate, the height h_k, and the shape parameters $\underline{p}_k$ and is given as

$$\underline{x}_k = \left[\underline{m}_k^T, \phi_k, \underline{w}_k^T, h_k, \underline{p}_k^T\right]^T. \tag{14}$$

Since we assume a symmetrical shape in the further course of the paper, the parameter vector

$$\underline{p}_k = [a_{00}, \ldots, a_{10}, \ldots, a_{01}, \ldots, a_{11}, \ldots]^T \tag{15}$$

only contains the coefficients of the cosine components of the Fourier series $\Psi_{nm}(\theta)$.

B. Measurement Equation

In our application, we assume that a measurement set

$$\mathcal{Y}_k = \left\{\underline{y}_{k,l}\right\}_{l=1}^{n_k} \tag{16}$$

of n_k measurements is gathered every time step k. To estimate the hidden system state $\underline{x}_k$ using the measurement set $\mathcal{Y}_k$, a measurement equation that relates the system state to a measurement is needed. In this paper, we assume a measurement source model [16]

$$\underline{y}_{k,l} = \underline{z}_{k,l} + \underline{v}_{k,l} \tag{17}$$

with measurement source $\underline{z}_{k,l}$ and white Gaussian zero-mean noise term $\underline{v}_{k,l} \sim \mathcal{N}\left(\underline{0}, \mathbf{C}_{\underline{v}}\right)$. The measurements of a specific measurement set are assumed to be mutually independent. Therefore, it is sufficient to define the measurement model for a single measurement and to perform n_k updates in the update step of the EKF. Using (11) and (17), a measurement source on the lateral surface of the target can be expressed as

$$\underline{y}_{k,l} = \underline{m}_k + \mathbf{R}_{\phi_k}\begin{bmatrix} f(\theta_{k,l}, u_{k,l})\cos(\theta_{k,l}) \\ f(\theta_{k,l}, u_{k,l})\sin(\theta_{k,l}) \\ u_{k,l}h_k \end{bmatrix} + \underline{v}_{k,l} \tag{18}$$

with the rotation matrix $\mathbf{R}_{\phi_k}$, the angular parameter $\theta_{k,l} \in [0, 2\pi]$, and the height parameter $u_{k,l} \in [0, 1]$ of a specific measurement source. With the range of the height parameter $u_{k,l}$, the position $\underline{m}_k$ is modeled as being at the bottom of the target.

C. Measurement Source Association

In EOT, the exact measurement source $[\theta_{k,l}, u_{k,l}]^T$ is unknown and we are therefore faced with the measurement source association problem [17]. For a measurement source model (18), three different association schemes in 3D space are known in the literature: a greedy association model (GAM), an extrusion RHM, and a partial information model (PIM) [16]. All these models yield different procedures for estimating the measurement source parameters $\left[\hat{\theta}_{k,l}, \hat{u}_{k,l}\right]^T$. Since a measurement source in 3D space is described with two parameters, different association models can arbitrarily be combined for those two parameters.

When applying an explicit likelihood for the parameter u in combination with a GAM for the parameter θ, named extrusion RHM, it was shown in [3] that the parameter u in (18) can be interpreted as multiplicative noise term. A height estimation with this combination can be expected to be unbiased [18]. In [19], [20], it was seen that the Kalman Filter and its nonlinear variants are inconsistent for systems modeled using a multiplicative noise term. A proposed solution to this is to apply a quadratic extension in the measurement equation. This extension, however, can only be applied to linear equations. If $f(\theta_{k,l}, u_{k,l}) = 1$ in (18), i.e. the shape is modeled as a cylinder, the quadratic extension can be applied for the linear z-coordinate of the measurement equation. In our model (18), however, the multiplicative noise not only affects the z-coordinate but also the xy-coordinates leading to inconsistency of the Kalman Filter in these coordinates as well. In [21], a polynomial approximation of a nonlinear measurement equation was presented for filtering systems corrupted by multiplicative noise. This or similar approaches were, to the best of our knowledge, however, never adapted to EOT. Another possibility to implement an extrusion RHM for our shape tracking model would be to apply a nonlinear filter that can handle arbitrary non-Gaussian and non-additive noise with no assumption for the mean value.

Due to the complex filtering and the ensuing high computational costs of an extrusion RHM, we propose to adapt a GAM for both shape parameters. The likelihood for both

models can for example be found in [5], [16], [22]. Therefore, the measurement

$$\tilde{\underline{y}}_{k,l} = \mathbf{R}_{\phi_k}^{-1}(\underline{y}_{k,l} - \underline{m}_k) \tag{19}$$

in local coordinates with $\tilde{\underline{y}}_{k,l} = \left[\tilde{y}_{k,l}^x, \tilde{y}_{k,l}^y, \tilde{y}_{k,l}^z\right]^T$ has to be calculated. Then, the angular associate

$$\hat{\theta}_{k,l} = \text{atan2}\left(\tilde{y}_{k,l}^y, \tilde{y}_{k,l}^x\right) \tag{20}$$

and the height parameter

$$\hat{u}_{k,l} = \begin{cases} 0 & \tilde{y}_{k,l}^z < 0 \\ \frac{\tilde{y}_{k,l}^z}{h_k} & 0 \leq \tilde{y}_{k,l}^z \leq h_k \\ 1 & \tilde{y}_{k,l}^z > h_k \end{cases} \tag{21}$$

equally to [13] can be examined. Since the Chebyshev polynomials are defined for the range $\hat{u}_{k,l} \in [-1, 1]$ we need to perform the parameter transform $f\left(\hat{\theta}_{k,l}, \hat{u}_{k,l}^*\right) = f\left(\hat{\theta}_{k,l}, 2\hat{u}_{k,l} - 1\right)$ to shift $\hat{u}_{k,l}$ to the desired range.

In [5], [17], [22], the measurement source is greedily associated to the nearest point on the contour with respect to the measurement. This point is assumed to be the most likely measurement source that has created the measurement. In 2D space, this can be done efficiently for arbitrary shapes by approximating the contour as a polygonal chain. In 3D space, an exact calculation for a specific shape is for example given in [22]. For arbitrary shapes in 3D space, however, calculating the nearest point on the contour is a nontrivial challenge. We have seen that our association of the measurement source is very efficient and gives good results. Approximating the nearest point on the contour as measurement source is, therefore, left for future work.

IV. Implementation

In this paper, we use an EKF for filtering rather than a sigma-point filter such as the unscented Kalman filter (UKF) [23] or its extensions [24]. The EKF was chosen since the filter performs quite accurately as will be seen later in Sec. V. Additionally, it is a very efficient filter in terms of computation time. Due to the fact that the xy-coordinates in (18) are expressed by a nonlinear function, a second-order EKF [25], [26] could be beneficial for future investigations.

The EKF equations are well-known in the literature [15] and are therefore not shown here in detail. Given a predicted system state $\hat{\underline{x}}_{k|k-1}$, the linearized measurement model is given as the Jacobian $H_k = \nabla_{\underline{x}^T} \underline{y}(\underline{x})|_{\underline{x}=\underline{x}_{k|k-1}}$. The derivatives for the Jacobian can be found in the appendix. In EOT, multiple measurements are gathered by the sensor. The update for the measurement set $\mathcal{Y}_k$ can then be performed sequentially or in a single update step. When sequentially updating the system state for each measurement, the results differ with different orderings of the measurements. We, therefore, perform a single update step for reproducible results. An overview of the procedure can be found in Algorithm 1.

Algorithm 1 Fourier-Chebyshev Shape Tracking EKF Update

1: Initialize stacked predicted measurement array and stacked Jacobian matrix.
2: **for** $l = 1, \ldots, n_k$ **do**
3: Measurement $\tilde{\underline{y}}_{k,l}$ in local coordinates using (19).
4: Angle parameter $\hat{\theta}_{k,l}$ using (20).
5: Height parameter $\hat{u}_{k,l}$ using (21).
6: Radius $f\left(\hat{\theta}_{k,l}, \hat{u}_{k,l}\right)$ using (11).
7: Measurement prediction using (18) and saving in stacked predicted measurement array.
8: Measurement Jacobian using appendix (24) to (44) and saving in stacked Jacobian matrix.
9: **end for**
10: Single EKF update using stacked predicted measurement array and stacked Jacobian matrix.

When implementing the Chebyshev polynomials, we used a list for the polynomials themselves and their derivatives (5–9) [14] rather than the explicit formulas (3) and (4) due to numerical issues.

V. Results

To evaluate our proposed shape tracking algorithm, we analyze the performance in a simulated and a real-world scenario, both in a maritime environment. We compare our method to shape tracking using a radial function in spherical coordinates represented as spherical DFS (SDFS) [9], GP [10], [11], and SH [12], and to our previous work [13] using CDFS. All four reference methods are implemented using a smart sampling Kalman filter (S²KF) [24]. In both scenarios, the performance is quantified using the intersection-over-union (IOU) measure [10], [27]. The IOU gives a good measure of the overall performance since the position and shape estimate can be ambiguous. In both scenarios, the CAD model of the "Solgenia" is applied as a reference. As motion model, a coordinated turn (CT) model [28] is deployed.

A. Simulation Results

In the simulated scenario, we generated a reference trajectory with 300 time steps and a measurement rate of 10 Hz over 30 s using the CT model. The CAD model of the "Solgenia" is moved along this trajectory and a simulated LiDAR sensor generates the surface points of the boat without any noise. The isotropic measurement noise with a standard deviation of $\sigma_{\underline{v}} = 0.05\,\text{m}$ is then added to the surface points generated by the artificial LiDAR. Every filter is initialized using the first measurement set. As xy-position, the mean

$$\underline{m}_1^{xy} = \frac{1}{n_1} \sum_{\underline{y}_{1,l} \in \mathcal{Y}_1} \underline{y}_{1,l}^{xy} \tag{22}$$

using the xy-coordinates $\underline{y}_{1,l}^{xy}$ of the measurements is applied. The z-position m_1^z and height h_1 are initialized using the minimum and maximum z-values. The covariance matrix

$$\mathbf{C}_{\underline{m}} = \text{Cov}\left(\underline{y}_1^x, \underline{y}_1^y\right) \tag{23}$$

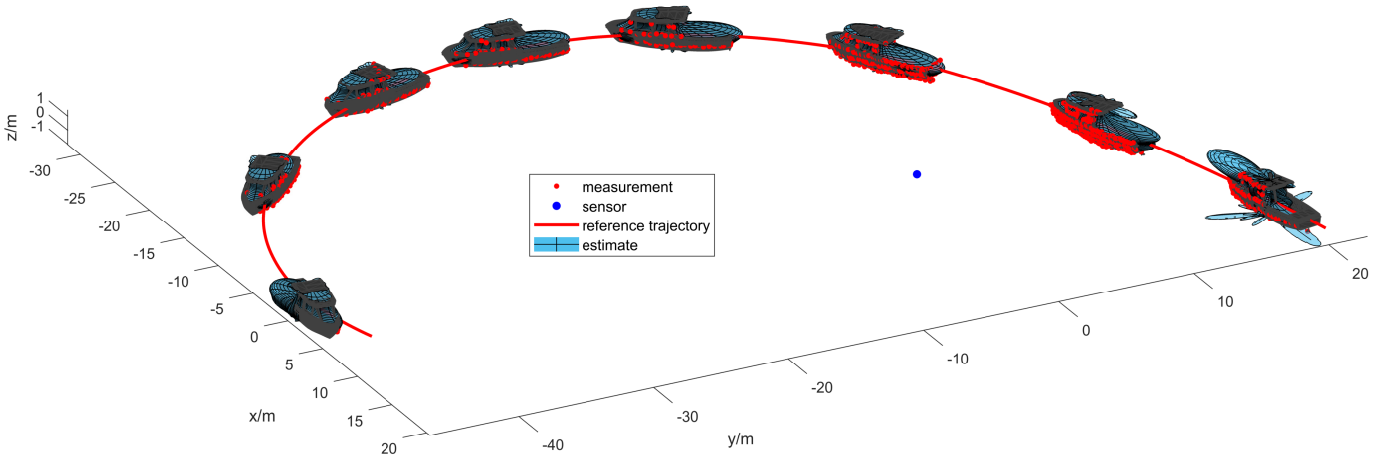

Fig. 3: Tracking results of the simulated scenario. CAD as black surface. Time steps $k = 5, 45, 85, \ldots$ are shown.

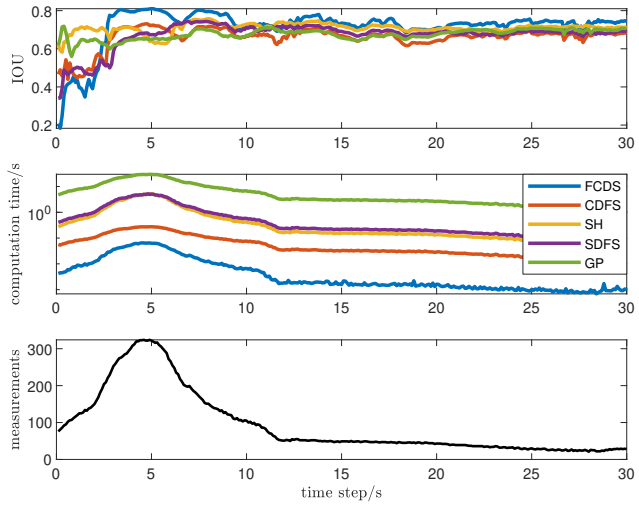

Fig. 4: IOU, computation time, and number of measurements of the MC simulation.

is used for the covariance initialization of the xy-position and the orientation of the covariance ellipse for the initialization of the yaw angle ϕ_1. The first coefficient of the shape parameters is set to be $a_{00_1} = 2$ while the rest of the coefficients are set to be 0. The kinematic state is initialized as $\underline{w}_1 = [2\,\text{m/s}, 1°/\text{s}]^T$ for velocity and yaw rate, and the variances as $\mathbf{C}_{\underline{m}}$ for the position in xy, $0.2\,\text{m}$ for the position in z, $5°$ for the orientation, $0.2\,\text{m/s}$ for the velocity, $5°/\text{s}$ for the yaw rate, and $0.3{\cdot}\mathbf{I}$ for the shape parameters. In future work, the initialization could be conducted by using a maximum likelihood estimator with the first measurement set. For GP, the initialization procedure proposed in [10] is adapted.

In Fig. 3, the simulation environment with the sensor position and the estimates for a specific scenario is depicted. The scenario starts on the right and ends on the left. The first time step shown on the right is at $k = 5$, then every 40th measurement set, reference, and estimate are shown. For the FCDS with a vertical plane of symmetry, the parameters are set to $n_u = 6$, and $n_\theta = 8$ resulting in 63 coefficients. The first time step shown in the figure still exhibits larger outbursts of the estimated shape. Due to the applied EKF, the state is linearized in the shape parameters and thus takes some time to adapt the correct shape coefficients. The procedure, however, reliably converges to a plausible shape estimate after a few time steps.

For a better overall investigation of the proposed method, a Monte Carlo (MC) simulation with 100 runs was performed. In this simulation, the reference trajectory of Fig. 3 was applied. The measurement noise, however, was drawn randomly in every simulation run. The shape tracking procedures for comparison were initialized with the same procedure presented before. For SH, we used 169, for spherical DFS 43, and for CDFS 54 coefficients. The difference in the number of coefficients can be tied to the fact that each algorithm requires more or fewer coefficients to achieve the same complexity in the shape representation. For GP, we used 250 basis points and restricted the elevation angle to the interval $[40°, 140°]$ as proposed in [11] to force the GP to focus more on the lateral surface where enough measurements can be expected. The number of coefficients was set to give good results for each method, but was not tuned. In practice, this tuning parameter highly depends on the application and maximum permissible computation time.

The results of the MC simulation can be seen in Fig. 4. In this figure, the mean IOU and mean update step computation time for every time step and the number of simulated measurements are depicted. It can be seen that FCDS has the best overall performance and the lowest computation time compared to the other methods. In the beginning, the procedure takes some time to converge but ends up being very consistent in accuracy. The IOU measure is the highest between 3s and 10s since most measurements are gathered in this period. The computation for GP is the highest since most shape coefficients are within the system state for this method. On the other hand, computation time is the lowest for FCDS not only due to the fact that fewer coefficients are needed for shape representation

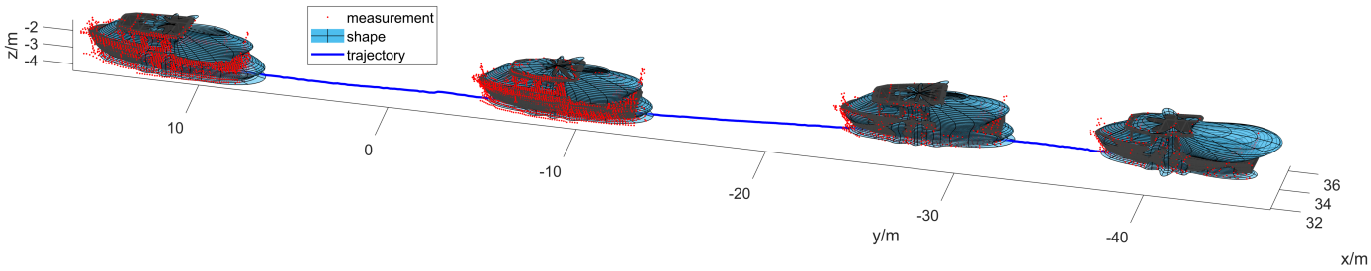

Fig. 5: Tracking results of the real-world scenario. CAD as black surface. Time steps $k = 1, 100, 200, 300$ are shown.

Fig. 6: Picture of the real-world scenario.

but also because the EKF is more efficient than the S²KF. Compared to our previous work [13], FCDS performs more accurately since the shape parametrization better fits the target than CDFS. The simulation was conducted using MATLAB R2022b on an Intel(R) Xeon(R) X5680 CPU with 3.33 GHz.

B. Real Data Results

For the real-world experiment, a multilayer LiDAR sensor with 128 Layers, 360° horizontal and 40° vertical field of view, a measurement range of up to 245 m, a frame rate of 10 Hz, a minimum angular resolution of 0.1°, and a horizontal angular resolution between 0.1° and 0.4° [29] was placed statically on the Rhine river bank in Constance. The research vessel "Solgenia" drove from right to left through the field of view of the LiDAR sensor. As a reference for the state of the "Solgenia", an RTK-GPS was placed on the roof of the vessel. Another RTK-GPS was placed on the sensor assembly where the LiDAR was mounted as well. Using the GPS positions and GPS times, we were able to transform the measurements and the reference position to the same coordinate system and interpolate the data to the same time steps. In Fig. 6, the "Solgenia" on the Rhine river recorded from a camera mounted on the sensor assembly can be seen.

In this investigation, we examined a scenario of 300 time steps over 30 s like in the simulated scenario. The motion prediction was performed using a CT model [28] with a standard deviation of 3 m/s in the velocity and $5\,^\circ/\mathrm{s}$ for the yaw rate. The z-position is modeled to be constant with a standard deviation of 0.2 m. The isotropic measurement noise was set to have a standard deviation of $\sigma_{\underline{v}} = 0.025\,\mathrm{m}$. The initialization procedure was taken from the simulated scenario. Also in the real-world scenario, we compared our shape tracking method to all shape representations presented before in the simulated scenario and used the same amount of coefficients for every procedure.

In Fig. 5, the qualitative results of this scenario are depicted. The CAD model of the "Solgenia" is again deployed as a reference. The blue trajectory in the figure is the position estimate. The scenario starts on the right with the first time step $k = 1$ and ends on the left with the last time step $k = 300$. In between, the time steps $k = 100$ and $k = 200$ are shown. The measurements shown were only recorded in the respective time step. The shape estimate is already very reasonable in the first time step and has high accuracy in the whole scenario. However, a difference between the CAD model and the real "Solgenia" is that the real one has a flagpole mounted on the stern of the boat that is not included in the CAD model. Since the radial function in cylindrical coordinates (11) is star-convex on every height section, the area above the rear of the boat is included in the overall shape estimate due to the measurements gathered by the flagpole. This causes the estimation performance of the real-world scenario to be slightly worse than in the simulated one. Visually, however, the estimation accuracy can be considered similar.

In Fig. 7, the quantitative results of the real-world scenario are depicted. For this, we applied the same measures, namely IOU and computation time, as in the simulated scenario before. The CAD model of the "Solgenia" is applied as a shape reference. The IOU measure using FCDS is slightly worse than in the simulated scenario. This, however, can be explained by the flagpole not included in the CAD model. Besides, the figure shows a reasonable tracking result of our proposed method over the whole scenario. The approaches with radial functions in spherical coordinates show very poor performance in this scenario. In comparison to the simulated scenario, a part of the boat is underwater leading to no measurements from the bottom part of the boat. In this area, the shape estimate with a radial function in spherical coordinates has very large outbursts and can never converge. This result illustrates very clearly the advantage of a shape function in cylindrical coordinates over a function in spherical coordinates. Our previous approach [13] diverges in this scenario since the shape parametrization using CDFS does not fit the target properly. Besides being the most accurate method, our shape tracking procedure is also one of the most efficient in this comparison. The calculations were performed with the same computer as before.

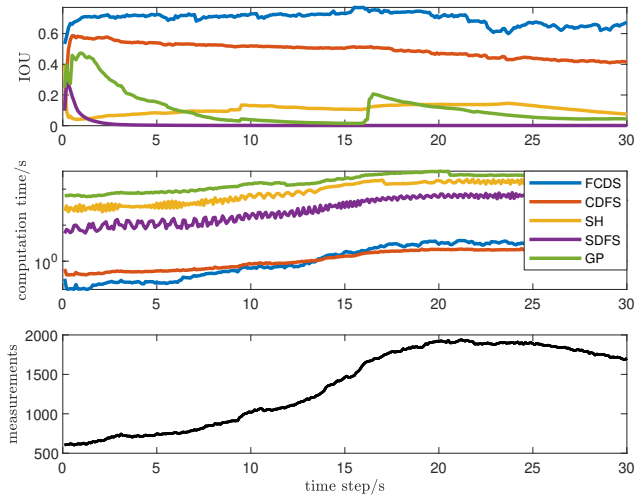

Fig. 7: IOU, computation time, and number of measurements of the real-world scenario.

VI. Conclusion and Future Work

In this paper, we proposed a new representation for 3D shapes using a radial function in cylindrical coordinates expanded as FCDS. Moreover, we utilized an EKF for shape tracking with a GAM for the measurement source association. Although an extrusion RHM is known to be more accurate for height estimation than a GAM, we have seen that our method performs very well in this regard. We pointed out several reasons why an extrusion RHM is much more costly in terms of computation time and complex to implement. We investigated our method in a simulated and a real-world scenario and compared it to shape representations using a radial function in spherical coordinates. We have seen that in both scenarios our method shows the best performance in terms of the largest IOU measure and is the most efficient one among the tested algorithms. Especially in the real-world scenario, our method had no problems in tracking the shape while the other investigated algorithms performed very poorly.

For future work, we would like to point out several options. First of all, we have seen that an EKF performs very well in this application. Theoretical investigations [25], [26], however, have shown that a second-order EKF calculates the correct mean and covariance when applying nonlinear systems. We, therefore, assume an even higher accuracy in our shape tracking procedure when applying a second-order EKF. Secondly, we suppose that we have a high redundancy in the measurements that contribute to the shape information over time. Especially for high-resolution sensors like multilayer LiDARs, procedures for selecting those elements from the whole measurement set with the highest information content would be highly recommended. Finally, a performance evaluation with more complex and faster maneuvers and under worse visibility conditions is also of high interest.

Appendix
Measurement Model Jacobian

The stacked Jacobian for a measurement set $\mathcal{Y}_k$ of n_k measurements is given as

$$H_k = \frac{\partial}{\partial \underline{x}_k}\left[\underline{y}_{k,1}^T, \cdots, \underline{y}_{k,n_k}^T\right]^T, \tag{24}$$

$$\left(\frac{\partial \underline{y}_{k,l}}{\partial \underline{x}_k}\right)^T = \left[\frac{\partial \underline{y}_{k,l}}{\partial \underline{m}_k}, \frac{\partial \underline{y}_{k,l}}{\partial \phi_k}, \frac{\partial \underline{y}_{k,l}}{\partial \underline{w}_k}, \frac{\partial \underline{y}_{k,l}}{\partial h_k}, \frac{\partial \underline{y}_{k,l}}{\partial \underline{p}_k}\right], \tag{25}$$

where $\frac{\partial \underline{y}_{k,l}}{\partial \underline{w}_k} = \mathbf{0}$. In the following, the time index k is omitted. The derivatives with respect to the position $\underline{m}$ are given as

$$\frac{\partial \underline{y}}{\partial \underline{m}} = I + \mathbf{R}_\phi \begin{bmatrix} \frac{\partial f(\hat{\theta},\hat{u}^*)}{\partial \underline{m}} \cos(\hat{\theta}) + f(\hat{\theta},\hat{u}^*)\frac{\partial cos(\hat{\theta})}{\partial \underline{m}} \\ \frac{\partial f(\hat{\theta},\hat{u}^*)}{\partial \underline{m}} \sin(\hat{\theta}) + f(\hat{\theta},\hat{u}^*)\frac{\partial sin(\hat{\theta})}{\partial \underline{m}} \\ \frac{\partial \hat{u}^*}{\partial \underline{m}} h \end{bmatrix}, \tag{26}$$

with the expressions

$$\frac{\partial f(\hat{\theta},\hat{u}^*)}{\partial \underline{m}} = \left[\frac{\partial f(\hat{\theta},\hat{u}^*)}{\partial \hat{\theta}}, \frac{\partial f(\hat{\theta},\hat{u}^*)}{\partial \hat{u}^*}\right] \begin{bmatrix} \frac{\partial \hat{\theta}}{\partial \underline{m}} \\ \frac{\partial \hat{u}^*}{\partial \underline{m}} \end{bmatrix}, \tag{27}$$

$$\begin{aligned} \frac{\partial f(\hat{\theta},\hat{u}^*)}{\partial \hat{\theta}} &= \frac{1}{2}\sum_{m=1}^{n_\theta} \frac{\partial \Psi_{0m}(\hat{\theta})}{\partial \hat{\theta}} \\ &\quad + \sum_{n=1}^{n_u}\sum_{m=1}^{n_\theta} T_n(\hat{u}^*)\frac{\partial \Psi_{nm}(\hat{\theta})}{\partial \hat{\theta}}, \end{aligned} \tag{28}$$

$$\frac{\partial \Psi_{nm}(\hat{\theta})}{\partial \hat{\theta}} = -m a_{nm}\sin(m\hat{\theta}), \tag{29}$$

$$\begin{aligned} \frac{\partial f(\hat{\theta},\hat{u}^*)}{\partial \hat{u}^*} &= \frac{1}{2}\sum_{n=1}^{n_u} a_{n0} T_n'(\hat{u}^*) \\ &\quad + \sum_{n=1}^{n_u}\sum_{m=1}^{n_\theta} T_n'(\hat{u}^*)\Psi_{nm}(\hat{\theta}), \end{aligned} \tag{30}$$

for the radial function $f(\theta, u)$, where

$$\frac{\partial \hat{\theta}}{\partial \underline{m}} = \frac{\partial \hat{\theta}}{\partial \underline{\tilde{y}}}\frac{\partial \underline{\tilde{y}}}{\partial \underline{m}}, \quad \frac{\partial \hat{u}^*}{\partial \underline{m}} = \frac{\partial \hat{u}^*}{\partial \underline{\tilde{y}}}\frac{\partial \underline{\tilde{y}}}{\partial \underline{m}}, \quad \frac{\partial \underline{\tilde{y}}}{\partial \underline{m}} = -\mathbf{R}_\phi^{-1}, \tag{31}$$

$$\frac{\partial \hat{\theta}}{\partial \underline{\tilde{y}}} = \left[-\frac{\tilde{y}^y}{\tilde{y}^{x^2}+\tilde{y}^{y^2}}, \frac{\tilde{y}^x}{\tilde{y}^{x^2}+\tilde{y}^{y^2}}, 0\right], \tag{32}$$

$$\frac{\partial \hat{u}^*}{\partial \underline{\tilde{y}}} = \left[0, 0, \begin{cases} 0 & 0 > \tilde{y}^z > h \\ \frac{2}{h} & \text{else} \end{cases}\right], \tag{33}$$

$$\frac{\partial \cos(\hat{\theta})}{\partial \underline{m}} = -\sin(\hat{\theta})\frac{\partial \hat{\theta}}{\partial \underline{m}}, \quad \frac{\partial \sin(\hat{\theta})}{\partial \underline{m}} = \cos(\hat{\theta})\frac{\partial \hat{\theta}}{\partial \underline{m}}. \tag{34}$$

The derivatives with respect to the orientation ϕ are given as

$$\begin{aligned} \frac{\partial \underline{y}}{\partial \phi} &= \frac{\partial \mathbf{R}_\phi}{\partial \phi}\begin{bmatrix} f(\hat{\theta},\hat{u}^*)\cos(\hat{\theta}) \\ f(\hat{\theta},\hat{u}^*)\sin(\hat{\theta}) \\ uh \end{bmatrix} \\ &\quad + \mathbf{R}_\phi \begin{bmatrix} \frac{\partial f(\hat{\theta},\hat{u}^*)}{\partial \phi}\cos(\hat{\theta}) + f(\hat{\theta},\hat{u}^*)\frac{\partial \cos(\hat{\theta})}{\partial \phi} \\ \frac{\partial f(\hat{\theta},\hat{u}^*)}{\partial \phi}\sin(\hat{\theta}) + f(\hat{\theta},\hat{u}^*)\frac{\partial \sin(\hat{\theta})}{\partial \phi} \\ \frac{\partial \hat{u}^*}{\partial \phi} h \end{bmatrix}, \end{aligned} \tag{35}$$

with the expressions

$$\frac{\partial f(\hat{\theta},\hat{u}^*)}{\partial \phi} = \left[\frac{\partial f(\hat{\theta},\hat{u}^*)}{\partial \hat{\theta}}, \frac{\partial f(\hat{\theta},\hat{u}^*)}{\partial \hat{u}^*}\right] \begin{bmatrix} \frac{\partial \hat{\theta}}{\partial \phi} \\ \frac{\partial \hat{u}^*}{\partial \phi} \end{bmatrix}, \tag{36}$$

$$\frac{\partial \hat{\theta}}{\partial \phi} = \frac{\partial \hat{\theta}}{\partial \underline{\tilde{y}}} \frac{\partial \underline{\tilde{y}}}{\partial \phi}, \quad \frac{\partial \hat{u}^*}{\partial \phi} = \frac{\partial \hat{u}^*}{\partial \underline{\tilde{y}}} \frac{\partial \underline{\tilde{y}}}{\partial \phi}, \tag{37}$$

$$\frac{\partial \underline{\tilde{y}}}{\partial \phi} = -\frac{\partial \mathbf{R}_\phi^{-1}}{\partial \phi}(\underline{y} - \underline{m}), \tag{38}$$

$$\frac{\partial \cos(\hat{\theta})}{\partial \phi} = -\sin(\hat{\theta})\frac{\partial \hat{\theta}}{\partial \phi}, \quad \frac{\partial \sin(\hat{\theta})}{\partial \phi} = \cos(\hat{\theta})\frac{\partial \hat{\theta}}{\partial \phi}. \tag{39}$$

The differentiation of the rotation matrix can be performed element-wise. The derivative with respect to the height h is given as

$$\frac{\partial \underline{y}}{\partial h} = \mathbf{R}_\phi \begin{bmatrix} \frac{\partial f(\hat{\theta},\hat{u}^*)}{\partial h}\cos(\hat{\theta}) \\ \frac{\partial f(\hat{\theta},\hat{u}^*)}{\partial h}\sin(\hat{\theta}) \\ \frac{\partial \hat{u}^*}{\partial h}h + \hat{u}^* \end{bmatrix}, \tag{40}$$

with

$$\frac{\partial f(\hat{\theta},\hat{u}^*)}{\partial h} = \frac{\partial f(\hat{\theta},\hat{u}^*)}{\partial \hat{u}^*}\frac{\partial \hat{u}^*}{\partial h}, \tag{41}$$

$$\frac{\partial \hat{u}^*}{\partial h} = \begin{cases} 0 & 0 > \tilde{y}^z > h \\ -\frac{2\tilde{y}^z}{h^2} & \text{else} \end{cases} \tag{42}$$

and for the shape parameter $\underline{p}$ as

$$\frac{\partial \underline{y}}{\partial \underline{p}} = \mathbf{R}_\phi \begin{bmatrix} \frac{\partial f(\hat{\theta},\hat{u}^*)}{\partial \underline{p}}\cos(\hat{\theta}) \\ \frac{\partial f(\hat{\theta},\hat{u}^*)}{\partial \underline{p}}\sin(\hat{\theta}) \\ 0 \end{bmatrix}, \tag{43}$$

with

$$\frac{\partial f(\hat{\theta},\hat{u}^*)}{\partial \underline{p}} = \Big[\frac{1}{4}, \frac{1}{2}T_1(\hat{u}^*), \cdots, \frac{1}{2}\cos(\hat{\theta}), \cdots \\ T_1(\hat{u}^*)\cos(\hat{\theta}), \cdots\Big] \tag{44}$$

for a symmetrical shape.

REFERENCES

[1] K. Granström, M. Baum, and S. Reuter, "Extended object tracking: Introduction, overview, and applications," *Journal of Advances in Information Fusion*, vol. 12, Dec. 2017.

[2] K. Granström and M. Baum, "A tutorial on multiple extended object tracking," *TechRxiv Preprint: https://doi.org/10.36227/techrxiv.19115858.v1*, 2022.

[3] F. Faion, M. Baum, and U. D. Hanebeck, "Tracking 3D shapes in noisy point clouds with random hypersurface models," in *15th International Conference on Information Fusion (FUSION)*, 2012, pp. 2230–2235.

[4] M. Baum and U. D. Hanebeck, "Random hypersurface models for extended object tracking," in *IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, 2009, pp. 178–183.

[5] A. Zea, F. Faion, and U. D. Hanebeck, "Tracking extended objects using extrusion random hypersurface models," in *Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, 2014.

[6] F. Faion, A. Zea, J. Steinbring, M. Baum, and U. D. Hanebeck, "Recursive Bayesian pose and shape estimation of 3D objects using transformed plane curves," in *Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, 2015.

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

[8] B. Naujoks, P. Burger, and H.-J. Wuensche, "Fast 3D extended target tracking using NURBS surfaces," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 1104–1109.

[9] T. Baur, J. Reuter, A. Zea, and U. D. Hanebeck, "Shape estimation and tracking using spherical double Fourier series for three-dimensional range sensors," in *2021 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, 2021, pp. 1–6.

[10] M. Kumru and E. Özkan, "Three-dimensional extended object tracking and shape learning using Gaussian processes," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, no. 5, pp. 2795–2814, 2021.

[11] F. Ebert and H. Wuensche, "Dynamic object tracking and 3D surface estimation using Gaussian processes and extended Kalman filter," in *IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 1122–1127.

[12] G. Kurz, F. Faion, F. Pfaff, A. Zea, and U. D. Hanebeck, "Three-dimensional simultaneous shape and pose estimation for extended objects using spherical harmonics," *arXiv Preprint: https://arxiv.org/abs/2012.13580*, 2020.

[13] T. Baur, J. Reuter, A. Zea, and U. D. Hanebeck, "Harmonic functions for three-dimensional shape estimation in cylindrical coordinates," in *2022 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, 2022, pp. 1–6.

[14] J. Mason and D. Handscomb, *Chebyshev Polynomials*. CRC Press, 2002.

[15] S. Challa, M. R. Morelande, D. Mušicki, and R. J. Evans, *Fundamentals of Object Tracking*. Cambridge University Press, 2011.

[16] F. Faion, "Tracking extended objects in noisy point clouds with application in telepresence systems," Ph.D. dissertation, Karlsruher Institut für Technologie (KIT), 2016.

[17] F. Faion, A. Zea, M. Baum, and U. D. Hanebeck, "Partial likelihood for unbiased extended object tracking," in *18th International Conference on Information Fusion (FUSION)*, 2015, pp. 1022–1029.

[18] F. Faion, A. Zea, M. Baum, and U. D. Hanebeck, "Bayesian estimation of line segments," in *2014 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, 2014, pp. 1–6.

[19] M. Baum, F. Faion, and U. D. Hanebeck, "Modeling the target extent with multiplicative noise," in *2012 15th International Conference on Information Fusion (FUSION)*, 2012, pp. 2406–2412.

[20] S. Yang and M. Baum, "Tracking the orientation and axes lengths of an elliptical extended object," *IEEE Transactions on Signal Processing*, vol. 67, no. 18, pp. 4720–4729, 2019.

[21] A. Germani, C. Manes, and P. Palumbo, "Polynomial extended Kalman filter," *IEEE Transactions on Automatic Control*, vol. 50, no. 12, pp. 2059–2064, 2005.

[22] T. Baur, J. Reuter, A. Zea, and U. D. Hanebeck, "Extent estimation of sailing boats applying elliptic cones to 3D lidar data," in *2022 25th International Conference on Information Fusion (FUSION)*, 2022, pp. 1–8.

[23] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, 2004.

[24] J. Steinbring and U. D. Hanebeck, "S2KF: The smart sampling Kalman filter," in *Proceedings of the 16th International Conference on Information Fusion (FUSION)*, 2013, pp. 2089–2096.

[25] S. Yang and M. Baum, "Second-order extended Kalman filter for extended object and group tracking," in *2016 19th International Conference on Information Fusion (FUSION)*, 2016, pp. 1178–1184.

[26] M. Roth and F. Gustafsson, "An efficient implementation of the second order extended Kalman filter," in *14th International Conference on Information Fusion*, 2011, pp. 1–6.

[27] M. Levandowsky and D. Winter, "Distance between sets," *Nature*, vol. 234, pp. 34–35, 1971.

[28] M. Roth, G. Hendeby, and F. Gustafsson, "EKF/UKF maneuvering target tracking using coordinated turn models with polar/Cartesian velocity," in *17th International Conference on Information Fusion (FUSION)*, 2014.

[29] *Alpha Prime Powering Safe Autonomy*, Velodyne Lidar, 2019, Rev. 1.