

Real-Time Pose Graph SLAM based on Radar

Martin Holder, Sven Hellwig, and Hermann Winner

Abstract—This work presents a real-time pose graph based Simultaneous Localization and Mapping (SLAM) system for automotive Radar. The algorithm constructs a map from Radar detections using the Iterative Closest Point (ICP) method to match consecutive scans obtained from a single, front-facing Radar sensor. The algorithm is evaluated on a range of real-world datasets and shows mean translational errors as low as 0.62 m and demonstrates robustness on long tracks. Using a single Radar, our proposed system achieves state-of-the-art performance when compared to other Radar-based SLAM algorithms that use multiple, higher-resolution Radars.

I. INTRODUCTION

One key aspect of autonomous driving technology is the ability to accurately determine the vehicle's location. This requires precise maps of the surrounding environment, which can be created with Simultaneous Localization and Mapping (SLAM) algorithms. SLAM aims to construct a consistent map of an unknown environment while simultaneously estimating the vehicle's pose within the map, see Fig. 1. Most modern SLAM systems use either laser scanner (Lidar) or camera-based approaches. There have been comparatively few attempts at SLAM using Radar sensors.

In contrast to cameras, Lidar, and ultrasonic sensors, Radar is suited for distinguishing between static and moving targets in one measurement cycle, due to measurement of the Doppler shift indicating relative radial velocity. This additional information appears to be attractive for discarding moving objects during the map building process. While Radar lacks the high range and angular resolutions offered by Lidar, it is more robust to adverse weather conditions and more affordable. Furthermore, Radar detections are more sparse than Lidar measurements. Radar detections are prone to distortions, as mirror targets and clutter returns occur frequently.

In this paper, we describe a SLAM system based on point-cloud-like measurements obtained from a single automotive-grade Radar of the sort used in driver assistance applications. We seek to determine the level of performance a SLAM application may achieve using a single, front-facing Radar. The remainder of the paper is organized as follows: After a brief review of existing Radar SLAM approaches, we introduce our pose graph SLAM concept for Radar. We evaluate our algorithm on three different scenarios recorded under real-world driving conditions. We conclude by summarizing the important characteristics of our algorithm and give an outlook to further work.

All authors are with Institute of Automotive Engineering, Technische Universität Darmstadt, 64287 Darmstadt, Germany {holder, winner}@fzd.tu-darmstadt.de sven.hellwig@online.de

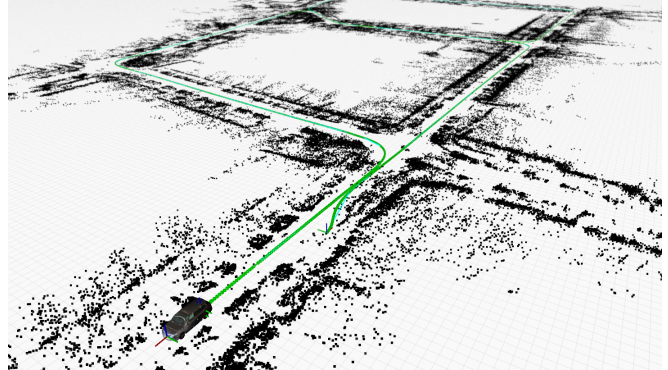


Fig. 1: Mapping process in an urban environment. The image shows the estimated trajectory and resulting point cloud map.

II. RELATED WORK

In one of the earliest applications of Radar for vehicle localization, highly visible Radar reflectors with known positions were processed by an extended Kalman filter (EKF) [1]. Reflectors were later used as landmarks in an EKF-based SLAM system [2]. A more recent scan-matching-based SLAM approach uses the Fourier-Mellin transformation to match consecutive Radar scans, where the power spectra are interpreted as 360° images [3]. The authors do not, however, present a solution for loop closing. Most other Radar-based SLAM systems use multiple Radar sensors for 360° coverage with range resolutions of up to 0.15 m accuracy. Many employ particle filters, such as FastSLAM [4], which renders an occupancy grid that can represent both occupied and free space, or memory efficient *Cluster-SLAM* [5] that merges Radar detections into larger *micro clusters*. A graph-based Radar SLAM is proposed in [6], where image features are extracted from a Radar scan and used as landmarks. 360° Radar sensors have also been used for ego-motion estimation [7]. Here, a point cloud of detected targets is extracted from the Radar's power-range spectra and the vehicle odometry is calculated by matching consecutive point clouds. Here too, loop closing is not considered. To the authors' best knowledge, this is the first work to conduct loop closing Radar SLAM using only a single, front-facing Radar sensor with a range resolution of approx. 0.5 m and an angular measurement range of approx. 60°.

III. METHOD

A major disadvantage of landmark based SLAM systems is the reliance on the extraction of suitable landmarks, which is difficult for Radar scans with medium range and angular resolution and high amounts of noise. Many Lidar-based

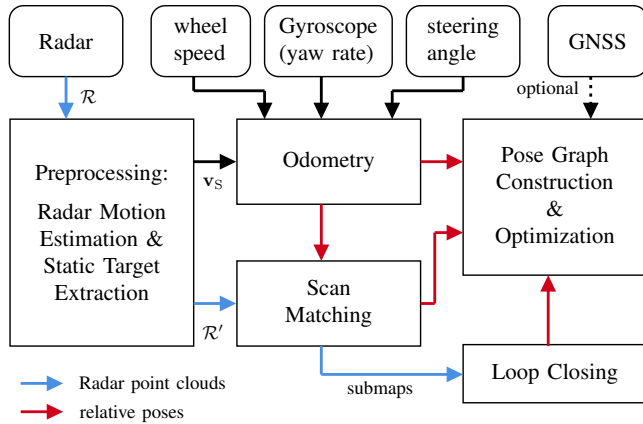


Fig. 2: Components of the SLAM system

SLAM systems forgo the problem of landmark extraction by matching point clouds directly [8]–[10]. However, despite successful application in Radar ego-motion estimation [7] and localization [11], point cloud matching approaches have not yet been applied to Radar-based SLAM. We therefore propose a graph-based SLAM system using the Iterative Closest Point (ICP) method for scan matching. Graph-based systems are the *de facto* standard in SLAM due to their reported superior performance and ease of use compared to filtering-based approaches [12]. Fig. 2 illustrates the structure of our SLAM system. The Radar preprocessing component separates static targets from moving targets and clutter while also estimating the Radar sensor’s velocity. The odometry component estimates relative transformations between consecutive poses by fusing wheel speed, steering wheel angle, and yaw rate measurements with the velocity information obtained by the Radar. The scan matching component identifies relative pose estimates by aligning sequential Radar scans. Lastly, the loop closing component calculates relative transformations with respect to previously visited places. These relative pose estimates are combined to construct a pose graph. The SLAM algorithm obtains an estimate of the vehicle trajectory by means of graph optimization. This process also produces a point cloud map composed of Radar detections. The relative pose measurements can optionally be augmented with position measurements from a GNSS receiver to improve localization. The following sections describe each of these components in detail.

A. Radar Preprocessing

The Radar sensor measurements are available as a list of detections, where Doppler ambiguities are resolved and to some extent, sensor artifacts such as clutter and invalid detections are removed. The sensor also compensates for alignment errors which justifies an all-flat road assumption. A Radar scan \mathcal{R} is a set of detections each consisting of the range r_i , the azimuth angle ϕ_i , and the relative radial velocity $v_{r,i}$.

The goal of the Radar preprocessing step is to find the velocity vector of the sensor in Cartesian sensor coordinates

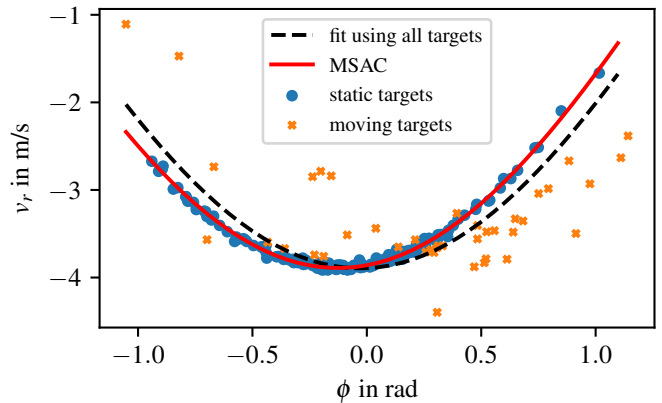


Fig. 3: Application of MSAC for filtering static targets by fitting a sinusoid.

$\mathbf{v}_S = [v_{S,x}, v_{S,y}]^\top$. Since we are only interested in static targets, the mapping process should discard moving targets such as other vehicles and pedestrians. To achieve this, we apply the Radar ego-motion estimation method proposed by Kellner *et al.* [13]. The true velocity of static targets as seen from the sensor has the same magnitude but opposite direction of \mathbf{v}_S . Only the radial components of the target velocities can be measured, leading to the following relationship between the sensor and target velocities:

$$-v_{r,i} = \cos(\phi_i) v_{S,x} + \sin(\phi_i) v_{S,y} \quad (1)$$

Since Eq. (1) holds only for static targets, using all detections would produce poor results when solving with ordinary linear least squares. Therefore, we employ a variant of the Random Sample Consensus (RANSAC) algorithm, M-Estimator Sample Consensus (MSAC) [14], that treats moving targets and clutter as outliers. The inlier set \mathcal{R}' contains static targets and is used in further processing. Fig. 3 shows the result of the MSAC regression.

B. Odometry

In order to estimate the vehicle’s velocity and relative poses between consecutive sensor measurements, the odometry component fuses rear wheel speeds, yaw rate, steering wheel angle, and the Radar velocity using an unscented Kalman filter (UKF). UKFs produce more accurate estimates in non-linear systems with measurement noise and do not require online calculations of Jacobians. The state vector \mathbf{x}_{odom} contains the vehicle 2D position, x and y , and heading ψ in a global, Cartesian coordinate system, the longitudinal velocity v_x in a coordinate system fixed to the rear axle, the yaw rate $\dot{\psi}$, and additional unknown system parameters such as a wheel speed correction factor k_{ws} , the steering ratio i_s , and steering wheel bias b_δ :

$$\mathbf{x}_{\text{odom}} = [x \ y \ \psi \ v_x \ \dot{\psi} \ k_{ws} \ i_s \ b_\delta]^\top \quad (2)$$

The correction factor k_{ws} accounts for wheel slip and the deviation of actual from nominal wheel radius. We assume that k_{ws} is similar for both wheels since the same type of tire with similar air pressure and wear is used. The estimation of

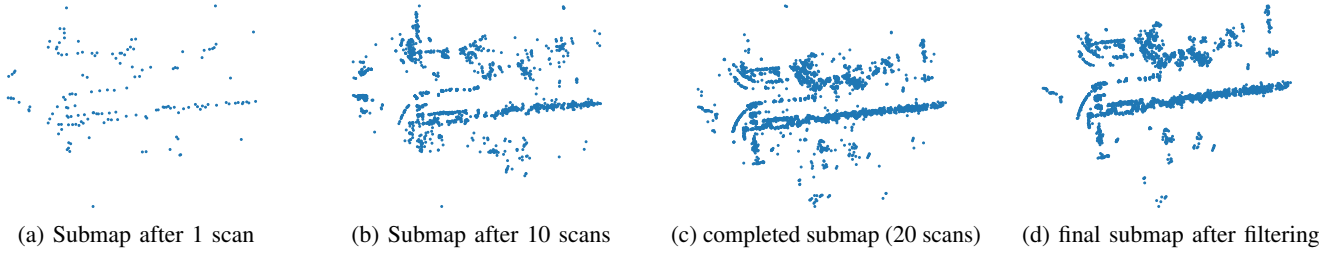


Fig. 4: Submap creation process. Radius outlier filtering is done with a radius of 2 m and 4 neighbors.

b_δ is necessary because the initial steering wheel position is unknown due to the usage of an incremental encoder. The filter uses a model that assumes that velocity, yaw rate, and system parameters stay constant over the sampling period. The update step is based on the following sensor models to predict the left and right rear wheel speeds v_{rL} and v_{rR} , the yaw rate reading of the gyroscope ω , the steering wheel angle δ_H and the Radar sensor velocity:

$$\begin{bmatrix} \hat{v}_{rL} \\ \hat{v}_{rR} \\ \hat{\omega} \\ \hat{\delta}_H \\ \hat{v}_{S,x} \\ \hat{v}_{S,y} \end{bmatrix} = \begin{bmatrix} k_{ws} \cdot \left(v_x - \frac{b}{2} \dot{\psi} \right) \\ k_{ws} \cdot \left(v_x + \frac{b}{2} \dot{\psi} \right) \\ \dot{\psi} + b_\omega \\ i_s \cdot \arctan \left(l \dot{\psi} v_x^{-1} \right) + b_\delta \\ \cos(\gamma_S) \left(v_x - \dot{\psi} y_S \right) + \sin(\gamma_S) \left(\dot{\psi} x_S \right) \\ -\sin(\gamma_S) \left(v_x - \dot{\psi} y_S \right) + \cos(\gamma_S) \left(\dot{\psi} x_S \right) \end{bmatrix} \quad (3)$$

with wheelbase l , rear axle track b , gyroscope bias b_ω and Radar sensor pose $[x_S, y_S, \gamma_S]$ relative to the center of the rear axle. The filter assumes additive white Gaussian noise for all sensor measurements and that the lateral velocity is negligible. In rare cases, the Radar motion estimation selects an incorrect inlier set which can lead to erroneous estimates of \mathbf{v}_S . As a safeguard, the UKF rejects residuals above a threshold as outliers, as described by Ting *et al.* [15].

C. Scan Matching

The main objective of the scan matcher is to estimate relative transformations between current and previous Radar scans using the 2D point-to-point ICP algorithm [16]. The key idea of ICP is to align two point sets by iteratively minimizing the distances between pairs of closest points. Let \mathbf{t}_0, θ_0 be an initial guess for the translation and rotation, and $\mathcal{P} = \{\mathbf{p}_1, \dots, \mathbf{p}_n\}$ a point set that is to be aligned with $\mathcal{Q} = \{\mathbf{q}_1, \dots, \mathbf{q}_m\}$. In each iteration, the algorithm first transforms \mathcal{P} according to the current estimates of θ_k and \mathbf{t}_k :

$$\mathcal{P}'_k = \{\mathbf{p}'_1, \dots, \mathbf{p}'_n\} \quad (4)$$

$$\text{with } \mathbf{p}'_i = \mathbf{R}_k \mathbf{p}_i + \mathbf{t}_k, \quad \mathbf{R}_k = \begin{bmatrix} \cos \theta_k & -\sin \theta_k \\ \sin \theta_k & \cos \theta_k \end{bmatrix} \quad (5)$$

It then forms n point pairs $\{\mathbf{p}'_i, \mathbf{q}_j\}$ by associating each point in \mathcal{P}'_k with its closest point in \mathcal{Q} and minimizes the sum of

the squared Euclidean distances:

$$\mathbf{t}_{k+1}, \theta_{k+1} = \arg \min_{\mathbf{t}, \theta} \sum_{(i,j)} \|\mathbf{R}(\theta) \mathbf{p}_i + \mathbf{t} - \mathbf{q}_j\|^2 \quad (6)$$

This process is repeated until convergence. In contrast to Lidar sensors with high angular resolution, where each scan contains several thousand points and considerably less noise, single Radar scans often comprise fewer than 100 detections after removing clutter and moving targets. Therefore, directly matching consecutive Radar scans leads to poor results. Instead, the scan matcher forms small partial maps of the environment (hereafter referred to as *submaps*) by merging several Radar scans, see Fig. 4. The positions of the static targets in each scan \mathcal{R} are transformed into a common Cartesian coordinate system using the odometry pose estimates and the known sensor position. Once a submap contains N scans, it is considered complete. N is a design parameter. A radius outlier filter reduces measurement noise by removing all points with fewer than a predefined number of neighbors within a certain radius. The submap is then aligned with a history of H previous submaps via ICP. Since the odometry component provides good initial values for the transformation, only point pairs with a distance below 2 m are considered in the ICP objective function (Eq. (6)). The relative pose estimate is rejected if the mean squared error after the final iteration exceeds a threshold. We use the Hessian method [17] for estimating the uncertainty of the resulting relative pose:

$$\text{Var}(\Phi^*) \approx \frac{J(\Phi^*)}{n-3} \left[\frac{1}{2} \mathbf{H} \right]_{\Phi^*}^{-1} \quad (7)$$

Here, J is the ICP objective function, n the number of pairs, $\Phi^* = [\mathbf{t}^*, \theta^*]^\top$ the ICP result after the final iteration and \mathbf{H} the Hessian of J w.r.t. to Φ .

D. Loop Closing

Recognizing previously visited places to close driven loops is key to reducing the drift that inevitably accumulates when using incremental pose estimation methods such as odometry and consecutive scan matching. A brute-force approach, i.e. matching the current with all previous submaps, quickly becomes computationally infeasible as the number of submaps grows. Instead, our system detects loops by applying Geometrical Landmark Relations (GLARE) [18], a technique for place recognition with 2D Lidar point clouds, to Radar scans and subsequently running a series of tests to

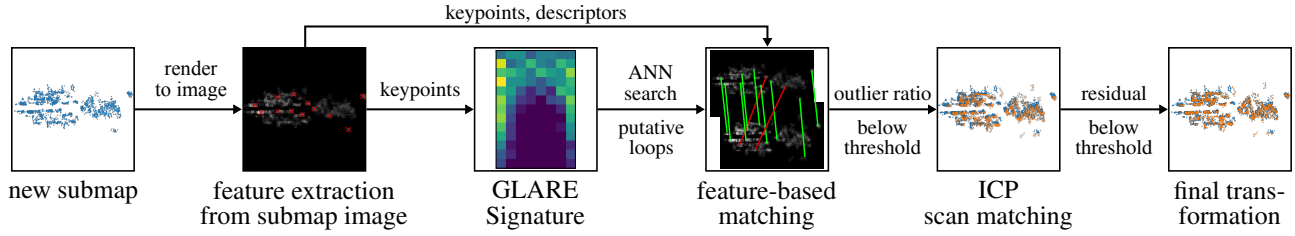


Fig. 5: Steps of the place recognition and loop closing process

confirm potential loops (see Fig. 5). GLARE extracts a set of keypoints from the scan and encodes their pairwise spatial relations, i.e. the Euclidean distances and bearings between each pair of points, by assigning them to histogram bins. The resulting histogram matrix serves as a scan signature and can be compared to other GLARE signatures via the L1-norm. In [18], the authors use Fast Laser Interest Region Transform (FLIRT) features [19] for keypoint extraction which are designed for 2D range data. However, FLIRT is not applicable for keypoint extraction from Radar since it is possible to detect occluded objects, i.e. obtain multiple range readings for a single azimuth angle. Instead, we render the Radar point cloud to a grayscale image and extract image features. The image is created by overlaying the point cloud with a pixel grid of predefined resolution in pixels per meter. Each point contributes to the intensity of the four nearest pixels by means of bilinear interpolation. The final image is the sum of all contributions divided by the total number of points and essentially represents the density distribution of the point cloud. For keypoint and descriptor extraction, we choose accelerated KAZE (AKAZE) features due to their rotation invariance and robustness to noise [20]. Furthermore, AKAZE uses binary descriptors that allow for fast calculation of correspondences. The keypoints are used to calculate GLARE signatures of each new submap. As suggested in [18], we utilize Approximate Nearest Neighbor Search (ANN) for fast retrieval of GLARE matches. For each new GLARE signature, the loop closer finds the K approximate nearest neighbors. If the L1 norm between two signatures is below a predefined threshold, the two corresponding submaps are considered as a loop candidate pair. We exclude a fixed number of previous submaps from the search, so as to avoid finding loops between immediately consecutive submaps. In order to find the transformation between a candidate pair, MSAC is applied for feature-based matching of the two submap images. MSAC finds the largest set of feature correspondences that are consistent with a 2D rigid transformation. Loop candidates with a low inlier ratio are rejected. The transformations of the remaining pairs are then refined using their full point clouds for ICP matching. Again, matches are only accepted if the ICP residual is below a fixed threshold.

E. Pose Graph SLAM

All previously described components of the SLAM system result in estimates of relative transformations between two poses of the vehicle at different times. The goal of pose

graph SLAM is to obtain an optimal estimate of the vehicle trajectory given the relative measurements. The trajectory is represented by a set $\mathcal{X} = \{\mathbf{x}_0, \dots, \mathbf{x}_T\}$ of 2D poses with $\mathbf{x}_i = [x_i, y_i, \psi_i]^\top$. The set \mathcal{Z} contains independent measurements \mathbf{z}_{ij} that describe relative transformations from \mathbf{x}_i to \mathbf{x}_j . The pose graph is a graphical representation of the unnormalized posterior distribution $p(\mathcal{X} | \mathcal{Z})$ in which each node corresponds to a pose \mathbf{x}_i , while edges between nodes represent relative pose measurements \mathbf{z}_{ij} and their uncertainty. Pose graph optimization involves finding the maximum *a posteriori* (MAP) estimate \mathcal{X}^* :

$$\begin{aligned} \mathcal{X}^* &= \arg \max_{\mathcal{X}} p(\mathcal{X} | \mathcal{Z}) \\ &= \arg \max_{\mathcal{X}} \prod_{ij} p(\mathbf{z}_{ij} | \mathcal{X}) \prod_k p(\mathbf{x}_k) \end{aligned} \quad (8)$$

In the following, we assume Gaussian or uniform noise for the priors $p(\mathbf{x}_k)$ and Gaussian noise for the relative measurements, i.e.

$$p(\mathbf{z}_{ij} | \mathcal{X}) \sim \exp\left(-\frac{1}{2} \|\mathbf{z}_{ij} - (\mathbf{x}_j \ominus \mathbf{x}_i)\|_{\Sigma_{ij}}^2\right) \quad (9)$$

with the pose difference operator \ominus and the covariance matrix Σ_{ij} . For this special case, Eq. (8) can be solved as a nonlinear least squares problem:

$$\begin{aligned} \mathcal{X}^* &= \operatorname{argmin}_{\mathcal{X}} -\ln(p(\mathcal{X} | \mathcal{Z})) \\ &= \operatorname{argmin}_{\mathcal{X}} \sum_{ij} \|\mathbf{e}_{ij}\|_{\Sigma_{ij}}^2 + \sum_k \|\mathbf{e}_k\|_{\Sigma_k}^2 \\ &\approx \operatorname{argmin}_{\mathcal{X}} \sum_{ij} \rho_{ij} \left(\|\mathbf{e}_{ij}\|_{\Sigma_{ij}} \right) + \sum_k \rho_k \left(\|\mathbf{e}_k\|_{\Sigma_k} \right). \end{aligned} \quad (10)$$

where ρ_{ij} and ρ_k are robust kernels that reduce the influence of outliers, and $\|\mathbf{e}\|_{\Sigma}^2 = \mathbf{e}^\top \Sigma^{-1} \mathbf{e}$. The errors \mathbf{e}_{ij} and \mathbf{e}_k denote the difference between measurement and estimation:

$$\mathbf{e}_{ij} = \mathbf{z}_{ij} - (\mathbf{x}_j \ominus \mathbf{x}_i), \quad \mathbf{e}_k = \mathbf{z}_k - \mathbf{x}_k \quad (11)$$

This lends itself to the intuitive interpretation that pose graph based SLAM finds the pose estimates that best explain the measurements. The graph is constructed from the relative transformations obtained by odometry and scan matching. Since ICP does not always converge to the global minimum, we use Cauchy kernels to reduce the influence of suboptimal matches [21]. Another problem arises from falsely detected loops. Although most false positives are rejected by the loop closer, even a single false loop can lead to an inconsistent map and large errors in the pose estimates. To prevent this, our

system uses Dynamic Covariance Scaling (DCS) kernels [22] that are specifically designed to reject false loop closures. For SLAM, it is usually assumed that the system has no prior knowledge about the environment except for the starting pose \mathbf{x}_0 . This means that all other priors $p(\mathbf{x}_1)$ to $p(\mathbf{x}_T)$ have uniform distributions and do not affect the minimization of the objective in Eq. 10. However, in many real-world scenarios, GNSS measurements are available that can reduce the uncertainty of the pose estimation. Our system thus optionally considers GNSS position measurements by including Gaussian priors on the appropriate poses in the objective function. For graph optimization, we use iSAM2 [23], which enables incremental graph updates in real time.

IV. EXPERIMENTS

A. Vehicle and Sensor Setup

All experiments were carried out using a 2008 Honda Accord, where the gyroscope, steering wheel, and wheel speed sensors were logged via the vehicle's Controller Area Network (CAN) bus. A 77 GHz automotive Radar is mounted at the front of the vehicle facing forward. The Radar has a measurement range of 100 m, with a point target range accuracy of 0.1 m and a range resolution of <0.5 m. It measures the azimuth angle for up to 60° with a resolution of up to 3.2° to 12.3° . A GeneSys ADMA-G-Pro+ equipped with Real-Time Kinematic (RTK) positioning was used to obtain a ground truth global position estimate with an accuracy of up to 2 cm.

B. Test Drives

Three test drives were recorded in and around the city of Darmstadt, Germany. The tests were designed in order to differ with respect to speed, driven distance, number of loops, amount of moving objects such as pedestrians and other vehicles, and the density of surrounding buildings.

1) *Lichtwiese*: The dataset recorded at the Lichtwiese campus of TU Darmstadt contains only a few moving objects such as pedestrians and cyclists. It comprises a short 700 m track where a large, tall building is circled twice in the same direction at 10 km/h to 20 km/h. The purpose of this scenario is to serve as a benchmark for the SLAM algorithm and in particular to test the loop closer's ability to detect the loop in a simple scenario.

2) *Paulusviertel*: This dataset has a total length of 1.1 km and was recorded in an urban, residential area at slow speed (10 km/h to 20 km/h). It contains three loops with the same driving direction and few moving objects. The scenario is challenging from the Radar perspective, as a large number of buildings and obstacles, such as parked cars, cause a high number of false targets due to multi-path reflections.

3) *Darmstadt - Roßdorf*: The rural road from Darmstadt to Roßdorf was driven twice in the same direction with the same start and end point at mid to high speeds (70 km/h to 100 km/h). The total covered distance is 13 km (6.5 km per round). The purpose of this scenario was to test the SLAM algorithm on a longer track comprising a multi-lane rural road in moderate traffic conditions. Also, we have not observed

an evaluation of previously reported Radar SLAM algorithms on longer tracks of similar length.

V. RESULTS

We use the Absolute Trajectory Error (ATE) as defined in [24] to evaluate our SLAM system. It calculates a measure for translational and rotational error at the trajectory level with respect to an appropriate ground truth estimate from the GNSS device. Its translational component is essentially the Euclidean distance between two trajectories and is particularly suitable for evaluating the global consistency of the localization and mapping result. All test runs were evaluated offline on a desktop computer with a contemporary hardware specification with an average computation time of 34.6 ms per frame. Real-time requirements are satisfied as the measurement update rate is about 70 ms.

A. Odometry and Trajectory estimates

The odometry as estimated by the UKF that serves as input for the SLAM algorithm is shown in Fig. 6, which also depicts the trajectories obtained from the SLAM algorithm and from ground truth. It can be seen that notable drift occurs only for the Roßdorf track, while the result for both the Lichtwiese and Paulusviertel tracks show only minor drift. A drift in odometry is expected due to sensor noise and offsets, model assumptions and simplifications, and numerical errors from integration. The nearly drift-free characteristics of the SLAM estimate stresses the successful loop detection and closing.

TABLE I: ATE results for the proposed SLAM

	Translational ATE in m			Rotational ATE in $^\circ$		
	mean	max	RMSE	mean	max	RMSE
Lichtwiese						
Odometry	1.05	2.41	1.18	1.18	3.23	1.31
SLAM	0.64	1.04	0.66	0.56	2.07	0.71
SLAM+GNSS	0.79	1.19	0.84	0.45	1.95	0.62
Paulusviertel						
Odometry	2.35	6.16	2.82	1.53	4.25	1.82
SLAM	1.04	1.93	1.15	0.61	2.59	0.72
SLAM+GNSS	0.62	1.76	0.69	0.31	2.85	0.45
Roßdorf						
Odometry	34.22	126.69	48.54	1.17	3.70	1.44
SLAM	9.52	14.84	10.35	0.35	2.21	0.47
SLAM+GNSS	4.94	11.63	5.86	0.34	1.61	0.43

B. SLAM

The quantitative distribution of the translational error by the SLAM system for each test drive is given in Fig. 7 and the full ATE results are stated in Table I. As outlined in [24], we emphasize that the rotational portion of ATE is only of minor significance, but is given for completeness. All values in the following refer to the mean translational portion of the ATE. The best performance, with an error of 0.64 m, was achieved on the Lichtwiese track. Incorporating GNSS, however, did not lead to an improvement as its signal was partly obscured from surrounding buildings. On the Paulusviertel track, the mean error settles at 1.04 m and incorporating GNSS, having

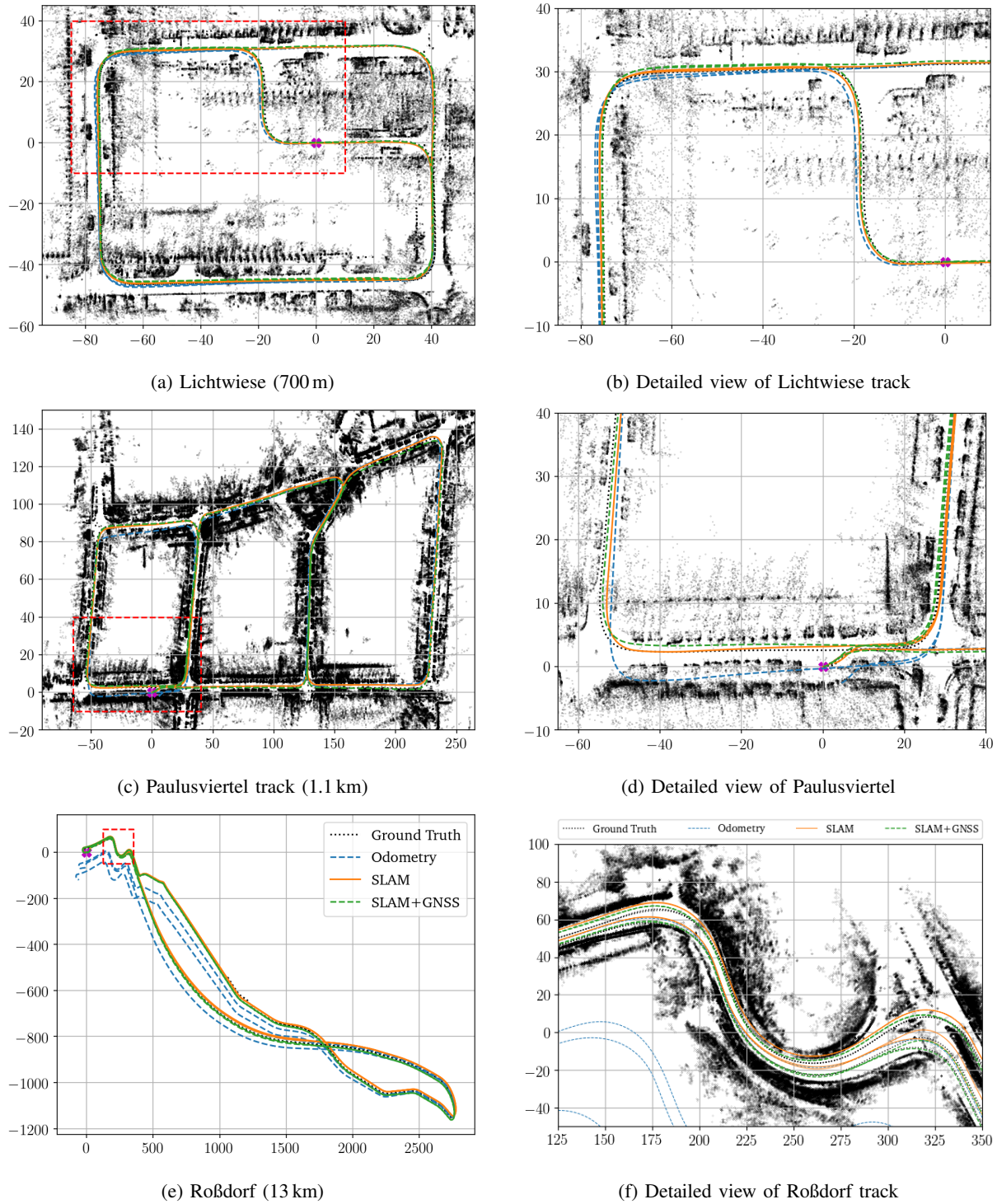


Fig. 6: Odometry analysis with origin (0,0) as starting and end point. Longitudinal and lateral distances are given in meter.

good reception, improves the result to 0.62 m. Lastly, the Roßdorf track showed higher translational errors of 9.52 m, but incorporating GNSS reduces the error to 4.94 m. Again, the GNSS signal was obscured during the parts of the track that lead to a forest, causing a high position variance, which is reflected in a higher maximum in the observed translational error. This test, however, demonstrates the robustness of our

SLAM algorithm on longer tracks. Also, a slight relation between track length and resulting translational error can be observed, but additional experiments must be carried out to verify this hypothesis.

VI. CONCLUSION

This paper provides a SLAM algorithm for Radar sensors that combines state-of-the-art concepts of pose graph SLAM

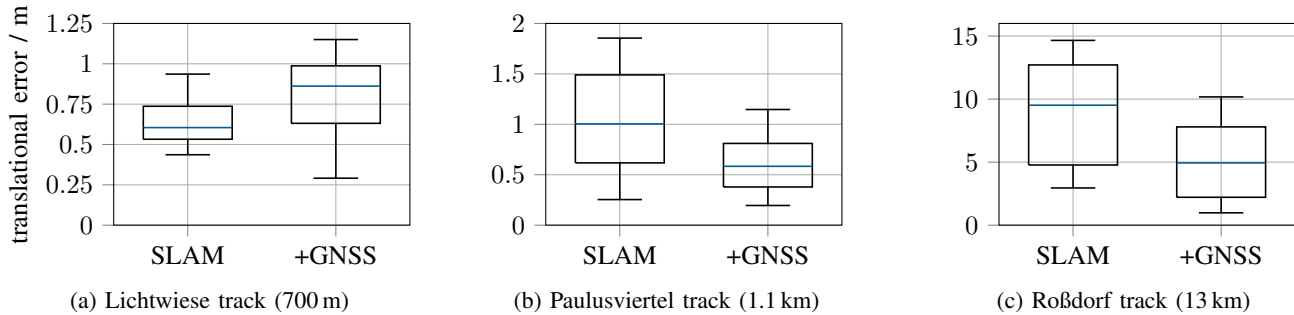


Fig. 7: Quantitative distribution of the translational error by the SLAM system during test drives. The mean, midspread are shown and the whiskers denote the 5% and 95% percentile.

and scan-point matching with ICP, where odometry information is inferred from an UKF. Experimental results obtained in three different real-world road scenarios demonstrate high accuracy with a mean translational ATE error as low as 0.62 m, while real-time capability, stability, and robustness of the algorithm are maintained on long tracks. A weakness of our algorithm is its dependency on many parameters, which require careful fine-tuning. Based on our results, we identified the potential of Radar sensors for SLAM applications. With the availability of higher range and angular resolutions, we expect Radar SLAM to establish itself as a cost efficient and robust alternative to Lidar and camera SLAM. In further work, we investigate potential benefits by adding additional Radars: A backwards-facing Radar would allow for the detection of loops regardless of driving direction and a second front-facing Radar would allow for the estimation of lateral velocity, which is unobservable with the current setup.

ACKNOWLEDGMENT

This work is part of the *aDDa 4 Students - Automated driving for Students at TU Darmstadt* initiative. The authors would like to thank Dorothea Koert and Philipp Rosenberger for fruitful discussions and gratefully acknowledge the helpful comments received from anonymous reviewers.

REFERENCES

- [1] S. Clark and H. Durrant-Whyte, "Autonomous land vehicle navigation using millimeter wave radar", in *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*.
- [2] M. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem", *IEEE Transactions on Robotics and Automation*, vol. 17, no. 3, pp. 229–241, Jun. 2001.
- [3] P. Checchin, F. Gérossier, C. Blanc, R. Chapuis, and L. Trassoudaine, "Radar Scan Matching SLAM Using the Fourier-Mellin Transform", in *Springer Tracts in Advanced Robotics*, Springer Berlin Heidelberg, 2010, pp. 151–161.
- [4] M. Schoen, M. Horn, M. Hahn, and J. Dickmann, "Real-Time Radar SLAM", in *11. Workshop Fahrerassistenzsysteme und automatisiertes Fahren*, Uni-DAS e. V., 2017, pp. 1–11.
- [5] F. Schuster, M. Worner, C. Keller, M. Haueis, and C. Curio, "Robust Localization based on Radar Signal Clustering", in *2016 IEEE Intelligent Vehicles Symposium*.
- [6] F. Schuster, C. G. Keller, M. Rapp, M. Haueis, and C. Curio, "Landmark based Radar SLAM using Graph Optimization", in *2016 IEEE International Conference on Intelligent Transportation Systems*.
- [7] S. H. Cen and P. Newman, "Precise Ego-Motion Estimation with Millimeter-Wave Radar Under Diverse and Challenging Conditions", in *2018 IEEE International Conference on Robotics and Automation*.
- [8] E. Mendes, P. Koch, and S. Lacroix, "ICP-based pose-graph SLAM", in *2016 IEEE International Symposium on Safety, Security, and Rescue Robotics*.
- [9] J.-E. Deschaud, *IMLS-SLAM: scan-to-model matching based on 3D data*, 2018. eprint: arXiv:1802.08633.
- [10] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time Loop Closure in 2D LIDAR SLAM", in *2016 IEEE International Conference on Robotics and Automation*.
- [11] E. Ward and J. Folkesson, "Vehicle localization with low cost radar sensors", in *2016 IEEE Intelligent Vehicles Symposium*.
- [12] C. Cadena, L. Carlone, H. Carrillo, et al., "Past, Present, and Future of Simultaneous Localization And Mapping: Towards the Robust-Perception Age", *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.
- [13] D. Kellner, M. Barjenbruch, J. Klappstein, J. Dickmann, and K. Dittmayer, "Instantaneous ego-motion estimation using Doppler radar", in *2013 IEEE International Conference on Intelligent Transportation Systems*.
- [14] P. Torr and A. Zisserman, "MLESAC: A New Robust Estimator with Application to Estimating Image Geometry", *Computer Vision and Image Understanding*, vol. 78, no. 1, pp. 138–156, Apr. 2000.
- [15] J.-A. Ting, E. Theodorou, and S. Schaal, "A kalman filter for robust outlier detection", in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*.
- [16] P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, Feb. 1992.
- [17] O. Bengtsson and A.-J. Baerveldt, "Robot localization based on scan-matching estimating the covariance matrix for the IDC algorithm", *Robotics and Autonomous Systems*, vol. 44, no. 1, pp. 29–40, Jul. 2003.
- [18] M. Himstedt, J. Frost, S. Hellbach, H.-J. Bohme, and E. Maehle, "Large scale place recognition in 2D LIDAR scans using Geometrical Landmark Relations", in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*.
- [19] G. D. Tipaldi and K. O. Arras, "Flirt - interest regions for 2d range data", in *2010 IEEE International Conference on Robotics and Automation*.
- [20] P. Alcantarilla, J. Nuevo, and A. Bartoli, "Fast Explicit Diffusion for Accelerated Features in Nonlinear Scale Spaces", in *Proceedings of the British Machine Vision Conference 2013*.
- [21] P. W. Holland and R. E. Welsch, "Robust regression using iteratively reweighted least-squares", *Communications in Statistics - Theory and Methods*, vol. 6, no. 9, pp. 813–827, 1977.
- [22] P. Agarwal, G. D. Tipaldi, L. Spinello, C. Stachniss, and W. Burgard, "Robust map optimization using dynamic covariance scaling", in *2013 IEEE International Conference on Robotics and Automation*.
- [23] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "iSAM2: Incremental smoothing and mapping using the bayes tree", *The International Journal of Robotics Research*, vol. 31, no. 2, pp. 216–235, Dec. 2011.
- [24] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-d SLAM systems", in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*.