Random Matrix-based Tracking of Rectangular Extended Objects with Contour Measurements

Simon Steuernagel, Kolja Thormann, and Marcus Baum
Institute of Computer Science
University of Göttingen
Göttingen, Germany
Email: {simon.steuernagel, kolja.thormann, marcus.baum}@cs.uni-goettingen.de

***Abstract*—A widely-used approach for extended object tracking is based on random matrices, where the scattering matrix, i.e., measurement spread, is used to update a symmetric positive definite random matrix representing an elliptic extent. However, for lidar data, a mismatch between the assumed measurement model and observed data hinders the estimation quality of the method. We propose adaptions to the random matrix approach in order to facilitate the application for tracking a rectangular extended object based on contour measurements. Specifically, we derive a suitable scaling factor for the scattering matrix of measurements in this setting. Furthermore, we propose a simple yet effective estimation scheme for the target center, adapting the shape estimate accordingly. The resulting algorithm closely follows the framework of the random matrix approach. A detailed comparison with a variety of state-of-the-art trackers is carried out in a simulation based on real-world lidar parameters, confirming the effectiveness of the approach.**

***Index Terms*—Extended object tracking, random matrix, Bayesian filtering, target tracking**

I. Introduction

Tracking of objects over time is a ubiquitous task in environment perception required in a variety of fields. Depending on the application and sensor used, the challenges faced by the tracking system are manifold. In particular, for high resolution sensors, it is necessary to track not only the kinematic properties, such as position and velocity, of the target, but to furthermore estimate its physical extent, i.e., its shape. This results in the problem of Extended Object Tracking (EOT), an overview of which can be found in [1]. In an automotive setting such sensors lending themselves to EOT are, e.g, lidar [2]–[4] or automotive radar [5]–[7]. In the context of EOT, typically restrictions regarding the target shape model are assumed. These can allow for a rather wide range of potential shapes, such as general star-convex models as adopted by, e.g., [8]–[10]. However, it can be beneficial to narrow down the target shape to a particular geometry. One common choice are ellipses, due to their flexibility in modeling a variety of objects ranging from pedestrians and cars to ships. For ellipses, two types of representation are common: Some works, such as [11]–[13], model the ellipse explicitly by means of an orientation and semi-axis lengths. Alternatively, a Semi Positive Definite (SPD) matrix can be used, which can be directly embedded into a Bayesian filtering framework based on the Wishart distribution. This approach is taken in the Random Matrix (RM) model, introduced in [14], [15].

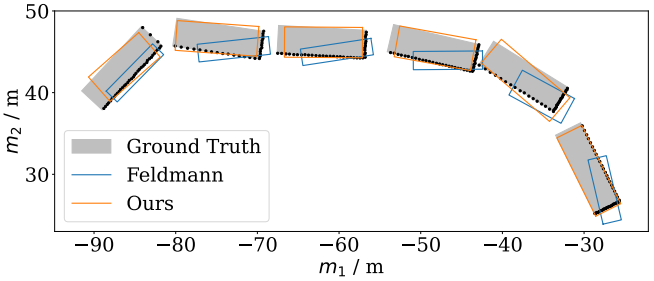


Fig. 1: Exemplary tracking results achieved by the standard RM algorithm by Feldmann et al. [15] (blue) and our proposed adaptions (orange). The contour measurements of a rectangle can not be accurately tracked by the standard reference method. Shown is a subset of time steps of the entire trajectory, and the vehicle is moving from the right to the left. The sensor is located at the origin.

Naturally, these representations can be converted into each other, even though the corresponding uncertainties can not. Furthermore, a rectangle can also be parameterized in the same fashion as an ellipse, i.e., models derived for ellipses can often be transferred into the context of tracking a rectangular target. This case arises in the automotive context of tracking, e.g., cars with high resolution lidar data. The contour of a vehicle is much closer to a rectangular shape than to an elliptic one. The measurements in this EOT setting will closely resemble contour measurements of a rectangle, which is the setting this work is concerned with. Most of the aforementioned methods are designed under the assumption that measurements can originate from the entire surface of the object. For lidar data, this usually does not hold. Reflection points are mainly distributed along the contour of the object, and are highly dependent on the sensor-to-target geometry, due to the self-occlusion effect. Therefore, many existing EOT approaches do not directly generalize well to the problem of tracking a rectangular extended object based on lidar measurements, as illustrated in Fig. 1.

A. Contribution

The contribution of this work is two-fold: Firstly, we propose modifications necessary to employ the RM filter for rectangular objects with a contour measurement model. The necessary alterations keep the main filtering framework intact, meaning that the adapted version can be directly inserted into

existing architectures building on the RM tracker. Furthermore, other extensions and improvements to the algorithm such as [16]–[18] can immediately incorporate the proposed adjustments. We derive the necessary scaling factor for the scattering matrix of measurements, and confirm its effectiveness in experiments. Moreover, we incorporate the nature of the contour measurements with target self-occlusion into the estimation process of the object center and shape by correcting the respective pseudo-measurements.

Secondly, an extensive comparison of several existing state-of-the-art trackers as well as the newly proposed adaption of the RM algorithm is carried out. To this end, realistic lidar data is simulated using a ray casting model with model statistics following a commercially available sensor, namely the Ouster OS1 [19].

B. Structure

The remainder of this paper is structured as follows: Next, in Section II, related works will be discussed. Following that, Section III will define the problem setting. In particular, the details of the simulated lidar measurement model will be given. Afterwards, the tracker will be proposed in Section IV. Section V contains the experimental results, comparing the proposed method with a variety of state-of-the-art reference algorithms. These results will then be discussed in Section VI, and the paper is concluded in Section VII.

II. Related Works

The RM approach [14], [15] makes use of the mean and scattering matrix of the measurements for estimating an SPD shape matrix, which is modeled with an inverse Wishart distribution. Thereby, the basic assumption is that the measurements stem from a Gaussian distribution modeled by the shape matrix. Measurements uniformly distributed over an elliptic surface can be modeled by means of a fixed scaling factor. An adaption of the RM algorithm to lidar data has been proposed in [4]. Combining geometric assumptions with an explicit integration of the beam-based lidar measurement model, a corrected estimate for the center of the target can be found. This is in turn used to compute a corrected scattering matrix, therefore allowing the RM approach to be employed for lidar data. The method was derived for elliptical shapes, however, it was successfully evaluated for rectangular targets as well. For automotive radar, a truncation model has been introduced to deal with parts of the object shape that are not observed [5], [6]. By making use of assumptions about the visible object parts based on the sensor-to-target geometry, pseudo-measurements can be sampled which are used to correct the output of the RM algorithm with regards to the mismatch between object shape and measurement model assumptions. A learning-based variant of this was introduced in [7]. Another possible approach to tackling the model mismatch between the RM approach and lidar data is to use a Virtual Measurement Model (VMM) [3]. Here, in parallel to the standard tracking filter, a second branch of the algorithm, the VMM, estimates a set of object parameters which, when input into the filter, would produce the observed results. This set of object parameters then forms the final output of the overall tracker. It is estimated in an iterative, optimization-based fashion. To avoid this, [20] proposed an alternative approach, based on a Gaussian Process (GP) model learned prior to the tracking process. Here, the GP parameters are trained using the VMM data, avoiding the costly online VMM computations. Whilst the authors observed slightly diminished tracking accuracy, initialization and computation time were found to improve. The VMM-based approach has also been employed for dealing with partially observed objects [21], where the case of objects shadowing each other w.r.t. the lidar is analyzed. For elliptical targets with a surface measurement model, in [11], an Extended Kalman Filter (EKF)-based tracker was derived. The distribution of the measurement sources was modeled using a multiplicative noise term. Introducing a Gaussian mixture for this noise term, in [12], a variant tailored to automotive radar was introduced, which manually models possible scattering centers across the surface and applies the update rules previously introduced in [11]. Another approach based on an EKF was proposed in [2], which focused on contour measurements as observed in lidar data. The core idea is to evaluate the measurement-to-target associations for a set of contour points with equidistant angles. Each possible association is represented by one component of a Gaussian mixture, and for each of these an EKF update is computed. Moment-matching is used to obtain a (single) Gaussian representation. Here, the orientation of the tracked object is assumed to align with its heading. Measurements are processed sequentially in an iterative fashion, as opposed to approaches such as RM, where measurements are processed in a single batch. Focusing on the challenges associated with tracking multiple extended objects rather than on the filtering accuracy for a single extended target, [22] propose a method for computing predicted measurements of both elliptical and rectangular objects.

The preceding works have focused on the (typically Bayesian) filtering task of EOT, where the properties of a target are to be estimated based on integration of measurements observed across a series of time steps. However, other works approach the problem from a single-frame perspective. Observing two sides of an object leads to a characteristic L-shape of measurements. Fitting such L-shapes, as done in, e.g., [23], [24], is one way to tackle the problem of determining a suitable rectangular shape given a set of single-target measurements. A search-based algorithm is used in [24] to minimize a suitable error criterion, such as the squared distance of measurement points to estimated box edges. In [23], the convex hull of the measurement is used as input to an algorithm based on rotating calipers. Afterwards, an EKF can be used to track the resulting target state across frames. The convex hull is also used as a starting point for a box fit in [25]. Using a heuristic approach based on the two-dimensional median of the measurements, the orientation of the resulting box is corrected. Order statistics are employed in [26] to estimate the centroid of a bounding box from lidar data. Motivated by the fact that

measurements concentrate around the target corner closest to the sensor, triangular likelihoods are used in the derivation of the method.

III. Problem Setting

We consider the tracking of a rectangular extended object using (simulated) lidar measurements in 2D space. In particular, tracking is to be carried out as a filtering task, i.e., at every time step measurements observed up to and including that time step are to be used in determining an estimate for the target state. The state at time k is defined as

$$\mathbf{x}_k = \begin{bmatrix} m_{1,k} & m_{2,k} & \dot{m}_{1,k} & \dot{m}_{2,k} & \theta_k & l_{1,k} & l_{2,k} \end{bmatrix}^{\mathrm{T}} \ , \quad (1)$$

where $m_{1,k}$ and $m_{2,k}$ represent the position of the target along the x- and y-axis, respectively, $\dot{m}_{1,k}$ and $\dot{m}_{2,k}$ the velocity, θ_k the orientation, and $l_{1,k}$ and $l_{2,k}$ the semi-axis lengths of the rectangle. We assume the target is moving with constant velocity, corrupted by zero-mean white noise. For the discrete time steps, we model this noise as a Gaussian with covariance $\mathbf{Q}$.

Measurements are assumed to be generated using a ray casting-based lidar model. In order for the simulation to be close to real-world conditions, we use statistics chosen according to the specification of an Ouster OS1 lidar scanner [19]. Throughout the work, we assume a sensor-centered coordinate system, with the sensor beginning it's revolution at the x-axis and returning measurements ordered by rotation. Every time step represents a single revolution of the sensor, and all values are chosen accordingly to match the scanning rate of 10 Hz of the OS1. The maximum range of the sensor is set to $100\,\mathrm{m}$, i.e., measurements beyond this threshold are dropped. In every time step, the sensor performs one full rotation, scanning a fix number of evenly spaced beams. If a beam intersects with the object, a measurement source $\mathbf{y}_k^i$ in polar coordinates is reported at the closest intersection point to the sensor. The number of beams per scan is fixed as n_{beams}. In the experiments described in Section V, different values for n_{beams} will be evaluated, following the specification of the OS1. The resulting set of measurement sources on the target contour generated at time k is denoted as $\mathbf{Y}_k$, with $|\mathbf{Y}_k| = N_k$. Based on these, the noise-corrupted reported measurements $\mathbf{Z}_k$ are computed following

$$\mathbf{z}_k^i = \mathbf{y}_k^i + \mathbf{v}_k^i, \quad i \in \{1, \ldots, N_k\} \ , \quad (2)$$

with $\mathbf{v}_k^i \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$. The measurement noise covariance matrix $\mathbf{R}$ is a diagonal matrix containing the variance of the range error and of the angular error. For the former, a standard deviation of $0.1\,\mathrm{m}$ was chosen. The standard deviation of the angular error was set to $0.01°$. Note that the fact that we model the sensor to be rotating means that generated measurements will follow a pattern of strictly monotonically increasing angles.

No prior information about the true target state $\mathbf{x}_0$ is known to the object tracker, i.e., the first set of measurements needs to be used for initialization of the tracker state.

IV. Proposed Method

In the following, the proposed algorithmic adjustments for the RM method will be presented. To this end, the necessary parts of the RM approach will be reviewed first. Afterwards, the required scaling factor for rectangle contour measurements will be derived, and the adapted pseudo-measurement computation will be proposed.

A. Random Matrix Method Review

For detailed explanation of the random matrix approach we refer to [1], [15]. In the filter, the kinematic and shape components of the state are treated separately. The former is modeled as a Gaussian random variable, and the latter based on an inverse Wishart distribution. To acquire an estimate of the shape state components $\begin{bmatrix} \theta_k & l_{1,k} & l_{1,k} \end{bmatrix}^{\mathrm{T}}$ from the matrix-valued estimate and degrees of freedom, e.g., Eigenvalue Decomposition (EVD) can be used. In the RM method, these three parameters represent an ellipse. Since rectangles and ellipses can be described by the same parameters, we can directly use these to represent the desired rectangular state[1]. Our implementation follows the standard approach originally proposed by [15], but we note that other adaptions or improvements to the method could also be employed (e.g., applying a rotation on the shape during the time update in [16], [17] or using a distorted shape during measurement update [18]), as our correction for the new measurement model only adapts very specific parts of the filter. For the measurement update, two pseudo-measurements are employed to correct the kinematic and shape estimates. The first is the mean of the measurements

$$\bar{\mathbf{z}}_k = \frac{1}{|\mathbf{Z}_k|} \sum_{\mathbf{z} \in Z_k} \mathbf{z} \ , \quad (3)$$

based on which the estimate of the center of the tracked object is updated, following the assumption that the center of mass of the measurements corresponds to the object center. Furthermore, the scattering matrix of the measurements, computed as

$$\bar{\mathbf{Z}}_k = \sum_{\mathbf{z} \in \mathbf{Z}_k} (\mathbf{z} - \bar{\mathbf{z}}_k)(\mathbf{z} - \bar{\mathbf{z}}_k)^{\mathrm{T}} \ , \quad (4)$$

is used as a pseudo-measurement for the shape update.

In [4], changes for these pseudo-measurements in order to adapt the filter to lidar data are proposed. Replacing (3), the center of the target is estimated using a geometric approach. A least squares-based line fit through the target center is computed first. This is then intersected with lines orthogonal to the tangents of the ellipse at the outermost measurements, in order to compute the center estimate. The scattering matrix is computed based on the estimated object mean rather than the measurement mean, as also discussed below in this work.

[1]Note that for the corner case of a circle, i.e., $l_1 = l_2$, this conversion is incorrect, as for the circle θ has no impact on the geometric shape, whereas for a square, it does.

B. Scaling Factor

Computing the sample covariance as in (4) results in a scattering matrix that corresponds to Gaussian distributed measurement sources. As this is not the case for extended objects, which have a finite spatial extent, a scaling factor c is employed. If, e.g., the target were elliptical and measurements were uniformly distributed across the entire surface, this scaling factor would be $\frac{1}{4}$ [15], [27]. For the target application of this work, a scaling factor for contour measurements of rectangular shapes needs to be derived.

In the following, the factor will be derived under the approximation that measurements are observed uniformly across the object contour of a symmetric, i.e., square, object with semi-axis length l, centered on the origin. The measurement source distributions of the four individual sides will be

$$f_{\text{upper}}(x,y) = f_1(x,y) = \delta(y-l)\cdot\mathcal{U}(x;-l,l) \ , \tag{5a}$$
$$f_{\text{lower}}(x,y) = f_2(x,y) = \delta(y+l)\cdot\mathcal{U}(x;-l,l) \ , \tag{5b}$$
$$f_{\text{left}}(x,y) = f_3(x,y) = \delta(x+l)\cdot\mathcal{U}(y;-l,l) \ , \tag{5c}$$
$$f_{\text{right}}(x,y) = f_4(x,y) = \delta(x-l)\cdot\mathcal{U}(y;-l,l) \ , \tag{5d}$$

where δ denotes the Dirac delta function. Due to the symmetry of the measurement sources on the target contour, we will consider only the x-axis, and marginalize out the y components. This yields

$$f_1(x) = f_2(x) = \mathcal{U}(x;-l,l) \ , \tag{6a}$$
$$f_3(x) = \delta(x+l) \ , \tag{6b}$$
$$f_4(x) = \delta(x-l) \ . \tag{6c}$$

We are interested in the variance σ^2 of the overall mixture distribution. As measurements are assumed to stem uniformly from each of the four sides, the mixture weights are uniform, i.e., $\omega_i = \frac{1}{4}$ for $i \in \{1,\dots,4\}$. The desired variance can be computed from the component weights, means, and variances as

$$\sigma^2 = \left(\sum_{i=1}^{4}\omega_i(\sigma_i^2+\mu_i^2)\right) - \mu^2 \ , \tag{7}$$

where the mixture mean μ can be computed as

$$\mu = \sum_{i=1}^{4}\omega_i\cdot\mu_i = \frac{1}{4}\cdot 0 + \frac{1}{4}\cdot 0 + \frac{1}{4}\cdot -l + \frac{1}{4}\cdot l = 0 \ . \tag{8}$$

The mean of the first two components, which are uniformly distributed, is $\mu_1 = \mu_2 = \frac{1}{2}\cdot(-l+l) = 0$. Their variance can be computed as

$$\sigma_1^2 = \sigma_2^2 = \frac{(l-(-l))^2}{12} = \frac{l^2}{3} \ . \tag{9}$$

For the third and fourth component, the variance will be 0. Their means $\mu_{3/4}$ will be $-l$ and l, respectively. From this, σ^2 can be computed as

$$\begin{aligned}\sigma^2 &= \frac{1}{4}\cdot\frac{l^2}{3} + \frac{1}{4}\cdot\frac{l^2}{3} + \frac{(-l)^2}{4} + \frac{l^2}{4} - 0 \\ &= \frac{l^2}{12} + \frac{l^2}{12} + \frac{3\cdot l^2}{12} + \frac{3\cdot l^2}{12} = \frac{2}{3}\cdot l^2 \ .\end{aligned} \tag{10}$$

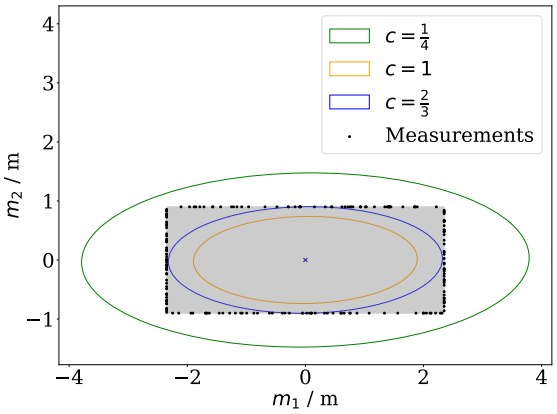


Fig. 2: Visualization of the sample covariance matrix of 200 noise-free measurements (black dots) originating from the contour of a rectangular object, divided by different scaling factors c. Results only match the semi-axis of the true object (filled gray) for the derived scaling factor of $\frac{2}{3}$.

Hence, to acquire the squared semi-axis length, corresponding to the shape matrix diagonal entries in the rotation-free multivariate case, the variance (or covariance in the multivariate case) needs to be divided by the scaling factor $\frac{2}{3}$.

Results for visualization of the scaled sample covariance matrix for different factors are shown in Fig. 2. Here, 200 measurements were drawn uniformly from the four sides of a rectangular target, with axis lengths of $4.7\,\text{m}$ and $1.8\,\text{m}$, respectively. The results match the true rectangle in axis lengths if and only if the previously derived scaling factor of $\frac{2}{3}$ is used.

C. Further Discussion

The presented derivation is based on assumptions which may not hold in practice. For different semi-axis lengths, rather than a square object, the derivation requires the distribution of measurements to be uniform across the object sides, regardless of their length. This can be seen in Fig. 2, where the density of measurements along the longer sides of the object is lower. In practice, a more intuitive assumption would be that measurements are distributed uniformly across the entire contour, resulting in more measurements from longer sides. In that case, the weights ω will not be uniform. Furthermore, the resulting scaling factor will not be a single scalar, but instead will vary depending on the semi-axis. Nevertheless, the very same derivation can be employed in that case. However, knowledge of the ratio of the axis lengths of the objects is required in order to compute ω. Incorporating the resulting set of scaling factors is not as straightforward as in the symmetric case. Instead, the individual semi-axis need to be scaled. Without knowledge of the object rotation, this can be facilitated by performing an EVD of the scattering matrix, followed by reconstruction of the matrix after applying the scaling factors to the corresponding semi-axis acquired from the eigenvalues. Following these steps, and given knowledge of the ratio between semi-axis lengths, the same derivation

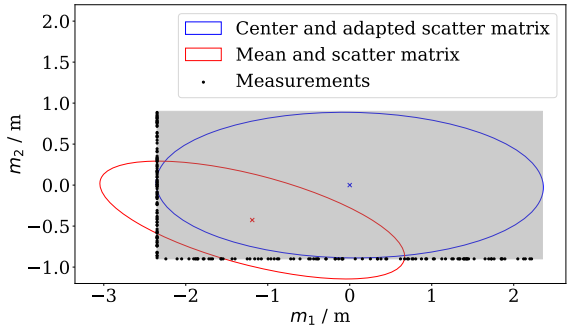


Fig. 3: Computation of the (scaled and normalized) scattering matrix based on the measurement mean (red ellipse) or the target center (blue ellipse). The 200 noise-free measurement (black dots) are drawn uniformly across the left and lower sides of the object.

as taken above can be employed to acquire a correctly scaled scattering matrix which can be employed for the RM tracker. Still, for lidar data, the entire assumption of a uniform measurement distribution only holds for highly specific sensor-to-target geometries. Hence, in the following, we choose to employ the straightforward approximation resulting in (10).

D. Adapted Pseudo-measurements

For measurements that stem from a lidar scanner, the assumption that the mean of the measurements coincides with the object center does not hold. Hence, a standard RM tracker using (3) and (4) will only filter the measurement mean and (scaled) measurement covariance, which do not correspond to the object location and extent.

As also done in, e.g., [4], given an estimate $\hat{\mathbf{m}}_k$ of the object center, it is possible to compute the scattering matrix centered on $\hat{\mathbf{m}}_k$ instead of $\bar{\mathbf{z}}_k$, i.e., as

$$\bar{\mathbf{S}}_k = \sum_{\mathbf{z} \in \mathbf{Z}_k} (\mathbf{z} - \hat{\mathbf{m}}_k)(\mathbf{z} - \hat{\mathbf{m}}_k)^{\mathrm{T}} \quad . \tag{11}$$

Naturally, for measurement models that assume $\bar{\mathbf{z}}_k = \hat{\mathbf{m}}_k$, there will be no difference between the two. Examples for these models are surface measurements of an (e.g., elliptical) surface, or contour measurements of a target with a shape that is point-symmetric around its center. For lidar data, however, the assumption generally does not hold, since due to the self-occlusion effect only parts of the object contour will be observed by the scanner.

This is visualized in Fig. 3, where only the left and lower sides of the contour are modeled as visible. Computing the measurement mean and (scaled) measurement sample covariance, as shown in red, does not yield an estimate for the target extent. By making use of (11) in combination with the true target center, the results visualized in blue are obtained. Clearly, the semi-axis of the target are estimated correctly.

Therefore, it is necessary to compute an accurate estimate $\hat{\mathbf{m}}$ for the center, taking the specifics of the measurement model into account. We make use of the fact that the measurements $\mathbf{Z}_k$ are ordered according to the rotation of the lidar. The center of the target can only be estimated from a single set of measurements if two sides were observed. In that case, the first and last observed measurements will stem from the measurement sources closest to the corners of the rectangle. Naturally, the mean of two opposite corners of a rectangle coincides with the rectangle center. Using $\mathbf{z}_k^1$ and $\mathbf{z}_k^{N_k}$ as an approximation for these corners, we obtain $\hat{\mathbf{m}}_k = 0.5 \cdot (\mathbf{z}_k^1 + \mathbf{z}_k^{N_k})$. Note that the uncertainty of $\hat{\mathbf{m}}_k$ is affected not only by the measurement noise of the two involved measurements, but also by the horizontal sensor resolution in combination with the unknown sensor-to-target geometry, affecting the measurement source positions on the contour and hence the accuracy of the approximation. As an approximation, we directly use $\hat{\mathbf{m}}$ as part of the RM algorithm, and do not incorporate such factors for now. If the object intersects with the beam at the start of the revolution of the scanner, i.e. at $0°$, this assumption does not hold, as in that case the first and last measurements may be anywhere on the object contour and will be close to each other. As in our sensor system, this beam is along the x-axis of the coordinate system, this case can be detected by checking for a measurement with a positive position along the x-axis and a sufficiently small (absolute) y value. We used $0.05\,\mathrm{m}$ as a threshold. If such a measurement exists, instead of $\mathbf{z}_k^1$ and $\mathbf{z}_k^{N_k}$, we use the two measurements with the largest angular distance between them in the computation of $\hat{\mathbf{m}}_k$. The same approach may also be taken if the assumption that measurements are received in an ordered manner is violated.

For a noise-free lidar with infinite horizontal resolution, the presented approximation would be accurate. In practice, we find the accuracy and resolution of the simulated Ouster OS1 to be sufficiently high in order to make use of the approximation, as validated below in Section V.

V. Experimental Analysis

The effectiveness of the proposed approach has been evaluated in an experimental evaluation, following the description in Section III. In the following, the reference methods and the details of the experimental setup will be introduced, and afterwards, results of the evaluation will be presented.

A. Compared Methods

The results of the adapted filter are compared with a set of reference algorithms. First of all, we use the standard RM algorithm [15], abbreviated as “Feldmann”. This algorithm serves as the baseline for an EOT system. Furthermore, we include two versions of the adaption of the RM approach proposed in [4]. The first (“Li”) is the version following the formulas of [4]. The second one (“LiImproved”) makes use of the same RM predict and update formulas as our proposed algorithm, but uses the pseudo-measurements of [4]. Not only does this improve the comparability of the methods, we additionally found results to improve. Both of these variants did not make use of the measurement weights proposed in [4] and used uniform weights instead, as this parameterization improved results. Next, the truncation based approach presented in [7] is included. In the experimental figures, it is referred to

as "TruncatedRM". Notably, for a fair comparison between methods, we make use of the previously derived scaling factor of $\frac{2}{3}$ for all of the RM based methods. As visible in Fig. 2, using an incorrect scaling factor for individual methods would artificially reduce their estimation accuracy. Finally, two EKF-based approaches are included. The first is the method of [2] ("AlqaderiEKF"), and the second the Multiplicative Error Model Extended Kalman Filter* (MEM-EKF*) variant using Gaussian mixtures proposed in [12] ("GM-MEM-EKF*"). The discussed VMM-based tracking algorithms are optimization-based, such as [3], or rely on learned models [20], and were hence not included in the evaluation.

All methods' parameters were tuned as good as possible for the investigated scenario. For brevity, we do not include the full details of all parameters here, and instead refer to our publicly available code[2]. Where methods were similar, parameters were chosen similarly as well. The measurement noise covariance $\mathbf{R}$ is in polar space, corresponding to the received range/angle measurements, but all evaluated filters work in Cartesian space. Therefore, throughout all experiments, all algorithms approximate the Cartesian measurement noise covariance at time k as $\hat{\mathbf{R}}_k$. To this end, we employed the Unscented Transform (UT) [28], which approximates the covariance after the nonlinear conversion from polar to Cartesian space via a set of sigma points. As the mean of the original distribution, the mean of all measurements is used.

B. Experimental Setup

For the evaluation, we simulated a set of random trajectories using the measurement generation process detailed in Section III. As described above, two settings were evaluated. In the first, measurements are sparse, with $n_{\text{beams}} = 512$ beams per lidar revolution. In the second setting, this number is increased to $n_{\text{beams}} = 1024$. The initial state of the tracked vehicle was uniformly sampled from

$$\mathbf{x}_0 \sim \mathcal{U}\left(\begin{bmatrix}-80\\-80\\0.5\\-0.3\\0\\2.5\\1.4\end{bmatrix}, \begin{bmatrix}-30\\-30\\0.7\\0.3\\2\pi\\6\\2.4\end{bmatrix}\right) . \tag{12}$$

The maximum trajectory length was set to 50. If a vehicle left the sensor range, the trajectory was stopped early. Due to the random sampling involved in the process, it is possible for the vehicle to cross the origin, i.e., the sensor location. In these rare cases, the trajectory was dropped and not counted. Objects move according to a constant velocity model with additive zero-mean Gaussian white noise. Object movement was not constrained in any way, hence it is not guaranteed that an L-shape of each object was visible in every time step. Instead, in some time steps, only a single side of the object may be visible.

[2]https://github.com/Fusion-Goettingen/

TABLE I: Runtime of a single predict/update cycle of the evaluated methods in ms, averaged over all steps of 25 Monte Carlo runs.

Name	Mean runtime / ms
Feldmann [15]	0.58
Ours	0.63
TruncatedRM [7]	4.40
LiImproved [4], [15]	10.72
Li [4]	10.93
GM-MEM-EKF* [12]	26.76
AlqaderiEKF [2]	212.41

The quality measure which was utilized for evaluation was the Intersection over Union (IoU). For two objects s_1, s_2, it is defined as

$$\text{IoU} = \frac{\text{area}(s_1 \cap s_2)}{\text{area}(s_1 \cup s_2)} . \tag{13}$$

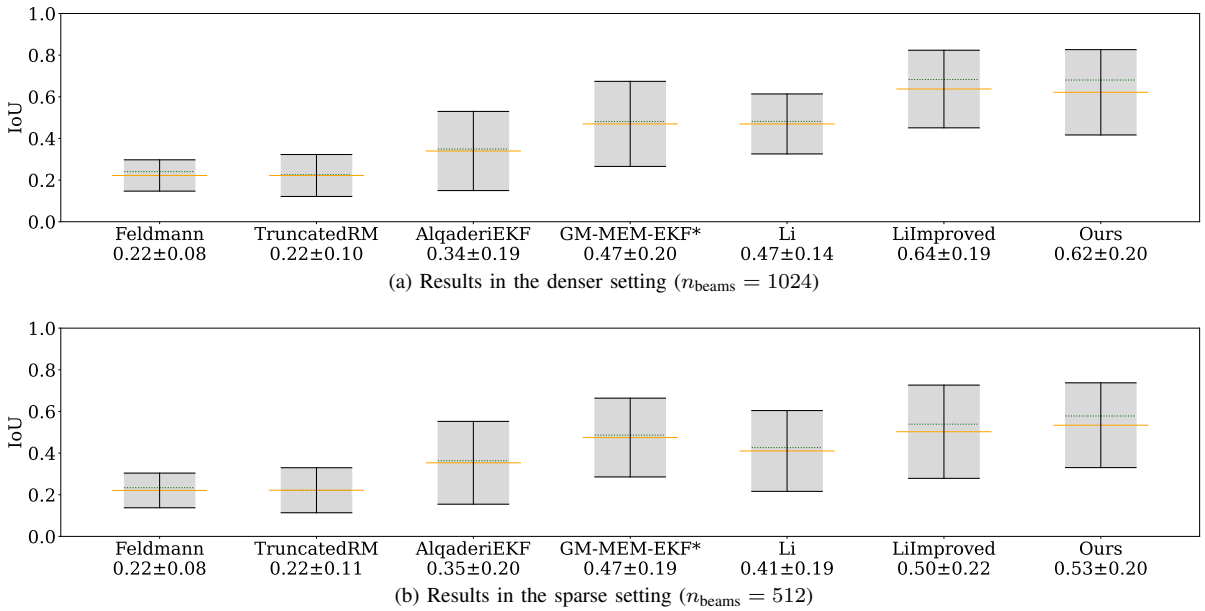
It results in a value in $[0, 1]$, with 0 meaning that the two objects do not overlap, and 1 meaning that the shapes perfectly overlap. Hence, higher IoU indicate a better result. For the evaluation, we compared the rectangle representation of the trackers based on their semi-axis lengths and orientation estimates with the ground truth rectangle representing the tracked object.

C. Results

Results of the previously discussed quantitative analysis are presented Table I and Fig. 4, where evaluation results for the two settings of n_{beams} are visualized in separate figures. The truncated approach [7], originally designed for automotive radar measurements, was adopted to the data for the given scenario. However, its performance is comparable with the baseline RM method [15].

For the investigated setting, the method proposed by Li et al. [4] performed noticeably better if the RM formulas according to [15] were used. In that case, it exhibited strong performance, slightly outperforming our proposed tracker for $n_{\text{beams}} = 1024$. However, for sparse data, the performance of the method of [4] is reduced by a larger margin than our method, accompanied by an increase in variance. Overall, the estimation quality of our version and [4] is highly comparable, even though our method is noticeably less complex. This furthermore requires using our derived scaling factor and RM formulas according to [15], whereas the original method exhibits worse tracking accuracy. Generally, the sparse setting introduces large outliers for most evaluated methods. Both EKF-based approaches suffer from a large spread of results already in the denser setting, but their estimation quality is not reduced further in the sparse setting. The GM-MEM-EKF* uses explicit assumptions about the sensor-to-target geometry to estimate observed sides of the contour. If these assumptions do not match the true distribution, the estimate can diverge. Similar behavior is exhibited by the EKF of [2].

Furthermore, an evaluation of the computational cost of each of the algorithms was conducted by means of an empirical runtime analysis, averaged over all time steps of 25 Monte Carlo runs. Results of this are given in Table I. All methods were implemented in python, and results were generated



(a) Results in the denser setting ($n_{\text{beams}} = 1024$)

(b) Results in the sparse setting ($n_{\text{beams}} = 512$)

Fig. 4: Results of all filters across 200 Monte Carlo runs. The mean of all methods is indicated by the solid orange line, the median by the dotted green line. The standard deviation around the mean is indicated by the gray box. Below the method's names, the mean and standard deviation are given. Each individual sub-figure covers a different horizontal resolution of the simulated lidar.

on an Intel i5-12600K. The iterative nature in combination with the need to evaluate each mixture component for each measurement means that the "AlqaderiEKF" method is vastly slower compared to all other algorithms. Similarly, the other EKF-based approach of [12] is the second slowest method. Of the RM based filters, our proposed one is the only one that does not add noticeable computational load compared to the baseline RM [15]. In particular, the method of Li et al. [4], which performed similarly with respect to the IoU, is over an order of magnitude slower than our proposed method. The original RM approach has a time complexity of $O(n)$, which is matched by our approach if measurements sorted by scanning order can be used, and otherwise is increased to $O(n \log n)$ due to the preliminary sorting needed for the computation of the center pseudo-measurement.

VI. Discussion

The results presented in Section V validate the effectiveness of our proposed approach in practice. Except for the method presented in [4], the evaluated reference methods could not be applied entirely successfully to the problem of tracking a rectangular extended object based on contour measurements. The method of Li et al. [4], when using the newly derived scaling factor and filter formulas following [15], matched the performance of our proposed method. For dense settings, it slightly outperformed our method. On the other hand, in sparse settings, our method outperformed the reference method. Even though outliers are present in both methods, the standard deviation of our approach did not increase between the settings, whereas the standard deviation of [4] increased. Noticeably, these similar results are achieved with a much less complex approach. Additionally, we found that in our setting the geometric approach to the center estimation problem by [4] can occasionally introduce strong errors, placing the estimated center far away from the vehicle extent. Specific sensor-to-target geometries introduce distributions of measurements concentrated on one edge of the vehicle, while a second edge is barely visible but detected. In some of these cases, the least-squares line fit that is supposed to pass through the object center is instead tilted towards the strongly observed edge, resulting in the large estimation errors. Potentially, this may be due to the fact that the method was originally derived for elliptical objects.

Moreover, the evaluation of the computational cost of the algorithms has shown that, in our setting, all existing state-of-the-art methods are at least an order of magnitude slower compared to the standard RM approach. Our method on the other hand barely increases the computational load of the tracker. Nevertheless, an overall problem that is exhibited by all methods, including ours, is that challenging situations cause the IoU to noticeably decrease.

One issue that is challenging to handle based on the RM approach is to manage the update for objects that are only visible from a single side. While information about one semi-axis, and to a lesser extent about the target orientation, is available, no information about the unobserved semi-axis is available. In the RM framework, the problem could be tackled using simply heuristics, utilizing information from previous time steps to adapt the pseudo-measurements. One possibility is to shift the center estimate according to the unobserved semi-axis length and to overwrite the collapsed semi-axis with the prior estimate. However, this may lead to overconfidence of the corresponding uncertainties, due to "double-counting" of prior information during the update.

Another potential issue is that the update step can not be carried our if less than three measurements are available, since then the pseudo-measurements can not be computed. However, for lidar data as considered in this work, this case typically only arises for objects at the edge of the sensor range.

VII. Conclusion

In this work, we proposed modifications of the RM algorithm for tracking a rectangular object with contour measurements. To this end, we first derived a suitable scaling factor based on moment-matching of rectangular contour measurements with a Gaussian. Then, we incorporated a simple center estimation scheme into the scattering matrix computation in order to obtain the required pseudo-measurements.

An extensive simulation based on a raycasting model with real-world lidar parameters was carried out. In situations with sparse measurements, our method was found to outperform all reference methods. In denser settings, only a single reference method yielded comparable results, and only if the newly derived scaling factor was used. Furthermore, the computational overhead of our algorithm compared to the original RM method is negligible, whereas all reference methods were found to be at least an order of magnitude slower. Since the method follows the standard RM formulations, any adaptions or downstream processing, such as a multiple object tracking system, developed for the RM algorithm could directly adopt the proposed changes.

For future work, a prime research direction is the extension of the method to three-dimensional space, since modern lidar sensors often cover more than a single vertical layer. Furthermore, the presented evaluation has not included clutter measurements. Evaluating the impact that clutter has on the object center estimation, and potentially improving its robustness, can be a further relevant research direction.

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] H. Alqaderi, F. Govaers, and R. Schulz, "Spacial elliptical model for extended target tracking using laser measurements," in *2019 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, Oct. 2019, pp. 1–6.

[3] P. Hoher, S. Wirtensohn, T. Baur, J. Reuter, F. Govaers, and W. Koch, "Extended target tracking with a lidar sensor using random matrices and a virtual measurement model," *IEEE Transactions on Signal Processing*, vol. 70, pp. 228–239, 2022.

[4] P. Li, C. Chen, C.-z. You, and J.-d. Qiu, "Modified extended object tracker for 2D lidar data using random matrix model," *Scientific Reports*, vol. 13, no. 1, p. 5095, Mar. 2023.

[5] Y. Xia, P. Wang, K. Berntorp, T. Koike-Akino, H. Mansour, M. Pajovic, P. Boufounos, and P. V. Orlik, "Extended object tracking using hierarchical truncation measurement model with automotive radar," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2020, pp. 4900–4904.

[6] Y. Xia, P. Wang, K. Berntorp, H. Mansour, P. Boufounos, and P. V. Orlik, "Extended object tracking using hierarchical truncation model with partial-view measurements," in *2020 IEEE 11th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, Jun. 2020, pp. 1–5.

[7] Y. Xia, P. Wang, K. Berntorp, L. Svensson, K. Granström, H. Mansour, P. Boufounos, and P. V. Orlik, "Learning-based extended object tracking using hierarchical truncation measurement model with automotive radar," *IEEE Journal of Selected Topics in Signal Processing*, vol. 15, no. 4, pp. 1013–1029, Jun. 2021.

[8] M. Baum and U. D. Hanebeck, "Extended object tracking with random hypersurface models," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 50, no. 1, pp. 149–159, Jan. 2014.

[9] N. Wahlström and E. Özkan, "Extended target tracking using Gaussian processes," *IEEE Transactions on Signal Processing*, vol. 63, no. 16, pp. 4165–4178, Aug. 2015.

[10] H. Alqaderi, F. Govaers, and W. Koch, "Symmetric star-convex shape tracking with Wishart filter," in *2021 IEEE 24th International Conference on Information Fusion (FUSION)*, Nov. 2021, pp. 1–8.

[11] 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, Sep. 2019.

[12] K. Thormann, S. Yang, and M. Baum, "Kalman filter based extended object tracking with a Gaussian mixture spatial distribution model," in *2021 IEEE Intelligent Vehicles Symposium Workshops (IV Workshops)*, Jul. 2021, pp. 293–298.

[13] M. Tesori, G. Battistelli, L. Chisci, and A. Farina, "L:OMEM - a fast filter to track maneuvering extended objects," in *2023 26th International Conference on Information Fusion (FUSION)*, Jun. 2023, pp. 1–8.

[14] W. Koch, "Bayesian approach to extended object and cluster tracking using random matrices," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, pp. 1042–1059, Jul. 2008.

[15] M. Feldmann, D. Fränken, 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] K. Granström and U. Orguner, "New prediction for extended targets with random matrices," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 50, no. 2, pp. 1577–1589, Apr. 2014.

[17] N. J. Bartlett and A. G. Wills, "A robust random matrix prediction model for extended object rotations," in *2021 IEEE 24th International Conference on Information Fusion (FUSION)*, Nov. 2021, pp. 1–8.

[18] J. Lan and X. R. Li, "Tracking of extended object or target group using random matrix: new model and approach," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, no. 6, pp. 2973–2989, Dec. 2016.

[19] Ouster, "Ouster OS1 specification," https://web.archive.org/web/20240221122133/https://data.ouster.io/downloads/datasheets/datasheet-rev7-v3p0-os1.pdf, (accessed Feb. 27, 2024).

[20] P. Hoher, J. Reuter, D. Dold, D. Griesser, F. Govaers, and W. Koch, "Extended target tracking with a lidar sensor using random matrices and a Gaussian processes regression model," in *2023 26th International Conference on Information Fusion (FUSION)*, 2023, pp. 1–8.

[21] P. Hoher, J. Reuter, F. Govaers, and W. Koch, "Tracking of partially visible elliptical objects with a lidar sensor using random matrices and a virtual measurement model," in *2022 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*. Bonn, Germany: IEEE, Oct. 2022, pp. 1–6.

[22] K. Granström, C. Lundquist, and U. Orguner, "Tracking rectangular and elliptical extended targets using laser measurements," in *14th International Conference on Information Fusion*, Jul. 2011, pp. 1–8.

[23] P. Kmiotek and Y. Ruichek, "Representing and tracking of dynamics objects using oriented bounding box and extended Kalman filter," in *2008 11th International IEEE Conference on Intelligent Transportation Systems*, Oct. 2008, pp. 322–328.

[24] X. Zhang, W. Xu, C. Dong, and J. M. Dolan, "Efficient L-shape fitting for vehicle detection using laser scanners," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2017, pp. 54–59.

[25] B. Naujoks and H.-J. Wuensche, "An orientation corrected bounding box fit based on the convex hull under real time constraints," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2018, pp. 1–6.

[26] K. Yang, Y. Bar-Shalom, P. Willett, and S. Hunt, "Order statistic estimation with application to tracking in autonomous driving," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 59, no. 4, pp. 3531–3538, Aug. 2023.

[27] M. Feldmann and W. Koch, "Comments on "Bayesian approach to extended object and cluster tracking using random matrices"," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 2, pp. 1687–1693, Apr. 2012.

[28] S. Julier, "The scaled unscented transformation," in *Proceedings of the 2002 American Control Conference (IEEE Cat. No.CH37301)*, vol. 6, May 2002, pp. 4555–4559 vol.6.