Do We Need to Compensate for Motion Distortion and Doppler Effects in Spinning Radar Navigation?

Keenan Burnett, Angela P. Schoellig, and Timothy D. Barfoot

***Abstract*—In order to tackle the challenge of unfavorable weather conditions such as rain and snow, radar is being revisited as a parallel sensing modality to vision and lidar. Recent works have made tremendous progress in applying spinning radar to odometry and place recognition. However, these works have so far ignored the impact of motion distortion and Doppler effects on spinning-radar-based navigation, which may be significant in the self-driving car domain where speeds can be high. In this work, we demonstrate the effect of these distortions on radar odometry using the Oxford Radar RobotCar Dataset and metric localization using our own data-taking platform. We revisit a lightweight estimator that can recover the motion between a pair of radar scans while accounting for both effects. Our conclusion is that both motion distortion and the Doppler effect are significant in different aspects of spinning radar navigation, with the former more prominent than the latter. Code for this project can be found at: https://github.com/keenan-burnett/yeti_radar_odometry**

***Index Terms*—Localization, range sensing, intelligent transportation systems.**

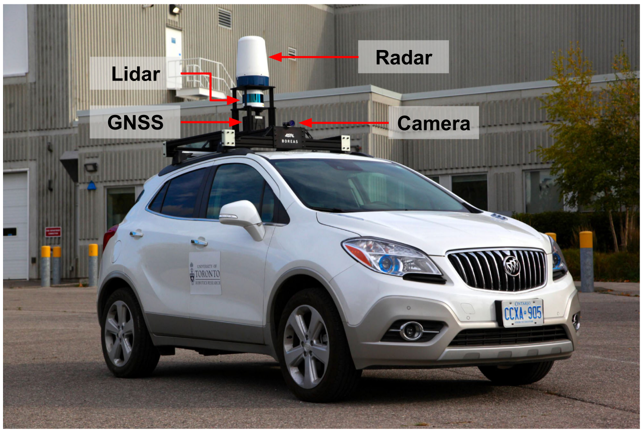


Fig. 1. Our data-taking platform, *Boreas*, which includes a Velodyne Alpha-Prime (128-beam) lidar, Navtech CIR204-H radar, FLIR Blackfly S monocular camera, and Applanix POSLV GNSS.

I. Introduction

As RESEARCHERS continue to advance the capabilities of autonomous vehicles, attention has begun to shift towards inclement weather conditions. Currently, most autonomous vehicles rely primarily cameras and lidar for perception and localization. Although these sensors have been shown to achieve sufficient performance under nominal conditions, rain and snow remain an open problem. Radar sensors, such as the one produced by Navtech [1], may provide a solution.

Due to its longer wavelength, radar is robust to small particles such as dust, fog, rain, or snow, which can negatively impact cameras and lidar sensors. Furthermore, radar sensors tend to have a longer detection range and can penetrate through some materials allowing them to see beyond the line of sight of lidar. These features make radar particularly well-suited for inclement weather. However, radar sensors have coarser spatial resolution than lidar and suffer from a higher noise floor making them challenging to work with.

Recent works have made tremendous progress in applying the Navtech radar to odometry [2]–[7] and place recognition [8]–[10]. However, all of these works make the simplifying assumption that a radar scan is collected at a single instant in time. In reality, the sensor is rotating while the vehicle is moving causing the radar scan to be distorted in a cork-screw fashion. Range measurements of the Navtech radar are also impacted by Doppler frequency shifts resulting from the relative velocity between the sensor and its surroundings. Both distortion effects become more pronounced as the speed of the ego-vehicle increases. Most automotive radar sensors are not impacted by either distortion effect. However, the Navtech provides 360° coverage with accurate range and azimuth resolution, making it an appealing navigation sensor.

In this paper, we demonstrate the effect that motion distortion can have on radar-based navigation. We also revisit a lightweight estimator, Motion-Compensated RANSAC [11], which can recover the motion between a pair of scans and remove the distortion. The Doppler effect was briefly acknowledged in [2] but our work is the first to demonstrate the impact on radar-based navigation and to provide a method for its compensation.

As our primary experiment to demonstrate the effects of motion distortion, we perform radar odometry on the Oxford Radar RobotCar Dataset [12]. As an additional experiment, we perform metric localization using our own data-taking platform, shown in Fig. 1. Qualitative results of both distortion effects are also provided. Rather than focusing on achieving state-of-the-art navigation results, the goal of this paper is to show that

Manuscript received October 15, 2020; accepted January 13, 2021. Date of publication January 18, 2021; date of current version February 1, 2021. This letter was recommended for publication by Associate Editor J. L. Blanco-Claraco and Editor J. Civera upon evaluation of the reviewers' comments. *(Corresponding author: Keenan Burnett.)*

The authors are with the University of Toronto Institute for Aerospace Studies (UTIAS), Toronto, Ontario M4Y2X6, Canada (e-mail: keenan.burnett@robotics.utias.utoronto.ca; schoellig@utias.utoronto.ca; tim.barfoot@utoronto.ca).

Digital Object Identifier 10.1109/LRA.2021.3052439

motion distortion and Doppler effects are significant and can be compensated for with relative ease.

The rest of this paper is organized as follows: Section II discusses related work, III provides our methodology to match two radar scans while compensating for motion distortion and the Doppler effect, IV has experiments, and V concludes.

II. Related Work

Self-driving vehicles designed to operate in ideal conditions often relegate radar to a role as a secondary sensor as part of an emergency braking system [13]. However, recent advances in Frequency Modulated Continuous Wave (FMCW) radar indicate that it is a promising sensor for navigation and other tasks typically reserved for vision and lidar. Jose and Adams [14] [15] were the first to research the application of spinning radar to outdoor SLAM. In [16], Checchin *et al.* present one of the first mapping and localization systems designed for a FMCW scanning radar. Their approach finds the transformation between pairs of scans using the Fourier Mellin transform. Vivet *et al.* [17] were the first to address the motion distortion problem for scanning radar. Our approach to motion distortion is simpler and we offer a more up-to-date analysis on a dataset that is relevant to current research. In [18], Kellner *et al.* present a method that estimates the linear and angular velocity of the ego-vehicle when the Doppler velocity of each target is known. The use of single-chip FMCW radar has recently become a salient area of research for robust indoor positioning systems under conditions unfavorable to vision [19], [20].

In [2], Cen *et al.* present their seminal work that has rekindled interest in applying FCMW radar to navigation. Their work presented a new method to extract stable keypoints and perform scan matching using graph matching. Further research in this area has been spurred by the introduction of the Oxford Radar RobotCar Dataset [12], which includes lidar, vision, and radar data from a Navtech radar. Other radar-focused datasets include the MulRan place recognition dataset [10] and the RADIATE object detection dataset [21].

Odometry has recently been a central focus of radar-based navigation research. Components of an odometry pipeline can be repurposed for mapping and localization, which is an ultimate goal of this research. In [3], Cen *et al.* present an update to their radar odometry pipeline with improved keypoint detection, descriptors, and a new graph matching strategy. Aldera *et al.* [4] train a focus of attention policy to downsample the measurements given to data association, thus speeding up the odometry pipeline. In [6], Barnes and Posner present a deep-learning-based keypoint detector and descriptor that are learned directly from radar data using differentiable point matching and pose estimation. In [22], Hong *et al.* demonstrate the first radar-SLAM pipeline capable of handling extreme weather.

Other approaches forego the feature extraction process and instead use the entire radar scan for correlative scan matching. Park *et al.* [7] use the Fourier Mellin Transform on Cartesian and log-polar radar images to sequentially estimate rotation and translation. In [5], Barnes *et al.* present a fully differentiable, correlation-based radar odometry approach. In their system, a binary mask is learned such that unwanted distractor features are ignored by the scan matching. This approach currently represents the state of the art for radar odometry performance.

Still others focus on topological localization that can be used by downstream metric mapping and localization systems to identify loop closures. Săftescu *et al.* [8] learn a metric space embedding for radar scans using a convolutional neural network. Nearest-neighbour matching is then used to recognize locations at test time. Gadd *et al.* [9] improve this place recognition performance by integrating a rotationally invariant metric space embedding into a sequence-based trajectory matching system previously applied to vision. In [23], Tang *et al.* focus on localization between a radar on the ground and overhead satellite imagery.

Other recent works using a spinning radar include the work by Weston *et al.* [24] that learns to generate occupancy grids from raw radar scans by using lidar data as ground truth. Kaul *et al.* [25] train a semantic segmentation model for radar data using labels derived from lidar- and vision-based semantic segmentation.

Motion distortion has been treated in the literature through the use of continuous-time trajectory estimation [11], [26]–[29] for lidars [30] and rolling-shutter cameras [31], but these tools are yet to be applied to spinning radar. TEASER [32] demonstrates a robust registration algorithm with optimality guarantees. However, it assumes that pointclouds are collected at single instances in time. Thus, TEASER is impacted by motion distortion in the same manner as rigid RANSAC.

Our work focuses on the problem of motion distortion and Doppler effects using the Navtech radar sensor which has not received attention from these prior works. Ideally, our findings will inform future research in this area looking to advance the state of the art in radar-based navigation.

III. Methodology

Section III-A describes our approach to feature extraction and data association. In Section III-B, we describe a motion-compensated estimator and a rigid estimator for comparison. Section III-C explains how the Doppler effect impacts radar range measurements and how to compensate for it.

A. Feature Extraction

Feature detection in radar data is more challenging than in lidar or vision due to its higher noise floor and lower spatial resolution. Constant False Alarm Rate (CFAR) [33] is a simple feature detector that is popular for use with radar. CFAR is designed to estimate the local noise floor and capture relative peaks in the radar data. One-dimensional CFAR can be applied to Navtech data by convolving each azimuth with a sliding-window detector.

As discussed in [2], CFAR is not the best detector for radar-based navigation. CFAR produces many redundant keypoints, is difficult to tune, and produces false positives due to the noise artifacts present in radar. Instead, Cen *et al.* [2] proposed a detector that estimates a signal's noise statistics and then scales the power at each range by the probability that it is a real detection. In [3], Cen *et al.* proposed an alternative detector that identifies continuous regions of the scan with high intensity

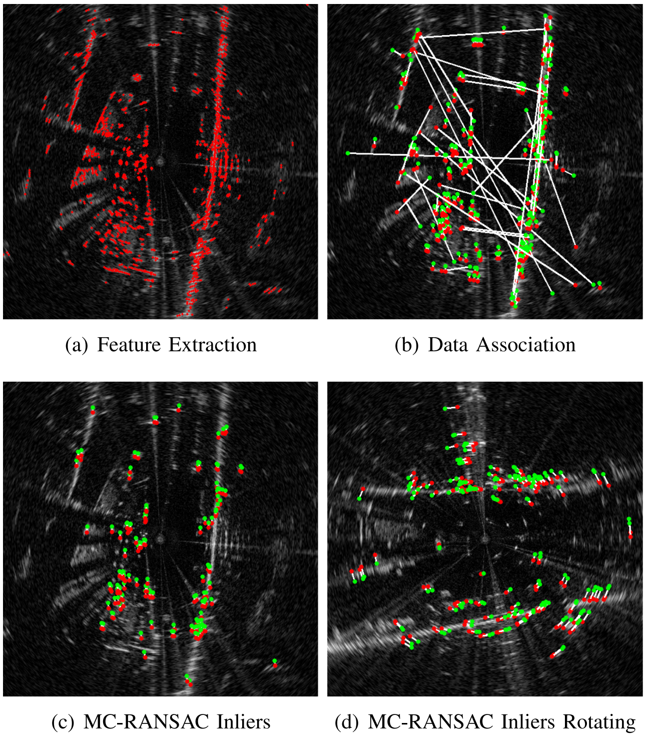


Fig. 2. This figure illustrates our feature extraction and matching process. (a) displays our raw extracted features. (b) displays the output of our ORB-descriptor-based data association. (c) and (d) show the inlier set resulting from motion-compensated RANSAC while driving straight and rotating.

and low gradients. Keypoints are then extracted by locating the middle of each continuous region. We will refer to these detectors as Cen2018 and Cen2019, respectively.

The original formulations of these detectors did not lend themselves to real-time operation. As such, we made several modifications to improve the runtime. For Cen2018, we use a Gaussian filter instead of a binomial filter, and we calculate the mean of each azimuth instead of using a median filter. We do not remove multipath reflections.

Cen2019 was designed to be easier to tune and have less redundant keypoints. However, we found that by adjusting the probability threshold of detections, Cen2018 obtained better odometry performance when combined with our RANSAC-based scan matching. Based on these preliminary tests, we concluded that Cen2018 was the best choice for our experiments. Fig. 2(a) shows Cen2018 features plotted on top of a Cartesian radar image.

We convert the raw radar scans output by the Navtech sensor, which are in polar form, into Cartesian form. We then calculate an ORB descriptor [34] for each keypoint on the Cartesian image. There may be better keypoint descriptors for radar data, such as the learned feature descriptors employed in [6]. However, ORB descriptors are quick to implement, rotationally invariant, and resistant to noise.

For data association, we perform brute-force matching of ORB descriptors. We then apply a nearest-neighbor distance ratio test [35] in order to remove false matches. The remaining matches are sent to our RANSAC-based estimators. Fig. 2(b) shows the result of the initial data association. Note that there are several outliers. Fig. 2(c), (d) shows the remaining inliers after performing RANSAC.

B. Motion Distortion

The output of data association is not perfect and often contains outliers. As a result, it is common to employ an additional outlier rejection scheme during estimation. In this paper, we use RANSAC [36] to find an outlier-free set that can then be used to estimate the desired transform. If we assume that two radar scans are taken at times $\bar{t}_1$ and $\bar{t}_2$, then the transformation between them can be estimated directly using the approach described in [37].

During each iteration of RANSAC, a random subset of size S is drawn from the initial matches and a candidate transform is generated. If the number of inliers exceeds a desired threshold or a maximum number of iterations is reached, the algorithm terminates. Radar scans are 2D and as such we use $S = 2$. The estimation process is repeated on the largest inlier set to obtain a more accurate transform. We will refer to this approach as *rigid* RANSAC.

Our derivation of motion-compensated RANSAC follows closely from [11]. However, we are applying the algorithm to a scanning radar in 2D instead of a two-axis scanning lidar in 3D. Furthermore, our derivation is shorter and uses updated notation from [38].

The principal idea behind motion-compensated RANSAC is to estimate the velocity of the sensor instead of estimating a transformation. We make the simplifying assumption that the linear and angular velocity between a pair of scans is constant. The combined velocity vector $\boldsymbol{\varpi}$ is defined as

$$\boldsymbol{\varpi} = \begin{bmatrix} \boldsymbol{\nu} \\ \boldsymbol{\omega} \end{bmatrix}, \tag{1}$$

where $\boldsymbol{\nu}$ and $\boldsymbol{\omega}$ are the linear and angular velocity in the sensor frame. To account for motion distortion, we remove the assumption that radar scans are taken at a single instant in time. Data association produces two sets of corresponding measurements, $\mathbf{y}_{m,1}$ and $\mathbf{y}_{m,2}$, where $m = 1 \ldots M$. Each pair of features, m, is extracted from sequential radar frames 1 and 2 at times $t_{m,1}$ and $t_{m,2}$. The temporal difference between a pair of measurements is $\Delta t_m := t_{m,2} - t_{m,1}$. The generative model for measurements is given as

$$\begin{aligned} \mathbf{y}_{m,1} &:= \mathbf{f}(\mathbf{T}_s(t_{m,1})\mathbf{p}_m) + \mathbf{n}_{m,1}, \\ \mathbf{y}_{m,2} &:= \mathbf{f}(\mathbf{T}_s(t_{m,2})\mathbf{p}_m) + \mathbf{n}_{m,2}, \end{aligned} \tag{2}$$

where $\mathbf{f}(\cdot)$ is a nonlinear transformation from Cartesian to cylindrical coordinates and $\mathbf{T}_s(t)$ is a 4 x 4 homogeneous transformation matrix representing the pose of the sensor frame $\underrightarrow{\mathcal{F}}_s$ with respect to the inertial frame $\underrightarrow{\mathcal{F}}_i$ at time t. $\mathbf{p}_m$ is the original landmark location in the inertial frame. We assume that each measurement is corrupted by zero-mean Gaussian noise: $\mathbf{n}_{m,1} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{m,1})$. The transformation between a pair of measurements is defined as

$$\mathbf{T}_m := \mathbf{T}_s(t_{m,2})\mathbf{T}_s(t_{m,1})^{-1}. \tag{3}$$

To obtain our objective function, we convert feature locations from polar coordinates into local Cartesian coordinates:

$$\mathbf{p}_{m,2} = \mathbf{f}^{-1}(\mathbf{y}_{m,2}), \tag{4}$$

$$\mathbf{p}_{m,1} = \mathbf{f}^{-1}(\mathbf{y}_{m,1}). \tag{5}$$

We then use the local transformation $\mathbf{T}_m$ to create a pseudomeasurement $\hat{\mathbf{p}}_{m,2}$,

$$\hat{\mathbf{p}}_{m,2} = \mathbf{T}_m \mathbf{p}_{m,1}. \tag{6}$$

The error is then defined in the sensor coordinates and summed over each pair of measurements to obtain our objective function:

$$\mathbf{e}_m = \mathbf{p}_{m,2} - \hat{\mathbf{p}}_{m,2}, \tag{7}$$

$$J(\boldsymbol{\varpi}) := \frac{1}{2}\sum_{m=1}^{M} \mathbf{e}_m^T \mathbf{R}_{\text{cart},m}^{-1} \mathbf{e}_m. \tag{8}$$

Here we introduce some notation for dealing with transformation matrices in $SE(3)$. A transformation $\mathbf{T} \in SE(3)$ is related to its associated Lie algebra $\boldsymbol{\xi}^\wedge \in \mathfrak{se}(3)$ through the exponential map:

$$\mathbf{T} = \begin{bmatrix} \mathbf{C} & \mathbf{r} \\ \mathbf{0}^T & 1 \end{bmatrix} = \exp(\boldsymbol{\xi}^\wedge), \tag{9}$$

where $\mathbf{C}$ is 3×3 a rotation matrix, $\mathbf{r}$ is a 3×1 translation vector, and $(\cdot)^\wedge$ is an overloaded operator that converts a vector of rotation angles $\boldsymbol{\phi}$ into a member of $\mathfrak{so}(3)$ and $\boldsymbol{\xi}$ into a member of $\mathfrak{se}(3)$:

$$\boldsymbol{\phi}^\wedge = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{bmatrix}^\wedge := \begin{bmatrix} 0 & -\phi_3 & \phi_2 \\ \phi_3 & 0 & -\phi_1 \\ -\phi_2 & \phi_1 & 0 \end{bmatrix}, \tag{10}$$

$$\boldsymbol{\xi}^\wedge = \begin{bmatrix} \boldsymbol{\rho} \\ \boldsymbol{\phi} \end{bmatrix}^\wedge := \begin{bmatrix} \boldsymbol{\phi}^\wedge & \boldsymbol{\rho} \\ \mathbf{0}^T & 1 \end{bmatrix}. \tag{11}$$

Given our constant-velocity assumption, we can convert from a velocity vector $\boldsymbol{\varpi}$ into a transformation matrix using the following formula:

$$\mathbf{T} = \exp(\Delta t \boldsymbol{\varpi}^\wedge). \tag{12}$$

In order to optimize our objective function $J(\boldsymbol{\varpi})$, we first need to derive the relationship between $\mathbf{T}_m$ and $\boldsymbol{\varpi}$. The velocity vector $\boldsymbol{\varpi}$ can be written as the sum of a nominal velocity $\overline{\boldsymbol{\varpi}}$ and a small perturbation $\delta\boldsymbol{\varpi}$. This lets us rewrite the transformation $\mathbf{T}_m$ as the product of a nominal transformation $\overline{\mathbf{T}}_m$ and a small perturbation $\delta\mathbf{T}_m$:

$$\mathbf{T}_m = \exp(\Delta t_m (\overline{\boldsymbol{\varpi}} + \delta\boldsymbol{\varpi})^\wedge) = \delta\mathbf{T}_m \overline{\mathbf{T}}_m. \tag{13}$$

Let $\mathbf{g}_m(\boldsymbol{\varpi}) := \mathbf{T}_m \mathbf{p}_{m,1}$, which is nonlinear due to the transformation. Our goal is to linearize $\mathbf{g}_m(\boldsymbol{\varpi})$ about a nominal operating point. We can rewrite $\mathbf{g}_m(\boldsymbol{\varpi})$ as:

$$\begin{aligned} \mathbf{g}_m(\boldsymbol{\varpi}) &= \exp(\Delta t_m \delta\boldsymbol{\varpi}^\wedge)\overline{\mathbf{T}}_m \mathbf{p}_{m,1}, \\ &\approx (\mathbf{1} + \Delta t_m \delta\boldsymbol{\varpi}^\wedge)\overline{\mathbf{T}}_m \mathbf{p}_{m,1}, \end{aligned} \tag{14}$$

where we use an approximation for small pose changes. We swap the order of operations using the $(\cdot)^\odot$ operator [38]:

$$\begin{aligned} \mathbf{g}_m(\boldsymbol{\varpi}) &= \overline{\mathbf{T}}_m \mathbf{p}_{m,1} + \Delta t_m (\overline{\mathbf{T}}_m \mathbf{p}_{m,1})^\odot \delta\boldsymbol{\varpi} \\ &= \overline{\mathbf{g}}_m + \mathbf{G}_m \delta\boldsymbol{\varpi}, \end{aligned} \tag{15}$$

$$\mathbf{p}^\odot = \begin{bmatrix} \boldsymbol{\rho} \\ \eta \end{bmatrix}^\odot = \begin{bmatrix} \eta\mathbf{1} & -\boldsymbol{\rho}^\wedge \\ \mathbf{0}^T & \mathbf{0}^T \end{bmatrix}. \tag{16}$$

We can now rewrite the error function from (7):

$$\begin{aligned} \mathbf{e}_m &\approx \mathbf{p}_{m,2} - \overline{\mathbf{g}}_m - \mathbf{G}_m \delta\boldsymbol{\varpi} \\ &= \overline{\mathbf{e}}_m - \mathbf{G}_m \delta\boldsymbol{\varpi}. \end{aligned} \tag{17}$$

By inserting this equation for the error function into the objective function from (8), and taking the derivative with respect to the perturbation and setting it to zero, $\frac{\partial J(\boldsymbol{\varpi})}{\partial \delta\boldsymbol{\varpi}^T} = 0$, we obtain the optimal update:

$$\delta\boldsymbol{\varpi}^\star = \left(\sum_m \mathbf{G}_m^T \mathbf{R}_{\text{cart},m}^{-1} \mathbf{G}_m\right)^{-1} \left(\sum_m \mathbf{G}_m^T \mathbf{R}_{\text{cart},m}^{-1} \overline{\mathbf{e}}_m\right), \tag{18}$$

where $\mathbf{R}_{\text{cart},m} = \mathbf{H}_m \mathbf{R}_{m,2} \mathbf{H}_m^T$ is the covariance in the local Cartesian frame, $\mathbf{h}(\cdot) = \mathbf{f}^{-1}(\cdot)$, and $\mathbf{H}_m = \frac{\partial \mathbf{h}}{\partial \mathbf{x}}|_{\overline{\mathbf{g}}_m}$. The optimal perturbation $\delta\boldsymbol{\varpi}^\star$ is used in a Gauss-Newton optimization scheme and the process repeats until $\boldsymbol{\varpi}$ converges.

This method allows us to estimate the linear and angular velocity between a pair of radar scans directly while accounting for motion distortion. These velocity estimates can then be used to remove the motion distortion from a measurement relative to a reference time using (12). MC-RANSAC is intended to be a lightweight method for showcasing the effects of motion distortion. A significant improvement to this pipeline would be to use the inliers of MC-RANSAC as an input to another estimator, such as [28]. The inliers could also be used for mapping and localization or SLAM.

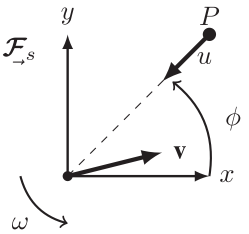


Fig. 3. This diagram illustrates the relationship between the ego-motion ($\mathbf{v}$, ω) and the radial velocity u.

C. Doppler Correction

In order to compensate for Doppler effects, we need to know the linear velocity of the sensor v. This can either be obtained from a GPS/IMU or using one of the estimators described above. As shown in Fig. 3, the motion of the sensor results in an apparent

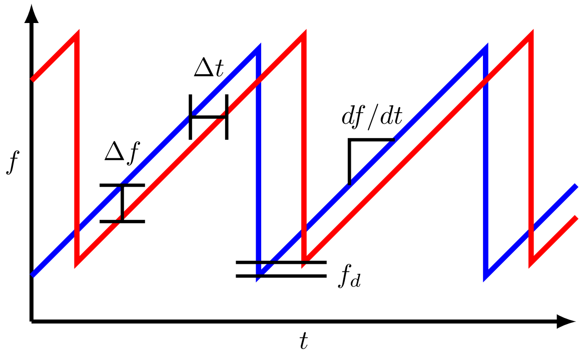


Fig. 4. This figure depicts the sawtooth modulation pattern of an FMCW radar. The transmitted signal is blue and the received signal is red.

relative velocity between the sensor and its surrounding environment. This relative velocity causes the received frequency to be altered according to the Doppler effect. Note that only the radial component of the velocity $u = v_x \cos(\phi) + v_y \sin(\phi)$ will result in a Doppler shift. The Radar Handbook by Skolnik [39] provides an expression for the Doppler frequency:

$$f_d = \frac{2\,u}{\lambda}, \tag{19}$$

where λ is the wavelength of the signal. Note that for an object moving towards the radar ($u > 0$) or vice versa, the Doppler frequency will be positive resulting in a higher received frequency. For FMCW radar such as the Navtech sensor, the distance to a target is determined by measuring the change in frequency between the received signal and the carrier wave Δf:

$$r = \frac{c\Delta f}{2(df/dt)}, \tag{20}$$

where df/dt is the slope of the modulation pattern used by the carrier wave and c is the speed of light. FMCW radar require two measurements to disentangle the frequency shift resulting from range and relative velocity. Since the Navtech sensor scans each azimuth only once, the measured frequency shift is the combination of both the range difference and Doppler frequency. From Fig. 4, we can see that a positive Doppler frequency f_d will result in an increase in the received frequency and in turn a reduction in the observed frequency difference Δf. Thus, a positive Doppler frequency will decrease the apparent range of a target.

The Navtech radar operates between 76 GHz and 77 GHz resulting in a bandwidth of 1 GHz. Navtech states that they use a sawtooth modulation pattern. Given 1600 measurements per second and assuming the entire bandwidth is used for each measurement, $df/dt \approx 1.6 \times 10^{12}$.

Hence, if the forward velocity of the sensor is 1 m/s, a target positioned along the x-axis (forward) of the sensor would experience a Doppler frequency shift of 510 Hz using (19). This increase in the frequency of the received signal would decrease the apparent range to the target by 4.8 cm using (20). Naturally, this effect becomes more pronounced as the velocity increases.

Let $\beta = f_t/(df/dt)$ where f_t is the transmission frequency ($f_t \approx$ 76.5 GHz). In order to correct for the Doppler distortion, the range of each target needs to be corrected by the following factor:

$$\Delta r_{\text{corr}} = \beta(v_x \cos(\phi) + v_y \sin(\phi)). \tag{21}$$

We use this simple correction in all our experiments with the velocity (v_x, v_y) coming from our motion estimator.

IV. Experimental Results

In order to answer the question posed by this paper, we have devised two experiments. The first is to compare the performance of rigid RANSAC and MC-RANSAC (with or without Doppler corrections) on radar odometry using the Oxford Radar RobotCar Dataset [12]. The second experiment demonstrates the impact of both distortion effects on localization using our own data. We also provide qualitative results demonstrating both distortion effects.

The Navtech radar is a frequency modulated continuous wave (FMCW) radar. For each azimuth, the sensor outputs the received signal power at each range bin. The sensor spins at 4 Hz and provides 400 measurements per rotation with a 163 m range, 4.32 cm range resolution, 1.8° horizontal beamwidth, and 0.9° azimuth resolution.

The Oxford Radar RobotCar Dataset is an autonomous driving dataset that includes two 32-beam lidars, six cameras, a GPS/IMU, and the Navtech sensor. The dataset includes thirty-two traversals equating to 280 km of driving in total.

A. *Odometry*

Our goal is to make a fair comparison between two estimators where the main difference is the compensation of distortion effects. To do this, we use the same number of maximum iterations (100), and the same inlier threshold (0.35 m) for both rigid RANSAC and MC-RANSAC. We also fix the random seed before running either estimator to ensure that the differences in performance are not due to the random selection of subsets.

For feature extraction, we use the same setup parameters for Cen2018 as in [2] except that we use a higher probability threshold, $z_q = 3.0$, and a Gaussian filter with $\sigma = 17$ range bins. We convert polar radar scans into a Cartesian image with 0.2592 m per pixel and a width of 964 pixels (250 m). For ORB descriptors, we use a patch size of 21 pixels (5.4 m). For data association, we use a nearest-neighbor distance ratio of 0.8. For Doppler corrections, we use $\beta = 0.049$. For each radar scan, extracting Cen2018 features takes approximately 35 ms. Rigid RANSAC runs in 1-2 ms and MC-RANSAC in 20-50 ms. Calculating orb descriptors takes 5 ms and brute-force matching takes 15 ms. These processing times were obtained on a quad-core Intel Xeon E3-1505 M 3.0 GHz CPU with 32 GB of RAM.

Odometry results are obtained by compounding the frame-to-frame scan matching results. Note that we do not use a motion prior or perform additional smoothing on the odometry. Three sequences were used for parameter tuning. The remaining 29 sequences are used to provide test results.

Table I summarizes the results of the odometry experiment. We use KITTI-style odometry metrics [40] to quantify the translational and rotational drift as is done in [6]. The metrics

TABLE I
RADAR ODOMETRY RESULTS. *ODOMETRY RESULTS FROM [6]

Method	Translational Error (%)	Rotational Error (deg/m)
RO Cen* [2]	3.7168	0.0095
Rigid RANSAC	3.9777	0.0141
MC-RANSAC	3.6042	0.0119
MC-RANSAC + Doppler	3.5847	0.0118

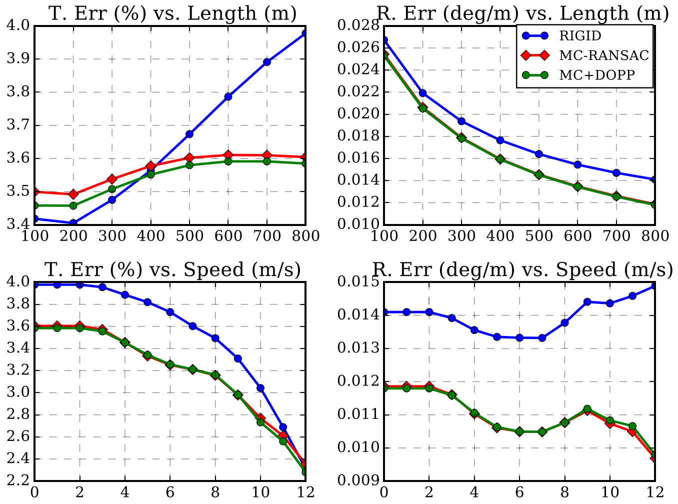


Fig. 5. This figure provides our KITTI-style odometry results on the Oxford dataset. We provide our translational and rotational drift as a function of path length and speed. MC-RANSAC: motion-compensated RANSAC, MC+DOPP: motion and distortion compensated.

are obtained by averaging the translational and rotational drifts for all subsequences of lengths (100, 200, …, 800) meters. The table shows that motion-compensated RANSAC results in a 9.4% reduction in translational drift and a 15.6% reduction in rotational drift. This shows that compensating for motion distortion has a modest impact on radar odometry. The table also indicates that Doppler effects have a negligible impact on odometry. Our interpretation is that Doppler distortion impacts sequential radar scans similarly such that scan registration is minimally impacted.

It should be noted that a large fraction of the Oxford dataset was collected at low speeds (0–5 m/s). This causes the motion distortion and Doppler effects to be less noticeable.

Fig. 5 depicts the translational and rotational drift of each method as a function of path length and speed. It is clear that motion compensation improves the odometry performance across most cases. Interestingly, MC-RANSAC does not increase in error as much as rigid RANSAC as the path length increases. Naturally, we would expect rigid RANSAC to become much worse than MC-RANSAC at higher speeds. However, what we observe is that as the speed of the vehicle increases, the motion tends to become more linear. When the motion of the vehicle is mostly linear, the motion distortion does not impact rigid RANSAC as much. Fig. 6 compares the odometry output of both estimators against ground truth. The results further illustrate that compensating for motion distortion improves performance.

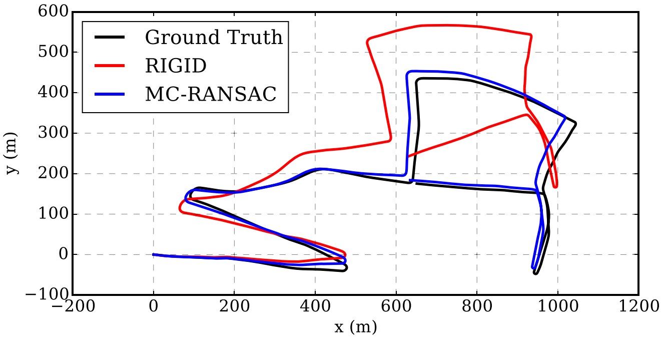


Fig. 6. This figure highlights the impact that motion distortion can have on the accuracy of radar-based odometry. Note that motion-compensated RANSAC (MC-RANSAC) is much closer to the ground truth.

B. Localization

The purpose of this experiment is to demonstrate the impact of motion distortion and Doppler effects on metric localization. As opposed to the previous experiment, we localize between scans taken while driving in opposite directions. While the majority of the Oxford Radar dataset was captured at low speeds (0–10 m/s), in this experiment we only use radar frames where the ego-vehicle's speed was above 10 m/s. For this experiment, we use our own data-taking platform, shown in Fig. 1, which includes a Velodyne Alpha-Prime lidar, Navtech CIR204-H radar, Blackfly S camera, and an Applanix POSLV GNSS. Individuals interested in using this data can fill out a Google form to gain access.[1]

Ground truth for this experiment was obtained from a 10 km drive using post-processed GNSS data provided by Applanix, which has an accuracy of 12 cm in this case. Radar scans were initially matched by identifying pairs of proximal scans on the outgoing and return trips based on GPS data. The Navtech timestamps were synchronized to GPS time to obtain an accurate position estimate.

Our first observation was that localizing against a drive in reverse is harder than odometry. When viewed from different angles, objects have different radar cross sections, which causes them to appear differently. As a consequence, radar scans may lose or gain features when pointed in the opposite direction. This change in the radar scan's appearance was sufficient to prevent ORB features from matching.

As a replacement for ORB descriptors, we turned to the Radial Statistics Descriptor (RSD) described in [2], [3]. Instead of calculating descriptors based on the Cartesian radar image, RSD operates on a binary Cartesian grid derived from the detected feature locations. This grid can be thought of as a radar target occupancy grid. For each keypoint, RSD divides the binary grid into M azimuth slices and N range bins centered around the keypoint. The number of keypoints (pixels) in each azimuth slice and range bin is counted to create two histograms. In [3], a fast Fourier transform of the azimuth histogram is concatenated with a normalized range histogram to form the final descriptor.

In our experiment, we found that the range histogram was sufficient on its own, with the azimuth histogram offering only

[1] [Online]. Available: https://forms.gle/ZGtQhKRXkxmcAGih9

TABLE II
METRIC LOCALIZATION RESULTS

Method	Trans. Error (m)	Rot. Error (deg)
No Compensation	2.2976	0.7110
Motion-Compensated	1.3403	1.0766
Motion-Compensated + Doppler-Compensated	0.4327	0.9984

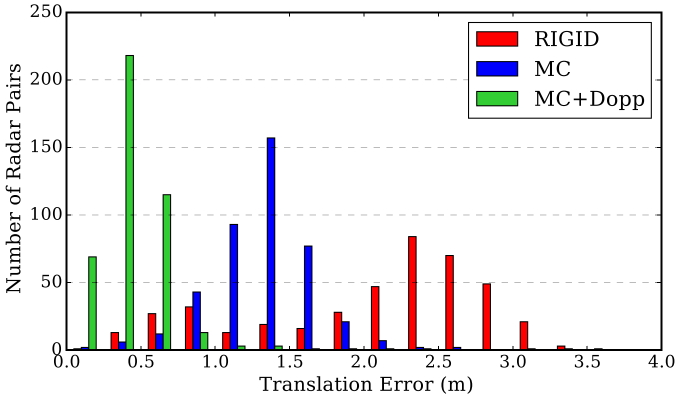


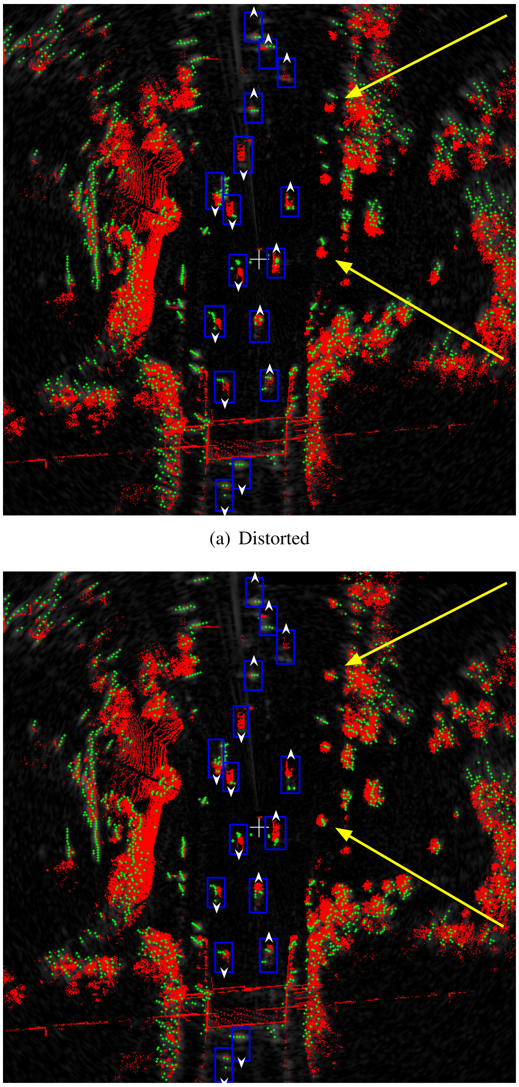
Fig. 7. By compensating for motion distortion alone (MC) or both motion and Doppler distortion (MC + DOPP), metric localization improves.

a minor improvement. It should be noted that these descriptors are more expensive to compute (60 ms) and match (30 ms) than ORB descriptors. These processing times were obtained using the same hardware as in Section IV-A.

The results of our localization experiment are summarized in Table II. In each case, we are using our RANSAC estimator from Section III. The results in the table are obtained by calculating the median translation and rotation error. Compensating for motion distortion results in a 41.7% reduction in translation error. Compensating for Doppler effects results in a further 67.7% reduction in translation error. Together, compensating for both effects results in a 81.2% reduction in translation error. Note that the scan-to-scan translation error is larger than in the odometry experiment due to the increased difficulty of localizing against a reverse drive. Fig. 7 depicts a histogram of the localization errors in this experiment.

C. Qualitative Results

In this section, we present qualitative results of removing motion distortion and Doppler effects from radar data. In Fig. 8, we have plotted points from our Velodyne lidar in red and extracted radar targets in green. In this example, the ego-vehicle is driving at 15 m/s with vehicles approaching from the opposite direction on the left-hand side of the road. In order to directly compare lidar and radar measurements, we have aligned each sensor's output spatially and temporally using post-processed GPS data and timestamps. Motion distortion is removed from the lidar points before the comparison is made. We use the lidar measurements to visualize the amount of distortion present in the radar scan. Fig. 8(a) shows what the original alignment looks like when the radar scan is distorted. Fig. 8(b) shows what the alignment looks like after compensating for motion distortion and Doppler effects. Note that static objects such as trees align much better with the lidar data in Fig. 8(b). It interesting to note that some of the moving vehicles (boxed in blue) are less aligned after removing distortion. Here we do not know the other vehicles' velocities and therefore the true relative velocity with respect to the ego-vehicle. These results indicate that additional velocity information for each object is required in order to align dynamic objects. We leave this problem for future work.


(a) Distorted

(b) Motion and Doppler Distortion Removed

Fig. 8. Lidar points are shown in red, radar targets are shown in green, vehicles are boxed in blue. (a) Both motion distortion and Doppler effects and present in the radar scan. (b) Motion distortion and Doppler effects have been removed from the radar data. Note that static objects (highlighted by yellow arrows) align much better in (b). Some of the moving vehicles (boxed in blue) are less aligned after removing distortion. Here we do not know the other vehicles' velocities and therefore the true relative velocity with respect to the ego-vehicle.

V. CONCLUSION

For the problem of odometry, compensating for motion distortion had a modest impact of reducing translational drift by 9.4%. Compensating for Doppler effects had a negligible effect on odometry performance. We postulate that Doppler effects are negligible for odometry because their effects are quite similar from one frame to the next. In our localization experiment, we observed that compensating for motion distortion and Doppler effects reduced translation error by 41.7% and 67.7% respectively with a combined reduction of 81.2%. We also provided qualitative results demonstrating the impact of both distortion effects. We noted that the velocity of each dynamic object is required in order to properly undistort points associated with dynamic objects. In summary, the Doppler effect can be safely ignored for the radar odometry problem but motion distortion should be accounted for to achieve the best results. For metric localization, especially for localizing in the opposite direction from which the map was built, both motion distortion and Doppler effects need to be compensated. Accounting for these effects is computationally cheap, but requires an accurate estimate of the linear and angular velocity of the sensor.

For future work, we will investigate applying more powerful estimators such as [28] to odometry and the full mapping and localization problem. We will also investigate learned features and the impact of seasonal changes on radar maps.

Acknowledgment

The authors would like to thank Goran Basic for designing and assembling our roof rack for Boreas. We would also like to thank Andrew Lambert and Keith Leung at Applanix for their help in integrating the POSLV system and providing the post-processed GNSS data. We thank General Motors for their donation of the Buick.

References

[1] "Safety is everything. - Navtech Radar," Accessed: Nov. 13, 2020. [Online]. Available: https://navtechradar.com/

[2] S. H. Cen and P. Newman, "Precise ego-motion estimation with millimeter-wave radar under diverse and challenging conditions," in *Proc. IEEE Int. Conf. Robot. Automat*., 2018, pp. 1–8.

[3] S. H. Cen and P. Newman, "Radar-only ego-motion estimation in difficult settings via graph matching," in *Proc. Int Conf. Robot. Automat*., 2019, pp. 298–304.

[4] R. Aldera, D. De Martini, M. Gadd, and P. Newman, "Fast radar motion estimation with a learnt focus of attention using weak supervision," in *Proc. Int Conf. Robot. Automat*., 2019, pp. 1190–1196.

[5] D. Barnes, R. Weston, and I. Posner, "Masking by moving: Learning distraction-free radar odometry from pose information," in *Proc. Conf. Robot Learn.*, 2020, pp. 303–316.

[6] D. Barnes and I. Posner, "Under the radar: Learning to predict robust keypoints for odometry estimation and metric localisation in radar," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 9484–9490.

[7] Y. S. Park, Y.-S. Shin, and A. Kim, "Pharao: Direct radar odometry using phase correlation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 2617–2623.

[8] Ş. Săftescu, M. Gadd, D. De Martini, D. Barnes, and P. Newman, "Kidnapped radar: Topological radar localisation using rotationally-invariant metric learning," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 4358–4364.

[9] M. Gadd, D. De Martini, and P. Newman, "Look around you: Sequence-based radar place recognition with learned rotational invariance," in *Proc. IEEE/ION Position, Location Navigation Symp.*, 2020, pp. 270–276.

[10] G. Kim, Y. S. Park, Y. Cho, J. Jeong, and A. Kim, "Mulran: Multimodal range dataset for urban place recognition," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 6246–6253.

[11] S. Anderson and T. D. Barfoot, "Ransac for motion-distorted 3 d visual sensors," in *Proc. IEEE/RSJ Int. Conf. Intell Robots Syst*., 2013, pp. 2093–2099.

[12] D. Barnes, M. Gadd, P. Murcutt, P. Newman, and I. Posner, "The oxford radar robotcar dataset: A. radar extension to the oxford robotcar dataset," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 6433–6438.

[13] K. Burnett *et al.*, "Zeus: A system description of the two-time winner of the collegiate SAE autodrive competition," *J. Field Robot.*, vol. 38, no. 1, pp. 139–166, 2021.

[14] E. Jose and M. D. Adams, "Relative radar cross section based feature identification with millimeter wave radar for outdoor slam," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, vol. 1, 2004, pp. 425–430.

[15] E. Jose, M. Adams, J. S. Mullane, and N. M. Patrikalakis, "Predicting millimeter wave radar spectra for autonomous navigation," *IEEE Sensors J.*, vol. 10, no. 5, pp. 960–971, May 2010.

[16] P. Checchin, F. Gérossier, C. Blanc, R. Chapuis, and L. Trassoudaine, "Radar scan matching slam using the Fourier-Mellin transform," in *Field and Service Robotics*, Berlin, Germany: Springer, 2010, pp. 151–161.

[17] D. Vivet, P. Checchin, and R. Chapuis, "Localization and mapping using only a rotating FMCW radar sensor," *Sensors*, vol. 13, no. 4, pp. 4527–4552, 2013.

[18] D. Kellner, M. Barjenbruch, J. Klappstein, J. Dickmann, and K. Dietmayer, "Instantaneous ego-motion estimation using doppler radar," in *Proc. 16th Int. IEEE Conf. Intell. Transp. Syst.*, 2013, pp. 869–874.

[19] A. Kramer, C. Stahoviak, A. Santamaria-Navarro, A.-A. Aghamohammadi, and C. Heckman, "Radar-inertial ego-velocity estimation for visually degraded environments," in *Proc. IEEE Int. Conf. Robot. Automat*., 2020, pp. 5739–5746.

[20] C. X. Lu *et al.*, "Milliego: Single-chip mmwave radar aided egomotion estimation via deep sensor fusion," in *Proc. 18th Conf. Embedded Networked Sensor Syst.*, 2020, pp. 109–122.

[21] M. Sheeny, E. De Pellegrin, S. Mukherjee, A. Ahrabian, S. Wang, and A. Wallace, "Radiate: A radar dataset for automotive perception," 2020, *arXiv:2010.09076*.

[22] Z. Hong, Y. Petillot, and S. Wang, "Radarslam: Radar based large-scale slam in all weathers," in *Proc. IEEE/RSJ Int. Conf. Intell Robots Syst.*, 2020.

[23] T. Y. Tang, D. De Martini, D. Barnes, and P. Newman, "Rsl-net: Localising in satellite images from a radar on the ground," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 1087–1094, Apr. 2020.

[24] R. Weston, S. Cen, P. Newman, and I. Posner, "Probably unknown: Deep inverse sensor modelling radar," in *Proc. Int Conf. Robot. Automat*., 2019, pp. 5446–5452.

[25] P. Kaul, D. De Martini, M. Gadd, and P. Newman, "Rss-net: Weakly-supervised multi-class semantic segmentation with FMCW radar," in *Proc. IEEE Intell. Veh. Symp.*, 2020, pp. 431–436.

[26] T. D. Barfoot, C. H. Tong, and S. Sarkka, "Batch continuous-time trajectory estimation as exactly sparse Gaussian process regression," in *Proc. Robot.: Sci. Syst.*, Berkeley, USA, 12–16, Jul. 2014.

[27] S. Anderson, T. D. Barfoot, C. H. Tong, and S. Sarkka, "Batch nonlinear continuous-time trajectory estimation as exactly sparse Gaussian process regression," *Auton. Robots*, vol. 39, no. 3, pp. 221–238, 2015.

[28] S. Anderson and T. D. Barfoot, "Full steam ahead: Exactly sparse Gaussian process regression for batch continuous-time trajectory estimation on se (3)," in *Proc. IEEE/RSJ Int. Conf. Intell Robots Syst*., 2015, pp. 157–164.

[29] P. T. Furgale, C. H. Tong, T. D. Barfoot, and G. Sibley, "Continuous-time batch trajectory estimation using temporal basis functions," *Int. J. Robot. Res.*, vol. 34, no. 14, pp. 1688–1710, 2015.

[30] M. Bosse and R. Zlot, "Map matching and data association for large-scale two-dimensional laser scan-based SLAM," *The Int. J. Robot. Res.*, vol. 27, no. 6, pp. 667–691, 2008.

[31] J. Hedborg, P.-E. Forssén, M. Felsberg, and E. Ringaby, "Rolling shutter bundle adjustment," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 1434–1441.

[32] H. Yang, J. Shi, and L. Carlone, "TEASER: Fast and certifiable point cloud registration," *IEEE Trans. Robot.*, 2020, pp. 1–20, doi: 10.1109/TRO.2020.3033695.

[33] H. Rohling, "Ordered statistic CFAR technique-an overview," in *Proc. 12th Int. Radar Symp*., 2011, pp. 631–638.

[34] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *Proc. Int. Conf. Comput. Vis*., 2011, pp. 2564–2571.

[35] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[36] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[37] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-D point sets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-9, no. 5, pp. 698–700, Sep. 1987.

[38] T. D. Barfoot, *State Estimation for Robotics*. Cambridge, U.K.: Cambridge Univ. Press, 2017.

[39] M. I. Skolnik, "Radar Handbook," 2nd ed. New York, NY, USA: McGrawHill, 1990.

[40] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, 2013.