On Runtime Reduction in 3D Extended Object Tracking by Measurement Downsampling

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

Uwe D. Hanebeck
Intelligent Sensor-Actuator-Systems Laboratory (ISAS)
Karlsruhe Institute of Technology (KIT)
Karlsruhe, Germany
uwe.hanebeck@kit.edu

***Abstract*—In 3D extended object tracking (EOT), well-established models exist for tracking the object extent using various shape priors. A single update, however, has to be performed for every measurement using these models leading to a high computational runtime for high-resolution sensors. In this paper, we address this problem by using various model-independent downsampling schemes based on distance heuristics and random sampling as pre-processing before the update. We investigate the methods in a simulated and real-world tracking scenario using two different measurement models with measurements gathered from a LiDAR sensor. We found that there is a huge potential for speeding up 3D EOT by dropping up to 95% of the measurements in our investigated scenarios when using random sampling. Since random sampling, however, can also result in a subset that does not represent the total set very well, leading to a poor tracking performance, there is still a high demand for further research.**

***Index Terms*—3D Extended Object Tracking, Elliptic Cylinder, Shape Tracking, Downsampling, Runtime Reduction.**

I. Introduction

In modern surveillance systems, high-resolution sensors such as LiDAR play an important role in localization and mapping, but also in tracking dynamic objects. The huge amount of measurements per object and time step captured by these sensors enables the size and shape of dynamic objects to be estimated simultaneously with their kinematic state. These algorithms can be summarized under the field of extended object tracking (EOT). An introduction for single EOT can be found in [1]. Extensions for tracking multiple extended objects are covered in [2].

In EOT, several fundamental approaches to derive a measurement model exist. An early and still popular approach is random matrices (RM) [3]. Because of the closed-form propagation of the extent distribution in RM, only a single update has to be performed for the whole measurement set in each time step. Thus, RM are very efficient with regard to computational runtime. The closed-form propagation, however, leads to the disadvantage that the approach is not flexible with respect to different shapes. A more flexible approach for estimating elliptical extents regarding various spatial measurement distributions is provided by [4]. A higher flexibility in prior shape assumptions can be achieved when using a random hypersurface model (RHM) [5]. Popular RHMs in 2D space employ Fourier series [6] or Gaussian processes [7] for expanding the shape as radial function.

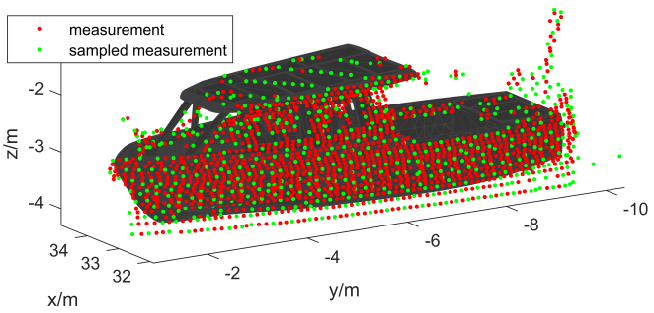


Fig. 1: Reduced measurement set. Reference as black surface.

The concept of RHMs was extended to extrusion RHMs in [8], [9] by a reformulation of the scaling parameter to model an extrusion in 3D space. These extrusion RHMs are, however, bound to fixed shape priors leading to poor tracking results when targets do not match these priors. Therefore, shape tracking procedures have been developed to overcome this problem by estimating the shape during the tracking process, which allows to model the measurement sources more accurately than when using fixed shape priors. First shape tracking approaches in 3D space employ radial functions in spherical coordinates expanding the shape functions using spherical harmonics [10], Gaussian processes [11], or spherical double Fourier series [12]. A representation in spherical coordinates, however, has the disadvantage that measurements from the top and bottom part of the target are needed for a viable shape estimate. Since these measurements are absent in a lot of scenarios, radial functions in cylindrical coordinates as shape representation were developed in [13], [14] to overcome this problem.

In summary, measurement models in 3D space are well-established and investigated in a lot of scenarios. An open and still unresolved problem, however, remains the fact that these measurement models require an update for each individual measurement, while the measurement sets of a time step can potentially contain thousands of measurements per object. For high-resolution sensors, this leads to an extensive computational burden, especially when multiple extended objects are to be tracked.

In this paper, we claim that a reduced measurement set of an object can achieve a similar tracking performance compared

to the full measurement set due to a high redundancy of the information encoded in the measurements. Furthermore, we show that the performance can even be increased by dropping measurements not well represented by the measurement model. According to that, there is still a high potential for speeding up and enhancing the performance of 3D EOT. In Fig. 1, an example for such a reduced measurement set from real LiDAR data in a maritime scenario is shown. It can be seen, that the reduced measurement set still covers the surface of the target very well what justifies our statement.

We investigate three downsampling schemes in a simulated and a real-world scenario. We use basic heuristic-based model-independent downsampling schemes that are well-known in the literature to get an intuition of the tracking performance when reducing the measurement set. The contributions of the paper can be summarized as follows:

- We investigate the farthest point sampling (FPS) [15] and voxel grid sampling (VGS) [16] schemes as pre-processing step.
- We use a random sampling (RS) as reference sampling approach.
- We implement two measurement models. An elliptic cylinder, that has not been shown before in the literature, and the shape tracking procedure from [14].
- Finally, we show results from a simulated and a real-world scenario.

Our results in Sec. V show that there remains a high potential to improve 3D EOT. To fasten and optimize the methods, however, further research has to be conducted.

II. State of the Art

In this section, we aim to review the usage of heuristic-based downsampling schemes such as the FPS or the VGS, and to present recent research activities on the topic of point cloud reduction. The algorithms itself will be presented in Sec. III.

The FPS strategy [15] is a deterministic and efficient approach of sampling a predefined number of points from the full set. Its main advantage is that it retains a uniformity criterion by iteratively sampling the points that are farthest away. Since the method is intuitive and easy to implement, it has been used in various point cloud detection and classification networks in a pre-processing step. One of the first networks was PointNet++ [17], using the FPS strategy to downsample partitions of the full point cloud to increase efficiency. The same strategy was used in the PointRCNN [18] and PV-RCNN [19] networks. In the 3DSSD network [20], a feature distance is added to the FPS strategy to preserve more informative points based on semantic feature extraction.

The VGS technique [16] is an approach based on voxelizing 3D space. By doing so, the space of interest is discretized by cubes of a predefined or adaptive size. In robotics, the voxel discretization of 3D space is a popular method often used for mapping and localization [21]. In [16], an adaptive VGS method is applied for speeding up the mapping procedure. The approach can also be applied as pre-processing step in a tracking and detection framework. In [22], it was used in a people detection and tracking algorithm for service robots.

Modern approaches use deep neural networks in order to learn the best sampling technique to retain points with the highest information content in the sampled subset. In [23], a network called S-NET is proposed that aims to produce a sampled subset of the point cloud with fixed size minimizing the objective function of a task network. The network, thus, produces downsampled points that are optimal for a specific superimposed application. A further developement of this approach can be found in [24]. In [25], an adaptive approach named Critical Points Layer is proposed. Using this approach, so called Critical Points with a maximum feature value are retained resulting in a set of points with a high information content. Finally, we would like to mention coreset approaches [26] as another method for point cloud reduction. These methods aim to compute a reduced set of points that approximates the full set by minimizing a specific geometric measure. By doing so, coreset methods try to ensure that superimposed algorithms produce approximately the same outcome with the reduced set as with to the full set.

III. Downsampling Methods

In this section, we briefly summarize the downsampling techniques that we apply as pre-processing before the update step of the investigated 3D EOT methods.

A. Farthest Point Sampling

The first downsampling scheme is the FPS technique [15] summarized in the pseudo-code in Alg. 1. The input to the algorithm is the point cloud and the cardinality k of the reduced point set, thus, the number of points that should be sampled. At first, the algorithm randomly selects the first index in line 3. Starting from this point, the next indexes of sampled points are selected iteratively by applying the min-max sense in lines 10–11 until the desired cardinality of the reduced point set is reached.

Algorithm 1 Farthest Point Sampling Algorithm

```
1:  function FarthestPointSampling(points, k)
2:      selected_indices ← empty array of size k
3:      selected_indices[1] ← random first index
4:      dist ← pdist2(points, points) pairwise distances
5:      min_dist ← ∞ array with length number of points
6:      for i ← 2 to k do
7:          last_index ← selected_indices[i − 1]
8:          last_point ← points[last_index]
9:          current_dist ← dist[last_index]
10:         min_dist ← min(min_dist, current_dist)
11:         farthest_index ← index of max value in min_dist
12:         selected_indices[i] ← farthest_index
13:     end for
14:     return selected_indices
15: end function
```

Advantages of the algorithm are that it is easy to implement, efficient and effective in sampling an approximately uniform

distributed subset within the full set of points. Additionally, a desired cardinality of the reduced set can be specified beforehand. This fact is especially interesting when specialized hardware is about to be designed that is limited by the number of measurements that should be processed. A crucial disadvantage is that the spatial distribution of the point cloud is not retained with this approach. As will be seen later, this fact can deteriorate tracking performance drastically.

B. Voxel Grid Sampling

The second applied downsampling scheme is the VGS [16] summarized in the pseudo-code of Alg. 2. In this method, the inputs are the point cloud and a fixed voxel size. In lines 3–11, each measurement is either assigned to an existing voxel grid, or to a newly created one if the grid cell does not exist. Afterwards, the mean values of all measurements within each voxel grid are evaluated and added to the set of downsampled measurements.

Algorithm 2 Voxel Grid Sampling Algorithm

```
1:  function VoxelGridSampling(points, voxel_size)
2:      voxel_grid ← empty voxel grid map
3:      for i ← 1 to size of points do
4:          point ← points[i]
5:          voxel_index ← floor(point/voxel_size)
6:          if voxel_index exists in voxel_grid then
7:              Append point to voxel_grid[voxel_index]
8:          else
9:              Create new entry in voxel_grid
10:         end if
11:     end for
12:     downsampled_points ← empty array
13:     for all voxel_indices in voxel_grid do
14:         voxel_points ← voxel_grid[voxel_indices]
15:         voxel_center ← mean(voxel_points)
16:         Append voxel_center to downsampled_points
17:     end for
18:     return downsampled_points
19: end function
```

The fact that the cardinality of the reduced measurement set cannot be specified beforehand can be interpreted as advantage or disadvantage. On the one hand, by not predetermining a specific cardinality the likelihood of retaining the spatial distribution of the measurement set is much higher since measurements are clustered and combined. On the other hand, hardware limits cannot be considered such as with the FPS. A clear disadvantage is that the reduced measurement set contains slightly different points than the full set. Additionally, the method is not as efficient as the FPS strategy because it has to handle the voxel grid and has to process every measurement twice.

C. Random Sampling

The last investigated downsampling technique is RS. For this technique, also the cardinality of the reduced measurement set and the measurement point cloud serve as inputs. When implementing RS, care must be taken to ensure that each measurement is only sampled once. RS has the advantage that the spatial distribution of the measurement set is retained on average. However, the reduction process can also result in very poor representations of the full set in a single run. Thus, RS is employed as reference method to exemplify the potential of downsampling a measurement set before performing a 3D EOT measurement update. In the following sections, the results will show the potential when using RS justifying further research on this topic.

IV. Investigated EOT Models

In this section, we present the two measurement models we implemented for our investigation. We apply these measurement models in maritime scenarios. In these scenarios, targets are partly underwater so that no measurements are taken from the bottom side of the target. This fact leads to error-prone estimates when target shapes are described in spherical coordinates because information from the bottom side is required for estimation [13], [14]. In the following, we use shape representations in cylindrical coordinates. Our system state is then given as

$$\underline{x}_k = \left[\underline{m}_k^T, \phi_k, \underline{w}_k^T, h_k, \underline{p}_k^T\right]^T. \tag{1}$$

with position $\underline{m}_k$, yaw angle ϕ_k, kinematic state $\underline{w}_k$, height h_k, and shape parameters $\underline{p}_k$. We assume that a measurement set

$$\mathcal{Y}_k = \left\{\underline{y}_{k,l}\right\}_{l=1}^{n_k} \tag{2}$$

of n_k measurements is gathered every time step k. For both measurement models, we assume the measurement source model

$$\underline{y}_{k,l} = \underline{z}_{k,l} + \underline{v}_{k,l} \tag{3}$$

with measurement source $\underline{z}_{k,l}$ and white Gaussian zero-mean noise term $\underline{v}_{k,l} \sim \mathcal{N}\left(\underline{0}, \mathbf{C}_{\underline{v}}\right)$.

A. Elliptic Cylinder Extrusion RHM

For the elliptic cylinder, the outer hull can completely be described with the half axes $\underline{p}_k = [a_k, b_k]^T$ of an ellipse and the height h_k. The measurement model is directly given as

$$\underline{y}_{k,l} = \underline{m}_k + \mathbf{R}_{\phi_k} \begin{bmatrix} a_k \cdot \cos(\theta_{k,l}) \\ b_k \cdot \sin(\theta_{k,l}) \\ u_{k,l} \cdot h_k \end{bmatrix} + \underline{v}_{k,l} \tag{4}$$

with rotation matrix $\mathbf{R}_{\phi_k}$, angular parameter $\theta_{k,l} \in [0, 2\pi]$, and height parameter $u_{k,l} \in [0, 1]$ of a specific measurement source. Since the parameters $\theta_{k,l}$, and $u_{k,l}$ are typically unknown, we are faced with the well-known measurement source association problem for EOT [27]. To solve this problem for the elliptic cylinder, we apply an extrusion RHM [8], [9]. The likelihood for the elliptic cylindrical shape is then given as

$$p\left(\underline{y}_{k,l}|\underline{x}_k\right) = \int_{u=0}^{1} \mathcal{N}\left(\underline{y}_{k,l}; \underline{z}_{k,l}(\hat{\theta}_{k,l}, u_{k,l}, \underline{x}_k), \mathbf{C}_{\underline{v}}\right) \cdot p\left(u_{k,l}|\underline{x}_k\right)\, du \tag{5}$$

with $p\left(u_{k,l}|\underline{x}_k\right)$ being the extrusion distribution of the height parameter $u_{k,l}$. For the elliptic cylinder, we assume a uniform distribution of the measurements in their respective heights. This uniform distribution is then approximated as Gaussian distribution and is finally given as

$$p\left(u_{k,l}|\underline{x}_k\right) = \mathcal{N}\left(u_{k,l}; \frac{1}{2}, \frac{1}{12}\right). \tag{6}$$

The angular parameter $\hat{\theta}_{k,l}$ is greedily associated using the closest point on the contour of the slice with height parameter $u_{k,l}$. This can be achieved by approximating the slice as closed polygonal chain, which yields a closed form projection of the closest point [9].

The extrusion RHM can be implemented using a smart sampling Kalman filter [28]. Using this filter, the extrusion distribution has to be sampled for every measurement using an augmented state space. An initial calculation of the samples is, however, very time consuming. We, thus, update the state using 10 measurements at once and perform several updates per measurement set.

B. Shape Tracking in Cylindrical Coordinates

For the shape tracking procedure in [14], the shape is represented using a Fourier-Chebyshev series given as

$$\begin{aligned} f(\theta, u) = \frac{a_{00}}{4} + \frac{1}{2}\sum_{n=1}^{n_u} a_{n0} T_n(u) + \frac{1}{2}\sum_{m=1}^{n_\theta} a_{0m}\cos(m\theta) \\ + \sum_{n=1}^{n_u}\sum_{m=1}^{n_\theta} a_{nm} T_n(u)\cos(m\theta) \end{aligned} \tag{7}$$

with $T_n(u)$ being the Chebyshev polynomials of the first kind [14]. The measurement equation is then given as

$$\underline{y}_{k,l} = \underline{m}_k + \mathbf{R}_{\phi_k}\begin{bmatrix} f(\theta_{k,l}, u_{k,l})\cdot\cos(\theta_{k,l}) \\ f(\theta_{k,l}, u_{k,l})\cdot\sin(\theta_{k,l}) \\ u_{k,l}\cdot h_k \end{bmatrix} + \underline{v}_{k,l} \tag{8}$$

with angular parameter $\theta_{k,l} \in [0, 2\pi]$ and height parameter $u_{k,l} \in [0, 1]$. For this measurement model, we are faced with exceptional problems if we want to implement an extrusion RHM. Details can be found in [14]. We therefore implement a greedy association model (GAM) for measurement source association. The likelihood for this association scheme is given as

$$p\left(\underline{y}_{k,l}|\underline{x}_k\right) = \mathcal{N}\left(\underline{y}_{k,l}; \underline{z}_{k,l}(\hat{\theta}_{k,l}, \hat{u}_{k,l}, \underline{x}_k), \mathbf{C}_{\underline{v}}\right). \tag{9}$$

For the GAM measurement source association, both parameters $\hat{\theta}_{k,l}$, and $\hat{u}_{k,l}$ are greedily associated to the surface of the predicted shape using the respective measurement. Thus, the height parameter $\hat{u}_{k,l}$ is associated so that the measurement in local coordinates and the measurement source are having the same z-value, and the angular parameter $\hat{\theta}_{k,l}$ so that the xy-values and the orientation enclose the same angle. The shape tracking procedure can then efficiently be implemented using an extended Kalman filter (EKF). Details for both the greedy measurement source association and the EKF implementation can be found in [14].

V. Results

We compare the three downsampling techniques presented in Sec. III in two scenarios using the measurement models presented in Sec. IV. We investigate the sampling techniques as pre-processing step in a dynamic simulated scenario and in a real-world scenario. For every scenario, varying values

$$\text{downsampling rate} = \frac{\#\,\text{retained measurements}}{\#\,\text{all measurements}} \tag{10}$$

between 0 and 1 are applied to reduce the measurement set, with # symbolizing the number of measurements. As motion model, we employ a coordinated turn (CT) model [29] for tracking and trajectory generation in the simulated scenario. The performance is measured using the intersection over union (IOU) [14] measure, as well as the computation time for update step and downsampling combined.

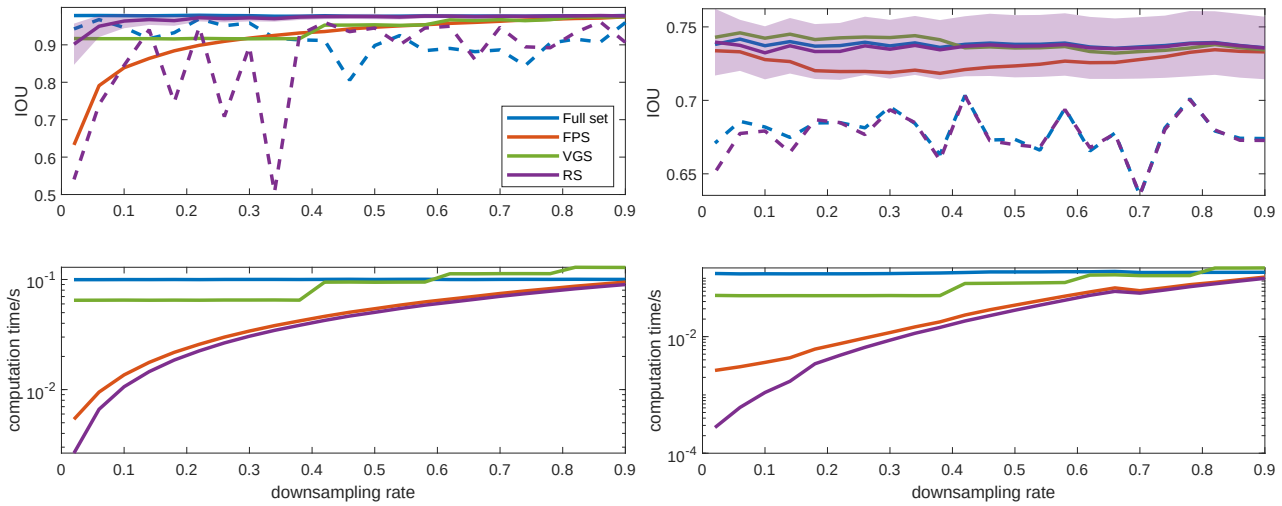
A. Simulation Results

In the first investigated scenario, we generate a reference trajectory of 300 time steps with a measurement rate of 10 Hz using the CT model. We then simulate an elliptic cylinder with a length of 8 m, a width of 4 m, and a height of 3 m. We generate measurements equally distributed on one half of the lateral surface with a standard deviation of $\sigma_{\underline{v}} = 0.1$ m. Each filter is initialized with the first measurement set using the procedure described in [14]. For both measurement models, we run four filters. One filter takes the complete measurement set for an update, the other filters each apply one of the downsampling methods described in Sec. III with a given downsampling rate (10) as pre-processing before an update. For the VGS we use the four voxel sizes $[0.15, 0.25, 0.35, 0.6]^T$, which have shown to produce a reduced measurement set of similar cardinality. In the simulated scenario, we generate 500 measurements in each time step.

In Fig. 2, the results for the simulated scenario are depicted with the results for the elliptic cylinder measurement model in Fig. 2a, and the results for the shape tracking measurement model in Fig. 2b. For these results we performed a Monte Carlo (MC) simulation of 100 runs for each downsampling rate. We then average over the whole scenario and each MC run. The solid lines show average values. For RS and the full measurement set, the worst outcome is shown with dashed lines. The standard deviation for RS is shown as purple patch.

We can see that the IOU of the full set using RS is met at a downsampling rate of approximately 0.2 for the elliptic cylinder and for each rate for the shape tracking model. The standard deviation is the same for each rate in the shape tracking scenario and gets very narrow for higher rates for the elliptic cylinder model. Interestingly, the worst outcomes of the full set and RS are almost the same in the shape tracking scenario. Overall, the other downsampling methods show poor results for both measurement models.

Of course, RS can be implemented very efficiently, however, already from the simulation results we can see how large the potential for runtime reduction can be expected if redundant measurements are removed, as the performance is the same for



(a) Results for the elliptic cylinder.

(b) Results for shape tracking.

Fig. 2: Results for the simulated scenario. Purple patches depict the standard deviations. Dashed lines show the worst results.

many reduced sets when using RS. From these results it is also evident how important it is to retain the spatial distribution of the full measurement set in the reduced measurement set, as both heuristic methods for the elliptic cylinder measurement model fail when it comes to maintaining the performance compared to the full measurement set.

B. Real-World Results

For the real-world scenario, we chose a maritime scenario that was recorded at the Rhine river in Constance with the research vessel "Solgenia". The LiDAR measurements and the CAD model of the "Solgenia" can be seen in Fig. 1. The scenario was already used in [14] and described in this paper in detail.

We use the same voxel sizes as before in the simulated scenario. The results can be seen in Fig. 3 with the results for the elliptic cylinder measurement model in Fig. 3a, and the results for the shape tracking measurement model in Fig. 3b. For FPS and VGS, we evaluate a single run with the average values of the whole scenario for every downsampling rate. Since the measurements of the LiDAR are sorted and only 10 measurements are used for an update of the elliptical cylinder, we used a random permutation of the measurements in each time step to avoid divergence. For RS, we performed a MC simulation with 100 runs for every time step to get the best performance on average. Hence, for RS the standard deviation and worst outcome are depicted like before in Fig. 3.

The potential for downsampling the measurement set to boost runtime with consistent performance can be seen clearly in these results. In addition to the simulated scenario, we can see that the performance can even get better with the reduced measurement set. These results can be ascribed to the fact that the measurement model does not describe all the measurements properly. Measurements from the inside of the target or from the top or bottom surfaces are not described with these models.

VI. Conclusion and Future Work

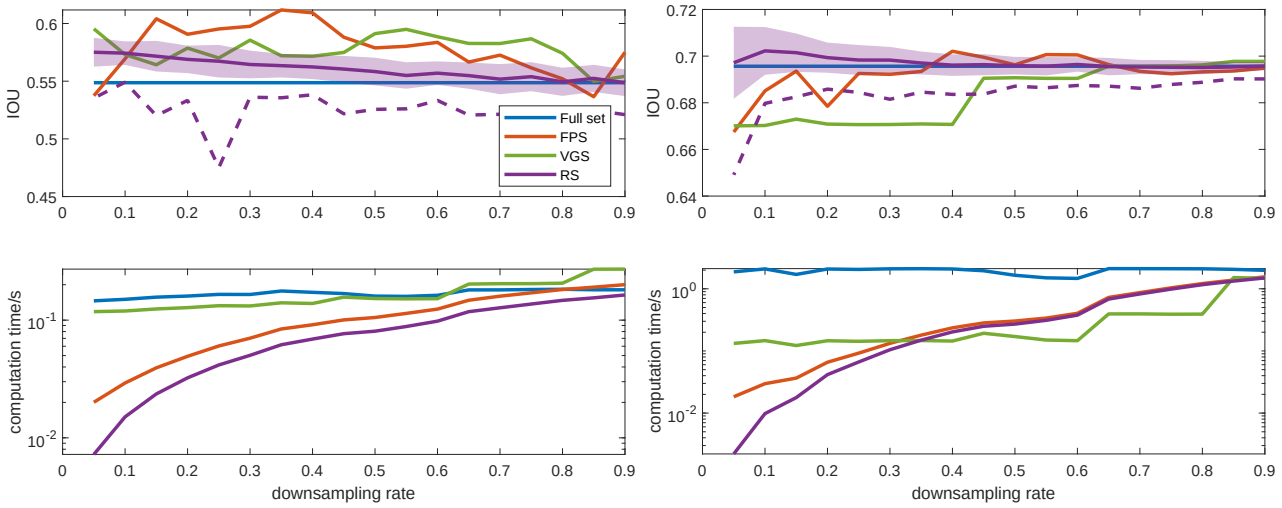
In this paper, we investigated measurement downsampling as pre-processing step before an update in 3D EOT to reduce runtime because of great redundancy in measurement sets with a high cardinality. We implemented three downsampling techniques, two measurement models, and examined our idea in a simulated and a real-world scenario. We have seen that there is a high and open potential to reduce runtime in 3D EOT by reducing the measurement set. If the spatial distribution of the measurements in the reduced set is retained, the tracking performance can be expected to be identical to the performance with the full set. Additionally, we have learned that eliminating measurements not properly described by the measurement model can improve the tracking performance further.

Future work on this topic will be diverse. First, we will work on techniques for eliminating measurements from the inside of a target and on modeling measurements from the top or bottom surfaces properly to improve the performance. Following, we will work on examining advanced downsampling methods described in Sec. III to further reduce runtime in single 3D EOT with consistent tracking performance. We expect the techniques to be also applicable in a multi-3D EOT scenario when measurements are properly associated to the respective tracks. Finally, we strive for an analytic examination of potential savings in computation time and a deeper understanding of the relation between the characteristics of the reference shape and the potential for runtime reduction. We would like to thank the reviewers for drawing our attention to the last point.

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.



(a) Results for the elliptic cylinder.

(b) Results for shape tracking.

Fig. 3: Results for the real-world scenario. Purple patches depict the standard deviations. Dashed lines show the worst results.

[3] 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, 2011.

[4] 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.

[5] M. Baum, B. Noack, and U. D. Hanebeck, "Extended object and group tracking with elliptic random hypersurface models," in *13th International Conference on Information Fusion (FUSION)*, 2010, pp. 1–8.

[6] M. Baum and U. D. Hanebeck, "Shape tracking of extended objects and group targets with star-convex RHMs," in *14th International Conference on Information Fusion (FUSION)*, 2011.

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

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] ——, "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] ——, "Shape tracking using Fourier-Chebyshev double series for 3D distance measurements," in *2023 26th International Conference on Information Fusion (FUSION)*, 2023, pp. 1–8.

[15] Y. Eldar, M. Lindenbaum, M. Porat, and Y. Zeevi, "The farthest point strategy for progressive image sampling," *IEEE Transactions on Image Processing*, vol. 6, no. 9, pp. 1305–1315, 1997.

[16] D. Lang, S. Friedmann, and D. Paulus, "Semantic 3D octree maps based on conditional random fields," in *IAPR International Workshop on Machine Vision Applications*, 2013.

[17] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 5105–5114.

[18] S. Shi, X. Wang, and H. Li, "PointRCNN: 3D object proposal generation and detection from point cloud," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 770–779.

[19] S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang, and H. Li, "PV-RCNN: Point-voxel feature set abstraction for 3D object detection," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 10 526–10 535.

[20] Z. Yang, Y. Sun, S. Liu, and J. Jia, "3DSSD: Point-based 3D single stage object detector," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 11 037–11 045.

[21] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, 2013.

[22] M. Munaro and E. Menegatti, "Fast RGB-D people tracking for service robots," *Auton. Robots*, vol. 37, no. 3, p. 227–242, oct 2014.

[23] O. Dovrat, I. Lang, and S. Avidan, "Learning to sample," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 2755–2764.

[24] I. Lang, A. Manor, and S. Avidan, "SampleNet: Differentiable point cloud sampling," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 7575–7585.

[25] E. Nezhadarya, E. Taghavi, R. Razani, B. Liu, and J. Luo, "Adaptive hierarchical down-sampling for point cloud classification," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 12 953–12 961.

[26] J. M. Phillips, "Coresets and sketches," in *Handbook of Discrete and Computational Geometry*, C. D. Toth, J. O'Rourke, and J. E. Goodman, Eds. CRC Press, Inc., 2017, ch. 48, pp. 1267–1286.

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

[28] J. Steinbring and U. D. Hanebeck, "LRKF revisited: The smart sampling Kalman filter (S2KF)," *Journal of Advances in Information Fusion*, vol. 9, no. 2, pp. 106–123, 2014.

[29] 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.