

# Radar-Based Localization For Autonomous Ground Vehicles In Suburban Neighborhoods

---

**Andrew J. Kramer**  
Amazon, LLC  
1988kramer@gmail.com

**Christoffer Heckman**  
University of Colorado, Boulder  
christoffer.heckman@colorado.edu

## Abstract

For autonomous ground vehicles (AGVs) deployed in suburban neighborhoods and other human-centric environments the problem of localization remains a fundamental challenge. There are well established methods for localization with GPS, lidar, and cameras. But even in ideal conditions these have limitations. GPS is not always available and is often not accurate enough on its own, visual methods have difficulty coping with appearance changes due to weather and other factors, and lidar methods are prone to defective solutions due to ambiguous scene geometry. Radar on the other hand is not highly susceptible to these problems, owing in part to its longer range. Further, radar is also robust to challenging conditions that interfere with vision and lidar including fog, smoke, rain, and darkness. We present a radar-based localization system that includes a novel method for highly-accurate radar odometry for smooth, high-frequency relative pose estimation and a novel method for radar-based place recognition and relocalization. We present experiments demonstrating our methods' accuracy and reliability, which are comparable with other methods' published results for radar localization and we find outperform a similar method as ours applied to lidar measurements. Further, we show our methods are lightweight enough to run on common low-power embedded hardware with ample headroom for other autonomy functions.

## 1 Introduction

Deploying autonomous ground vehicles (AGVs) in human-centric environments like sidewalks and multi-use paths remains a significant challenge. High accuracy localization is essential because the difference between the robot driving safely on the sidewalk and drifting into traffic is often just tens of centimeters. Unfortunately, GPS does not meet this accuracy requirement when it is available, and of course one can't assume GPS will be available at all times. Many methods have been published addressing the problem of global localization using cameras and lidar. However, even in ideal conditions these sensors have their weaknesses. Visual odometry is susceptible to low light and motion blur while visual place recognition still has significant problems with appearance changes due to weather, lighting changes, and seasonal changes. Lidar-based methods, on the other hand, are prone to defective solutions when presented with ambiguous scene geometry. These problems may be dismissed as edge cases, but even in a modest-sized fleet of AGVs these edge cases occur quite frequently.

Millimeter-wave radar has the potential to sidestep many of the issues encountered other localization methods. It is not affected by visual challenges like darkness, fog, or smoke. Further, radar is not affected by ambiguous scene geometry because it is able to directly measure motion and sense material properties. Despite these advantages, though, radar has seen limited adoption in the robotic localization literature. Many

of the existing methods use the NavTech scanning radar sensor including (Cen and Newman, 2018; Cen and Newman, 2019; Barnes and Posner, 2020). (Cen and Newman, 2018; Cen and Newman, 2019) focus on the odometry problem, (Hong et al., 2020) proposes a SLAM method, and (Barnes and Posner, 2020) adds place recognition and relocalization. Similar works focusing on scanning radars for place recognition (Kim et al., 2020; Kim et al., 2021) have shown remarkable performance, approaching that of lidar-based place recognition. Unfortunately, the NavTech sensor all of these methods depend on is large, power-hungry, and expensive.

Automotive-grade millimeter-wave radar sensors are small, low-power, and inexpensive, making them a better option for AGVs. (Schuster et al., 2016b) uses automotive radar sensors to create maps by using stream clustering to identify landmarks in radar scans. They can then reliably align radar scans to those maps for global localization. (Schuster et al., 2016a) aggregates radar detections into pseudo-images and uses the BASD feature detector (Rapp et al., 2016) to detect and match features in those pseudo images. (Kramer et al., 2020) fuses radar and IMU measurements to estimate a vehicle’s velocity and orientation in 3D, and (Kramer and Heckman, 2021) extends the method by adding a radar-based occupancy mapping method. The method of (Kramer and Heckman, 2021) is similar to (Schuster et al., 2016a) in that they aggregate radar detections into discretized maps, but (Kramer and Heckman, 2021) uses a learned noise filter based on PointNet (Qi et al., 2016) to remove spurious radar detections and they use a 3D octomap (Hornung et al., 2013) rather than 2D dense pseudo images. However, (Kramer and Heckman, 2021) only uses their radar-based map for obstacle detection and does not attempt to localize against it. (Doer and Trommer, 2020b) and (Doer and Trommer, 2020a) present a similar radar-inertial odometry (RIO) method using a Doppler-based radar constraint. A fault with all of these Doppler-based methods is they only use gyroscope measurements to constrain the robot’s heading estimate, so their heading estimate drifts severely over longer trajectories. (Doer and Trommer, 2021) propose to get around this issue by making the restrictive assumption that the robot is operating in a Manhattan (grid) world. Other techniques (Kung et al., 2021; Lu et al., 2020) have demonstrated impressive performance, but the former does not include an IMU which would likely improve performance, while the latter treats the IMU with a neural network. There are many ongoing research directions related to the use of automotive radar being developed continuously (Harlow et al., 2023), some of which also include loop closure detection with federated back-ends (Zhang et al., 2023) and are resistant to dynamic obstacles and map elements (Chen et al., 2023). To the author’s knowledge, so far no method has solved the problem of a general heading constraint in RIO using low-cost automotive sensors.

Our contributions beyond these works are as follows:

1. A method for RIO that includes a novel method for dynamic outlier rejection and a unique radar-based heading constraint.
2. A method for creating radar-based maps that effectively removes noise and preserves repeatable environmental features.
3. A method for estimating the robot’s global pose by registering local-to-global radar maps with accuracy and reliability comparable to lidar-based methods.

## 2 Radar-Inertial Odometry

Our RIO method takes inputs from an arbitrary array of synchronized radar sensors and a single IMU. It estimates the robot’s state in 6DoF using a nonlinear optimization over a sliding window of recent states and measurements implemented with iSAM2 (Kaess et al., 2011). The overall structure of the problem is shown as a factor graph in Figure 1. In subsequent sections we give further detail on the constraints used in the optimization problem including our dynamic object detection method and our Doppler, IMU, and heading constraints. However, we’ll first give a brief overview of the notation used in subsequent sections and defined the estimated states and sensor measurements.

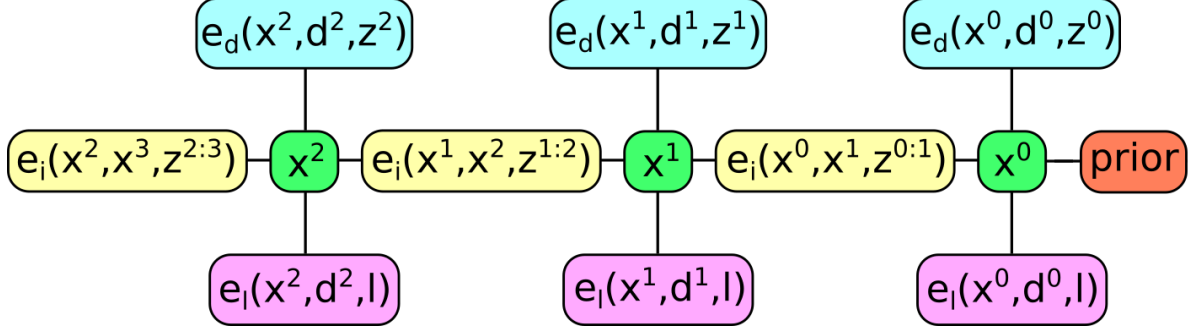


Figure 1: Factor graph used for the RIO estimator, with increasing time going left. The Doppler and Landmark errors  $e_d$  and  $e_l$  provide constraints on the state that results in a nonlinear minimization problem over these errors by adjusting the state variables  $x^i$ . The states are connected through binary constraints  $e_i$  induced by IMU measurements. A prior is formed from marginalized states prior to  $x^0$  which have fallen off of the sliding window, thereby maintaining a constant-time algorithm.

## 2.1 Notation, State, and Measurements

We note scalars in lower case as in  $r$ , vectors in lower case bold as in  $\mathbf{v}$ , and matrices in upper case bold as in  $\mathbf{R}$ . The frame in which a quantity is expressed is noted as a pre-superscript. For instance,  ${}^I\mathbf{p}$  denotes vector  $\mathbf{p}$  expressed in frame  $I$ . For velocities, the pre-superscript notes both the frame for and in which velocity is being expressed; note that velocities are always relative to a global reference frame. As an example,  ${}^R\mathbf{v}$  is the velocity of frame  $R$  relative to the global reference frame expressed in frame  $R$ . We represent rotations as either 3D rotation matrices  $\mathbf{R}$  or quaternions  $\mathbf{q}$ . Rotation frames are noted as pre-scripts. So  ${}^I_R\mathbf{q}$  is the quaternion describing the rotation from frame  $R$  to frame  $I$ . The frames used in this work include  $O$ , the global reference frame;  $I$ , the IMU reference frame; and  $R$  the frame of a radar sensor. Lastly, estimated quantities are noted with a  $\hat{\cdot}$ .

The proposed system optimizes a sliding window of  $N$  robot states  $\mathbf{X} = [\mathbf{x}^0, \dots, \mathbf{x}^N]$  where the state at timestep  $k$  is  $\mathbf{x}^k = [{}^O\mathbf{v}^T, {}^O_I\mathbf{q}^T, \mathbf{b}_a^T, \mathbf{b}_g^T]^T$ , the robot's velocity, orientation, and IMU biases. The measurement at a single timestep from a radar sensor is referred to as a scan, and a scan consists of a set of detections  $\mathcal{D}$ . Each detection represents a target detected by the CFAR (Scharf and Demeure, 1991) detector running onboard the radar sensor and detections are parameterized as  $\mathbf{d} = [{}^R\mathbf{p}^T, {}^R\dot{r}]^T$ , where  $\mathbf{p} \in \mathbb{R}^3$  refers to the 3-dimensional vector between the radar sensor and the detection, and  $\dot{r} \in \mathbb{R}$  refers to the Doppler measurement along the vector (i.e., the velocity of the detection along the ray cast between the sensor and the target). A measurement from the IMU is parameterized as  $\mathbf{z} = [{}^I\mathbf{a}^T, {}^I\boldsymbol{\omega}^T]^T$  where  ${}^I\mathbf{a}$  and  ${}^I\boldsymbol{\omega}$  are the acceleration and angular velocity in the IMU frame respectively.

## 2.2 Dynamic Object Rejection

A crucial part of our RIO system is our method for detecting and discarding radar detections which represent dynamic objects like cars, cyclists, and pedestrians. A common way to do this across many sensor types is random sample consensus (RANSAC) (Fischler and Bolles, 1981). There is a simple linear relationship between a static detection's range rate to the sensor's velocity,  ${}^R\dot{r} = {}^R\mathbf{v} \cdot \frac{{}^R\mathbf{p}}{\|{}^R\mathbf{p}\|}$  and it is easily incorporated into a RANSAC model to solve the problem:

$${}^R\hat{\mathbf{v}} = \underset{{}^R\mathbf{v}}{\operatorname{argmin}} \left( \sum_{\mathbf{d} \in \mathcal{D}} {}^R\dot{r}_{\mathbf{d}} - {}^R\mathbf{v} \cdot \frac{{}^R\mathbf{p}_{\mathbf{d}}}{\|{}^R\mathbf{p}_{\mathbf{d}}\|} \right)^2 \quad (1)$$

The above formulation has been used in other radar odometry methods including (Kramer et al., 2020). It can reliably differentiate between static and dynamic objects in a scan from a single sensor and provide a good initial guess at that sensor’s velocity. Using this model with multiple synchronized sensors would require running a separate RANSAC problem for each sensor, but this presents a problem. If a scan from one sensor does not contain a sufficient number of inliers then that scan must be discarded entirely, even if there is a sufficient number of inliers from all the sensors combined. Further, this method results in different estimates of the local frame velocities for each sensor  ${}^{R_s}\mathbf{v}$  but it would be more useful to estimate a single IMU frame velocity  ${}^I\mathbf{v}$ .

To address these issues we consider all detections from each sensor in the same RANSAC problem. To do this we first transform the detection’s location in the radar frame to the IMU frame as  ${}^I\mathbf{p} = {}^{I}\mathbf{R}{}^{R_s}\mathbf{p} + {}^{I}\mathbf{t}$  where  ${}^{I}\mathbf{R}$  is the rotation from the frame of radar sensor  $s$  to the IMU frame and  ${}^{I}\mathbf{t}$  is the translation. We then offset the range rate measurement with the velocity induced by the lever-arm between the radar sensor and IMU as

$${}^I\dot{r} = {}^{R_s}\dot{r} + \left( \left( {}^{R_s}{}^I\mathbf{t} \right)^\times {}^{R_s}{}^I\mathbf{R} {}^I\boldsymbol{\omega} \right) \cdot \frac{{}^{R_s}\mathbf{p}}{\|{}^{R_s}\mathbf{p}\|} \quad (2)$$

where  $(\cdot)^\times$  returns the skew-symmetric matrix corresponding to a given 3D vector. This allows us to process all of the detections from each radar sensor in the same RANSAC problem as

$${}^I\hat{\mathbf{v}} = \underset{{}^I\mathbf{v}}{\operatorname{argmin}} \left( \sum_{s \in \mathcal{S}} \sum_{\mathbf{d} \in \mathcal{D}_s} {}^I\dot{r}_{\mathbf{d}} - {}^I\mathbf{v} \cdot \frac{{}^I\mathbf{p}_{\mathbf{d}}}{\|{}^I\mathbf{p}_{\mathbf{d}}\|} \right)^2 \quad (3)$$

where  $\mathcal{S}$  is the set of radar scans and  $\mathcal{D}_s$  is the set of detections from radar sensor  $s$ . Outliers identified in solving this RANSAC problem are assumed to represent dynamic objects and are discarded.

### 2.3 Doppler Constraint

After discarding detections identified as dynamic in the RANSAC step, radar detections are used to create constraints on the odom frame velocity  ${}^O\mathbf{v}$ . The Doppler error function is

$$e_d(\hat{\mathbf{x}}^k, \mathbf{d}, \mathbf{z}^k) = {}^{R}\dot{r} - \left( {}^R\hat{\mathbf{R}}^k {}^O\hat{\mathbf{v}}^k + \left( {}^R{}^I\mathbf{t} \right)^\times {}^R{}^I\mathbf{R} \left( {}^I\boldsymbol{\omega}^k - \hat{\mathbf{b}}_g^k \right) \right) \cdot \frac{{}^R\mathbf{p}}{\|{}^R\mathbf{p}\|} \quad (4)$$

where  ${}^R\hat{\mathbf{R}}^k = {}^R\mathbf{R}{}^I\hat{\mathbf{R}}^k$  and  ${}^I\hat{\mathbf{R}}^k$  is the global frame to IMU frame rotation, which is estimated as part of the state vector; and  ${}^R{}^I\mathbf{R}$  is the IMU frame to radar frame rotation, which is constant. An IMU measurement interpolated to match the timestamp of the radar measurement  $\mathbf{z}^k$  is also required to estimate the effect of the sensor’s angular rate on the Doppler measurement.

### 2.4 IMU Constraint

In our system, IMU measurements are used to measure the system’s change in global-frame velocity and orientation between radar measurements. Our IMU model is very similar to that provided in GTSAM

(Forster et al., 2015), except we do not use the IMU to measure the change in the system’s full pose, only its velocity and orientation.

The IMU provides measurements at many times the rate of the radar sensors and the IMU and radar measurements are not synchronized. To address this we first interpolate between the IMU measurements immediately before and after the radar measurements to obtain estimated IMU readings that align temporally with the radar measurements at timesteps  $k$  and  $k + 1$ . We then use the method for on-manifold IMU pre-integration presented in (Forster et al., 2015) to combine the set of IMU measurements between times  $k$  and  $k + 1$  into a single relative constraint which is used to predict the state  $\mathbf{x}^{k+1}$  from  $\hat{\mathbf{x}}^k$ . The IMU error is then defined as the difference between the estimated state  $\hat{\mathbf{x}}^{k+1}$  and the IMU propagated state

$$\mathbf{e}_i(\hat{\mathbf{x}}^k, \hat{\mathbf{x}}^{k+1}, \mathbf{z}^{k:k+1}) = \begin{bmatrix} 2 \left[ \begin{array}{c} {}^O_I \hat{\mathbf{q}}^{k+1} \otimes ({}^O_I \mathbf{q}^{k+1})^{-1} \\ {}^O_{\hat{\mathbf{v}}}^{k+1} - {}^O_{\mathbf{v}}^{k+1} \end{array} \right]_{1:3} \\ \hat{\mathbf{b}}_g - \mathbf{b}_g \\ \hat{\mathbf{b}}_a - \mathbf{b}_a \end{bmatrix} \quad (5)$$

where the  $\otimes$  operator is as defined in (Barfoot et al., 2011).

## 2.5 Heading Constraint

With only the Doppler and IMU constraints our state estimation problem is underdetermined. While the gyroscope biases in pitch and roll are constrained by the gravity direction, there is no constraint on the gyroscope bias in yaw. This will cause the system’s heading estimate to drift over longer trajectories. To address this we implement a novel means of constraining the system’s heading estimate by identifying and tracking persistent, repeatable landmarks in radar scans. This consists of two components: a unique method for associating new radar detections to existing landmarks, and a series of checks to determine if the confidence in a given landmark is high enough to be used to create constraints in the estimator.

To identify candidate matches between the set of new radar detections  $\mathcal{D}$  and tracked landmarks from the last timestep  $\mathcal{L}$  we first construct a dense matrix of the distances  $\mathbf{S} \in \mathbb{R}^{|\mathcal{D}| \times |\mathcal{L}|}$  between each detection  $\mathbf{d}$  and landmark  $\mathbf{l}$ . We use a unique weighted polar distance metric first proposed in (Minguez et al., 2006):

$$\begin{aligned} \phi(\mathbf{p}) &= \text{atan2}(\mathbf{p}^y, \mathbf{p}^x) \\ \mathbf{S}_{\mathbf{d}, \mathbf{l}} &= \left( L^2 (\phi({}^I \mathbf{p}_{\mathbf{d}}) - \phi({}^I \mathbf{p}_{\mathbf{l}}))^2 \right. \\ &\quad \left. + (\|{}^I \mathbf{p}_{\mathbf{d}}\| - \|{}^I \mathbf{p}_{\mathbf{l}}\|)^2 \right)^{\frac{1}{2}} \end{aligned} \quad (6)$$

This distance metric is calculated in polar coordinates and it allows us to control the relative importance of the azimuth and range dimensions by setting the coefficient  $L$ . This allows us to more accurately model the noise distribution of our radar sensors when matching new detections to existing landmarks.

This is the only appearance of the position of the Doppler targets  $\mathbf{p}_{\mathbf{d}}$  in our algorithm. This is because typical frame-to-frame tracking of these targets is generally not reliable over long durations due to the nature of noise affecting radar measurements. Landmarks are generally not consistently trackable for long traces in time, however they *are* very consistent through rotations. In fact, through rotation in yaw, radar targets are remarkably consistent, likely as a result of the radar cross section of the target not undergoing significant change compared with other types of motion with respect to hypothetical targets. We leverage this fact to support localization through rotation in yaw specifically.

At this point  $\mathbf{S}$  can be viewed as a weighted bipartite graph and a set of potential matches between detections and landmarks is found with the Hungarian algorithm (Munkres, 1957). A potential match is validated if the distance between the detection and landmark found by equation 6 is below a user-defined threshold. New detections for which no valid match is found are initialized as new landmarks. Tracked landmarks are discarded if the time elapsed since the last time the landmark was successfully matched to a new detection is above a threshold value.

Each valid match between a landmark and detection is not necessarily used to create a new constraint in the odometry estimator. We track two quantities for each landmark to determine if a new match should be used to create a new constraint. The number of times the landmark was previously observed must be above a threshold value and the maximum error in the observations of the landmark must be below a threshold value. If the matched landmark meets these criteria the matched pair  $[\mathbf{d}, \mathbf{l}]$  is added to the set of active matches  $\mathcal{S}$ . For each active match an error term of the form:

$$e_l(\hat{\mathbf{x}}^k, \mathbf{d}, \mathbf{l}) = \phi({}^I_R \mathbf{R}^R \mathbf{p}_d + {}^I_R \mathbf{t}) - \phi({}^I_O \hat{\mathbf{R}}^k {}^O \mathbf{p}_l + {}^I_O \mathbf{t}) \quad (7)$$

is added to the estimator. Note the landmark position  ${}^O \mathbf{p}_l$  and the IMU frame to global frame translation  ${}^O_I \mathbf{t}$  are not estimated, they are treated as constants.  ${}^O \mathbf{p}_l$  is set to the first observed position of the landmark and  ${}^O_I \mathbf{t}$  is found by integrating the estimated velocities  ${}^O \hat{\mathbf{v}}$ . This greatly simplifies the optimization and ensures the landmark error terms only influence the system’s orientation estimate  ${}^O_I \hat{\mathbf{q}}$ . Empirical testing showed estimating  ${}^O \mathbf{p}_l$  and  ${}^O_I \mathbf{t}$  as part of the system’s state actually has negative effects on the accuracy of the system’s odometry estimate. Since this method is inevitably paired with a technique that corresponds current radar measurements with a prior map, long-term longitudinal tracking introduces error that creates tension with the prior map tracking, and is therefore not utilized.

### 3 Radar Mapping and Matching

Our radar-inertial odometry method provides high-frequency, smooth relative motion estimates but does not include a method to constrain drift or localize the robot in a global frame. In the following sections we describe our method for solving these problems. This consists of two components: a method for constructing maps from radar detections described in Section 3.1 and a method for estimating the robot’s pose by registering local radar maps to global radar maps described in Section 3.2.

#### 3.1 Radar Mapping

The principal challenge in creating useful maps from radar data is in distinguishing spurious detections from those that represent real features of the environment. Spurious detections are very common and they arise from many sources including multipath reflections, reflections from dynamic objects, reflections from the ground, ringing from signal processing, and others. This results in a very low repeatability between radar scans. This is demonstrated in Figure 3a, which shows the raw detections logged over the course of a short mission transformed to a common reference frame. The image shows very little identifiable structure in the aligned detections, which seem to be distributed evenly over the whole environment.

An important characteristic of radar detections is not apparent from Figure 3a, however. As the robot moves through the environment true detections tend to appear in the same places while spurious detections tend to be spatially sparse. This means the structure of the environment becomes more apparent if the detections are binned into a spatial histogram where each voxel stores the number of detections that occur within it. Figure 3b shows one such histogram colored by the number of detections per voxel, where detections are summed into the spatial histogram at the frequency of the sensor. One can see the structure of the environment is

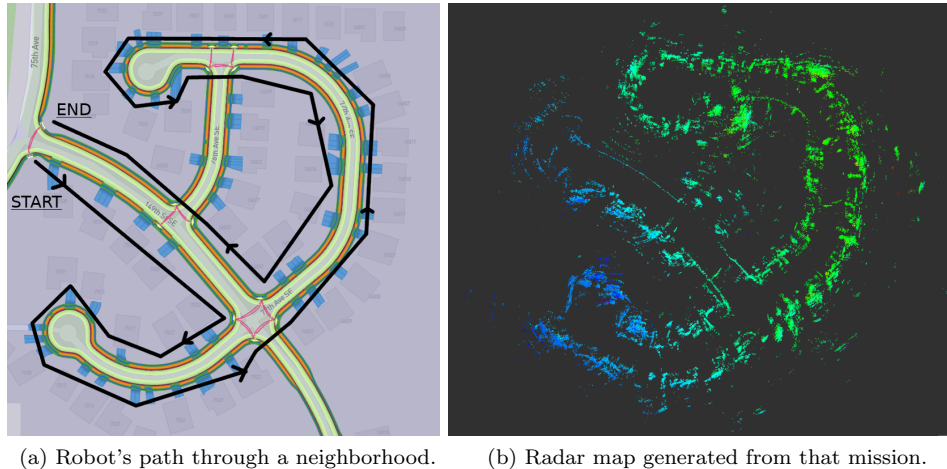
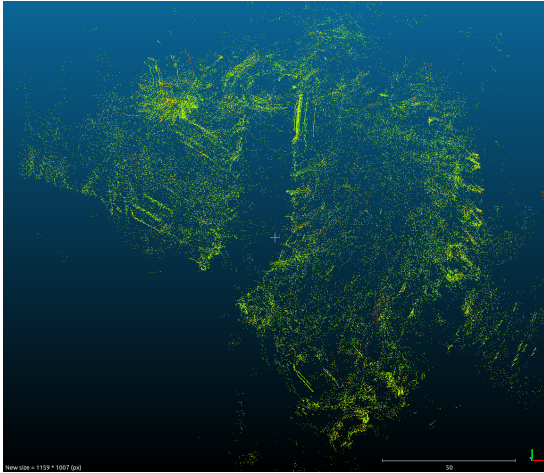


Figure 2: A neighborhood trajectory and single global map generated using the volumetric mapping method.

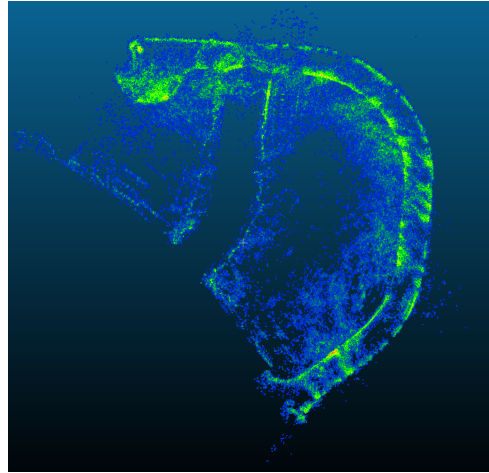
much more apparent in that image, with curved sidewalks and some primitive geometry of the neighborhood becoming clearer.

We take advantage of this property in our mapping method by first adding new radar detections to an occupancy grid map (OGM), which is a voxelized grid of 3D space parametrized by a probability from 0 to 1 which represents the probability that a specific voxel is occupied (Hornung et al., 2013). The overall OGM then has a tunable set of parameters, namely the likelihood threshold to consider a certain voxel “occupied,” and the probability of occupancy  $P(n|z_t)$  where  $n$  is a given voxel and  $z_t$  is the current measurement, either a detection or non-detection. A probabilistic fusion is then applied using the posterior  $P(n|z_{1:t-1})$  and an environmental prior  $P(n)$  which is commonly assumed to be 0.5 (a uniform prior probability); our approach to this fusion is identical to the method of (Moravec and Elfes, 1985; Hornung et al., 2013). We have found that through this method we can generate richly detailed radar maps that we have observed to remove map errors generally resulting from multipath reflections, ground reflections, and other radar-specific noise, while retaining the structure of the environment. An example OGM of a large area can be seen in Figure 2b. Figure 4 shows detail on a specific area of a radar OGM along with an aerial view of the same area to show the correspondence between features in the map and the real world.

Estimating the robot’s pose involves registering a small local map of the robot’s immediate surroundings, referred to as a query map, to a much larger map covering the whole area in which the robot is operating, referred to as a global map. The requirements for these two types of maps are different and the process for creating them differs as well. Creating a global map requires very high metric accuracy and global consistency over an area spanning whatever may be reasonably loaded onto the robot. In many applications, memory and runtime limitations drive this requirement, which may mean the map covers an area of 1–10 km<sup>2</sup>. However, such a large map might contain millions of points, resulting in high memory overhead at runtime. Therefore, we further subdivide the neighborhood maps into chunks, which tessellate the global map and cover a smaller area, approximately 250m<sup>2</sup>. The area to be mapped is traversed by a special mapping robot as shown in Figure 2b. The radar sensor arrangement on the mapping robot is identical to that on the production robots, but the mapping robot’s state estimation may differ; it may for instance be equipped with a high-accuracy GPS+INS system whose outputs are batch-optimized offline, or it may have a full data stream over a mission from which a fully bundle-adjusted trajectory may be resolved. In either case, as in many autonomous vehicle operations where prior maps are used for localization, the trajectory of the mapping robot must be centimeter-accurate in a global frame. Over the course of a mapping robot mission the detections from the robot’s three radar sensors are also logged. After the mission the optimized trajectory is used to transform the logged detections to a common frame of reference. These transformed detections are then consumed into an OGM; the global map is represented as the point centers of the occupied



(a) Raw radar detections from a short mission transformed into a global reference frame.



(b) The detections in Figure 3a binned into a spatial histogram colored by the number of detections per voxel.

Figure 3: Example of binning raw radar detections into spatial histograms to elucidate the structure of the environment.

cells of the OGM.

Query maps on the other hand must rely on only information available at hand to construct the OGMs, namely robot odometry, and therefore are subject to drift over time. Relative pose estimates from the RIO method described in 2 have sufficient accuracy for transforming radar detections to the frame of the query map over a limited scale. Since many automotive-grade radars have accuracy on the order of 10cm at 30m range, query maps are only collected over an axis-aligned bounding box of  $x, y \in 40\text{m} \times 40\text{m}$  centered at the robot’s current position. This clipping of data accomplishes two goals: first it limits the amount of odometric error that can infiltrate map generation, and second it bounds map size, since all voxels in the query map outside of this bounding box are cleared from memory. Note that this full node-traversal is required regardless, as the full data structure containing the map must be traversed in order to populate a point cloud for later consumption by map matching.

### 3.2 Radar Map Matching

The process for registering a query map to a global map proceeds in two stages during a mission. First, a “Full” comparison is performed, where coarse alignment is performed via a discretized brute-force search of possible alignments between the query and global maps. Then the best candidate alignment is refined using a smaller fine alignment, which also relies on brute-force searching, but terminates using the iterative closest point (ICP) method (Recherche et al., 1992). Second, once the “Full” search has concluded successfully, subsequent matching for later query maps may be performed using a “Tracking” comparison, where only the fine alignment and ICP steps are invoked. Since Tracking relies on a smaller number of comparisons and over a much smaller spatial scale, it concludes much more rapidly than the Full comparison, which generally is not required unless Tracking results in a failure to find a match.

During the coarse alignment phase, a discretized space of possible translations and rotations between the query map and global map are evaluated via brute-force search. For each possible alignment the points in the query map  $\mathbf{p}_q$  are associated to their nearest neighbor in the global map  $\mathbf{p}_g$  to create the set of matched points  $\mathcal{P}$ . A score  $s$  for each alignment is calculated as the the number of points in the query map that are within  $d$  of the global map:

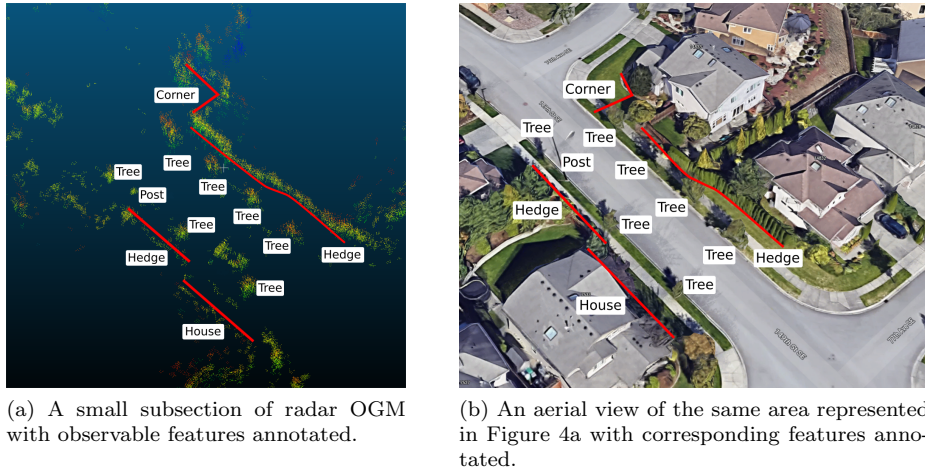


Figure 4: Detailed view of a radar-based OGM showing correspondences to an aerial view of the same area.

$$s = \sum_{\mathbf{p}_q, \mathbf{p}_g \in \mathcal{P}} \begin{cases} 1, & \text{if } \|\mathbf{p}_q - \mathbf{p}_g\| < d \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

where  $d$  is a threshold for a point to be an inlier.

Spatial pyramiding (Olson, 2015) is employed to make the brute-force search process more impervious to defective matches resulting from local minima of the cost surface induced by Eq. (8). The top  $K$  scoring alignments from pyramid level  $n$  are refined in pyramid level  $n + 1$  and the process is repeated until the highest-resolution pyramid level is reached. In our methodology we search within  $\pm\pi/2$  in yaw and  $\pm 5m$  in translation for the “Full” search, and in “Tracking” limit our search to  $\pm\pi/4$  and  $\pm 1m$  which we find to yield few losses of tracking, generally conditioned on the environment and the presence of map features to be matched.

The highest scoring alignment from the brute-force search process are then refined using ICP with the standard point-to-point metric. A score is calculated for the refined alignment using equation 8. If this score is equal to or above a threshold value, the alignment is considered to have been successful.

## 4 Experiments

We conducted a series of experiments evaluating the accuracy, reliability, and computational requirements of our odometry and relocalization methods. Both experiments used the same autonomous ground vehicle which was equipped with 3 automotive radars, a Lord Microstrain 3DM-GX5 IMU, and a high-accuracy Novatel GPS+INS system for groundtruth. The vehicle and sensor locations are detailed in Figure 5 and specifications of the radar sensors are detailed in Table 1. We acknowledge the existence of some excellent datasets for radar (Choi et al., 2023; Kramer et al., 2021; Barnes et al., 2020), each relying on their own radar sensor(s) to encourage the development of novel capabilities. In our case, our radar was a proprietary blend of hardware and signal processing techniques, limiting the capability of our testing on different datasets.

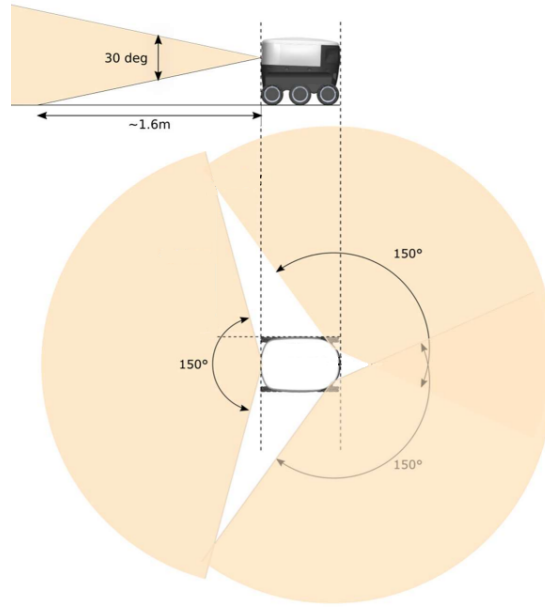


Figure 5: Fields of view of the robot’s radar sensors. Above a side view of the robot shows the rear sensor’s vertical field of view and below a top-down view shows all three sensors’ horizontal fields of view.

#### 4.1 Radar-Inertial Odometry Evaluation

To evaluate the RIO method we use four missions carried out in residential neighborhoods. Collectively, the missions cover approximately 12km in length and 4 hours in duration. We report statistics on the error accumulated by our RIO method over non-overlapping 10m subsegments of each trajectory using the GPS+INS system for groundtruth. We report the median (p50), 95th percentile (p95), 99th percentile (p99), and maximum drift for both translation and heading for each mission in Table 2.

With median translational drift as low as 0.013m/m and heading drift as low as  $0.021^\circ/\text{m}$ , our RIO method is impressively accurate. It is very lightweight as well. When running on an Intel Core i7-10870H processor our RIO method’s mean update time is just 0.86ms. On an ARM Cortex-A57 (a common low-power embedded processor) its mean update time is still only 4ms. Our radar sensor’s update period is 50ms, so even on low-power embedded hardware our RIO method leaves plenty of headroom for other processes.

##### 4.1.1 Ablation Study

To evaluate the importance of different components of the proposed RIO system we re-ran the system offline with different components de-activated. To evaluate the benefit of using multiple synchronized radar sensors we include an experiment in which only the front-left sensor is used. To evaluate the benefit of the heading constraint described in Section 2.5 we include an experiment in which the heading constraint is not used. Table 3 shows the results of these experiments.

We can see from Table 3 that using only one radar still performs reasonably well. The median and even the 95th percentile drift statistics for the single radar version are fairly close to the three radar version. The benefit to using multiple radar sensors seems to be in the tails of the error distribution because the 99th percentile and maximum drift statistics for the single radar version are far higher than the three radar version. The version with the heading constraint de-activated does not perform nearly as well, however. The translation drift statistics for the no-heading-constraint version are 3 to 5 times those for the three radar

max range	80m
range resolution	0.3m
range accuracy	0.03m
Doppler min/max	$\pm 30\text{m/s}$
Doppler resolution	0.4m/s
Doppler accuracy	0.04m/s
azimuth FOV	$\pm 75^\circ$
azimuth resolution	$2^\circ$
azimuth accuracy	$0.2^\circ$
elevation FOV	$\pm 15^\circ$
elevation resolution	$2.5^\circ$
elevation accuracy	$0.25^\circ$
Update Rate	20Hz
Frequency	76 – 77GHz

Table 1: Specifications of the radar sensors used in field experiments.

		Translation Drift (m/m)	Heading Drift ( $^\circ/\text{m}$ )
Mission 1	p50	0.013	0.021
	p95	0.027	0.084
	p99	0.052	0.254
	max	0.275	0.483
Mission 2	p50	0.017	0.037
	p95	0.043	0.123
	p99	0.064	0.194
	max	0.127	0.224
Mission 3	p50	0.015	0.033
	p95	0.035	0.105
	p99	0.042	0.162
	max	0.078	0.99
Mission 4	p50	0.017	0.034
	p95	0.041	0.119
	p99	0.050	0.176
	max	0.072	0.248

Table 2: Translation and heading drift statistics for each mission.

		Translation Drift (m/m)	Heading Drift (°/m)
Three Radar	p50	0.017	0.034
	p95	0.041	0.119
	p99	0.050	0.176
	max	0.072	0.248
Single Radar	p50	0.017	0.059
	p95	0.070	0.195
	p99	0.141	0.309
	max	0.314	0.493
No Heading Constraint	p50	0.049	0.526
	p95	0.257	2.258
	p99	0.322	4.714
	max	0.392	15.59

Table 3: Translation and heading drift statistics for mission 4 with different components of the odometry system de-activated.

version and the heading drift statistics are 16 to 60 times higher. Clearly the heading constraint is crucial for accurate odometry estimation.

## 4.2 Radar Map Matching Evaluation

To evaluate our radar mapping and matching method we created a global map of a neighborhood shown in Figure 2b using our mapping robot. Then the robot traverses the same path twice continuously creating local query maps and attempting to register them to the global map. The mapping robot’s Novatel GPS+INS system is used both to create the global map, and to evaluate the accuracy of alignments between the query and global maps. The global map covers a square area roughly 220m on a side, the query maps are roughly 20m on a side, and both the global and query maps use a 0.2m resolution in their OGMs. Within this global map region, the robot traverses a trajectory of 1.6km.

We evaluate the quality of our map matching method by comparing the transforms calculated from successful map registrations to groundtruth transforms from the Novatel GPS+INS system. We also note the availability of map matches as the number of successful matches over the total number of attempted matches. We compare the results for radar map matching to a our own lidar map matching method, which uses LOAM-generated submaps (Zhang and Singh, 2014) as the query map (rather than an OGM) but otherwise uses the same coarse-to-fine Full and Tracking comparison methods as our radar map matching method.

Table 4 shows the errors in translation estimates for radar and lidar map matching for both missions. One can see the median errors for radar and lidar map matching are similar, although lidar map matching gives lower median error. Radar map matching clearly outstrips lidar at the tails of the error distribution though. Radar’s 90th and 99th percentile errors are far lower than those for lidar.

The advantage of radar map matching in accuracy is somewhat compensated by its lower availability, however. Out of 1715 attempted matches, our radar map matching method succeeds 1389 times for a total availability of 0.81. Lidar, on the other hand matches 1652 out of 1826 for a total availability of 0.90. This is demonstrated in Figure 6, which shows the poses estimated by lidar map matching in red, radar map matching in black, and a groundtruth pose for each radar pose in green. One can see from this image that lidar map matching successfully finds matches more often, but its pose estimates can also be less accurate.

		Radar Translation Error (m)		Lidar Translation Error (m)	
		$x$	$y$	$x$	$y$
Mission 1	p50	0.1757	0.199	0.1791	0.2111
	p90	0.5056	0.4903	1.8316	1.3199
	p99	0.7995	0.9167	8.324	4.9173
	max	1.6315	2.9356	10.5059	6.8023
Mission 2	p50	0.2205	0.2161	0.1675	0.1932
	p90	0.6081	0.54	1.3718	1.932
	p99	0.8813	0.7492	4.6913	6.5304
	max	1.0016	0.8195	5.808	7.5731

Table 4: Translation error for radar and lidar map matching techniques.

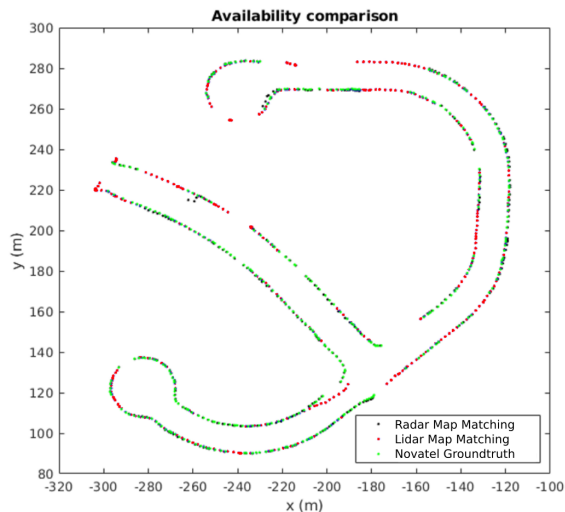


Figure 6: Comparison of the poses estimated by our radar map matching method with lidar map matching with radar pose estimates in black, lidar pose estimates in red, and groundtruth poses for each estimated radar pose in green.

## 5 Conclusion

In this paper we presented novel radar-based methods for odometry and global localization. For our radar-inertial odometry (RIO) method we combine an established Doppler velocity constraint with a novel method for dynamic outlier rejection and a unique heading constraint based on radar landmarks. We demonstrate the combination of these elements results in unprecedented relative pose estimation accuracy for a method using automotive sensors. Our RIO method’s accuracy is comparable to that of some methods based on high-resolution NavTech sensors. For example, our RIO method achieves median translation drift of 1.7% in the worst case while *Under The Radar* achieves 2.1% (Barnes and Posner, 2020) and *Masking By Moving* achieves 1.6% (Barnes et al., 2019) according to their evaluation. We note however that a direct comparison on their data is not possible given the unique experimental platform for which our method was developed. Further, our proposed RIO method is shown to be impressively lightweight; it can be used on low-power embedded hardware with ample headroom for other processes.

Our radar-based mapping method is able to aggregate radar detections into a representation of the robot’s environment that is repeatable and highly descriptive. This is demonstrated by our method for registering radar maps, which is able to achieve relative pose estimation accuracy similar to lidar-based methods. Unlike lidar, radar is sensitive to the properties of materials in addition to their geometry. This means radar maps

can represent changes in material properties where there is no corresponding change in geometry as in the cases of access hole covers, storm drain grates, etc. This makes radar map registration less susceptible to geometric ambiguity, as demonstrated by the much lower 99th percentile and maximum pose estimation errors of radar map matching relative to lidar. Our primary finding is that by balancing the IMU, Doppler, and landmark factors in RIO, as well as utilizing an incrementally-built OGM for map matching, we are able to achieve localization performance acceptable for suburban neighborhood navigation.

Together, our RIO and radar map matching methods provide a highly robust localization system for autonomous ground vehicles (AGVs) using only radar and inertial measurements. Radar map matching is able to globally localize the vehicle with respect to a pre-built map, and RIO can provide smooth and continuous pose updates between map matches. These developments could allow for the operation of AGVs in far more diverse and challenging environments from complete darkness to rain, snow, smoke and fog.

## Acknowledgments

This work was done for and supported by Amazon’s Scout project.

## References

- Barfoot, T., R. Forbes, J., and Furgale, P. (2011). Pose estimation using linearized rotations and quaternion algebra. *Acta Astronautica*, 68:101–112.
- Barnes, D., Gadd, M., Murcutt, P., Newman, P., and Posner, I. (2020). The oxford radar robotcar dataset: A radar extension to the oxford robotcar dataset. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6433–6438. IEEE.
- Barnes, D. and Posner, I. (2020). Under the radar: Learning to predict robust keypoints for odometry estimation and metric localisation in radar. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Paris.
- Barnes, D., Weston, R., and Posner, I. (2019). Masking by moving: Learning distraction-free radar odometry from pose information. *CoRR*, abs/1909.03752.
- Cen, S. H. and Newman, P. (2018). Precise Ego-Motion Estimation with Millimeter-Wave Radar Under Diverse and Challenging Conditions. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, Brisbane, QLD. IEEE.
- Cen, S. H. and Newman, P. (2019). Radar-only ego-motion estimation in difficult settings via graph matching. *arXiv:1904.11476 [cs]*.
- Chen, H., Liu, Y., and Cheng, Y. (2023). Drio: Robust radar-inertial odometry in dynamic environments. *IEEE Robotics and Automation Letters*, 8(9):5918–5925.
- Choi, M., Yang, S., Han, S., Lee, Y., Lee, M., Choi, K. H., and Kim, K.-S. (2023). Msc-rad4r: Ros-based automotive dataset with 4d radar. *IEEE Robotics and Automation Letters*.
- Doer, C. and Trommer, G. (2021). x-río: Radar inertial odometry with multiple radar sensors and yaw aiding. *Gyroscope and Navigation*, 12:329–339.
- Doer, C. and Trommer, G. F. (2020a). An ekf based approach to radar inertial odometry. In *2020 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 152–159.
- Doer, C. and Trommer, G. F. (2020b). Radar inertial odometry with online calibration. In *2020 European Navigation Conference (ENC)*, pages 1–10.

- Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24:381–395.
- Forster, C., Carlone, L., Dellaert, F., and Scaramuzza, D. (2015). Imu preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation. In *Robotics: Science and Systems*.
- Harlow, K., Jang, H., Barfoot, T. D., Kim, A., and Heckman, C. (2023). A new wave in robotics: Survey on recent mmwave radar applications in robotics.
- Hong, Z., Petillot, Y., and Wang, S. (2020). Radarslam: Radar based large-scale slam in all weathers. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5164–5170.
- Hornung, A., Wurm, K. M., Bennewitz, M., Stachniss, C., and Burgard, W. (2013). Octomap: An efficient probabilistic 3d mapping framework based on octrees. *Autonomous robots*, 34(3):189–206.
- Kaess, M., Johannsson, H., Roberts, R., Ila, V., Leonard, J., and Dellaert, F. (2011). isam2: Incremental smoothing and mapping with fluid relinearization and incremental variable reordering. In *2011 IEEE International Conference on Robotics and Automation*, pages 3281–3288.
- Kim, G., Choi, S., and Kim, A. (2021). Scan context++: Structural place recognition robust to rotation and lateral variations in urban environments. *IEEE Transactions on Robotics*.
- Kim, G., Park, Y. S., Cho, Y., Jeong, J., and Kim, A. (2020). Mulran: Multimodal range dataset for urban place recognition. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6246–6253. IEEE.
- Kramer, A., Harlow, K., Williams, C., and Heckman, C. (2021). Coloradar: The direct 3d millimeter wave radar dataset.
- Kramer, A. and Heckman, C. (2021). Radar-inertial state estimation and obstacle detection for micro-aerial vehicles in dense fog. In Siciliano, B., Laschi, C., and Khatib, O., editors, *Experimental Robotics*, pages 3–16, Cham. Springer International Publishing.
- Kramer, A., Stahoviak, C., Santamaria-Navarro, A., Agha-mohammadi, A.-a., and Heckman, C. (2020). Radar-inertial ego-velocity estimation for visually degraded environments. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5739–5746.
- Kung, P.-C., Wang, C.-C., and Lin, W.-C. (2021). A normal distribution transform-based radar odometry designed for scanning and automotive radars. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 14417–14423. IEEE.
- Lu, C. X., Saputra, M. R. U., Zhao, P., Almalioglu, Y., De Gusmao, P. P., Chen, C., Sun, K., Trigoni, N., and Markham, A. (2020). milliego: single-chip mmwave radar aided egomotion estimation via deep sensor fusion. In *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*, pages 109–122.
- Minguez, J., Montesano, L., and Lamiraux, F. (2006). Metric-based iterative closest point scan matching for sensor displacement estimation. *IEEE Transactions on Robotics*, 22(5):1047–1054.
- Moravec, H. and Elfes, A. (1985). High resolution maps from wide angle sonar. In *Proceedings. 1985 IEEE international conference on robotics and automation*, volume 2, pages 116–121. IEEE.
- Munkres, J. R. (1957). Algorithms for the assignment and transportation problems. *Journal of The Society for Industrial and Applied Mathematics*, 10:196–210.
- Olson, E. (2015). M3rsm: Many-to-many multi-resolution scan matching. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5815–5821. IEEE.
- Qi, C. R., Su, H., Mo, K., and Guibas, L. J. (2016). Pointnet: Deep learning on point sets for 3d classification and segmentation. *CoRR*, abs/1612.00593.

- Rapp, M., Dietmayer, K., Hahn, M., Schuster, F., Lombacher, J., and Dickmann, J. (2016). Fscd and basd: Robust landmark detection and description on radar-based grids. In *2016 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM)*, pages 1–4.
- Recherche, E., Automatique, E., Antipolis, S., and Zhang, Z. (1992). Iterative point matching for registration of free-form curves. *Int. J. Comput. Vision*, 13.
- Scharf, L. L. and Demeure, C. (1991). *Statistical signal processing: detection, estimation, and time series analysis*. Addison-Wesley Publishing Company.
- Schuster, F., Keller, C. G., Rapp, M., Haueis, M., and Curio, C. (2016a). Landmark based radar slam using graph optimization. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 2559–2564.
- Schuster, F., Wörner, M., Keller, C. G., Haueis, M., and Curio, C. (2016b). Robust localization based on radar signal clustering. pages 839–844.
- Zhang, J. and Singh, S. (2014). Loam: Lidar odometry and mapping in real-time. In *Robotics: Science and Systems*, volume 2, pages 1–9. Berkeley, CA.
- Zhang, J., Zhuge, H., Wu, Z., Peng, G., Wen, M., Liu, Y., and Wang, D. (2023). 4dradarslam: A 4d imaging radar slam system for large-scale environments based on pose graph optimization. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8333–8340.