

FMCW Radar on LiDAR map localization in structural urban environments

Yukai Ma  | Han Li | Xiangrui Zhao | Yaqing Gu | Xiaolei Lang | Lajian Li | Yong Liu

April Laboratory, Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou, China

Correspondence

Yong Liu, Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou, China.

Email: yongliu@ipc.zju.edu.cn

Funding information

National Natural Science Foundation of China, Grant/Award Number: NSFC 62088101

Abstract

Multisensor fusion-based localization technology has achieved high accuracy in autonomous systems. How to improve the robustness is the main challenge at present. The most commonly used LiDAR and camera are weather-sensitive, while the frequency-modulated continuous wave Radar has strong adaptability but suffers from noise and ghost effects. In this paper, we propose a heterogeneous localization method called Radar on LiDAR Map, which aims to enhance localization accuracy without relying on loop closures by mitigating the accumulated error in Radar odometry in real time. To accomplish this, we utilize LiDAR scans and ground truth paths as Teach paths and Radar scans as the trajectories to be estimated, referred to as Repeat paths. By establishing a correlation between the Radar and LiDAR scan data, we can enhance the accuracy of Radar odometry estimation. Our approach involves embedding the data from both Radar and LiDAR sensors into a density map. We calculate the spatial vector similarity with an offset to determine the corresponding place index within the candidate map and estimate the rotation and translation. To refine the alignment, we utilize the Iterative Closest Point algorithm to achieve optimal matching on the LiDAR submap. The estimated bias is subsequently incorporated into the Radar SLAM for optimizing the position map. We conducted extensive experiments on the Mulran Radar Data set, Oxford Radar RobotCar Dataset, and our data set to demonstrate the feasibility and effectiveness of our proposed approach. Our proposed scan projection descriptors achieves homogeneous and heterogeneous place recognition and works much better than existing methods. Its application to the Radar SLAM system also substantially improves the positioning accuracy. All sequences' root mean square error is 2.53 m for positioning and 1.83° for angle.

KEYWORDS

3D scan registration, GPS denied, localization

1 | INTRODUCTION

Localization is a critical aspect of autonomous driving, and it relies on various sensors, including global positioning system (GPS), cameras, Radar, and LiDAR. However, there are limitations to the direct application of some sensors in specific environments. For example, changes in illumination may compromise visual localization accuracy, and GPS can become unreliable in urban canyons.

Out of these sensors, LiDAR shines for its high precision and widespread usage in robotics and autonomous driving, specifically for localization and mapping tasks. LiDAR achieves accurate mapping by employing point cloud registration, which allows it to function effectively even in changing lighting conditions.

Aside from real-time simultaneous localization and mapping (SLAM), LiDAR measurements taken in favorable weather conditions can be utilized to construct offline prebuilt maps. These maps can be augmented with data from sensors such as inertial measurement units (IMUs) and GPS to enhance both local and global information.

Nevertheless, LiDAR still faces accuracy challenges in extreme weather conditions such as rain, snow, and fog, which results in reduced performance. To counter this issue, millimeter-wave Radar, utilizing longer-wavelength microwaves, has been integrated into SLAM applications. Although Radar has its drawbacks, including multipath phenomena, ghost reflections, higher noise levels, and lower accuracy compared to LiDAR, it demonstrates exceptional robustness and is capable of long-range detection and map construction during severe weather conditions. Research in this area has gained significant attention in recent years (Hong et al., 2020, 2021).

Therefore, Radar localization on prebuilt LiDAR maps will complement each other (Yin, Chen, et al., 2022; Yin, Xu, et al., 2021; Yin et al., 2020). LiDAR maps can compensate for the noise and sparsity of Radar data. Moreover, since most existing maps are constructed using LiDAR, the proposed method in this paper avoids redundant mapping or calibration efforts, significantly improving efficiency. Furthermore, integrating Radar data for positioning enhances localization robustness in all weather conditions. However, there are two challenges in matching and aligning the Radar data and the LiDAR map: (1) The Navtech frequency-modulated continuous wave (FMCW) Radar can only obtain 2D information of the sweep line plane, which is one dimension less than the LiDAR. (2) LiDAR point clouds can provide detailed outlines of even small objects, whereas Radar polar data can only approximate changes in reflectivity in a scene. This disparity results in a lack of direct correspondence between LiDAR and Radar points in space.

To establish a standard for measuring the similarity between 2D and 3D data, we consider the concept of projection-based dimensionality reduction. We employ vectors with offsets to map the heterogeneous data to a unified vector space. To address the occlusions and ghost reflections in Radar images, we extract keypoints (Burnett et al., 2021) from each frame and stack the features of consecutive frames. The pose estimation process of the system can be divided into four steps. First, the initial pose estimation

is obtained from Radar odometry. Second, a LiDAR frame similar to the Radar keyframe is identified, and its extrinsic parameters are calculated. Next, the deviation between the current position and the map pose is determined. Finally, an optimization method using a heterogeneous pose graph is introduced to refine the pose estimation. To verify the feasibility and effectiveness, we validate our algorithm on the Mulran Data set (Kim et al., 2020), Oxford Radar RobotCar Data set (Maddern et al., 2017), and our ZJU Radar Data sets (Figure 7). Figure 1 shows the effect of our method on the Oxford data set.

In general, the contribution of this paper can be summarized as follows:

- Radar-to-LiDAR localization: We utilize Radar-to-LiDAR localization to eliminate odometry drift. By leveraging the complementary strengths of Radar and LiDAR sensors, we can improve the accuracy and robustness of the SLAM system.
- RoLM: We introduce a new feature description and matching method called Radar on LiDAR Map (RoLM). This method allows us to retrieve the corresponding position index from historical LiDAR observations and estimate the coarse transformation. By incorporating RoLM, we enhance the accuracy of the SLAM system.
- Sensor association in pose graph optimization: Heterogeneous sensor association is added to the sliding window pose graph optimization. By considering the measurements from multiple sensors, we effectively improve the localization accuracy of the SLAM system.
- Availability of new data sets: We have created a new mobile cart Radar data set (Figure 7), which is publicly available.¹ This data set, along with extensive experiments conducted on the Mulran Radar Data set (Kim et al., 2020) and the Oxford Radar RobotCar Data set (Barnes et al., 2020; Maddern et al., 2017), validate the effectiveness and feasibility of the proposed system.

2 | RELATED WORK

2.1 | Radar SLAM

Radar SLAM has been a hotspot in recent years. Radar sensors can provide multiple levels of data, including signals, images, or point clouds. Many algorithms have been adapted from vision or LiDAR platforms (Marck et al., 2013; Săftescu et al., 2020) to process millimeter wave data as images or point clouds for the front-end processing of Radar data. There are two general Radar feature extraction approaches: traditional (Cen & Newman, 2018, 2019) and neural network (Aldera et al., 2019; Barnes & Posner, 2020) methods. Traditional methods typically involve hand-crafted feature extraction techniques, such as edge detection, corner detection, or other image processing techniques, applied to Radar images or point clouds.

¹<https://github.com/HR-zju/ZJU-Radar-Dataset.git>

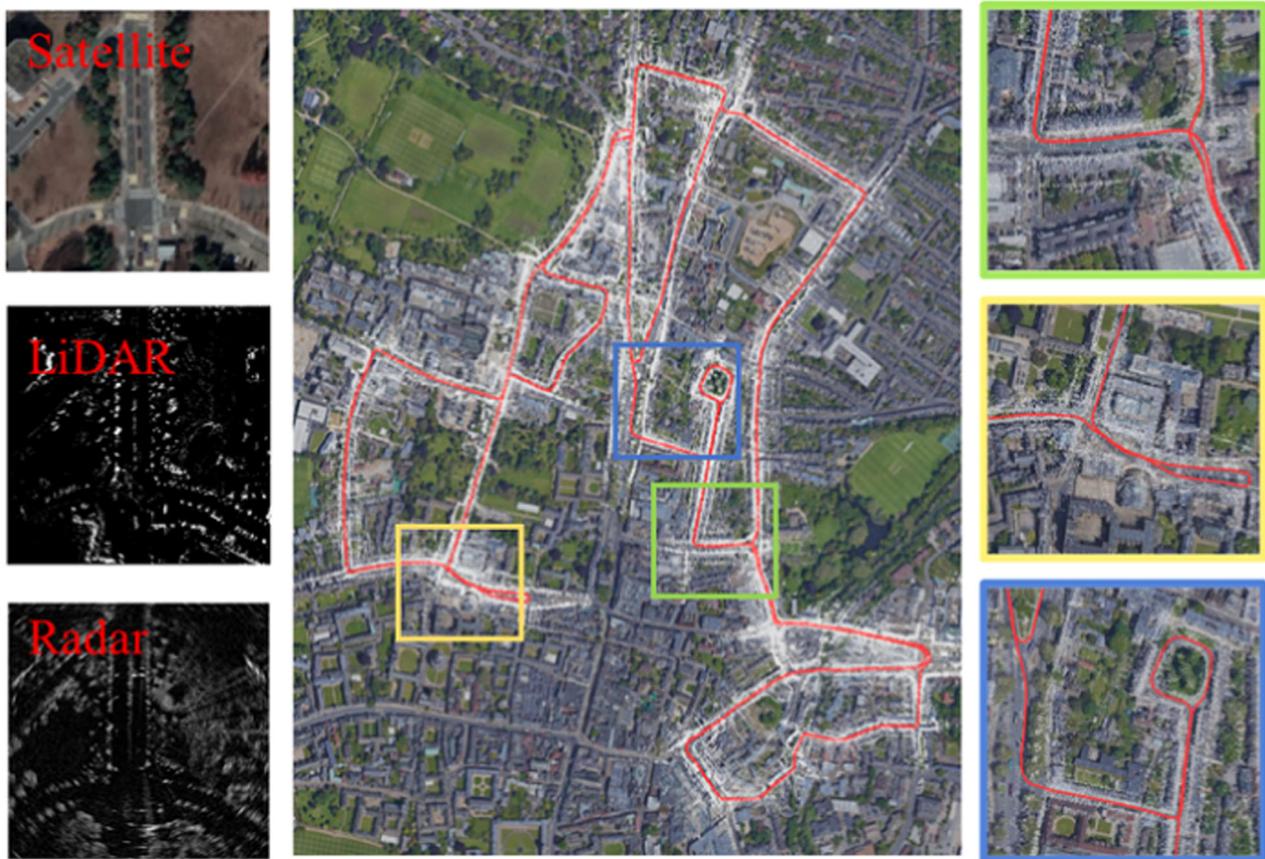


FIGURE 1 Radar odometry generated using RoLM in which the colorful box shows some details. The left side of the figure provides the difference between LiDAR data and Radar data in the same scene.

These methods rely on explicit rules or mathematical algorithms to extract meaningful features from Radar data. On the other hand, neural network methods utilize deep learning techniques to automatically learn features from Radar data. Deep neural networks can be trained to extract features directly from Radar images or point clouds without relying on explicit rules. These methods have shown promising results in Radar SLAM, as they can capture complex patterns and representations in Radar data, leading to improved accuracy and robustness in feature extraction tasks. Traditional and neural network methods have advantages and limitations in Radar feature extraction for SLAM. Traditional methods may be computationally efficient and interpretable but need help to capture complex and abstract features in Radar data. On the other hand, neural network methods may offer higher accuracy and flexibility in feature extraction, but they may require more extensive training data and computational resources. The approach choice depends on the Radar SLAM system's specific requirements and constraints and the available data and resources.

In this paper, we have focused on traditional methods for Radar feature extraction in the context of Radar SLAM. Cen et al. proposed a feature detection method in 2018 that scales the Radar power spectrum according to its truth probability to address the issue of redundant keypoints and false positives generated by the constant

false alarm rate (CFAR) (Cen & Newman, 2018). They later proposed an updated detector in 2019 that identifies regions with high intensity and low gradient in the continuously scanned region (Cen & Newman, 2019). Based on Cen's work, Burnett et al. introduced the Yeti Radar Odometry algorithm to eliminate motion distortion and the Doppler effect in Radar data using a Gaussian filter instead of a binomial filter. This method also mitigated the impact of multipath reflections. The researchers found that Cen (2018), combined with their RANSAC-based matching method, performs well. After feature extraction, the original Radar data in polar coordinates are converted into Cartesian form. The ORB descriptor is then computed for each keypoint using the ORB descriptor method for violent matching, and mismatches are removed using a distinctive feature-based method (Lowe, 2004; Rublee et al., 2011). The remaining matches are sent to an MC-RANSAC-based estimator to exclude outliers while correcting motion distortion (Anderson & Barfoot, 2013).

There being few pieces of research on unstructured, disordered, and sparse point clouds currently, Kim from KAIST University proposed the Scan Context method for 3D point cloud relocation and scene recognition (Kim & Kim, 2018). The main idea of Scan Context is to compress the 3D information of the scene and convert it from Cartesian coordinates to polar coordinates for calculation. However, the original Scan Context method has limitations in

handling lateral motion and efficiency. Therefore, the authors proposed Scan Context++, which uses two descriptor representations of polar and Cartesian contexts to handle rotation and translation robustly, and sub-descriptors for efficient information compression (Kim et al., 2022).

2.2 | Localization on prebuilt maps

The concept of localization on prebuilt maps is closely related to SLAM, with high real-time requirements. A prebuilt map can eliminate the need for repeated online mapping in long-term fixed systems, thereby improving efficiency. Existing localization algorithms on prebuilt maps include visual localization on visual and LiDAR maps (Ding et al., 2019; Huang et al., 2020), and LiDAR localization on LiDAR maps (Yin et al., 2019). However, these methods still need to be revised in terms of robustness. Compared to LiDAR, Radar has the advantage of being able to penetrate smoke and dust, making it suitable for all-weather and anti-interference localization research in both indoor and outdoor scenes (Clark & Dissanayake, 1999; Jose & Adams, 2005). In recent years, Navtech has provided Radar sensors with higher accuracy, less motion distortion, and a 360° range for research, resulting in rich data sets (Barnes et al., 2020; Kim et al., 2020; Maddern et al., 2017; Sheeny et al., 2021) and various algorithms (Burnett et al., 2021; Hong et al., 2020). Nevertheless, Radar also has limitations, such as being susceptible to the Doppler effect and noise, and its accuracy may be lower than LiDAR. As a result, Radar-based localization algorithms often require graph optimization (Holder et al., 2019; Schuster et al., 2016) or sensor fusion with information from other sensors to improve accuracy and robustness.

Yin proposed a Radar-on-LiDAR localization algorithm in their work (Yin et al., 2020), which utilized a conditional generative adversarial network (GAN) called pix2pix (Isola et al., 2017). The GAN

was trained to transfer Radar data to fake LiDAR points. Subsequently, a Monte Carlo localization (MCL) system is built to achieve accurate localization on prebuilt LiDAR maps. Later, Yin also proposed an end-to-end learning system for localization in their work (Yin, Chen, et al., 2022). This system used back-propagation of gradients from pose supervision to achieve localization and also incorporated a Kalman Filter to improve accuracy. A recent study by Yin, Xu, et al. (2021) introduced a heterogeneous place recognition method via joint learning (JL). This method involved joint training to extract shared embeddings from Radar and LiDAR data for place recognition. Furthermore, heterogeneous prior constraints are added to the factor graph for global optimization, enhancing the accuracy of the localization. A recent study by Yin, Xu, et al. (2021) introduced a heterogeneous place recognition method via JL. This approach utilizes satellite maps as an input for the localization algorithm, providing additional information for accurate localization.

Overall, there are several algorithms proposed in the literature that utilize Radar, LiDAR, or satellite maps for prebuilt map localization, employing techniques such as GANs, Monte Carlo localization, end-to-end learning, Kalman Filter, and joint training to improve accuracy and robustness in various ways.

3 | SYSTEM DESIGN

3.1 | Overview and motivation

This section introduces the proposed system. Figure 2 illustrates the proposed RoLM framework of the system. Unlike existing methods for matching heterogeneous sensor information, we use scan projection descriptors (SPD) to describe their similarity without using GPU for acceleration. For a set of LiDAR scans and Radar polar data, we preprocess them separately. The LiDAR data is processed into a

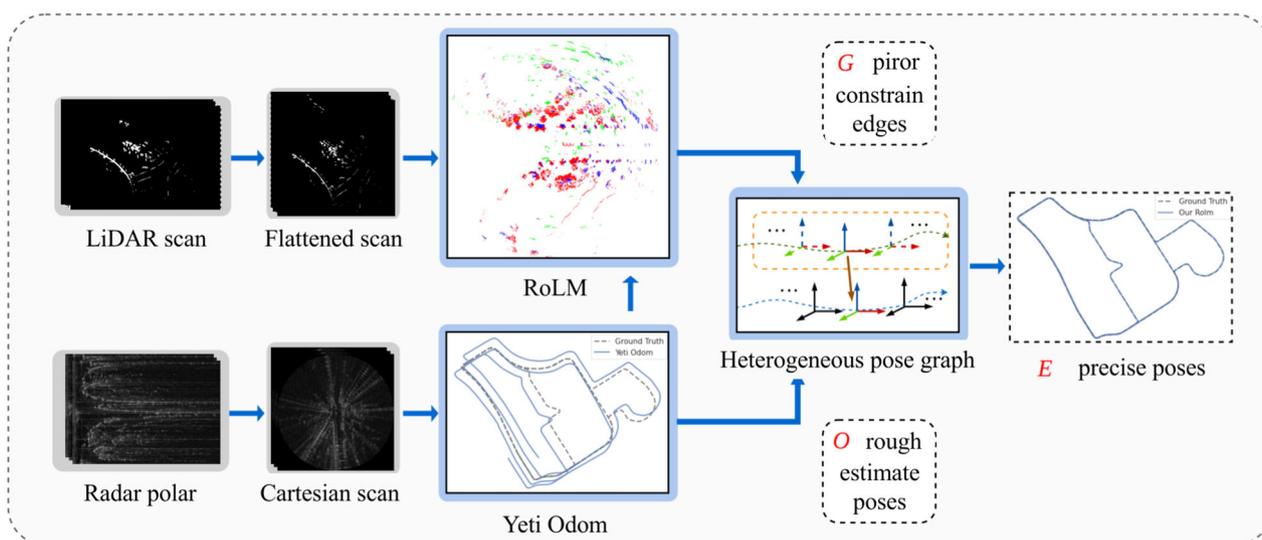


FIGURE 2 The overall framework. Given the raw range measurements, RoLM can find the corresponding location index from a set of locations in the map and compute the pose bias to add to pose graph optimization.

Teach sequence, while the Radar data is processed into a Repeat sequence, as explained in Section 3.2. To perform Radar odometry (Burnett et al., 2021), we employ our proposed RoLM to extract scene descriptions from both Radar and LiDAR data (Section 3.4). Subsequently, we align these descriptions using a coarse-to-fine method (Section 3.5). This alignment produces a heterogeneous constrained edge incorporated into the pose graph optimization (Section 3.6). The algorithm's main steps are presented in Algorithm 1.

using the rough transform ${}^{SP}T_{r \rightarrow l}$. g_k^w denotes the pose of the most similar LiDAR scan $L_{k,s}$.

The Scan Context descriptor, proposed by Kim and Kim (2018), is designed explicitly for closed-loop detection in LiDAR odometry and has shown good performance in urban environments. The descriptor uses the highest point within each point cloud block as a bin and measures the similarity between scenes based on the distances between columns of ray vectors. In a related work by Kim et al. (2022), intensity information is encoded on the Cartesian Radar

Algorithm 1: Multi-modal Localization

Input: A set of Radar Cartesian images \mathcal{R} and initial pose \mathbf{O}_0^w

Output: 3DOF precise pose at every imaging time \mathbf{E}_i

Thread 1:

for *image* $i \in \mathcal{R}$ do

```

  Pre-processing
  Feature extractions
  Feature tracking across images
  Feature matching based Yeti odometry
  if isKeyframe( $i$ ) then
    Keyframe submap  ${}^wF_k$  generated by Equation 2
     $\mathcal{W} \leftarrow \mathcal{W} + \{[{}^wF_k, \mathbf{O}_k]\}$ 

```

// Pre-processing & Radar Odometry

Thread 2:

for $w({}^wF_k, \mathbf{O}_k) \in \mathcal{W}$ do

```

  Candedated LiDAR scans  $L_k \leftarrow \text{radiusSearch}(\mathbf{O}_k, R_{\text{research}})$ 
   $S_R \leftarrow$  Equation 3 ( ${}^wF_k$ );  $S_L \leftarrow$  Equation 3 ( $\{L_{k,i}\}$ )
  Calculate the smallest  $D(S_R, S_L)$  by Equation 6 to find the most similar LiDAR scan  $L_{k,s}$ 
  Calculate coarse transformation  ${}^{SP}T_{r \rightarrow l}$  by Equation 7
   ${}^wF'_k \leftarrow \text{transformPointCloud}({}^wF_k, {}^{SP}T_{r \rightarrow l})$ 
   ${}^{icp}T_{r \rightarrow l} \leftarrow \text{ICP}({}^wF'_k, L_{k,s}); *T_{r \rightarrow l} \leftarrow {}^{SP}T_{r \rightarrow l} \cdot {}^{icp}T_{r \rightarrow l}$ 
  if gtFactorUseful() then
     $\mathcal{G} \leftarrow \mathcal{G} + \{[g_k^w, *T_{r \rightarrow l}]\}$ 

```

// Coarse-to-Fine Multi-model alignment

Thread 3:

Create Optimizer

Add odometer node \mathbf{O}_i and edge \mathbf{O}_{ij}

for $g_k^w \in \mathcal{G}$ do

```

  Add prior node  $g_k^w$  and edge  $T_{r \rightarrow l, k}$ 

```

$\mathbf{E} \leftarrow \text{updatePoses}(\mathbf{O})$

// Heterogeneous Pose Graph Optimization

In our algorithm, we utilize the following variables: wF_k represents the Radar submap centered on the middle frame in the k th sliding window. \mathbf{O}_k represents the middle position of the scans within the window. L_k represents the set of LiDAR candidate frames for the k th Radar keymap within a given radius R_{research} . S_R and S_L are descriptors of the local Radar submap and its corresponding set of candidate LiDAR point clouds, respectively. ${}^{SP}T_{r \rightarrow l}$ and ${}^{ICP}T_{r \rightarrow l}$ are the SE3 transforms of the Radar point clouds computed using SPD and ICP methods, respectively, concerning the LiDAR point clouds. ${}^wF'_k$ represents the transformed wF_k in the LiDAR coordinate system,

image instead of using height information. However, there is no direct connection between the power of Radar points and the height of LiDAR points.

When considering a nontransparent object in 3D space, it appears with a clear outline and geometric structure in the LiDAR point cloud. On the other hand, its edge seems to be blurred in the Radar point cloud. Although the correspondence between LiDAR and Radar points may be vague, it tends to be explicit between clusters of their point clouds. The density of a point cloud can indicate various properties of an object, such as its size, thickness, and hollowness.

For example, a tree's point cloud density will be sparser than a wall's, making it more easily distinguishable when viewed from a top-down perspective.

3.2 | Teach and repeat

In Figure 3, \mathcal{R} denotes the set of Radar keymaps. Subscript k indicates the keyframes during Radar odometry. \mathcal{M} denotes the LiDAR fixation frames on the Teach sequence, which consist of m vertices connected by edges representing relative positional transformations. Once all the Radar scan frame $\mathcal{R}_{k,i} \in \mathcal{R}_k$ are collected, we estimate the transformation from \mathcal{R}_k to the map frame \mathcal{M} . This estimation combines the odometry-estimated transformation ${}^*T_{r \rightarrow l}$ from \mathcal{R}_{k-1} to \mathcal{R}_k with the edge $T_{k-1 \rightarrow k}$, and optimally adjusts the entire pose graph. In publicly available datasets that provide ground truth, we can directly select keyframes as vertices in \mathcal{M} based on changes in distance and angle. In our collection of sequences, our primary framework of choice was LIO-SAM (Shan et al., 2020). However, we incorporated specific components from Fast-LOAM (Wang et al., 2020) for point cloud alignment. By fusing GPS and IMU information, we constructed highly accurate instructional maps.

3.2.1 | Teaching and map-building on LiDAR

The prebuilt map is generated using LiDAR scans. The map is constructed using a series of LiDAR keyframes with ground truth information. During the mapping process, if there are sufficient translations or rotations between two scans, a new vertex \mathcal{M}_m is added, connected by a new edge $T_{\mathcal{M},m-1 \rightarrow m}$. To calculate the true transformation relationship between these two keyframes $T_{\mathcal{M},m-1 \rightarrow m}$, we utilize the ground truth provided by the data set. We employ localization and mapping techniques in our collected data sequence using LiDAR, IMU, and GNSS data. These methods allow us to

accurately determine the ground truth, which serves as a reference for subsequent Radar-based localization.

3.2.2 | Repeating and localization on radar

In the repeating and localization on Radar step, the sequential Radar keymaps $\mathcal{R}_k (k = 1, 2, 3, \dots)$ are processed. First, an initial transformation $T_{k-1 \rightarrow k}$ is estimated based on Radar odometry between consecutive scans \mathcal{R}_{k-1} and \mathcal{R}_k . Subsequently, the most similar LiDAR frame \mathcal{M}_{m-1} that corresponds to the current Radar keymap \mathcal{R}_k is identified using the proposed method. The prior relationship ${}^*T_{r \rightarrow l}$ between the two frames is computed by employing the SPD and iterative closest point (ICP) algorithm. Finally, the transform estimation is refined by adding the a priori transform T_k to the pose graph optimization, using the preconstructed LiDAR map as a reference.

3.3 | Radar keyframe generation

The Radar image is affected by noise and ghost reflections caused by multipath return. Extracting an accurate environment description from the Radar is crucial for aligning the Radar point cloud with the LiDAR point cloud. Typically, noise is filtered out in a single frame. However, this single-frame approach does not eliminate ghost reflections, and it also mistakenly classifies the peak parts of white noise as tiny objects, which impacts the final result. We extract keypoints from each Radar image frame to avoid occlusion and ghost reflections and fuse the feature points obtained from multiple frames.

3.3.1 | Keypoint extraction

Using a more appropriate feature extractor is advantageous in extracting Radar feature points to represent the structural features of

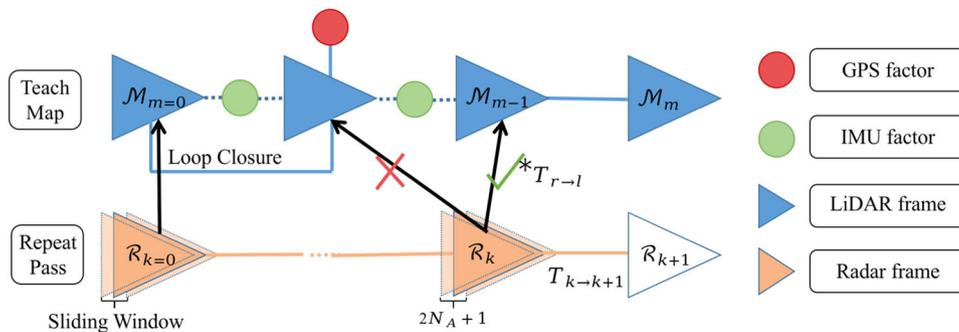


FIGURE 3 The structure of T&R. Orange triangles depicts Radar odometry, with each triangle representing a radar frame pose. The opaque triangles represent the selected middle frame within the sliding windows, with the point clouds of the surrounding transparent triangles aligned to them. Notably, each Radar submap frame is annotated with a subscript k to signify its corresponding pose. In contrast, the blue triangles represent LiDAR keyframe poses, which incorporate additional GPS and IMU constraints. Furthermore, the alignment between Radar and LiDAR is denoted by ${}^*T_{r \rightarrow l}$, highlighted with a green check mark, signifying the accurate identification of a corresponding similar frame.

a scene. According to Cen's concept (Cen & Newman, 2018), CFAR (Rohling, 2011) is not the most optimal detector for Radar due to its generation of numerous redundant points. The raw polar images of Navtech FMCW Radar are acquired through rotational scanning of the millimeter wave transceiver. Based on this property, Cen and Newman (2019) introduced an efficient feature detector that eliminates redundancy. Cen requires the maximum number of key points l_{max} to be specified for each angle, eliminating the need for complex input parameters. We show the whole process of Radar keypoint extraction in Figure 4a. Initially, the Prewitt operator calculates the gradient G in the range direction, and G' is obtained by normalizing G using the formula $G' = \frac{|G|}{|G|_{max}}$. Simultaneously, the original image S_{raw} undergoes mean subtraction to yield $S' = S_{raw} - \text{mean}(S_{raw})$. Ultimately, $H = (1 - G') \times S'$ is calculated to identify the high-intensity and low-gradient regions. The pixels exceeding the average value in H are retained and sorted in descending order to label the continuous regions along each ray. Subsequently, starting from these pixel positions, an extension is made on S' towards both ends of the ray, with all positions greater than 0 being considered as connected regions $'M$. Each ray can produce a maximum of l_{max} continuous marked segments. For every serial region $'m_a \in 'M$, the range bin r with the highest value is designated as the keypoint $'k_a \in 'K$.

3.3.2 | Motion and Doppler correction

To improve odometry positioning accuracy and Radar LiDAR matching performance, we eliminate Radar scanning distortion

caused by sensor motion and the Doppler effect. We implement the improved Motion-Compensated RANSAC (MC-RANSAC) proposed in Cen and Newman (2019). MC-RANSAC (as described in Figure 4b) directly estimates the sensor's velocity $\varpi = [v \ \omega]^T$. Instead of transforming, we employ ϖ to eliminate the motion distortion in Radar measurements, following the method described in Cen and Newman (2019). During the RANSAC iteration process, we estimate and update the displacement caused by the Doppler effect in the current subset of points and eliminate it using the latest v and Equation (1):

$$\Delta r_{\text{corr}} = \beta(|v_x| \cos(\phi) + |v_y| \sin(\phi)), \quad (1)$$

where velocity vector $v = (v_x + v_y)$ is obtained from the optimal MC-RANSAC estimate, while ϕ represents the angle between the ray direction and the x -axis. Let $\beta = f_t / (df/dt)$, with f_t representing the transmission frequency ($f_t \approx 76.5$ GHz) and df/dt representing the rate of change of frequency ($df/dt \approx 1.6 \times 10^{12}$). The Radar keypoints in a single frame suffering ghost reflections are sparse. To construct a keymap as the environmental representation of the keyframe, we consider converting the multiframe features to the sensor coordinates of the intermediate position. A sliding window consists of n Radar frames. Each frame contains a feature point cloud $'F_i$ obtained by converting polar coordinates ($'K$) to Cartesian coordinates and scaling by a factor. The middle position serves as the reference coordinate for the submap. We register the crucial point clouds of all frames within the window to this coordinate system, forming the Radar local feature point cloud map under the sliding window, represented as $'F_k$:

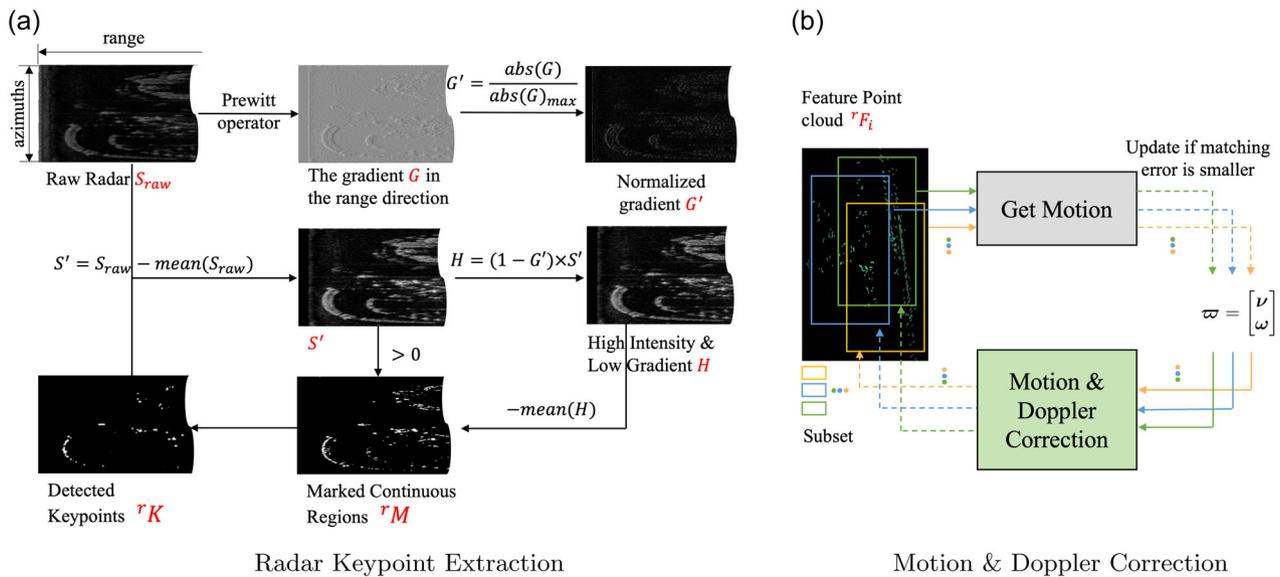


FIGURE 4 Illustration of Radar keypoint extraction and MC-RANSAC. In (a), we depict identifying the brightest points within connected regions at each azimuth by utilizing a mask that exhibits high brightness and low gradient, which are regarded as feature points. During iterations in (b), we randomly choose a subset of the key point cloud to calculate the sensor velocities directly. If the new velocity estimate results in a more minor alignment error, we update the velocity estimate accordingly. We correct the motion and Doppler aberrations of the subset of point clouds using the latest ϖ value at each iteration.

$${}^rF_k = {}^rF_{k,0}^k \dots {}^rF_{k,N_A-1}^k \cup {}^rF_{k,N_A}^k \cup {}^rF_{k,N_A+1}^k \dots {}^rF_{k,2N_A}^k, \quad (2)$$

where $2N_A + 1$ is equal to Radar frame numbers in the window, and ${}^rF_j^k$ means the register of rF_j at wO_k .

3.4 | Scan projection descriptor for RoLM

Building upon (Kim et al., 2022), we replace the value of each bin with the normalized point density of each patch. Initially, we rasterize the XY plane of a single point cloud frame. Next, we calculate the point count in all grids. Finally, we normalize the point count across all grids to obtain the point cloud descriptor for this frame:

$$\begin{cases} d_{R_{i,j}}^{rec} = \frac{D_{i,j}^{rec}}{\max D_{rec}} & (i, j = 1, 2, \dots, n), \\ d_{R_{i,j}}^{arc} = \frac{D_{i,j}^{arc}}{\max D_{arc}} & (i = 1, 2, \dots, m, j = 1, 2, \dots, n). \end{cases} \quad (3)$$

The variables D^{rec} and D^{arc} represent the density of point cloud blocks defined by rectangles and arcs, respectively. The variables i and j represent the indices of the grid. For example, in Section 4, we use grids of either 60×20 sectors with dimensions of $6^\circ \times 2\text{m}$ or 200×200 sectors with dimensions of $200\text{m} \times 200\text{m}$. The point cloud descriptor obtained through projection is denoted as S .

The resolution of the descriptor is influenced by the size and number of rasters that have 1 DOF in the row vector direction separating them. Descriptors can be divided into two categories according to the DOF (as shown in Figure 5):

- Polar projection (PP): The PP utilizes polar coordinates, with the angle θ serving as the horizontal axis and the radius r as the vertical axis. The descriptor is filled by counting the number of points falling into each arc. It contains 1 DOF ($\Delta\theta$) in the heading direction.
- Cartesian projection (CP): In this method, the vertical axis is represented by the x -axis of the sensor coordinates, and the y -axis represents the horizontal axis. The descriptor is filled by counting the number of points falling into the rectangular box. It contains 1 DOF Δy in the y -direction, typically to the left of the car when its front is facing forward.

The two descriptors mentioned above do not include the x -axis for Radar odometry. However, in a large-scale urban road scene, the translation at the lane level has minimal impact on the calculation results of PP. By evaluating the score of PP, we can align the two frames of point clouds on the x -axis.

3.5 | Scan projection estimate

Despite clarifying in Section 3.1 that there is a visible correlation between the dense region of the point cloud and the bright spot of the Radar point cloud, an accurate numerical relationship still needs to be improved. The similarity between the descriptor column vectors is first compared. The distances of each column vector are summed to provide an equitable representation of the two complete descriptors. By utilizing Equation 3, one can obtain the Radar

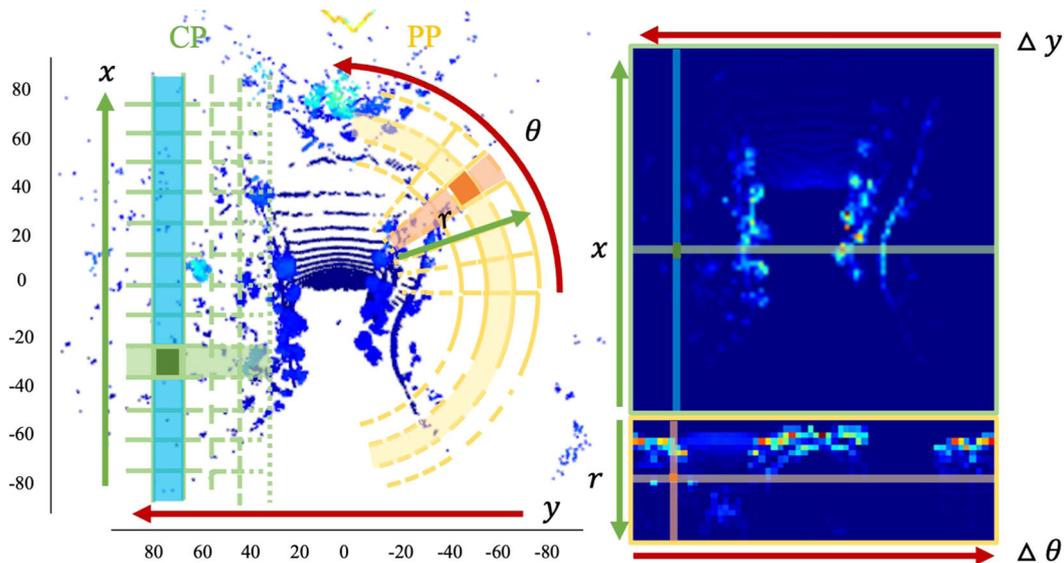


FIGURE 5 Summary of the transformation of Radar/LiDAR scans to SPD. The Radar/LiDAR point cloud is partitioned based on the raster regions illustrated in the figure. The green edges represent cartesian projection, while the yellow edges represent polar projection. Equation (3) determines the number of points within each raster, and the counts are then normalized to generate the SPD representation on the right side. When comparing the similarity of SPDs of the same type, the SPD description can be shifted along the red axis. SPD, symmetric positive definite.

descriptor S_R and the LiDAR descriptor S_L , and subsequently, calculate their distance using the following expression:

$$d_i(S_{R,i}, S_L) = \frac{1}{N} \sum_j = \frac{1}{N} \left(1 - \frac{S_{R,i}^j \cdot S_L^j}{\|S_{R,i}\| \cdot \|S_L\|} \right). \quad (4)$$

According to Section 3.4, $d(S_R, S_L)$ also has 1 DOF along the horizontal axis ($\Delta y, \Delta \theta$). $S_{R,i}$ is an SPD with shifted columns, where each column is shifted by an amount i compared to the original.

The LiDAR keyframes used for comparison in Section 4 are obtained by taking one frame every 0.5 m in all scans. All these keyframes are used to construct the complete LiDAR map. The alignment result is represented by n_{align} :

$$D(S_R, S_L) = \min_{i \in [N]} d_i(S_{R,i}, S_L), \quad (5)$$

$$n_{align} = \arg \min_{i \in [N]} d_i(S_{R,i}, S_L).$$

The function $D(\cdot, \cdot)$ computes the minimum distance between two descriptors for all offsets. The set S_L is defined as the collection of descriptions S_L for all candidate LiDAR scans located within a radius of $R_{research}$ from the location of the Radar scan described by S_R . By iteratively examining each candidate LiDAR scan $S_{L,j}$ within S_L , we determine the LiDAR scan that exhibits the closest proximity to the current Radar scan, referred to as $S_{L,s}$. The Radar and LiDAR scans' alignment result is $n_{align,s}$:

$$S_{L,s} = \arg \min_{S_{L,j} \in S_L} D(S_R, S_{L,j}), \quad (6)$$

$$n_{align,s} = \arg \min_{i \in [N]} d_i(S_{R,i}, S_{L,s}).$$

Thus, based on their PP and CP scores, we can calculate the rotation and translation of any key measurement ${}^w F_k$ relative to similar LiDAR frames. Furthermore, the process of selecting similar LiDAR frames enables us to make a rough estimation of the

translation along the x -axis. The rotation $\theta_{n_{align}}$ can be obtained using the formula $n_{align,s} \times \frac{360^\circ}{N}$. Similarly, the translation $y_{n_{align}}$ can be calculated as $(n_{align,s} - N) \times \frac{2 \times R_y}{N}$, where $R_y = 100\text{m}$ represents the farthest distance of the point used to calculate S_R in Section 4. To express this estimation, we use the transformation matrix ${}^{SP}T_{r \rightarrow i}$:

$${}^{SP}T_{r \rightarrow i} = \begin{bmatrix} \cos \theta_{n_{align}} & -\sin \theta_{n_{align}} & 0 & 0 \\ \sin \theta_{n_{align}} & \cos \theta_{n_{align}} & 0 & y_{n_{align}} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (7)$$

3.5.1 | Precise alignment using ICP

The accuracy of the initial rotation matrix depends on the choice of parameters n, m in Equation (3), which may bring a significant mistake to the final estimate. Based on the alignment results presented in Equation (6), it is highly probable to achieve a coarse registration with a rotation error of less than 7° and a lateral translation error of less than 1 m, as demonstrated in Figure 8. So, we can employ ICP with RANSAC to further improve the registration within a limited range. This iterative refinement process allows us to fine-tune the alignment and enhance the accuracy of the registration:

$${}^*T_{r \rightarrow i} = {}^{SP}T_{r \rightarrow i} \cdot {}^{ICP}T_{r \rightarrow i}, \quad (8)$$

where ${}^*T_{r \rightarrow i}$ indicates the corresponding between Radar and LiDAR poses. The entire initial alignment process can be referred to the Figure 6. Using the SPD algorithm for coarse point cloud alignment yields satisfactory outcomes, necessitating the subsequent employment of ICP for fine-tuning. For our experimental setup in Section 4, we configured the ICP parameters: the maximum alignment distance

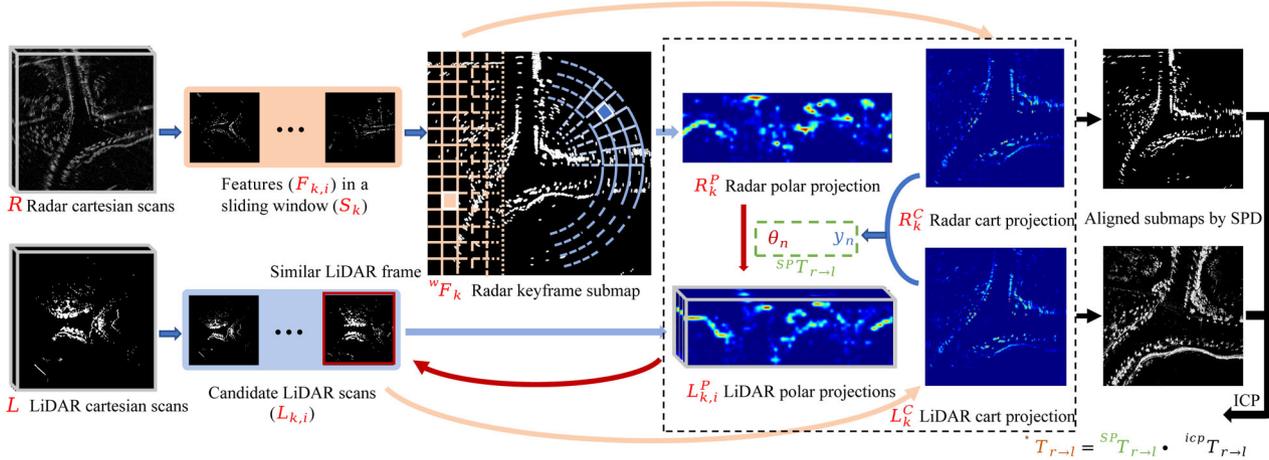


FIGURE 6 Scan projection based rough estimate. Given the initial measurement set R , the feature points $F_{k,i}$ in the sliding window S_k are spliced into a keyframe self-map ${}^w F_k$. The most similar LiDAR frame is selected from the candidate list using polar and Cartesian projection descriptors, respectively, and the rotation θ_n and translation y_n are calculated. On this basis, ICP is used to complete the alignment to obtain the primary edge constrain edges ${}^*T_{r \rightarrow i}$.

was set to 5 m, and the maximum number of iterations and RANSAC iterations were set to 100. Moreover, the ICP algorithm terminated iterations when the root mean square error between consecutive iterations fell below 10^{-4} .

3.6 | Heterogeneous pose graph optimization

The system's optimized estimation (Figure 3) can be divided into two parts. (1) Radar odometry: Provide initial pose estimation and the Radar point cloud keyframe. (2) Radar on LiDAR localization: Find a LiDAR frame similar to the Radar keyframe and calculate the extrinsic parameters of the two to get the deviation between the current position and ground truth.

3.6.1 | Radar odometry edge

The keypoints are extracted from the new Radar scan using the feature extraction method described in Section 3.3.1. These keypoints are then converted from polar coordinates to the Cartesian coordinate system. All Radar frames ${}^iF_{k,i}$ in the sliding window are registered at keyframe coordinates according to the estimated pose $\mathbf{O}_{k,i}$, forming a local keymap wF_k . Using each frame as a keyframe will be computationally expensive, affecting the algorithm's real-time performance. The interval between keyframes and the size of the sliding window is adjustable. We define the residual edge between Radar odometry frame i and j minimally as:

$$r_{i,j}(\mathbf{o}_i^w, \gamma\phi_i, \mathbf{o}_j^w, \gamma\phi_j) = \begin{bmatrix} \mathbf{R}(0, 0, \gamma\phi_i)^{-1}(\mathbf{o}_j^w - \mathbf{o}_i^w) - \hat{\mathbf{o}}_{ij}^i \\ \gamma\phi_j - \gamma\phi_i - \gamma\hat{\phi}_{ij} \end{bmatrix}, \quad (9)$$

where $\hat{\mathbf{o}}_{ij}^i$ is the relative position, and $\gamma\hat{\phi}_{ij}$ is the fixed estimate of the yaw angle value of rotation we estimated.

3.6.2 | Radar-LiDAR edge

The most similar LiDAR scan of the Teach path can provide a prior factor for the Radar keymap in the current sliding window. Prior Factor acts like a high-frequency loop closure factor. It differs from the odometry factor as it provides prior constraints on the SE3 6 DOF:

$$\mathbf{e}_{k,k}(\mathbf{o}_k^w, \mathbf{g}_k^w) = T_{r \rightarrow l,k} {}^*T_{r \rightarrow l,k}^T, \quad (10)$$

where ${}^*T_{r \rightarrow l,k}^T$ is the relative estimate of transformation, which is obtained from Equation (8), and $T_{r \rightarrow l,k}$ is the actual transformation between the current Radar and LiDAR frame.

The whole graph is optimized by minimizing sequential edges and Radar-LiDAR edges:

$$\min_{\mathbf{o}, \Phi} \left\{ \sum_{(i,j) \in S} \|r_{i,j}^2\| + \sum_{k \in H} \rho_k b_k \|e_{k,k}^2\| \right\}. \quad (11)$$

4 | EXPERIMENTAL VALIDATION

4.1 | Implementation strategy

We evaluated the performance of our RoLM system on the Mulran (Kim et al., 2020) and Oxford (Barnes et al., 2020; Maddern et al., 2017) Data sets. Additionally, we present our data set, obtained from the Zijingang campus of ZJU, which involved the deployment of the Navtech Radar CIR sensor and a 32-ring LiDAR (see Figure 7). We also conducted experiments on the identical route sequences collected at different times in the Oxford Radar Data set. It is distinct from the multiple sequential Mulran data sets that collected different routes within the same area. Besides, Sejong-02 tests the performance of our RoLM over 23 km.

To validate the algorithm's efficacy across different sensor types and onboard platforms, a test vehicle was constructed (Figure 7a). The platform is equipped with various sensors, including the Navtech CTS350-X millimeter wave Radar, RoboSense RS-LiDAR-32 LIDAR, CHCNAV X6 RTK, Xsens MTi-680G IMU, and FLIR Blackfly BFS-U3-16S2C-CS Camera. The sensor's initial extrinsic parameters were set based on the design specifications of the sensor module. During the LiDAR and IMU-based map construction, online optimization of their inter-parameter relationships was performed. However, no refinement was conducted for the extrinsic parameters of the remaining sensors. The sensors employed a software-triggered approach and utilized the ROS interface provided by the manufacturers for data recording. Additionally, the system clock served as a uniform source of synchronization.

Our LiDAR sensor has a vertical field of view of 40° , with a vertical angle resolution of less than 0.33° and a horizontal resolution ranging from 0.1° to 0.4° . It can detect 600,000 points per second within a measurement range of 0.2 to 200 m. On the other hand, the Navtech Radar operates based on an FMCW principle, providing a horizontal angle resolution of 0.9° and a distance resolution of 5.96 cm. It can accurately measure distances of up to 200 m at a rate of 4 Hz.

The datasets utilized in our experiments were gathered at two distinct locations: the Zhejiang University campus and Huanglong Sports Center. These sequences encompass various environments, including urban buildings, flat grasslands, and dense forests. We employed postprocessing techniques to obtain ground truth poses that involved integrating data from GNSS, RTK, IMU, and LiDAR. RTK subscription was utilized to achieve centimeter-level accuracy in positioning without the need for satellite base stations. The collection route can refer to Figure 7a–k. Table 1 shows some details about the data series. It is worth mentioning that the sensor types and locations on vehicles are different in each data set, and all experiments are conducted on the same system with an Intel[®] Core[™] i7-9700 CPU @3.00 GHz \times 8.

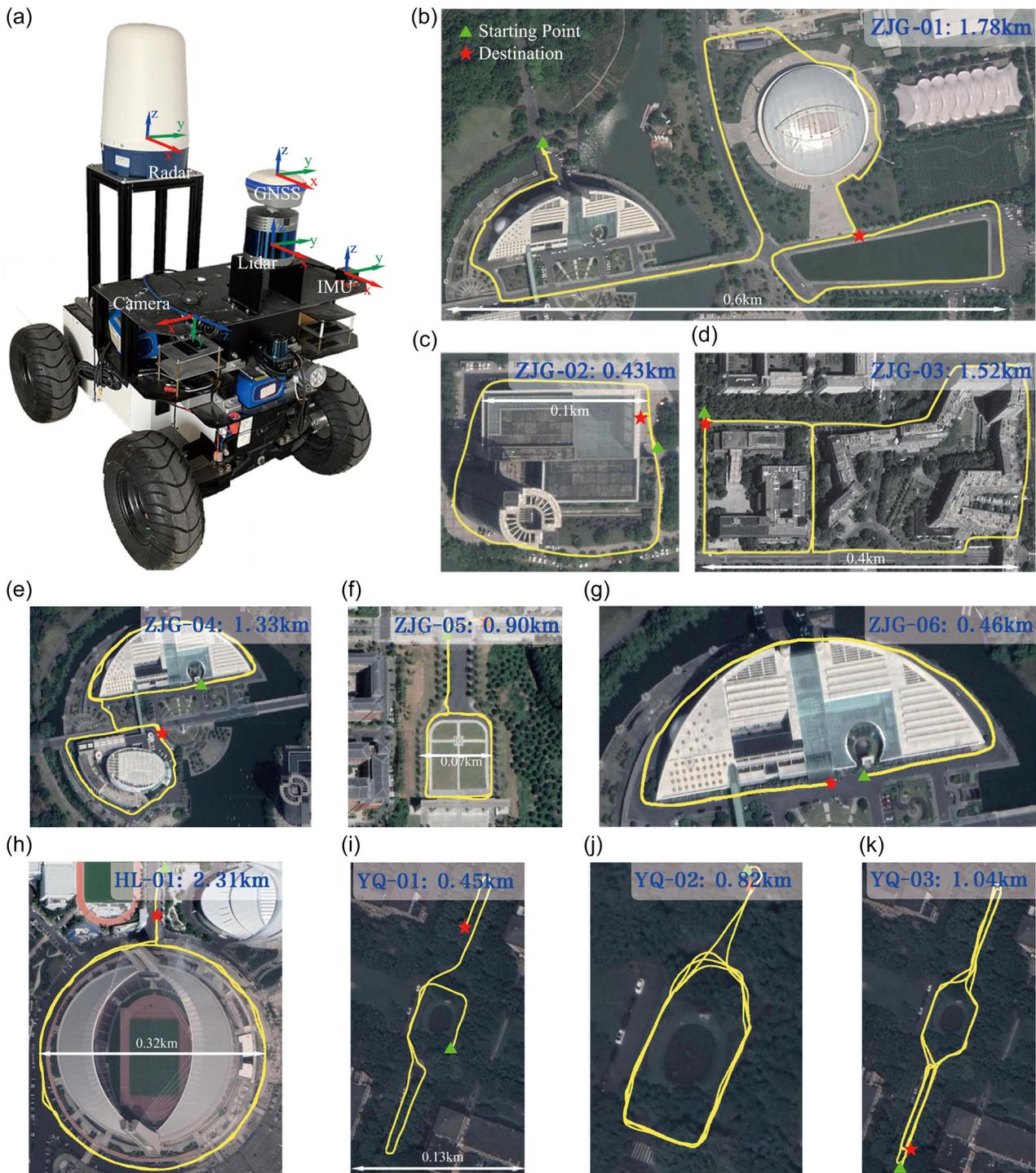


FIGURE 7 (a) Our test vehicle with Radar, LiDAR, IMU, and RTK sensors. (b)–(k) The paths contained in our data set, collected in Zhejiang University and Huanglong Sports Center.

4.2 | SPD performances

Before officially starting the experiment, we conducted some simple experiments to demonstrate the effectiveness of SPD. First, we use the Radar local point cloud image with different frame numbers to match the LiDAR to determine the frame numbers $2N_A + 1$ in the window.

We employ a technique that involves stitching together multiple frames of Radar feature point clouds to address the issue of sparse and obscured single-frame point clouds. However, incorporating too many frames can introduce high system latency, so selecting appropriate frames for stitching is crucial. In this experiment, we select every five frames Radar F_k as the central point and stitch

TABLE 1 Information on the ZJU data set.

Sequence	Time	Scene	Weather	Dynamic objects	Direction	Length (m)
ZJG-01	2022-01-15-14-11	Zijingang	Sunny	Many	Both	1784.620
ZJG-02	2022-01-15-15-40	Zijingang	Sunny	Many	Anticlockwise	462.844
ZJG-03	2022-01-15-16-06	Zijingang	Sunny	Many	Both	1515.229
ZJG-04	2022-05-19-14-42	Zijingang	Cloudy	Many	Both	1328.263
ZJG-05	2022-05-19-16-58	Zijingang	Cloudy	Many	Anticlockwise	901.609
ZJG-06	2022-05-19-14-42	Zijingang	Cloudy	Few	Anticlockwise	457.800
HL-01	2022-05-24-17-00	Huanglong	Sunny	Many	Anticlockwise	2312.648
YQ-01	2022-05-21-14-55	Yuquan	Cloudy	Few	Both	450.571
YQ-02	2022-05-21-15-55	Yuquan	Cloudy	Few	Both	819.383
YQ-03	2022-05-23-14-31	Yuquan	Cloudy	Few	Clockwise	1035.923

together $N_A = 1, 3, 5, 10, 15, 20$ into a point cloud denoted as ${}^W F_k$, using Equation (2). We randomly apply uniform lateral translations $\Delta y \in U(0, 8)$ ranging from 0 to 8 m to simulate lane changes. Next, we search for the LiDAR scan that is closest in time and calculate the transformations θ_{nalign} and y_{align} using Equations (3) and (4) (theoretical values being $\theta_{nalign} = \hat{\theta} = 0^\circ$ and $y_{align} = \hat{y} = \Delta y$). Since the calculation of the fraction of alignment results for all angles has already been performed by SPD, random rotation points are not necessary for this experiment. To account for the maximum range of lane changes, we disregard results beyond 15 m when calculating the CP error $y_{align} - \hat{y}$. The experimental results are presented in Figure 8.

Our findings indicate that our proposed SPD system is highly effective in most ideal urban autonomous driving scenarios. Additionally, incorporating a limited number of stitched Radar point clouds significantly enhances the accuracy of coarse alignment. Among the different stitching options, $N_A = 5$ and $N_A = 10$ demonstrate the best results, with $N_A = 5$ being the optimal choice. However, beyond this point, the gains achieved are marginal and not proportional to the cost involved. Therefore, for the experiments conducted in Section 4.4, we opted to use $N_A = 5$.

We employed different search ranges to validate SPD's effectiveness on multiple sequence data. The search localization strategy of RoIM involves traversing all LiDAR scans in the teaching sequence $\{\mathbf{T}_{teach,i}(\mathbf{R}_{teach,i}, \mathbf{t}_{teach,i}) \mid \|\mathbf{t}_{teach,i} - \mathbf{t}_{repeat,k}\| < R_{search}\}$ within a radius R_{search} , based on the currently estimated position $\mathbf{T}_{repeat,k}(\mathbf{R}_{repeat,k}, \mathbf{t}_{repeat,k})$ to locate similar frames $\mathbf{T}_{teach,s}(\mathbf{R}_{teach,s}, \mathbf{t}_{teach,s})$. During the experiment, a pair of matches with a distance of less than 10 m, that is, $\|\mathbf{t}_{repeat,k} - \mathbf{t}_{teach,s}\| < 10\text{m}$, were considered as true positives. Specifically, we evaluated the performance using recall@1, which is calculated as follows:

$$\text{recall@1} = \frac{\text{True positive samples}}{\text{Number of query scans}}. \quad (12)$$

Figure 9 demonstrates the strong performance of SPD on multiple sequence data, yielding high recall@1 values. Additionally,

our observation reveals a decreasing recall@1 as the R_{search} value increases. However, SPD exhibits limited effectiveness in degraded scenes with numerous bridges, resulting in sparse Radar feature points and challenging scene recognition. We ensure localization accuracy in Radar SLAM by adjusting the threshold and discarding sparse point cloud keyframes.

We employ Yin's experimental framework (Yin, Xu, et al., 2021) to examine the match scores between candidate LiDAR and Radar frames. Following a coarse-to-fine strategy, we select a subset of LiDAR frames that accounted for 1% of the total frames. We consider the top-1 positive match as true positive when the inter-frame distance is less than 3 m. We generate precision-recall curves by adjusting the score thresholds, as depicted in Figure 10. Furthermore, we report the maximum F1 scores, as summarized in Table 2. Our experiments involve comparing the performance of our proposed methods against established techniques such as Scan Context (Kim et al., 2022; Kim & Kim, 2018), DiSCO (Xu et al., 2021), and JL (Yin, Xu, et al., 2021). The evaluation is conducted on three benchmark sequences: Oxford, Riverside, and KAIST.

Notably, the original Scan Context and DiSCO methods exclusively apply to isomorphic sensors. Consequently, attempting to directly apply these methods to LiDAR and Radar data independently resulted in unsatisfactory outcomes. So, we test R2L using the signatures from L2L and R2R models we train separately. While designed for Radar on LiDAR localization, the JL method still exhibited suboptimal performance on the test sequences.

To compare the approach proposed in this paper, we conducted a series of place recognition tests using Yin's JL (Yin, Xu, et al., 2021). These tests included both the single-session loop detection test and the multisession localization validation mentioned in Yin's paper. These tests utilize pretrained models that have been made available as open source. Figure 11a illustrates the results of Radar-to-Radar loop detection on a single session, specifically KAIST01 on KAIST01, with the exclusion of the adjacent 100 frames. Subsequently, the

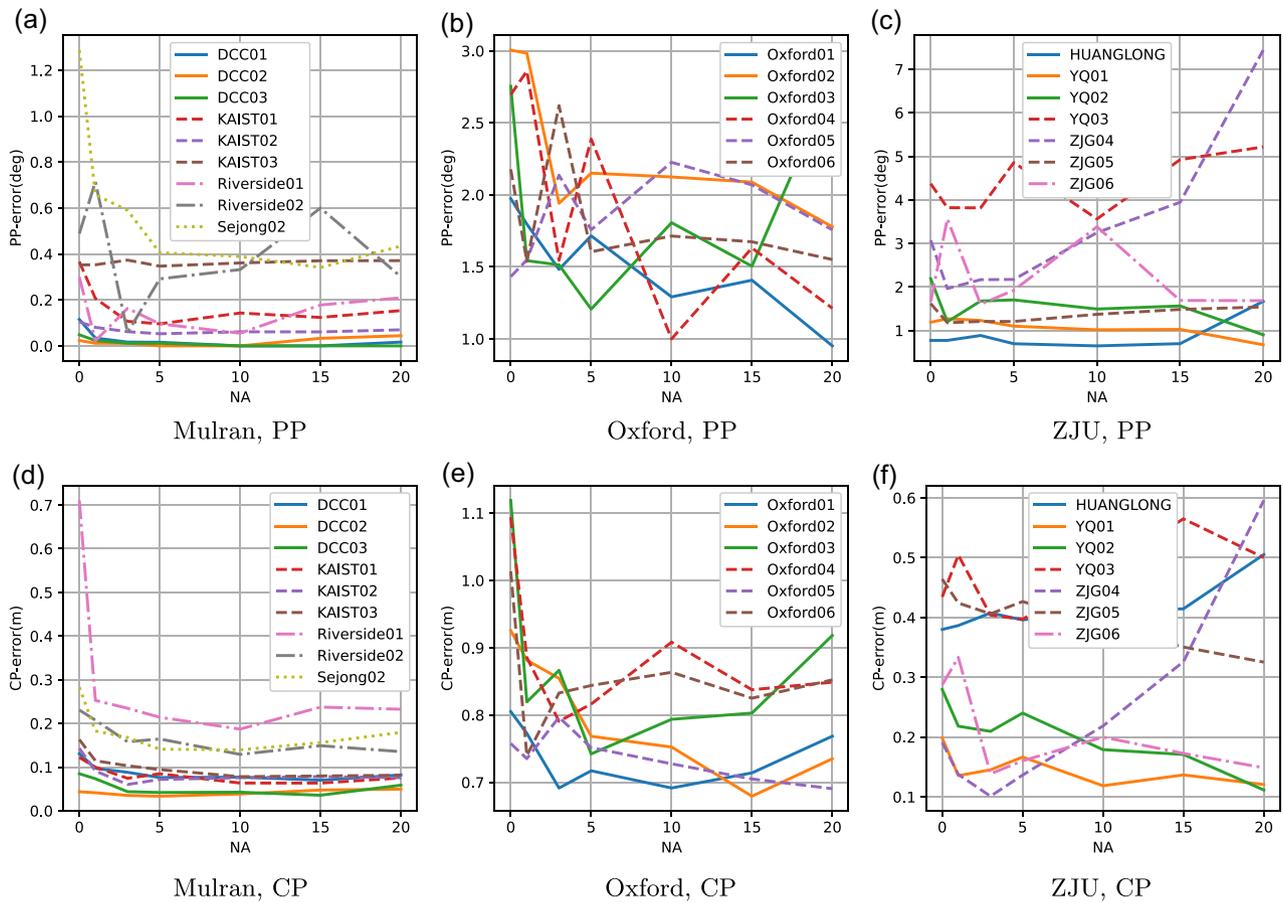


FIGURE 8 Illustration of frame-error. The figure demonstrates the impact of multiframe splicing on PP (a)–(c) and CP (d)–(f). Our findings reveal that the optimal splicing is achieved at approximately $N_A = 5$ under the specific experimental conditions outlined in this paper.

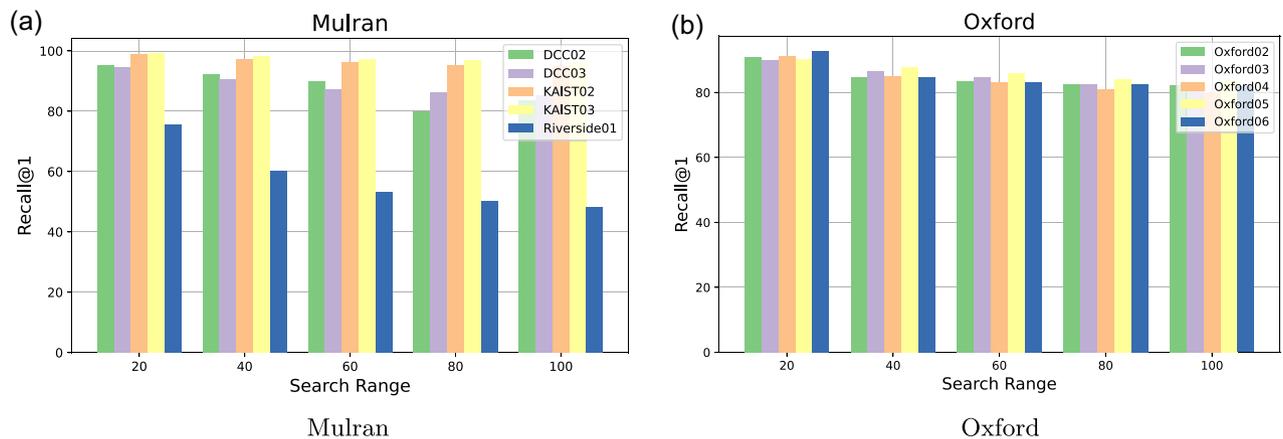


FIGURE 9 The performance (recall@1) of SPD on multiple sequence data. The Teach sequence of DCC02 and DCC03 is DCC01, the Teach sequence of KAIST02 and KAIST03 is KAIST01, and the Teach sequence of Oxford02–06 is Oxford01.

evaluation is expanded to encompass multisession scenarios, specifically KAIST03 on KAIST01. We conduct Radar-to-Radar tests (Figure 11b) and Radar-to-LiDAR tests (Figure 11c). The figures clearly show that their method produces favorable outcomes in loop

detection for single-session scenarios. However, it exhibits significant degradation when confronted with changes in the scene and sensors. Retrieving accurate map poses becomes challenging when employing heterogeneous sensor data. Similarly, Radar-to-LiDAR localization

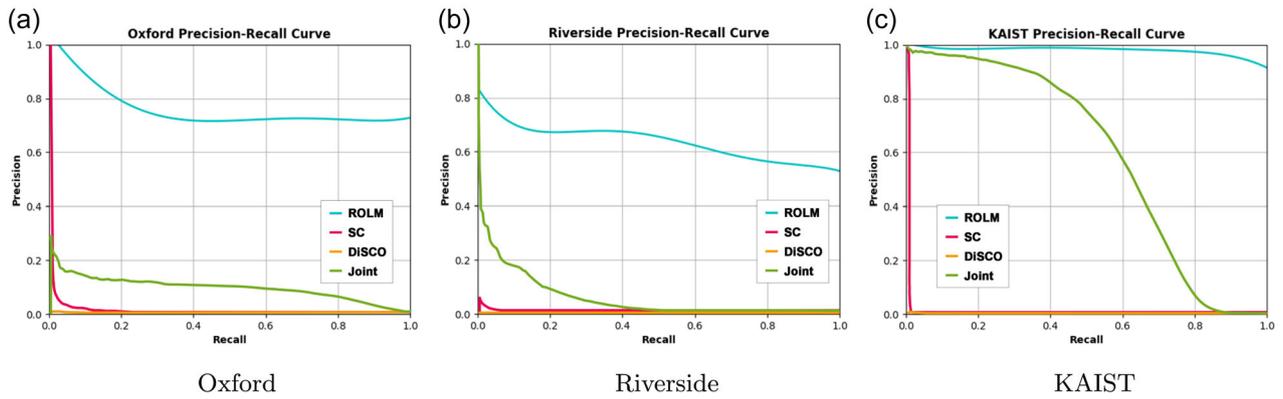


FIGURE 10 Precision-recall curve. We compare the SPD with the three methods (Kim et al., 2022; Xu et al., 2021; Yin, Xu, et al., 2021). We consider the top-1 positive match from 1% coarse candidates of the database as true positive when the inter-frame distance is less than 3 meters. The closer the curve is to the upper right, the better method.

TABLE 2 Maximum F1 score of precision-recall curves.

Sequence	Scan context (Kim et al., 2022)	DISCO (Xu et al., 2021)	JL (Yin, Xu, et al., 2021)	RoLM (Ours)
Oxford	0.04	0.02	0.18	0.84
Riverside	0.02	0.00	0.14	0.69
KAIST	0.02	0.01	0.61	0.96

tests were performed on various sequences using our RoLM methods detailed in this paper. The results demonstrate good qualitative accuracy in the tests of KAIST03 on KAIST01 (Figure 11d). We conducted experiments to assess the impact of Radar noise on SPD by introducing random Gaussian white noise, following $N(t) \sim \mathcal{N}(0, \sigma^2)$, at each pixel of the polar images derived from Radar scans. We introduced noise levels of $\sigma^2 = 0.1, 0.2, 0.3$, as depicted in Figure 12a. Beyond $\sigma^2 = 0.4$, Radar odometry ceased to function effectively. Experiments were carried out on the oxford01 sequence to identify similar LiDAR data using SPD through Radar scanning with added noise levels of $\sigma^2 = 0.1, 0.2, 0.3$. A search radius of $R_{research}$ was set at 20 m, pairs with distances D exceeding 0.8 were excluded, and the success rate of pairs within 5 m was calculated. The results are shown in Figure 12b, demonstrating that PP is very insensitive to noise and CP is relatively more sensitive.

To explore the impact of displacement on SPD, we manually translated the Radar scans within a range of 0–8 meters before performing LiDAR-based positioning. The average of the entire sequence was then calculated every 10 frames. To more accurately assess the magnitude of the increase in descriptor distance $D(S_R, S_L)$, we normalize the value of $1 - D(S_R, S_L)$ to a percentage to represent the similarity of S_R and S_L . Figure 12c depicts the decreasing trend in similarity for PP and CP within the Oxford01 sequence. The findings suggest that CP descriptors have low sensitivity to lateral translations, while PP descriptors are sensitive

to displacement. In our approach to determining a successful match, we establish that the distances of both frames must be below a predefined threshold. We then select the frame with the highest score from the pool of available candidates. Finally, we empirically assess the appropriateness of the a priori edge, as outlined in Section 4.3.

4.3 | Experiment setup and hypotheses

We set the sliding window size to 11 for all experiments and acquired a new window every 10 Radar frames. The heterogeneous prior constraint weight $\rho = 1$. In general, we consider each heterogeneous constraint to be credible unless any of the following situations occurs:

- If the SPD similarity $D(\cdot, \cdot)$ is below the threshold τ , the obtained result is considered a false match.
- U-turns rarely occur during daily driving. To eliminate the mismatch, we stipulate that if the difference between the current constraint g_k^w and the current body position o_k^w heading angle is more significant than 120° , then ignore it.
- As a rule of thumb, when the car drives normally on the road, it will not swerve suddenly at high speeds. Therefore, when the PP alignment result is $n_{align} \in (5, 25) \cup (35, 55)$, we set $b_k = 0$.

The current strategy needs an initial positioning for the first frame. Thus, it is necessary to ensure that the starting point of the Radar odometry is just a short distance from the map origin. Otherwise, the initial offset must be manually provided. We leveraged the KD tree to propose all the map poses and c candidates were selected for retrieval. The selection of candidate LiDAR maps will be adjusted according to the vehicle's speed and pavement information. For example, in Riverside and Sejong, we take $c = 100$ for bridges and mountain roads where road information is not abundant and $c = 50$ for structured urban scenarios.

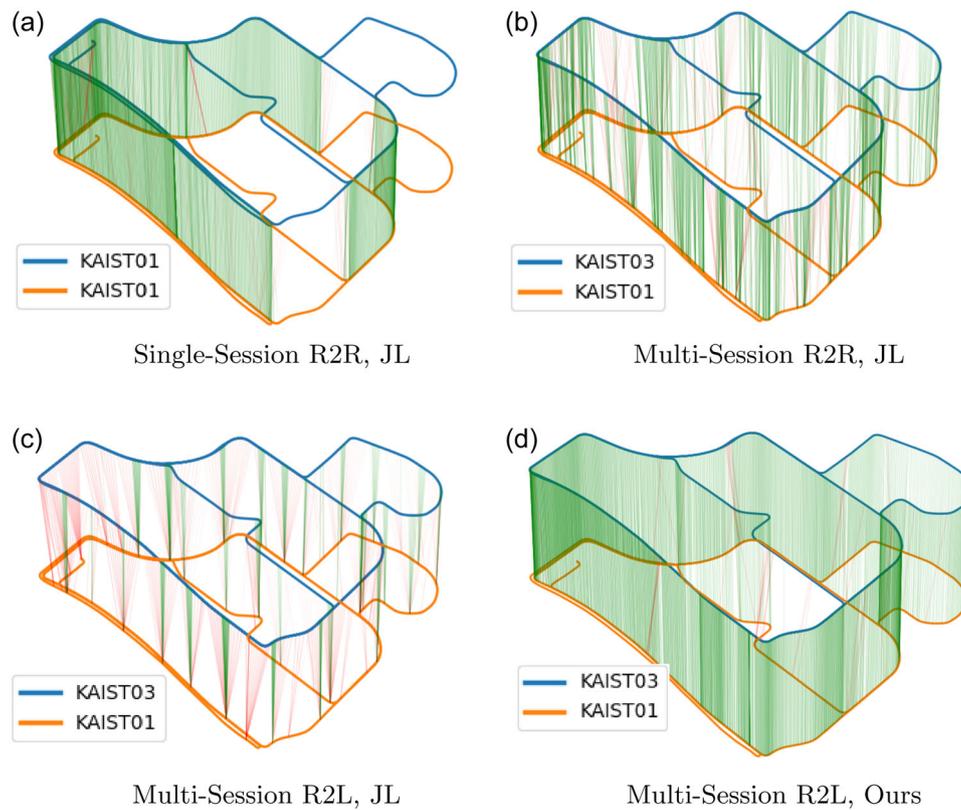


FIGURE 11 Visualization of localization results of different methods on multisequence data. A search radius of $R_{research} = 50$ m was used, with positive pairs connected by green lines and negative pairs connected by red lines using a threshold of $thr_{pos} = 10$ m. Tests (a), (b), and (c) were conducted using Joint Learning (Yin, Xu, et al., 2021), a method proposed by Yin. The result of our RoLM method is shown in (d).

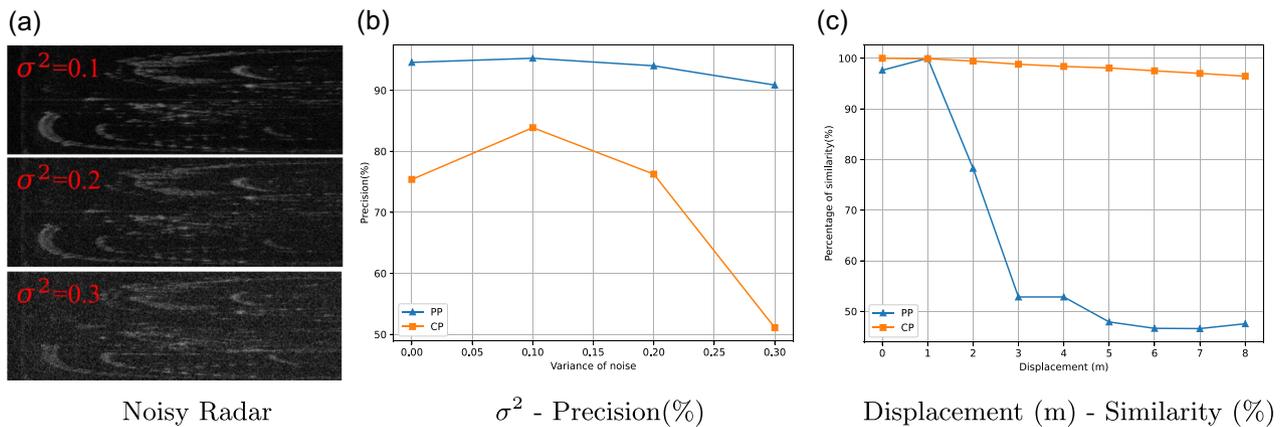


FIGURE 12 Ablation experiments on the expressivity of SPD. (a) Radar scan polar images at various noise levels. (b) The relationship between noise level and the accuracy of SPD relocation. (c) The relationship between the similarity of Radar and LiDAR scans at various additional offsets.

As for the LiDAR ground truth of the public data set referenced in the experiment, we transform the ground truth and LiDAR point cloud provided by the data set into the Radar coordinate system with the provided extrinsic. Moreover, we used a loosely coupled scheme based on LIO-SAM (Shan et al., 2020) and GPS for the self-built data set to obtain the LiDAR ground truth.

We construct the pose graph using the GTSAM (Dellaert & Contributors, 2022) and optimize it with the default Levenberg-Marquardt optimizer. We introduce the LiDAR map frames as fixed priority factors. Conversely, the odometry keyframes are incorporated as odometry factors, constrained by their self-motion estimations and the prior map information.

4.4 | Experiments results evaluation

We compared the proposed system with the three methods on two public data sets (Barnes et al., 2020; Kim et al., 2020) and data from ZJU data set. These competitive methods include RO (Burnett et al., 2021), RO with loop detection (Kim & Kim, 2018), and RaLL (Yin, Chen, et al., 2022). We added the data from ZJG01-03 to fine-tune the weight of RaLL's pretrained model. We also verified the effectiveness of the proposed descriptor through ablation

experiments. The results are presented in Table 3. We use the open-source tool (Grupp, 2017) for error calculation. An overview of the trajectory estimation results on some sequences is shown in Figure 13. The Scan Context method is significantly better than Yeti Odom in repeated road sections because it includes closed-loop detection. RoLM exhibits the most minor trajectory error compared to all other methods. In addition, the second and third rows of Figure 13 show our trajectory's translational and rotational relative errors. Specifically, the relative errors equal the mean translation and

TABLE 3 RMSE of global trajectories.

Sequence	Yeti Odom (Burnett et al., 2021)		Scan Context (Kim et al., 2022)		RaLL (Yin, Chen, et al., 2022)		RoLM(SPD) (Ours)		RoLM(SPD+ICP) (Ours)	
	Trans.(m)	Rot.(°)	Trans.(m)	Rot.(°)	Trans.(m)	Rot.(°)	Trans.(m)	Rot.(°)	Trans.(m)	Rot.(°)
Oxford-01	95.45	13.13	28.29	5.74	train	train	1.07	1.19	1.11	1.17
Oxford-02	34.25	5.46	14.14	3.66	0.98	1.45	0.84	0.93	0.92	1.01
Oxford-03	118.38	16.06	99.39	14.68	1.14	1.62	1.12	1.08	1.07	1.04
Oxford-04	201.01	26.30	185.53	23.95	1.71	1.93	1.22	1.29	1.41	1.33
Oxford-05	95.92	8.55	53.73	5.33	1.11	1.48	1.22	1.30	1.06	1.15
Oxford-06	148.29	22.37	120.02	19.45	1.14	1.52	1.24	1.14	1.29	1.23
RobotCar-all	126.38	16.95	102.09	14.38	1.23	1.60	1.13	1.16	1.15	1.16
DCC-01	30.60	2.61	17.76	2.79	2.11	1.97	2.93	1.09	0.97	1.17
DCC-02	26.72	4.49	20.15	4.16	4.71	2.01	1.17	1.06	1.02	0.95
DCC-03	19.94	4.02	12.63	2.53	5.14	2.55	1.36	1.44	0.78	1.24
KAIST-01	34.78	5.86	19.86	4.86	1.30	1.71	0.75	1.61	0.81	1.60
KAIST-02	31.99	6.61	5.55	2.5	1.30	1.71	0.66	1.06	0.66	1.05
KAIST-03	30.55	3.50	4.94	2.41	1.27	1.50	0.72	1.05	0.70	1.00
Riverside-01	40.40	5.97	8.10	3.00	4.12	2.84	2.55	2.01	2.50	1.99
Riverside-02	37.56	3.40	11.47	3.29	2.52	1.93	5.54	3.44	3.67	1.78
MulRan-part	32.78	4.83	13.27	3.27	3.12	2.02	2.60	1.84	1.83	1.44
Sejong-02	2893.17	38.14	2847.81	37.40	-	-	8.90	3.02	5.20	1.43
ZJG-01	51.26	48.02	50.32	47.59	train	train	8.87	6.98	1.10	6.94
ZJG-02	171.60	157.14	-	-	train	train	1.17	9.11	2.38	8.50
ZJG-03	137.25	178.48	-	-	train	train	2.46	6.55	2.36	6.53
ZJG-04	40.19	23.98	22.07	12.25	1.35	2.63	0.67	2.96	0.41	2.78
ZJG-05	30.04	25.49	8.20	5.53	1.46	3.69	0.68	3.05	0.46	3.11
ZJG-06	137.25	178.48	-	-	1.46	2.85	0.58	3.20	0.38	2.95
YQ-01	14.96	12.86	14.74	13.93	1.34	3.4	0.68	2.30	0.45	2.72
YQ-02	11.73	16.25	8.37	10.95	0.79	3.03	0.54	2.56	0.40	3.25
YQ-03	137.25	178.48	-	-	1.21	3.47	1.17	3.14	1.13	2.79
Huanglong	105.48	33.75	72.10	20.41	2.50	3.62	1.09	3.71	0.97	1.98
ZJU-all	96.89	102.23	47.86	26.70	5.40	3.32	3.75	4.89	1.25	4.62
All-sequence	1197.16	86.70	1177.17	19.58	2.77	2.01	4.18	2.29	2.53	1.83

Note: Meanings in bold are the best metrics among the different methods in the current experimental setup.

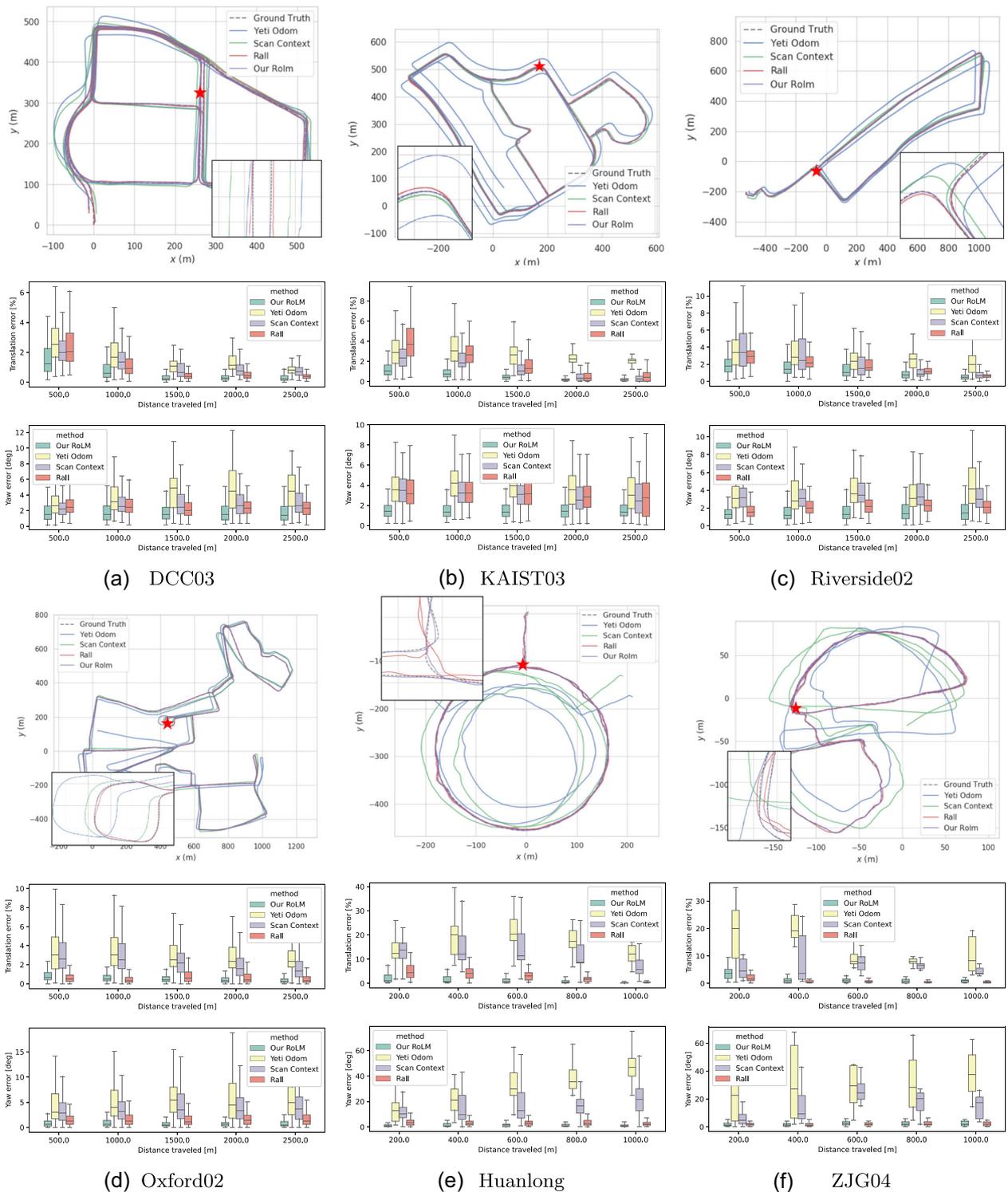


FIGURE 13 Evaluation of four different methods on the MulRan (a)–(c), Oxford (d), and ZJU (e, f) Data set. Each subplot has three rows of images. The first row presents a comparison between the estimated trajectories of the four methods and the ground truth, while the second and third rows display the percentages of relative translation errors and relative heading errors, respectively. Details of the trajectory for the marked part of the pentagram are shown.

rotation errors from 500 to 2500m with the incremental distance traveled.

Our method stands out in evaluating the absolute error of trajectories in all sequences, which has the minor RMSE of experimental results in most of the sequences (Table 3).

Our system performs better in structured urban scenes (i.e., Oxford, DDC, KAIST) while it has a more significant error in degraded scenes (i.e., Riverside, Sejong). Among them are many bridge scenes in the Riverside sequence, which is an excellent challenge for relocation. Therefore, when the number of point clouds in W_{F_k} is tiny,

set $b_k = 0$. Additionally, each Radar ray in ZJU data set has no exact timestamp, which results in the Doppler effect and motion distortion elimination challenges—the uncertain ray timestamp results in a significant error in the heading angle estimate. The Yeti odometry even provides a nonsmooth trajectory, which also causes the Scan Context algorithm to crash.

Finally, we also designed a set of ablation experiments. We tested RoLM (SPD) and RoLM (SPD+ICP), respectively, showing that SPD significantly improves the system, and ICP can make it more stable. Our RoLM has succeeded on a wide range of Radar and LiDAR models and is highly inclusive of vehicle speeds and lane changes.

5 | CONCLUSIONS AND FUTURE WORK

We first construct the Teach path using the well-established LIDAR-based multisensor fusion SLAM, which contains keyframe scans and a position reference. The Repeat path is constructed using the Radar sequence, which first maintains a Radar odometry that provides the raw position estimation result, accumulating a drift error over time. The most similar LiDAR references are found using our proposed SPD description, and their previous transformation relationships are computed from coarse to fine and added as a priori constraints to the pose graph optimization. We design a series of experiments to confirm the excellent performance of the SPD on both single sequences and multiple sequences and incorporate it into a Radar SLAM system to verify its enhancement of localization accuracy. The RMSE of localization across all our sequences is 2.53 m for positioning and 1.83° for angular measurement. Compared to Rall on the Oxford Radar RobotCar Data set (Barnes et al., 2020; Maddern et al., 2017), the positioning error demonstrates an 8.1% reduction, while the angular error experiences a substantial 27.5% decrease. Excluding the Sejong sequences, the Mulran Radar Data set (Kim et al., 2020) reveals a remarkable 41.3% reduction in translational error and a notable 28.7% reduction in angular error. Notably, the translational error in the ZJU data set shows a remarkable reduction of 76.9%. We demonstrate the reliability of the proposed localization system and its advantages over other methods in multisession multiscenario and our collected data sets. On the other hand, there are promising breakthroughs in the system to improve the practicability of Radar. First, only prior constraints are added to the middle frame of the sliding window during the system's operation. In contrast, the latest frame in the sliding window cannot be verified, and the algorithm has a certain lag. Second, we intend to implement Radar scene recognition on LiDAR based on the existing Radar (Kim et al., 2022) and cross-sensor (Yin, Xu, et al., 2021) global relocalization method in the future.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China under Grant NSFC 62088101.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are openly available in ZJU-Radar-Dataset at <https://github.com/HR-zju/ZJU-Radar-Dataset.git>.

ORCID

Yukai Ma  <http://orcid.org/0000-0001-8135-9012>

REFERENCES

- Aldera, R., De Martini, D., Gadd, M. & Newman, P. (2019) Fast radar motion estimation with a learnt focus of attention using weak supervision. In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 1190–1196.
- Anderson, S. & Barfoot, T.D. (2013) Ransac for motion-distorted 3d visual sensors. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 2093–2099.
- Barnes, D., Gadd, M., Murcutt, P., Newman, P. & 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)*. IEEE, pp. 6433–6438.
- Barnes, D. & Posner, I. (2020) Under the radar: Learning to predict robust keypoints for odometry estimation and metric localisation in radar. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 9484–9490.
- Burnett, K., Schoellig, A.P. & Barfoot, T.D. (2021) Do we need to compensate for motion distortion and Doppler effects in spinning radar navigation? *IEEE Robotics and Automation Letters*, 6(2), 771–778.
- Cen, S.H. & 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)*. IEEE, pp. 6045–6052.
- Cen, S.H. & Newman, P. (2019) Radar-only ego-motion estimation in difficult settings via graph matching. In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 298–304.
- Clark, S. & Dissanayake, G. (1999) Simultaneous localisation and map building using millimetre wave radar to extract natural features. In: *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)*. vol. 2. IEEE, pp. 1316–1321.
- Dellaert, F. & GTSAM, Contributors (2022) *borglab/gtsam*.
- Ding, X., Wang, Y., Xiong, R., Li, D., Tang, L., Yin, H. & Zhao, L. (2019) Persistent stereo visual localization on cross-modal invariant map. *IEEE Transactions on Intelligent Transportation Systems*, 21(11), 4646–4658.
- Grupp, M. (2017) *evo: python package for the evaluation of odometry and SLAM*. <https://github.com/MichaelGrupp/evo>
- Holder, M., Hellwig, S. & Winner, H. (2019) Real-time pose graph SLAM based on radar. In: *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, pp. 1145–1151.
- Hong, Z., Petillot, Y., Wallace, A. & Wang, S. (2021) Radar SLAM: a robust SLAM system for all weather conditions. arXiv preprint arXiv:2104.05347.
- Hong, Z., Petillot, Y. & Wang, S. (2020) Radarslam: Radar based large-scale SLAM in all weathers. In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 5164–5170.
- Huang, H., Ye, H., Sun, Y. & Liu, M. (2020) Gmmloc: structure consistent visual localization with Gaussian mixture models. *IEEE Robotics and Automation Letters*, 5(4), 5043–5050.
- Isola, P., Zhu, J.-Y., Zhou, T. & Efros, A.A. (2017) Image-to-image translation with conditional adversarial networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 1125–1134.

- Jose, E. & Adams, M.D. (2005) An augmented state SLAM formulation for multiple line-of-sight features with millimetre wave radar. In: *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 3087–3092.
- Kim, G., Choi, S. & Kim, A. (2022) Scan context++: structural place recognition robust to rotation and lateral variations in urban environments. *IEEE Transactions on Robotics*, 38(3), 1856–1874.
- Kim, G. & Kim, A. (2018) Scan context: egocentric spatial descriptor for place recognition within 3d point cloud map. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 4802–4809.
- Kim, G., Park, Y.S., Cho, Y., Jeong, J. & Kim, A. (2020) Mulran: multimodal range dataset for urban place recognition. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 6246–6253.
- Lowe, D.G. (2004) Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2), 91–110.
- Maddern, W., Pascoe, G., Linegar, C. & Newman, P. (2017) 1 year, 1000 km: the Oxford robotcar dataset. *The International Journal of Robotics Research (IJRR)*, 36(1), 3–15.
- Marck, J.W., Mohamoud, A., vd Houwen, E. & van Heijster, R. (2013) Indoor radar SLAM a radar application for vision and GPS denied environments. In: *2013 European Radar Conference*. IEEE, pp. 471–474.
- Rohling, H. (2011) Ordered statistic CFAR technique-an overview. In: *2011 12th International Radar Symposium (IRS)*. IEEE, pp. 631–638.
- Rublee, E., Rabaud, V., Konolige, K. & Bradski, G. (2011) Orb: an efficient alternative to sift or surf. In: *2011 International Conference on Computer Vision*. IEEE, pp. 2564–2571.
- Săftescu, Ș., Gadd, M., De Martini, D., Barnes, D. & Newman, P. (2020) Kidnapped radar: topological radar localisation using rotationally-invariant metric learning. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 4358–4364.
- Schuster, F., Keller, C.G., Rapp, M., Haueis, M. & Curio, C. (2016) Landmark based radar SLAM using graph optimization. In: *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, pp. 2559–2564.
- Shan, T., Englot, B., Meyers, D., Wang, W., Ratti, C. & Daniela, R. (2020) Lio-sam: tightly-coupled lidar inertial odometry via smoothing and mapping. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 5135–5142.
- Sheeny, M., De Pellegrin, E., Mukherjee, S., Ahrabian, A., Wang, S. & Wallace, A. (2021) Radiate: a radar dataset for automotive perception in bad weather. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 1–7.
- Wang, H., Wang, C., Chen, C. & Xie, L. (2020) F-loam: fast lidar odometry and mapping. In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.
- Xu, X., Yin, H., Chen, Z., Li, Y., Wang, Y. & Xiong, R. (2021) Disco: differentiable scan context with orientation. *IEEE Robotics and Automation Letters*, 6(2), 2791–2798.
- Yin, H., Chen, R., Wang, Y. & Xiong, R. (2022) Rall: end-to-end radar localization on Lidar map using differentiable measurement model. *IEEE Transactions on Intelligent Transportation Systems*, 23(7), 6737–6750.
- Yin, H., Wang, Y., Ding, X., Tang, L., Huang, S. & Xiong, R. (2019) 3d lidar-based global localization using siamese neural network. *IEEE Transactions on Intelligent Transportation Systems*, 21(4), 1380–1392.
- Yin, H., Wang, Y., Tang, L. & Xiong, R. (2020) Radar-on-lidar: metric radar localization on prior lidar maps. In: *2020 IEEE International Conference on Real-time Computing and Robotics (RCAR)*. IEEE, pp. 1–7.
- Yin, H., Xu, X., Wang, Y. & Xiong, R. (2021) Radar-to-lidar: heterogeneous place recognition via joint learning. *Frontiers in Robotics and AI*, 8, 101.

How to cite this article: Ma, Y., Li, H., Zhao, X., Gu, Y., Lang, X., Li, L. et al. (2024) FMCW Radar on LiDAR map localization in structural urban environments. *Journal of Field Robotics*, 41, 699–717. <https://doi.org/10.1002/rob.22291>