Are We Ready for Radar to Replace Lidar in All-Weather Mapping and Localization?

Keenan Burnett*, Yuchen Wu*, David J. Yoon, Angela P. Schoellig, Timothy D. Barfoot

Abstract— We present an extensive comparison between three topometric localization systems: radar-only, lidar-only, and a cross-modal radar-to-lidar system across varying seasonal and weather conditions using the Boreas dataset. Contrary to our expectations, our experiments showed that our lidar-only pipeline achieved the best localization accuracy even during a snowstorm. Our results seem to suggest that the sensitivity of lidar localization to moderate precipitation has been exaggerated in prior works. However, our radar-only pipeline was able to achieve competitive accuracy with a much smaller map. Furthermore, radar localization and radar sensors still have room to improve and may yet prove valuable in extreme weather or as a redundant backup system. Code for this project can be found at: `https://github.com/utiasASRL/vtr3`

I. INTRODUCTION

Many autonomous driving companies leverage detailed semantic maps to drive safely. These maps may include the locations of lanes, pedestrian crossings, traffic lights, and more. In this case, the vehicle no longer has to detect each of these features from scratch in real-time. Instead, given the vehicle's current position, the semantic map can be used as a prior to simplify the perception task. However, it then becomes critical to know the pose of the robot within the map with sufficient accuracy and reliability.

Dense lidar maps can be built using offline batch optimization while incorporating IMU measurements for improved local alignment and GPS for improved global alignment [1]. Highly accurate localization can be subsequently performed by aligning a live lidar scan with a pre-built map with reasonable robustness to weather conditions [2], [3]. Vision-based mapping and localization is an alternative that can be advantageous in the absence of environment geometry. However, robustness to large appearance change (e.g., lighting) is a difficult and on-going research problem [4]. Radar-based systems present another compelling alternative.

Models of atmospheric attenuation show that radar can operate under certain adverse weather conditions where lidar cannot [5], [6]. These conditions may include heavy rain ($>$25mm/hr), dense fog ($>$0.1g/m^3), or a dust cloud ($>$10g/m^3). Existing literature does not describe the operational envelope of current lidar or radar sensors for the task of localization. Prior works have assumed that lidar localization is susceptible to moderate rain or snow necessitating the use of radar. In this paper, we attempt to

Fig. 1. Our platform, *Boreas*, includes a Velodyne Alpha-Prime (128-beam) lidar, a FLIR Blackfly S camera, a Navtech CIR304-H radar, and an Applanix POS LV GNSS-INS.

shed some light on this topic by comparing the performance of three topometric localization systems: radar-only, lidar-only, and a cross-modal radar-to-lidar system. We compare these systems across varying seasonal and weather conditions using our own publicly available dataset collected using the vehicle shown in Figure 1 [7]. Such a comparison of topometric localization methods has not been shown in the literature before and forms our primary contribution.

II. RELATED WORK

Automotive radar sensors now offer range and azimuth resolutions approximately on par with mechanically actuated radar. It is possible to replace a single 360 degree rotating radar with several automotive radar panelled around a vehicle [8]. Each target will then enjoy a relative (Doppler) velocity measurement, which can be used to estimate ego-motion [9]. However, recent work [10], [11] seems to indicate that the target extraction algorithms built into automotive radar may not necessarily be optimal for mapping and localization. Thus, sensors that expose the underlying signal data offer greater flexibility since the feature extraction algorithm can be tuned for the desired application.

Extracting keypoints from radar data and subsequently performing data association has proven to be challenging. The first works to perform radar-based localization relied on highly reflective objects installed within a demonstration area [12] [13]. These reflective objects were thus easy to discriminate from background noise. Traditional radar filtering tech-

niques such as Constant False Alarm Rate (CFAR) [14] have proven to be difficult to tune for radar-based localization. Setting the threshold too high results in insufficient features, which can cause localization to fail. Setting the threshold too low results in a noisy radar pointcloud and a registration process that is susceptible to local minima.

Several promising methods have been proposed to improve radar-based localization. Jose and Adams [15] demonstrated a feature detector that estimates the probability of target presence while augmenting their Simultaneous Localization and Mapping (SLAM) formulation to include radar cross section as an additional discriminating feature. Chandran and Newman [16] maximized an estimate of map quality to recover both the vehicle motion and radar map. Rouveure et al. [17] and Checchin et al. [18] eschewed sparse feature extraction entirely by matching dense radar scans using 3D cross-correlation and the Fourier-Mellin transform. Callmer et al. [19] demonstrated large-scale radar SLAM by leveraging vision-based feature descriptors. Mullane et al. [20] proposed to use a random-finite-set formulation of SLAM in situations of high clutter and data association ambiguity. Vivet et al. [21] and Kellner et al. [9] proposed to use relative Doppler velocity measurements to estimate the instantaneous motion. Schuster et al. [22] demonstrated a landmark-based radar SLAM that uses their Binary Annular Statistics Descriptor to match keypoints. Rapp et al. [23] used Normalized Distributions Transform (NDT) to perform probabilistic ego-motion estimation with radar.

Cen and Newman [24] demonstrated low-drift radar odometry over a large distance that inspired a resurgence of research into radar-based localization. Several datasets have been created to accelerate research in this area including the Oxford Radar RobotCar dataset [25], MulRan [26], and RADIATE [27]. We have recently released our own dataset, the Boreas dataset[1], which includes over 350km of data collected on a repeated route over the course of 1 year.

More recent work in radar-based localization has focused on either improving aspects of radar odometry [10], [28]–[37], developing better SLAM pipelines [38]–[40], or performing place recognition [41]–[43]. Barnes et al. [31] trained an end-to-end correlation-based radar odometry pipeline. Barnes and Posner [32] demonstrated radar odometry using deep learned features and a differentiable singular value decomposition (SVD)-based estimator. In [34], we quantified the importance of motion distortion in radar odometry and showed that Doppler effects should be removed during mapping and localization. Subsequently, in [35], we demonstrated unsupervised radar odometry, which combined a learned front-end with a classic probabilistic back-end.

Alhashimi et al. [37] present the current state of the art in radar odometry. Their method builds on prior work by Adolfsson et al. [36] by using a feature extraction algorithm called Bounded False Alarm Rate (BFAR) to add a constant offset b to the usual CFAR threshold: $T = a \cdot Z + b$.

Fig. 2. The 8km Glen Shields route[3] in Toronto. The yellow stars correspond to UTIAS, Dufferin, and Glen Shields (left to right) as in Figure 5.

The resulting radar pointclouds are registered to a sliding window of keyframes using an Iterative Closest Point (ICP)-like optimizer while accounting for motion distortion.

Other related work has focused on localizing radar scans to satellite imagery [44]–[46], or to pre-built lidar maps [47], [48]. Localizing live radar scans to existing lidar maps built in ideal conditions is a desirable option as we still benefit from the robustness of radar without incurring the expense of building brand new maps. However, the global localization errors reported in these works are in the range of 1m or greater. We demonstrate that we can successfully localize live radar scans to a pre-built lidar map with a relative localization error of around 0.1m.

In this work, we implement topometric localization that follows the Teach and Repeat paradigm [49], [50] without using GPS or IMU measurements. Hong et al. [40] recently compared the performance of their radar SLAM to SuMa, surfel-based lidar SLAM [3]. On the Oxford RobotCar dataset [25], they show that SuMa outperforms their radar SLAM. However, in their experiments, SuMa often fails partway through a route. Our interpretation is that SuMa losing track is more likely due to an implementation detail inherent to SuMa itself rather than a shortcoming of all lidar-based SLAM systems. It should be noted that Hong et al. did not tune SuMa beyond the original implementation which was tested on a different dataset. In addition, Hong et al. tested SuMa using 32-beam lidar whereas the original implementation used a 64-beam lidar. Furthermore, Hong et al. only provide a qualitative comparison between their radar SLAM and SuMA in rain, fog, and snow whereas our work provides a quantitative comparison across varying weather conditions. In some of the qualitative results they presented, it is unclear whether SuMa failed due to adverse weather or due to geometric degeneracy in the environment which is a separate problem. Importantly, our results seem to conflict with theirs by showing that lidar localization can operate successfully in even moderate to heavy snowfall. Although, it is possible that topometric localization is more robust to adverse weather since it uses both odometry and localization to a pre-built map constructed in nominal weather.

III. METHODOLOGY

A. Lidar/Radar Teach and Repeat Overview

Teach and Repeat is an autonomous route following framework that manually teaches a robot a network of traversable paths [49], [51], [52]. A key enabling idea is the construction of a *topometric map* [53] of the taught paths, represented as a pose graph in Figure 3. In the teach pass, a sequence of sensor data (i.e., lidar or radar) from a driven route is processed into local submaps stored along the path (vertices), and are connected together by relative pose estimates (edges). In the repeat pass, a new sequence of sensor data following the same route is processed into a new branch of the pose graph while simultaneously being localized against the vertices of the previous sequence to account for odometric drift. By localizing against local submaps along the taught paths, the robot can accurately localize and route-follow without the need for an accurate global reconstruction. In this paper, we focus on the estimation pipeline of Teach and Repeat (Figure 4). We divide the pipeline into: *Preprocessing*, *Odometry and Mapping*, and *Localization*

1) Preprocessing: This module performs feature extraction and filtering on raw sensor data, which in our work is from either lidar or radar sensors. More sensor-specific information is provided in III-B.

2) Odometry and Mapping: During both teach and repeat passes, this module estimates the transformation between the submap of the latest (local) vertex frame, $\mathcal{F}_k$, and the latest live sensor scan at the current moving robot frame, $\mathcal{F}_r$ (i.e., $\hat{\mathbf{T}}_{rk}$ in Figure 3(a)). If the translation or rotation of $\hat{\mathbf{T}}_{rk}$ exceeds a predefined threshold (10m / 30 degrees), we add a new vertex $\mathcal{F}_{k+1}$ connected with a new edge $\mathbf{T}_{k+1,k} = \hat{\mathbf{T}}_{rk}$. Each edge consists of both the mean relative pose and its covariance (uncertainty) estimate. The new submap stored at $\mathcal{F}_{k+1}$ is an accumulation of the last $n = 3$ processed sensor scans. All submaps are motion compensated and stored in their respective (local) vertex frame. The live scan is also motion compensated and sent as input to the *Localization* module. We present the details of our motion-compensated odometry algorithm, Continuous-Time Iterative Closest Point (CT-ICP), in Section III-C.

3) Localization: During the repeat pass, this module localizes the motion-compensated live scan of the robot frame, $\mathcal{F}_r$, against the submap of the spatially closest vertex frame, $\mathcal{F}_m$, of the previous sequence (i.e., $\hat{\mathbf{T}}_{rm}$ as shown in Figure 3(b)). Vertex frame $\mathcal{F}_m$ is chosen by leveraging our latest odometry estimate and traversing through the pose graph edges. Given the pose graph in Figure 3(b), the initial estimate to $\mathcal{F}_m$ is

$$\check{\mathbf{T}}_{rm} = \mathbf{T}_{rk}\mathbf{T}_{kk'}\mathbf{T}_{k'm'}\mathbf{T}_{m'm}. \quad (1)$$

We localize using ICP with $\check{\mathbf{T}}_{rm}$ as a prior, resulting in $\hat{\mathbf{T}}_{rm}$. If ICP alignment is successful, we add a new edge between the vertex of $\mathcal{F}_m$ and the latest vertex of the current sequence, $\mathcal{F}_k$, by compounding the mean localization result with the latest odometry result,

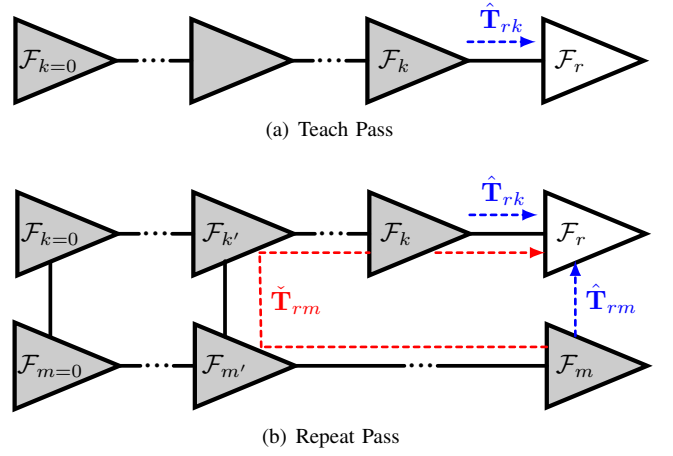$$\hat{\mathbf{T}}_{km} = \hat{\mathbf{T}}_{rk}^{-1}\hat{\mathbf{T}}_{rm}, \quad (2)$$

(a) Teach Pass

(b) Repeat Pass

Fig. 3. The structure of the pose graph during (a) the teach pass and (b) the repeat pass. $\mathcal{F}_r$ is the moving robot frame and others are vertex frames. We use subscript k for vertex frames from the current pass (teach or repeat) and m for vertex frames from the reference pass (always teach). During both teach and repeat passes, we estimate the transformation from the latest vertex frame $\mathcal{F}_k$ to the robot frame $\mathcal{F}_r$, $\hat{\mathbf{T}}_{rk}$, using odometry. For repeat passes only, we define $\mathcal{F}_{k'}$ to be the latest vertex frame that has been successfully localized to the reference pass, $\mathcal{F}_{m'}$ the corresponding map vertex of $\mathcal{F}_{k'}$, and $\mathcal{F}_m$ the spatially closest map vertex to $\mathcal{F}_r$. A prior estimate of the transform from $\mathcal{F}_m$ to $\mathcal{F}_r$, $\check{\mathbf{T}}_{rm}$, is generated by compounding transformations through $\mathcal{F}_{m'}$, $\mathcal{F}_{k'}$, and $\mathcal{F}_k$, which is then used to compute the posterior $\hat{\mathbf{T}}_{rm}$.

as well as their covariances. We present the details of the ICP optimization in Section III-D.

B. Raw Data Preprocessing

1) Lidar: For each incoming lidar scan, we first perform voxel downsampling with voxel size $dl = 0.3$m. Only one point that is closest to the voxel center is kept. Next, we extract plane features from the downsampled pointcloud by applying Principle Component Analysis (PCA) to each point and its neighbors from the raw scan. We define a feature score from PCA to be

$$s = 1 - \lambda_{\min}/\lambda_{\max}, \quad (3)$$

where $\lambda_{\min}$ and $\lambda_{\max}$ are the minimum and maximum eigenvalues, respectively. The downsampled pointcloud is then filtered by this score, keeping no more than 20,000 points with scores above 0.95. We associate each point with its eigenvector of $\lambda_{\min}$ from PCA as the underlying normal.

2) Radar: For each radar scan, we first extract point targets from each azimuth using the Bounded False Alarm Rate (BFAR) detector as described in [37]. BFAR adds a constant offset b to the usual Cell-Averaging CFAR threshold: $T = a \cdot Z + b$. We use the same (a, b) parameters as [37]. For each azimuth, we also perform *peak detection* by calculating the centroid of contiguous groups of detections as is done in [24]. We obtained a modest performance improvement by retaining the maximum of the left and right sub-windows relative to the cell under test as in (greatest-of) GO-CFAR [14]. These polar targets are then transformed into Cartesian coordinates and are passed to the Odometry and Mapping module without further filtering.
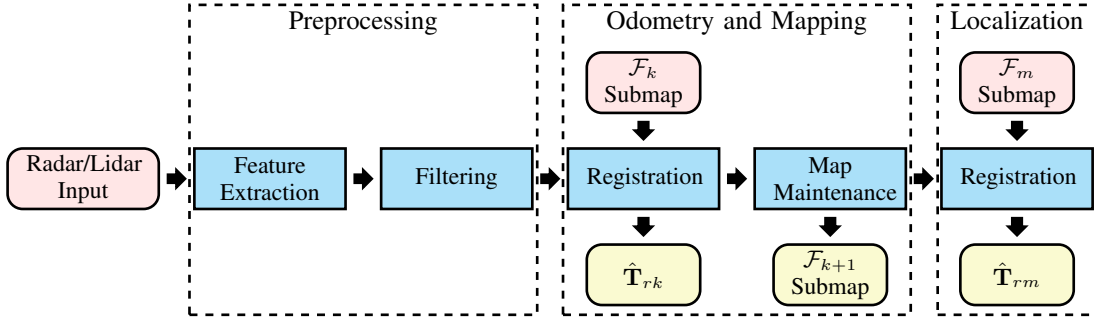
Fig. 4. The data processing pipeline of our Teach and Repeat implementation, divided into three modules: *Preprocessing*, *Odometry and Mapping*, and *Localization*. See III-A for a detailed description of each module.

C. Continuous-Time ICP

Our odometry algorithm, CT-ICP, combines the iterative data association of ICP with a continuous-time trajectory represented as exactly sparse Gaussian Process regression [54]. Our trajectory is $\mathbf{x}(t) = \{\mathbf{T}(t), \boldsymbol{\varpi}(t)\}$, where $\mathbf{T}(t) \in SE(3)$ is our robot pose and $\boldsymbol{\varpi}(t) \in \mathbb{R}^6$ is the body-centric velocity. Following Anderson and Barfoot [54], our GP motion prior is

$$\dot{\mathbf{T}}(t) = \boldsymbol{\varpi}(t)^{\wedge}\mathbf{T}(t),$$
$$\dot{\boldsymbol{\varpi}} = \mathbf{w}(t), \quad \mathbf{w}(t) \sim \mathcal{GP}(\mathbf{0}, \mathbf{Q}_c \delta(t - \tau)), \quad (4)$$

where $\mathbf{w}(t) \in \mathbb{R}^6$ is a zero-mean, white-noise Gaussian process, and the operator, $\wedge$, transforms an element of $\mathbb{R}^6$ into a member of Lie algebra, $\mathfrak{se}(3)$ [55].

The prior (4) is applied in a piecewise fashion between an underlying discrete trajectory of pose-velocity state pairs, $\mathbf{x}_i = \{\mathbf{T}_i, \boldsymbol{\varpi}_i\}$, that each correspond to the representative timestamp of the ith sensor scan. Each pose, $\mathbf{T}_i$, is the relative transform from the latest vertex, $\mathcal{F}_k$, to the robot frame, $\mathcal{F}_r$, that corresponds to the ith sensor scan (i.e., $\mathbf{T}_{rk}$). Likewise, $\boldsymbol{\varpi}_i$ is the corresponding body-centric velocity. We seek to align the latest sensor scan i to the latest vertex submap in frame $\mathcal{F}_k$ (see Figure 3).

We define a nonlinear optimization for the latest state $\mathbf{x}_i$, locking all previous states. The cost function is

$$J_{\text{odom}} = \phi_{\text{motion}} + \underbrace{\sum_{j=1}^{M} \left(\frac{1}{2}\mathbf{e}_{\text{odom},j}^T \mathbf{R}_j^{-1} \mathbf{e}_{\text{odom},j} \right)}_{\text{measurements}}. \quad (5)$$

In the interest of space, we refer readers to Anderson and Barfoot [54] for the squared-error cost expression of the motion prior, ϕ_{motion}. Each measurement error term is

$$\mathbf{e}_{\text{odom},j} = \mathbf{D} \left(\mathbf{p}_k^j - \mathbf{T}(t_j)^{-1}\mathbf{T}_{rs}\mathbf{q}_j \right), \quad (6)$$

where $\mathbf{q}_j$ is a homogeneous point with corresponding timestamp t_j from the ith sensor scan, $\mathbf{T}_{rs}$ is the extrinsic calibration from the sensor frame $\mathcal{F}_s$ to the robot frame $\mathcal{F}_r$, $\mathbf{T}(t_j)$ is a pose from our trajectory queried at t_j[4], $\mathbf{p}_k^j$ is a homogeneous point from the kth submap associated to $\mathbf{q}_j$

---

[4]Through interpolation, $\mathbf{T}(t_j)$ depends on the state variables $\mathbf{x}_i = \{\mathbf{T}_i, \boldsymbol{\varpi}_i\}$ and $\mathbf{x}_{i-1} = \{\mathbf{T}_{i-1}, \boldsymbol{\varpi}_{i-1}\}$ since $t_j > t_{i-1}$ [54].

and expressed in $\mathcal{F}_k$, and $\mathbf{D}$ is a constant projection that removes the 4th homogeneous element. We define $\mathbf{R}_j^{-1}$ as either a constant diagonal matrix for radar data (point-to-point) or by using the outer product of the corresponding surface normal estimate for lidar data (point-to-plane).

We optimize for $\mathbf{x}_i = \{\mathbf{T}_i, \boldsymbol{\varpi}_i\}$ iteratively using Gauss-Newton, but with nearest-neighbours data association after every Gauss-Newton iteration. CT-ICP is therefore performed with the following steps:

1) Temporarily transform all points q_j to frame $\mathcal{F}_k$ using the latest trajectory estimate (motion undistortion).
2) Associate each point to its nearest neighbour in the map to identify its corresponding map point p_k^j in $\mathcal{F}_k$.
3) Formulate the cost function J in (5) and perform a single Gauss-Newton iteration to update $\mathbf{T}_i$ and $\boldsymbol{\varpi}_i$.
4) Repeat steps 1 to 3 until convergence.

The output of CT-ICP at the timestamp of the latest sensor scan is then the odometry output $\hat{\mathbf{T}}_{r,k}$.

D. Localization ICP

We use ICP to localize the motion-compensated live scan of the robot frame, $\mathcal{F}_r$, against the submap of the spatially closest vertex frame, $\mathcal{F}_m$, of the previous sequence. The resulting relative transformation is $\hat{\mathbf{T}}_{rm}$, as shown in Figure 3(b). The nonlinear cost function is

$$J_{\text{loc}} = \phi_{\text{pose}} + \sum_{j=1}^{M} \left(\frac{1}{2}\mathbf{e}_{\text{loc},j}^T \mathbf{R}_j^{-1} \mathbf{e}_{\text{loc},j} \right), \quad (7)$$

where we use $\check{\mathbf{T}}_{rm}$ from (1) as an initial guess and a prior:

$$\phi_{\text{pose}} = \frac{1}{2}\ln(\check{\mathbf{T}}_{rm}\mathbf{T}_{rm}^{-1})^{\vee T} \mathbf{Q}_{rm}^{-1} \ln(\check{\mathbf{T}}_{rm}\mathbf{T}_{rm}^{-1})^{\vee}, \quad (8)$$

where $\ln(\cdot)$ is the logarithm map and the operator $\vee$ is the inverse of $\wedge$ [55]. The covariance $\mathbf{Q}_{rm}$ can be computed by compounding the edge covariances corresponding to the relative transformations in (1). Since all pointclouds are already motion-compensated, the measurement error term is simply

$$\mathbf{e}_{\text{loc},j} = \mathbf{D} \left(\mathbf{p}_m^j - \mathbf{T}_{rm}^{-1}\mathbf{T}_{rs}\mathbf{q}_j \right), \quad (9)$$

where $\mathbf{q}_j$ is a homogeneous point from the motion-compensated live scan, and $\mathbf{p}_m^j$ is the nearest neighbour submap point to $\mathbf{q}_j$. See Section III-C for how we define $\mathbf{T}_{rs}$, $\mathbf{D}$, and $\mathbf{R}_j^{-1}$.
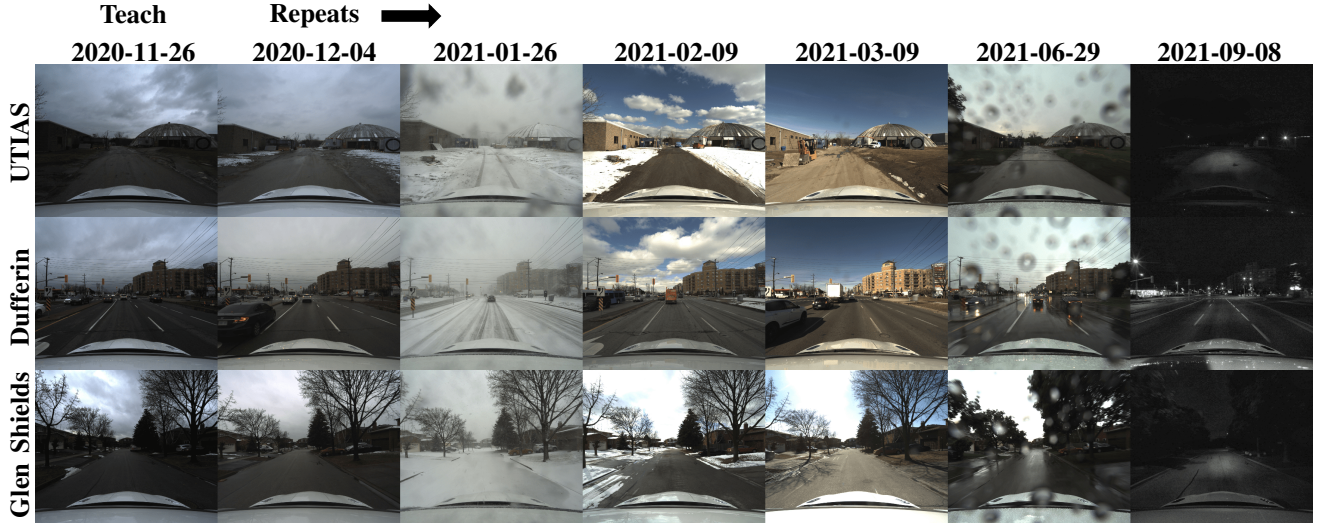
Fig. 5. Our test sequences were collected by driving a repeated route over the source of one year. In our experiments, we use 2020-11-26 as our reference sequence for building maps. The remaining six sequences, which include sequences with rain and snow, are used to benchmark localization performance, which amounts to 48km of test driving. These camera images are provided for context.
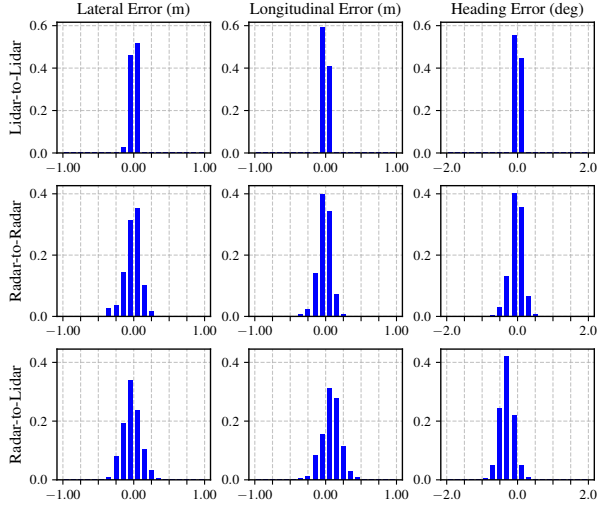

Fig. 6. These histograms show the spread of the localization error for lidar-to-lidar, radar-to-radar, and radar-to-lidar, during the snowstorm sequence 2021-01-26.
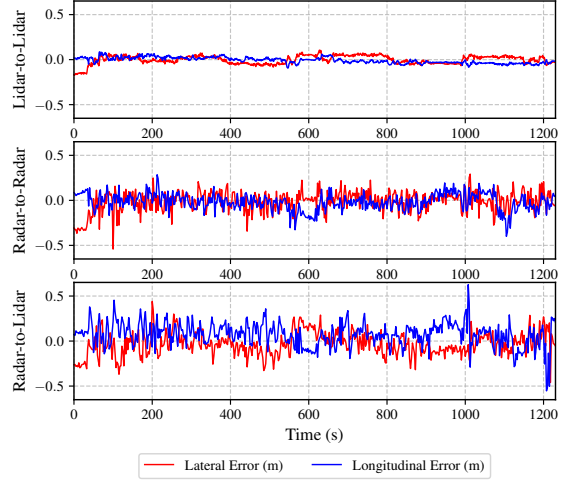

Fig. 7. Here we plot metric localization errors during the snowstorm sequence 2021-01-26. Note that the lidar localization estimates remain accurate even with 1/4 of its field of view being blocked by a layer of ice as shown in Figure 9.

E. Doppler-Compensated ICP

In our prior work [34], we showed that current Navtech radar sensors are susceptible to Doppler distortion and that this effect becomes significant during mapping and localization. A relative velocity between the sensor and the surrounding environment causes the received frequency to be altered according to the Doppler effect. If the velocity in the sensor frame is known, this effect can be compensated for using a simple additive correction factor $\Delta \mathbf{q}_j$. In this work, we include this correction factor, which depends on a continuous-time interpolation of the estimated body-centric velocity $\boldsymbol{\varpi}(t)$ at the measurement time of each target t_j, in the measurement error term for radar CT-ICP:

$$\mathbf{e}_{\text{odom},j} = \mathbf{D}\left(\mathbf{p}_k^j - \mathbf{T}(t_j)^{-1}\mathbf{T}_{rs}(\mathbf{q}_j + \Delta\mathbf{q}_j)\right) \quad (10)$$

$$\text{where } \Delta\mathbf{q}_j = \mathbf{D}^T\beta\mathbf{a}_j\mathbf{a}_j^T\mathbf{D}\mathbf{q}_j^{\odot}\text{Ad}(\mathbf{T}_{sr})\boldsymbol{\varpi}(t_j), \quad (11)$$

β is Doppler distortion constant inherent to the sensor [34], and $\mathbf{a}_j$ is a 3×1 unit vector in the direction of $\mathbf{q}_j$. The $\odot$ operator allows one to swap the order of the operands associated with the $\wedge$ operator [55]. $\text{Ad}(\cdot)$ represents the adjoint of an element of SE(3) [55].

IV. EXPERIMENTAL RESULTS

In this section, we compare the performance of radar-only, lidar-only, and radar-to-lidar topometric localization. Our experimental platform, depicted in Figure 1, includes
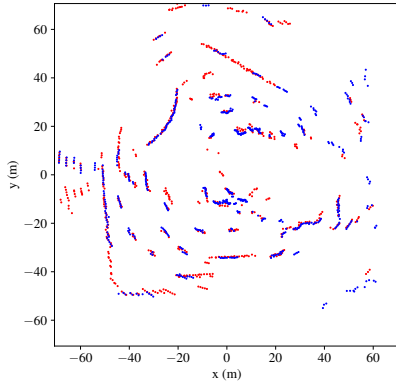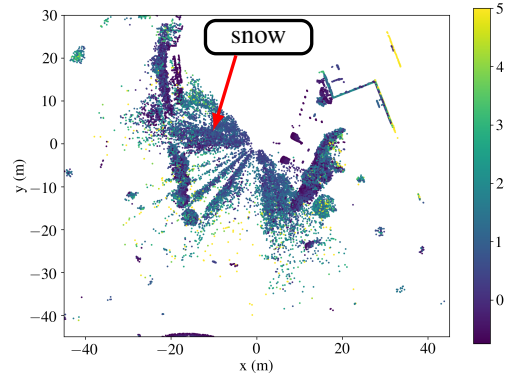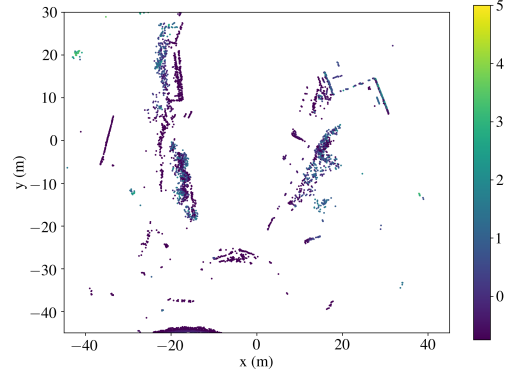
(a) Radar-to-Radar
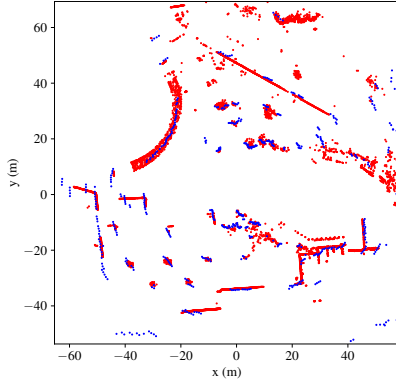
(b) Radar-to-Lidar

Fig. 8. This figure shows the live radar pointcloud (blue) registered to a submap (red) built during the teach pass. In (a) we are performing radar-to-radar localization and so the submap is made up of radar points. In (b) we are localizing a radar pointcloud (blue) to a previously built lidar submap.

(a) Lidar with snow detections

(b) Lidar with outliers removed

Fig. 9. This figure illustrates the noisy lidar data that was used to localize during the 2021-01-26 sequence. Points are colored by their z-height. In (a) the ground plane has been removed and the pointcloud has been randomly downsampled by 50% to highlight the snowflake detections. Note that a large forward section of the lidar's field of view is blocked by a layer of ice. However, as the results in Table I and Figure 7 show, lidar localization remains quite robust under these adverse conditions. (b) shows the lidar pointcloud after filtering points by their normal score as in Equation 3 and only retaining the inliers of truncated least squares. Note that the snowflake detections seem to disappear, illustrating the robustness of our lidar pipeline.

a 128-beam Velodyne Alpha-Prime lidar, a FLIR Blakfly S monocular camera, a Navtech radar, and an Applanix POS LV GNSS-INS. Our lidar has a 40° vertical field of view, 0.1° vertical angular resolution, 0.2° horizontal angular resolution, and produces roughly 220k points per revolution at 10Hz up to 300m. The Navtech is a frequency modulated continuous wave (FMCW) radar with a 0.9° horizontal angular resolution and 5.96cm range resolution, which provides measurements up to 200m at 4Hz. The test sequences used in this paper are part of our new Boreas dataset, which contains over 350km of driving data collected by driving a repeated route over the course of one year [7]. Ground truth poses are obtained by post-processing GNSS, IMU, and wheel encoder measurements. An RTX subscription was used to achieve cm-level accuracy without a base station. RTX uses data from a global network of tracking stations to calculate corrections. The residual error of the post-processed poses reported by Applanix is typically 2-4cm in nominal conditions.

In this experiment, we used seven sequences of the Glen Shields route (shown in Figure 2) chosen for their distinct weather conditions.. These sequences are depicted in Figure 5. During the teach pass, a map is constructed using the reference sequence 2020-11-26. The radar-only and lidar-only pipelines use their respective sensor types to construct

the map. No GPS or IMU information is required during the map-building process. Note that our test sequences include a significant amount of seasonal variation with ten months separating the initial teach pass and the final repeat pass. Sequences 2021-06-29 and 2021-09-08 include trees with full foliage while the remaining sequences lack this. 2021-01-26 was collected during a snowstorm, 2021-06-29 was collected in the rain, and 2021-09-08 was collected at night.

During each of the repeat traversals, our topometric localization outputs a relative localization estimate between the live sensor frame s_2 and a sensor frame in the map s_1: $\hat{\mathbf{T}}_{s_1,s_2}$. We then compute root mean squared error (RMSE) values for the relative translation and rotation error as in [7]. We separate translational error into lateral and longitudinal components. Since the Navtech radar is a 2D sensor, we restrict our comparison to SE(2) by omitting z errors and reporting heading error as the rotation error.

Figure 6 depicts the spread of localization error during sequence 2021-01-26. Note that, although lidar-to-lidar localization is the most accurate, radar-to-radar localization

remains reasonably competitive. When localizing radar scans to lidar maps, both the longitudinal error and heading error incur a bias. The longitudinal bias could be due to some residual Doppler distortion effects, and the heading bias could be the result of an error in the radar-to-lidar extrinsic calibration. A video showcasing radar mapping and localization can be found at this link[5].

Figure 7 shows the localization errors as a function of time during the snowstorm sequence 2021-01-26. Surprisingly, lidar localization appears to be unperturbed by the adverse weather conditions. During the snowstorm sequence, the lidar pointcloud becomes littered with detections associated with snowflakes and a large section of the horizontal field of view becomes blocked by a layer of ice as shown in Figure 9 (a). However, in Figure 9 (b), we show that in actuality, these snowflake detections have little impact on ICP registration after filtering by normal score and only retaining the inliers of truncated least squares. Charron et al. [56] previously demonstrated that snowflake detections can be removed from lidar pointclouds, although we do not use their method here. The robustness of our lidar pipeline to both weather and seasonal variations is reflected in the RMSE results displayed in Table I.

Our results show that, contrary to the assertions made by prior works, lidar localization can be robust to moderate levels of precipitation and seasonal variation. Clearly, more work is required by the community to identify operational conditions where radar localization has a clear advantage. These conditions may include very heavy precipitation, dense fog, or dust clouds. Nevertheless, we demonstrated that our radar localization is reasonably competitive with lidar. Furthermore, radar localization may still be important as a redundant backup system in autonomous vehicles. Figure 8 illustrates the live scan and submap during radar-to-radar and radar-to-lidar localization.

It is important to recognize that the results reported in this work are taken at a snapshot in time. Radar localization is not as mature of a field as lidar localization and radar sensors themselves still have room to improve. Note that incorporating IMU or wheel encoder measurements would improve the performance of all three compared systems. The detector we used, BFAR [37], did not immediately work when applied to a new radar with different noise characteristics. It is possible that a learning-based approach to feature extraction and matching may improve performance. Switching to a landmark-based pipeline or one based on image correlation may also be interesting avenues for comparison.

Radar-to-lidar localization is attractive because it allows us to use existing lidar maps, which many autonomous driving companies already have, while taking advantage of the robustness of radar sensing. Radar-based maps are not as useful as lidar maps since they lack sufficient detail to be used to create semantic maps.

In Table II, we show the computational and storage requirements of the different pipelines discussed in this work.

[5] https://youtu.be/okS7pF6xX7A

TABLE I
METRIC LOCALIZATION RMSE RESULTS
REFERENCE SEQUENCE: 2020-11-26

	Lidar-to-Lidar		
	lateral (m)	longitudinal (m)	heading (deg)
2020-12-04	0.057	0.082	0.025
2021-01-26	0.047	0.034	0.034
2021-02-09	0.044	0.037	0.029
2021-03-09	0.049	0.040	0.022
2021-06-29	0.052	0.058	0.042
2021-09-08	0.060	0.041	0.030
mean	**0.052**	**0.049**	**0.030**
	Radar-to-Radar		
	lateral (m)	longitudinal (m)	heading (deg)
2020-12-04	0.130	0.127	0.223
2021-01-26	0.118	0.098	0.201
2021-02-09	0.111	0.089	0.203
2021-03-09	0.129	0.093	0.213
2021-06-29	0.163	0.155	0.241
2021-09-08	0.155	0.148	0.231
mean	**0.134**	**0.118**	**0.219**
	Radar-to-Lidar		
	lateral (m)	longitudinal (m)	heading (deg)
2020-12-04	0.132	0.179	0.384
2021-01-26	0.130	0.153	0.360
2021-02-09	0.126	0.152	0.389
2021-03-09	0.140	0.152	0.360
2021-06-29	0.159	0.161	0.415
2021-09-08	0.169	0.165	0.401
mean	**0.143**	**0.161**	**0.385**

TABLE II
COMPUTATIONAL AND STORAGE REQUIREMENTS

	Odom. (FPS)	Loc. (FPS)	Storage (MB/km)
Lidar-to-Lidar	3.6	3.0	86.4
Radar-to-Radar	**5.3**	**5.1**	**5.6**
Radar-to-Lidar	N/A	5.1	86.4

We used a Lenovo P53 laptop with Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz and 32GB of memory. A GPU was not used. Our radar-based maps use significantly less storage (5.6MB/km) than our lidar-based maps (86.4MB/km).

V. CONCLUSIONS

In this work, we compared the performance of lidar-to-lidar, radar-to-radar, and radar-to-lidar topometric localization. Our results showed that radar-based pipelines are a viable alternative to lidar localization but lidar continues to yield the best results. Surprisingly, our experiments showed that lidar-only mapping and localization is quite robust to adverse weather such as a snowstorm with a partial sensor blockage due to ice. We identified several areas for future work and noted that more experiments are needed to identify conditions where the performance of radar-based pipelines exceeds that of lidar.

REFERENCES

[1] J. Levinson and S. Thrun, "Robust vehicle localization in urban environments using probabilistic maps," in *ICRA*. IEEE, 2010.
[2] R. W. Wolcott and R. M. Eustice, "Fast lidar localization using multiresolution gaussian mixture maps," in *ICRA*. IEEE, 2015.
[3] J. Behley and C. Stachniss, "Efficient surfel-based slam using 3d laser range data in urban environments." in *Robotics: Science and Systems*, 2018.

[4] M. Gridseth and T. D. Barfoot, "Keeping an eye on things: Deep learned features for long-term visual localization," *RA-L*, 2021.

[5] G. M. Brooker, S. Scheding, M. V. Bishop, and R. C. Hennessy, "Development and application of millimeter wave radar sensors for underground mining," *IEEE Sensors journal*, vol. 5, no. 6, pp. 1270–1280, 2005.

[6] G. Brooker, M. Bishop, and S. Scheding, "Millimetre waves for robotics," in *Australian Conference for Robotics and Automation*, 2001.

[7] K. Burnett, D. J. Yoon, Y. Wu, A. Z. Li, H. Zhang, S. Lu, J. Qian, W.-K. Tseng, A. Lambert, K. Y. Leung, A. P. Schoellig, and T. D. Barfoot, "Boreas: A multi-season autonomous driving dataset," *arXiv preprint arXiv:2203.10168*, 2022.

[8] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *CVPR*, 2020.

[9] D. Kellner, M. Barjenbruch, J. Klappstein, J. Dickmann, and K. Diet-mayer, "Instantaneous ego-motion estimation using doppler radar," in *ITSC*. IEEE, 2013.

[10] P.-C. Kung, C.-C. Wang, and W.-C. Lin, "A normal distribution transform-based radar odometry designed for scanning and automotive radars," in *ICRA*. IEEE, 2021.

[11] P. Gao, S. Zhang, W. Wang, and C. X. Lu, "Accurate automotive radar based metric localization with explicit doppler compensation," *arXiv preprint arXiv:2112.14887*, 2021.

[12] S. Clark and H. Durrant-Whyte, "Autonomous land vehicle navigation using millimeter wave radar," in *ICRA*. IEEE, 1998.

[13] M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (slam) problem," *T-RO*, 2001.

[14] H. Rohling, "Radar cfar thresholding in clutter and multiple target situations," *IEEE transactions on aerospace and electronic systems*, 1983.

[15] E. Jose and M. D. Adams, "Relative radar cross section based feature identification with millimeter wave radar for outdoor slam," in *IROS*. IEEE, 2004.

[16] M. Chandran and P. Newman, "Motion estimation from map quality with millimeter wave radar," in *IROS*. IEEE, 2006.

[17] R. Rouveure, M. Monod, and P. Faure, "High resolution mapping of the environment with a ground-based radar imager," in *RADAR*. IEEE, 2009.

[18] P. Checchin, F. Gérossier, C. Blanc, R. Chapuis, and L. Trassoudaine, "Radar scan matching slam using the fourier-mellin transform," in *FSR*. Springer, 2010.

[19] J. Callmer, D. Törnqvist, F. Gustafsson, H. Svensson, and P. Carlbom, "Radar slam using visual features," *EURASIP Journal on Advances in Signal Processing*, 2011.

[20] J. Mullane, B.-N. Vo, M. D. Adams, and B.-T. Vo, "A random-finite-set approach to bayesian slam," *T-RO*, 2011.

[21] D. Vivet, P. Checchin, and R. Chapuis, "Localization and mapping using only a rotating fmcw radar sensor," *Sensors*, 2013.

[22] F. Schuster, C. G. Keller, M. Rapp, M. Haueis, and C. Curio, "Landmark based radar slam using graph optimization," in *ITSC*. IEEE, 2016.

[23] M. Rapp, M. Barjenbruch, M. Hahn, J. Dickmann, and K. Dietmayer, "Probabilistic ego-motion estimation using multiple automotive radar sensors," *Robotics and Autonomous Systems*, 2017.

[24] S. H. Cen and P. Newman, "Precise ego-motion estimation with millimeter-wave radar under diverse and challenging conditions," in *ICRA*. IEEE, 2018.

[25] 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 *ICRA*. IEEE, 2020.

[26] G. Kim, Y. S. Park, Y. Cho, J. Jeong, and A. Kim, "Mulran: Multi-modal range dataset for urban place recognition," in *ICRA*. IEEE, 2020.

[27] M. Sheeny, E. De Pellegrin, S. Mukherjee, A. Ahrabian, S. Wang, and A. Wallace, "Radiate: A radar dataset for automotive perception in bad weather," in *ICRA*. IEEE, 2021.

[28] S. H. Cen and P. Newman, "Radar-only ego-motion estimation in difficult settings via graph matching," in *ICRA*. IEEE, 2019.

[29] R. Aldera, D. De Martini, M. Gadd, and P. Newman, "What could go wrong? introspective radar odometry in challenging environments," in *ITSC*. IEEE, 2019.

[30] ——, "Fast radar motion estimation with a learnt focus of attention using weak supervision," in *ICRA*. IEEE, 2019.

[31] D. Barnes, R. Weston, and I. Posner, "Masking by moving: Learning distraction-free radar odometry from pose information," in *Conference on Robot Learning*, 2020.

[32] D. Barnes and I. Posner, "Under the radar: Learning to predict robust keypoints for odometry estimation and metric localisation in radar," in *ICRA*. IEEE, 2020.

[33] Y. S. Park, Y.-S. Shin, and A. Kim, "Pharao: Direct radar odometry using phase correlation," in *ICRA*, 2020.

[34] K. Burnett, A. P. Schoellig, and T. D. Barfoot, "Do we need to compensate for motion distortion and doppler effects in spinning radar navigation?" *RA-L*, 2021.

[35] K. Burnett, D. J. Yoon, A. P. Schoellig, and T. D. Barfoot, "Radar odometry combining probabilistic estimation and unsupervised feature learning," in *Robotics: Science and Systems*, 2021.

[36] D. Adolfsson, M. Magnusson, A. Alhashimi, A. J. Lilienthal, and H. Andreasson, "Cfear radarodometry-conservative filtering for efficient and accurate radar odometry," in *IROS*. IEEE, 2021.

[37] A. Alhashimi, D. Adolfsson, M. Magnusson, H. Andreasson, and A. J. Lilienthal, "Bfar-bounded false alarm rate detector for improved radar odometry estimation," *arXiv preprint arXiv:2109.09669*, 2021.

[38] M. Holder, S. Hellwig, and H. Winner, "Real-time pose graph slam based on radar," in *IV*. IEEE, 2019.

[39] Z. Hong, Y. Petillot, and S. Wang, "Radarslam: Radar based large-scale slam in all weathers," in *IROS*. IEEE, 2020.

[40] Z. Hong, Y. Petillot, A. Wallace, and S. Wang, "Radarslam: A robust simultaneous localization and mapping system for all weather conditions," *IJRR*, 2022.

[41] D. De Martini, M. Gadd, and P. Newman, "kradar++: Coarse-to-fine fmcw scanning radar localisation," *Sensors*, 2020.

[42] Ş. Săftescu, M. Gadd, D. De Martini, D. Barnes, and P. Newman, "Kidnapped radar: Topological radar localisation using rotationally-invariant metric learning," in *ICRA*. IEEE, 2020.

[43] M. Gadd, D. De Martini, and P. Newman, "Look around you: Sequence-based radar place recognition with learned rotational invariance," in *PLANS*, 2020.

[44] T. Y. Tang, D. De Martini, D. Barnes, and P. Newman, "Rsl-net: Localising in satellite images from a radar on the ground," *RA-L*, 2020.

[45] T. Y. Tang, D. De Martini, S. Wu, and P. Newman, "Self-supervised localisation between range sensors and overhead imagery," in *Robotics: Science and Systems*, 2020.

[46] T. Y. Tang, D. De Martini, and P. Newman, "Get to the point: Learning lidar place recognition and metric localisation using overhead imagery," in *Robotics: Science and Systems*, 2021.

[47] H. Yin, Y. Wang, L. Tang, and R. Xiong, "Radar-on-lidar: metric radar localization on prior lidar maps," in *2020 IEEE International Conference on Real-time Computing and Robotics (RCAR)*. IEEE, 2020.

[48] H. Yin, R. Chen, Y. Wang, and R. Xiong, "Rall: end-to-end radar localization on lidar map using differentiable measurement model," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[49] P. Furgale and T. D. Barfoot, "Visual teach and repeat for long-range rover autonomy," *JFR*, 2010.

[50] M. Paton, K. MacTavish, M. Warren, and T. D. Barfoot, "Bridging the appearance gap: Multi-experience localization for long-term visual teach and repeat," in *IROS*, 2016.

[51] M. Paton, K. MacTavish, L.-P. Berczi, S. K. van Es, and T. D. Barfoot, "I can see for miles and miles: An extended field test of visual teach and repeat 2.0," in *FSR*. Springer, 2018.

[52] P. Krüsi, B. Bücheler, F. Pomerleau, U. Schwesinger, R. Siegwart, and P. Furgale, "Lighting-invariant adaptive route following using iterative closest point matching," *JFR*, 2015.

[53] H. Badino, D. Huber, and T. Kanade, "Visual topometric localization," in *IV*, 2011.

[54] S. Anderson and T. D. Barfoot, "Full steam ahead: Exactly sparse gaussian process regression for batch continuous-time trajectory estimation on se(3)," in *IROS*, 2015.

[55] T. D. Barfoot, *State Estimation for Robotics*. Cambridge University Press, 2017.

[56] N. Charron, S. Phillips, and S. L. Waslander, "De-noising of lidar point clouds corrupted by snowfall," in *CRV*. IEEE, 2018.