RADAR TRACKER FOR HUMAN LEGS BASED ON GEOMETRIC AND INTENSITY FEATURES

1st Nils Mandischer
IGMR, RWTH Aachen University
Aachen, Germany
mandischer@igmr.rwth-aachen.de

2nd Ina Koop
IGMR, RWTH Aachen University
Aachen, Germany
ina.koop@rwth-aachen.de

3rd Anna Granich
IHF, RWTH Aachen University
Aachen, Germany
granich@ihf.rwth-aachen.de

4th Dirk Heberling
IHF, RWTH Aachen University
Aachen, Germany
heberling@ihf.rwth-aachen.de

5th Burkhard Corves
IGMR, RWTH Aachen University
Aachen, Germany
corves@igmr.rwth-aachen.de

***Abstract*—Human leg detection and tracking is a common topic in autonomous driving and mobile robotics. When dealing with heavy rain, fog or smoke, approaches based on vision and laser are denied. Radar sensing can solve problems even in the harshest environments. Many radar tracking methods are based on measuring the human motion state by utilizing the Doppler effect. However, in applications with pivoting sensors or sensors with low bandwidth interfaces, this information may not be available. This paper proposes a method to detect legs with only geometric and intensity-based features and assesses tracking methods to form a leg tracking system to be used in a wide range of applications with special emphasis on emergency response and firefighting.**

***Index Terms*—radar, tracking, leg detection, robotics**

I. Introduction

Firefighting and emergency response is demanding on the human operator. Especially in firefighting scenarios, the attack squad can only operate for a few minutes while using respirators. Our effort is to allow mobile robots to support the operators in a variety of rescue tasks. When dealing with fire and smoke, conventional sensor principals will not produce feasible data. In fact, most smoke or heavy weather scenarios completely deny the usage of laser or vision-based systems. Thus, in emergency response, alternative methods have to be utilized.

Radar leg tracking is a less frequently researched field. Most methods originate from (autonomous) driving and are concerned with forecasting possible collisions with pedestrians. Hence, the focus of these methods is more towards short-term tracking, i.e. over short periods, where continuous tracking is less important. When dealing with long-term tracking, more laser or vision-based methods are used. Independent of the time frame, trackers are always split into two main operations: Detection and Tracking. Detection is concerned with finding the desired object in the observed space, while Tracking establishes a continuous temporal relation between those objects. In general, first, possible objects are detected. Secondly, objects are accepted or denied based on their spatial relation and afterwards combined into leg pair objects. Finally, the leg pairs are tracked to establish a semantic relation between consecutive detections.

In this paper, we introduce a novel and comprehensive radar leg tracking pipeline that is solely based on geometric and intensity features. Thus, our method is also available in situations, where the radar sensor cannot compute or detect Doppler shift, hence the relative velocity of its target objects. In this paper, first, the related work is presented concentrating on the two major concepts Detection and Tracking. Afterwards, our pipeline is discussed and evaluated in divers scenarios, scoring an accuracy of up to 79.18%.

A. Related Work: Leg Detection

In general, radar leg detection is based on motion detection using Doppler shift and Fourier transform techniques, like spectogram analysis. Majer et al. [1] propose a system for pedestrian detection, combining ultra-wideband radar and laser. The method clusters legs based on the Euclidean relation between radar scan points and rejects/accepts them according to the laser detections. Afterwards, the clusters are classified in a Decision Tree using seven distinct features. Zhao et al. [2] propose to use Density-Based Spatial Clustering of Applications with Noise (DBSCAN) to generate the point clusters instead. In situations, where Doppler data is not available, e.g. for pivoting radars or in applications with low bandwidth interfaces, these methods are not usable.

If Doppler shift shall not be an inherit requirement, also laser-based methods can be exploited. Laser sensing approaches fall in three categories based on their clustering method: Blob Analysis [3]–[5], Distance Thresholding [6] and Local Neighborhood Clustering [7]–[9]. Xavier et at. [7] define a leg as a non-occluded circle with diameter between 0.1 and 0.25 m. Taipalus and Ahtiainen [8] define legs based on data density, min./max. leg cross-section and circumferences. Chung et al. [9] detect legs based on the cluster contour, involving its width and depth. Arras, Mozos and Burgard [10] propose a more sophisticated method for leg detection, where the legs are characterized by eleven geometric (e.g.

width, point count) and three motion features (e.g. mean speed, jumping distance). AdaBoost is then used to classify the legs. AdaBoost is a supervised learning technique based on combining multiple weighted weak classifiers into a single strong classifier. The leg detection method is referenced in multiple works, e.g. [11], [12].

B. Related Work: Leg Tracking

Leg tracking is a challenging topic when the human motion state cannot be measured directly but has to be estimated in a state observer. For the estimation of the motion state of the tracked person, mostly Bayesian filters are applied. Common filters are Kalman Filter (KF), Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF). KF is a suitable model, when dealing with a Gaussian sensor model and linear motion [13]. Saho [14] proposes to use position-velocity-measured KF to improve the accuracy. However, the abstraction in KF towards non-linear motion is still limited. Zhai, Wu and Wang [15] propose to also model statistical system and observation noise by adapting the noise parameters to bring the model closer to the real-world. Linder et al. [11] use an EKF instead. This filter overcomes the linearity constraints of the KF with low complexity, but low precision and slow convergence speed [15]. Meuter et al. [16] use an UKF and demonstrate its real-time capabilities, while handling non-linearity. Schubert, Richter and Wanielik [17] propose to combine multiple motion models in the UKF to overcome each model's weaknesses. Linder et al. [11] compare trackers based on their data association method, i.e. Greedy Nearest Neighbor (GrNN), Simple Nearest Neighbor (SNN) and Multi-Hypothesis Tracker (MHT), and observe that there is no single method for all applications.

Leigh et al. [18] propose the laser-based *leg_tracker* framework[1], which is open-access in the Robot Operating System (ROS). They use a combination of Nearest Neighbor Clustering (NNC) with the Hungarian algorithm [19] and a Random Forrest Classifier with 15 features for leg detection, based on the works of Arras, Mozos and Burgard [10] and Lu and Smart [20]. Afterwards, they generate tracks based on EKF and Global Nearest Neighbor (GNN) data association with a Mahalanobis gate (see Section II-C2).

II. Radar Leg Tracker

In this section, the proposed radar leg tracker is described. The method is based on the approach by Leigh et al. [18]. The tracker should fulfill online time constraints. Hence, a pre-processing step is introduced to shrink the data load. Further, detection and tracker modules are presented.

A. Pre-Filtering

The main objective of the pre-filtering step is to reduce the data load, while maintaining a certain informative value. In the worst case, leg detections are filtered out before the tracking even starts. The simplest filter is a distance filter, which deletes far and close points but keeps all data in between. This is feasible as the regular distance for people following is 1-3 m. A more sophisticated method is to, first, filter by distance, and afterwards apply a threshold on the reflection intensity and only keep those points that exceed the static threshold. In [21] this approach is extended by Otsu Thresholding. The objective of an Otsu Threshold is to split the data set into two groups with maximal inter-class and minimal intra-class variance. Hereby, the data table is annotated intensities over number of points in each intensity class. Hill Climbing optimization is used to find the corresponding threshold. The major benefit of the dynamic threshold is that it can react to the surrounding environment. With a static threshold, strong reflections nearby may suppress other data points.

B. Radar Leg Detection

The leg detector consists of two steps: Clustering and Classification. Clustering is concerned with structuring the scan into multiple objects. Afterwards, potential legs are identified using the classification method.

1) Clustering: In this work, we compare and evaluate two clustering algorithms (see Section III-A). In [18], NNC is used. Points are clustered if their Euclidean distance to another point in the cluster is below a threshold. The threshold is selected in such a way that it can distinguish legs while generating no more than two clusters per person. Later on, people tracks can be initialized based on one or two clusters, so generating two clusters is no mandatory requirement but simplifies the track matching process.

For radar tracking Zhao et al. [2] propose to use DBSCAN. In DBSCAN the cluster is grown from a core point. If a point lies within a distance threshold to the core point, it is grouped to the core cluster. Other points are labeled noise. DBSCAN is an iterative scheme, where noise points can be post-clustered later on when a core cluster is grown. The approach is more complex but generates better clusters. However, if the clusters have varying density, we observed a dependency on a suitable set of parameters, which is supported by Schubert et al. [22].

2) Random Forest Classification: Once clusters have been generated, legs can be detected. For detection, a Random Forest Classifier is utilized. A Random Forest (RF) consists of multiple Binary Trees (Decision Trees). The Binary Trees are made from the same pool of features with varying sequence using supervised learning. The majority vote in the RF decides if a cluster is labeled a leg detection. The features in our RF are based on, but extend (emphasized) the ones used by Leigh et al. [18]. Geometric features are:

1. Number of points
2. Standard deviation
3. Mean average deviation from median
4. Cluster width
5. Linearity
6. Circularity
7. Radius of best-fit circle
8. Boundary length
9. Boundary regularity
10. Mean curvature
11. Mean angular difference
12. *Inscribed angular variance*
13. *Standard deviation of inscribed angle*
14. Distance from sensor
15. *Distance relative to number of points*

[1] https://github.com/angusleigh/leg_tracker

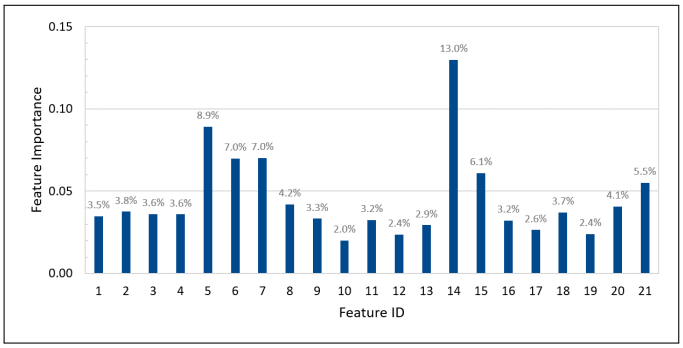

Fig. 1: Influence of all features on the leg decision

While these features are based on the geometric relations in the cluster, a radar detection is further characterized by the reflection intensities. Thus, we add intensity features. These are based on, but extend the work of Majer et al. [1]:

16. Mean
17. *Median*
18. Standard deviation
19. *Minimum*
20. *Maximum*
21. *Average on cluster hull*

The influence of each feature on the leg decision is analyzed and listed in Figure 1. Distance and cluster linearity have the most impact on the decision. It is to be noted that the training data only features ranges of up to 8 m, hence this feature might be overfitted on the common distance in human following applications.

C. Leg Tracker

The leg tracker consists of three sub-methods: Motion modeling, data association and track management. First, motion modeling is used to estimate the motion state of the leg object. Afterwards, the legs are matched based on the motion state and forwarded to the track management, which is concerned with maintaining and deleting old/new tracks.

1) Motion Modeling: Motion prediction is commonly performed with Bayesian filters. Within this class of filters, KFs are often used in tracking applications. KFs use a two-stage approach, where first the human motion is predicted and afterwards updated based on the real-world observations, i.e. sensor detections. The classic KF process is depicted in Figure 2. KF models are assumed to be optimal, i.e. with white noise and known noise covariance. For processes with high level of noise, the update step can be exchanged by the Joseph form [23]. This form can handle KFs that are not optimal. The new update step is adapted to

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k|k-1}(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)^T + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^T. \quad (1)$$

In addition to the Joseph form, the Adaptive Sage-Husa Kalman Filter (ASHKF) allows the noise to be modeled closer to the real system [15]. In this variation the observation noise is computed by

$$\mathbf{R}_k = (1 - d_k)\mathbf{R}_{k-1} + d_k(\tilde{\mathbf{y}}_k\tilde{\mathbf{y}}_k^T - \mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^T), \quad (2)$$

with the fading memory index $d_k = (1 - b)/(1 - b^k)$.

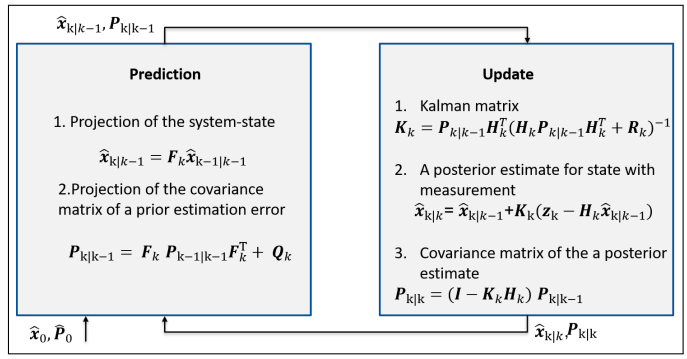

Fig. 2: Two-step approach in the classic KF with relevant math, including system matrix $\boldsymbol{F}$, process noise $\boldsymbol{Q}$, measurement noise $\boldsymbol{R}$, innovation covariance $\boldsymbol{S}$ and observation matrix $\boldsymbol{H}$

2) Data Association: Data association is concerned with matching the projected track (from motion model) and the incoming measurements to establish a continuous track. Before matching the track, a gating procedure is applied to eliminate outliers. Gating establishes a relation between the old track and update candidates by applying a distance threshold. In this case, the Mahalanobis distance metric d_m is used, given by

$$d_m(\mathbf{x_1}, \mathbf{x_2}) = \sqrt{(\mathbf{x_1} - \mathbf{x_2})^T\mathbf{S}^{-1}(\mathbf{x_1} - \mathbf{x_2})}, \quad (3)$$

where $\mathbf{x}_i$ is a data point and $\mathbf{S}$ the covariance matrix. The covariance is the distribution of a data set, but can cover more elaborate characteristics like noise, if required. In an experimental study, the gate is set to $d_m = 1.3$ m.

For track-to-detection matching, GrNN is used. Nearest Neighbor methods find the best match based on neighborhood relations. First, the Mahalanobis distances between all detections and tracks are mapped in the association distance matrix. By applying the gate, some relations are rejected. From the matrix a Bipartite Graph is generated. The transition costs are based on the mapped Mahalanobis distances. The data association problem is then solved by selecting the smallest distances (greedy) until all detections are matched. Unassigned tracks are either categorized as lost or new depending on the prior state. GrNN is feasible in problems with few possible combinations [24], which can be assumed for people tracking. In more complex problems, GrNN may converge to non-optimal results. In this case data association may be shifted to GNN utilizing the Hungarian algorithm [19] instead.

3) Track Management: Track management is concerned with initializing, maintaining and deleting tracks. Some of the associated matches need to be confirmed. To initialize a track, the person has to be seen in eight consecutive scans (after initially being detected) with a covered average distance of 5-18 cm per frame. In case of a static sensor, the thresholds can be relaxed. To avoid a person being represented by multiple tracks, the method proposed by Leigh et al. [18] is used. Here, a person track can be matched with up to two detections. In this cas,e the track is updated with the mean position of the single leg detections.

In our tracker, a track is deleted in following cases:

i. Track confirmed < 10 frames, track rejected > 10 frames
ii. Track not confirmed > 40 frames
iii. Track confirmed > 10 frames, covered distance < 5 cm per frame (mean)
iv. Covered distance < 5 cm for > 15 frames
v. Covered distance between two frames > 0.5 m

Old tracks (track confirmed > 50 times) are only considered for deletion, if the track was rejected more than 15 times and traveled less than 5 cm for more than ten frames. The parameters were determined and validated in the training scenarios presented below.

III. Validation

The system is validated in indoor (A, B, C) and outdoor (D, E) scenarios over varying time periods. The robot used is a Robotnik Summit XL-Steel equipped with an indurad iSDR-C pivotal radar sensor. The sensor outputs five 360°-scans per second and operates within a range of 72 - 82 GHz. Tests are performed in five environments (see Figure 3):

A. Large office room with tables and chairs
B. Lab with many reflective objects
C. Wide corridor with low noise
D. Walkway between two buildings, controlled disturbances
E. Walkway along a street, uncontrolled disturbances

A. Detection

The leg detection is validated with the confusion matrix. Here, the expected output is compared with the leg detection and classified as True Positives (TP; leg expected and detected), True Negatives (TN; no leg expected and none detected) or their inversions (FP: False Positives, FN: False Negatives). From these classes, four scores are computed:

$$\text{Accuracy} \qquad Ac = \frac{TP + TN}{TP + TN + FP + FN} \tag{4a}$$

$$\text{Precision} \qquad Pr = \frac{TP}{TP + FP} \tag{4b}$$

$$\text{Recall} \qquad Re = \frac{TP}{TP + FN} \tag{4c}$$

$$\text{F1-Score} \qquad F1 = \frac{2 * Re * Pr}{Re + Pr} \tag{4d}$$

The leg detector is trained with 4464 positive and 2195 negative examples. Positive examples originate from a single person moving freely and continuous in front of the sensor. Negative examples are arbitrary clusters from the environment. The training sets are generated in environments A and B and cover five different people (male and female). For testing, two test sets are made with 155 frames each. One covers a known, another a known and an unknown person. Both sets are generated in environment B. The results are listed in Table I. While the combination DBSCAN and distance filter offers the best overall score, it is unable to be run in real time. Thus, the combination NNC and Otsu filter is used, which offers a high score with low computational complexity. The NNC is parameterized with a distance threshold of 0.45 m and a minimal cluster size of five points. The feature importance analysis in Figure 1 is generated using this exact combination.

TABLE I: Comparison of clustering and filter combinations

Cluster	Filter	Ac	Pr	Re	F1	Cycle
DBCSAN	Otsu	94.9%	57.4%	62.9%	60.0%	106.0 ms
DBSCAN	Dist.	97.1%	75.6%	75.1%	75.3%	857.1 ms
NNC	Otsu	97.8%	68.3%	75.8%	71.9%	45.1 ms
NNC	Dist.	94.8%	34.9%	73.3%	47.3%	154.4 ms

TABLE II: Comparison of ASHKF combinations with different data association algorithms and gates

Data Association	Gate	FP	MIS	ID	MOTA
Global NN	Euclidean	43	56	2	73.97%
	Mahalanobis	10	170	8	75.33%
Greedy NN	Euclidean	42	58	2	73.71%
	Mahalanobis	13	59	1	81.19%

B. Tracking

For tracking validation, the Multi Object Tracking Accuracy (MOTA) measure is used, which is computed by

$$MOTA = 1 - \frac{\sum_k (MIS_k - FP_k + ID_k)}{\sum_k GT_k}. \tag{5}$$

Here, MIS is the number of misses, i.e. mismatch between track and detection, and ID is the number of ID switches, i.e. when a track is mistakenly assigned a new ID. GT is the ground truth.

The tracking performance using the ASHKF is validated in environments B and C with multiple people and a static sensor. The results are listed in Table II. For comparison, also GNN and both gating methods - Euclidean and Mahalanobis - are tested. The proposed combination of GrNN and Mahalanobis gate shows the best MOTA score and fewest ID switches.

C. Full System

The full system is evaluated in two test cases. In test case I (environment D), the robot is stationary and two people move around it, crossing the FOV of the sensor. This is a long-term test, spanning over 10 min. The conditions are controlled. Test case II is as close to a real-world scenario as possible. In this case, the robot is following a single person (manually controlled) in environments D and E. In environment E the conditions are uncontrolled with people arbitrarily passing by. The results of the two test cases are listed in Table III. The system shows a good reliability in all test cases. In environment D, lamp posts along the walkway induce many FP, due to their leg-like shape. This has a significant impact on the MOTA scores in test cases I and II-D. It is to be noted, that the system maintains the same ID for the target person and never looses her track in test case II.

Further, the impact of multiple people on the system is evaluated. The test is performed under the conditions of test case I. The results are listed in Table IV. It is observed, that the MOTA is lowered the more people are tracked, while the impact on run time is minimal. Similar scores in

Fig. 3: Environments used in the validation (from left: A to E)

TABLE III: Evaluation of the full system in the two test cases

Test Case	Environment	Scans	FP	MIS	ID	MOTA
I	D	6010	402	1275	27	71.65%
II	D	836	165	104	0	68.75%
II	E	1047	94	124	0	79.18%

TABLE IV: Impact of people count on system performance

People Count	Scans	FP	MIS	ID	MOTA	Cycle
2	650	93	225	3	75.31%	39.89ms
3	655	57	617	9	65.24%	34.11ms
4	660	56	917	13	62.65%	36.44ms
5	659	424	990	34	56.05%	45.03ms
6	657	351	1357	38	55.71%	47.31ms

cases with four and five people indicate that the MOTA may eventually reach a plateau. In addition, comparing test case I and the test case with two people, the tracking period may have an impact on the performance (loss of 4%). However, the system exceeds the results by Leigh et al. [18] and is comparable to the results by Linder et al. [11], who apply the MOTA score on laser tracking. This indicates that the proposed radar tracker is at least on par with typical laser leg tracker applications. Comparison to radar trackers is difficult as most utilize Doppler shift to measure the motion state.

IV. Conclusion

This paper proposes a radar-based approach for human leg detection and people tracking using just geometric and intensity features, estimating the motion state instead of measuring. Leg detection is based on a combination of a Random Forest Classifier with 21 features and Nearest Neighbor Clustering with an Otsu filter. The tracker is based on the Adaptive Sage-Huse Kalman Filter combined with Greedy Nearest Neighbor data association and a Mahalanobis gate. The system is evaluated in five environments with varying recording time. The overall score indicates good usability in people following and for stationary tracking systems in outdoor and indoor applications. In future research, the tracking system shall be tested in a firefighting scenario under real conditions.

References

[1] F. Majer, Z. Yan, G. Broughton, Y. Ruichek, and T. Krajnik, "Learning to see through haze: Radar-based human detection for adverse weather conditions," in *2019 European Conference on Mobile Robots*, 2019.

[2] P. Zhao, C. X. Lu, J. Wang, C. Chen, W. Wang, N. Trigoni, and A. Markham, "mid: Tracking and identifying people with millimeter wave radar," in *15th International Conference on Distributed Computing in Sensor Systems*, 2019.

[3] A. Fod, A. Howard, and M. Mataric, "A laser-based people tracker," in *IEEE International Conference on Robotics and Automation*, 2002.

[4] N. Bellotto and H. Hu, "Computationally efficient solutions for tracking people with a mobile robot: an experimental evaluation of bayesian filters," *Autonomous Robots*, vol. 28, no. 4, pp. 425–438, 2010.

[5] M. Scheutz, J. McRaven, and G. Cserey, "Fast, reliable, adaptive, bimodal people tracking for indoor environments," in *IEEE International Workshop on Intelligent Robots and Systems*, 2004.

[6] B. Kluge, C. Kohler, and E. Prassler, "Fast and robust tracking of multiple moving objects with a laser range finder," in *IEEE International Conference on Robotics and Automation*, 2001, pp. 1683–1688.

[7] J. Xavier, M. Pacheco, D. Castro, A. Ruano, and U. Nunes, "Fast line, arc/circle and leg detection from laser scan data in a player driver," in *IEEE International Conference on Robotics and Automation*, 2005.

[8] T. Taipalus and J. Ahtiainen, "Human detection and tracking with knee-high mobile 2d lidar," in *IEEE International Conference on Robotics and Biomimetics*, 2011, pp. 1672–1677.

[9] W. Chung, H. Kim, Y. Yoo, C.-B. Moon, and J. Park, "The detection and following of human legs through inductive approaches for a mobile robot with a single laser range finder," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 8, pp. 3156–3166, 2012.

[10] K. O. Arras, O. M. Mozos, and W. Burgard, "Using boosted features for the detection of people in 2d range data," in *IEEE International Conference on Robotics and Automation*, ser. Proceedings, 2007.

[11] T. Linder, S. Breuers, B. Leibe, and K. O. Arras, "On multi-modal people tracking from mobile platforms in very crowded and dynamic environments," in *IEEE International Conference on Robotics and Automation*, 2016.

[12] C. Dondrup, N. Bellotto, D. Jovan, and M. Hanheide, "Real-time multisensor people tracking for human-robot spatial interaction," in *Workshop on Machine Learning for Social Robotics at International Conference on Robotics and Automation (ICRA)*, 2015.

[13] J. Mochnac, S. Marchevsky, and P. Kocan, "Bayesian filtering techniques: Kalman and extended kalman filter basics," in *RADIOELEKTRONIKA '09. 19th International Conference*. IEEE, 2009.

[14] K. Saho, "Kalman filter for moving object tracking: Performance analysis and filter design," in *Kalman Filters - Theory for Advanced Applications*, G. L. d. O. Serra, Ed. IntechOpen, 2018.

[15] G. Zhai, C. Wu, and Y. Wang, "Millimeter wave radar target tracking based on adaptive kalman filter," in *IEEE Intelligent Vehicles Symposium*, 2018.

[16] M. Meuter, U. Iurgel, S.-B. Park, and A. Kummert, "The unscented kalman filter for pedestrian tracking from a moving host," in *IEEE Intelligent Vehicles Symposium*, 2008.

[17] R. Schubert, E. Richter, and G. Wanielik, "Comparison and evaluation of advanced motion models for vehicle tracking," in *2008 11th International Conference on Information Fusion*, 2008, pp. 1–6.

[18] A. Leigh, J. Pineau, N. Olmedo, and H. Zhang, "Person tracking and following with 2d laser scanners," in *Ieee International Conference on Robotics and Automation*, 2015.

[19] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[20] D. V. Lu and W. D. Smart, "Towards more efficient navigation for robots and humans," in *International Conference on Intelligent Robots and Systems*, 2013.

[21] N. Mandischer, S. Charaf Eddine, M. Huesing, and B. Corves, "Radar slam for autonomous indoor grinding," in *IEEE Radar Conference*, 2020.

[22] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "Dbscan revisited, revisited," *ACM Transactions on Database Systems*, 2017.

[23] R. Zanetti and K. J. DeMars, "Joseph formulation of unscented and quadrature filters with application to consider states," *Journal of Guidance, Control, and Dynamics*, vol. 36, no. 6, pp. 1860–1864, 2013.

[24] P. C. Austin, "A comparison of 12 algorithms for matching on the propensity score," *Statistics in medicine*, vol. 33, no. 6, 2014.