# FMCW Radar on LiDAR Map Localization in Structual Urban Environments

**Yukai Ma**
Institute of Cyber-Systems and Control
Zhejiang University
Hangzhou, China
yukaima@zju.edu.cn

**Han Li**
Institute of Cyber-Systems and Control
Zhejiang University
Hangzhou, China
hanley@zju.edu.cn

**Xiangrui Zhao**
Institute of Cyber-Systems and Control
Zhejiang University
Hangzhou, China
xiangruizhao@zju.edu.cn

**Yaqing Gu**
Institute of Cyber-Systems and Control
Zhejiang University
Hangzhou, China
3190104610@zju.edu.cn

**Xiaolei Lang**
Institute of Cyber-Systems and Control
Zhejiang University
Hangzhou, China
jerry_locker@zju.edu.cn

**Laijian Li**
Institute of Cyber-Systems and Control
Zhejiang University
Hangzhou, China
lilaijian@zju.edu.cn

**Yong Liu** *
Institute of Cyber-Systems and Control
Zhejiang University
Hangzhou, China
yongliu@iipc.zju.edu.cn

## Abstract

Multi-sensor 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 FMCW 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 (RoLM), which aims to enhance localization accuracy without relying on loop closures by mitigating the accumulated error in Radar odometry in real time. 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 (ICP) algorithm to achieve optimal matching on the LiDAR submap. We conducted extensive experiments on the Mulran Radar Dataset, Oxford Radar RobotCar Dataset, and our dataset to demonstrate the feasibility and effectiveness of our proposed approach.

*Corresponding author: Yong Liu, email: yongliu@iipc.zju.edu.cn

# 1 Introduction

Localization is an indispensable component of autonomous driving. Sensors such as GPS, camera, Radar, and LiDAR are widely employed in current systems. However, there are many limitations to the direct application of certain sensors in specific environments. For instance, visual localization accuracy may be compromised in the presence of illumination changes, and GPS can become unreliable in urban canyons.

Among these sensors, LiDAR is known for its high precision and is widely used in robotics and autonomous driving for localization and mapping tasks. Point cloud registration allows LiDAR to achieve accurate mapping even under varying illumination conditions.

In addition to real-time simultaneous localization and mapping (SLAM), offline pre-built maps can be constructed using LiDAR measurements taken in favorable weather conditions, augmented with information from sensors like inertial measurement units (IMUs) and GPS, to enhance local and global information.

However, LiDAR can still encounter accuracy issues in extreme weather conditions such as rain, snow, and fog, due to reduced performance. To address this challenge, millimeter wave Radar, which employs longer wavelength microwaves for observation, has been introduced into the field of SLAM. Despite potential drawbacks such as multipath phenomena and ghost reflections, as well as higher noise levels and lower accuracy compared to LiDAR, Radar exhibits exceptional robustness and is capable of long-range detection and map building in severe weather conditions. [Hong et al., 2020, Hong et al., 2021]. It has also become a hotspot of research in recent years.

Therefore, Radar localization on pre-build LiDAR maps will complement each other [Yin et al., 2020, Yin et al., 2021b, Yin et al., 2021a]. The use of LiDAR maps can compensate for the noise and sparsity of Radar data, and 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 the robustness of localization in all weather conditions [Hong et al., 2020, Hong et al., 2021]. However, there are two challenges in matching and aligning the Radar data and the LiDAR map: 1. 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, resulting in a lack of direct correspondence between LiDAR points 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 issue of 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. Firstly, the initial pose estimation is obtained from Radar odometry. Secondly, a LiDAR frame similar to the Radar keyframe is identified, and its external 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 Dataset [Kim et al., 2020], Oxford Radar RobotCar Dataset [Maddern et al., 2017], and our ZJU Radar Datasets (Fig. 5).

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

- We propose a multimodal Radar SLAM system that utilizes Radar-to-LiDAR relocalization to eliminate odometry drift.
- A new feature description and matching method of Radar on LiDAR Map (RoLM) is offered. It

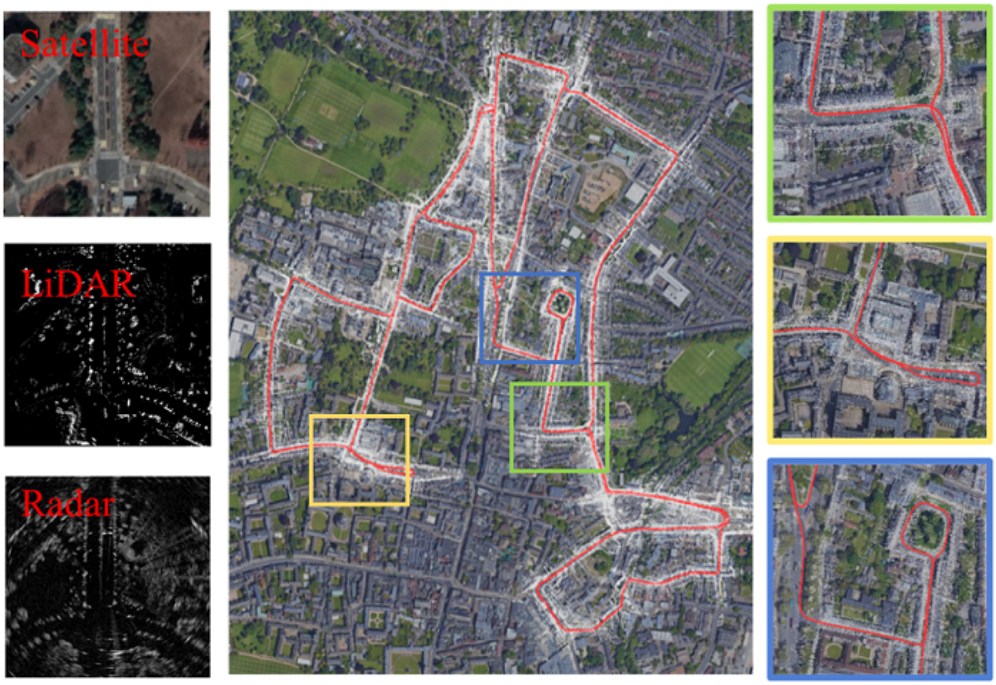



Figure 1: Radar odometry generated using RoLM in which the colorful box shows some details. The left side of the figure provides the difference between LiDAR data and Radar data in the same scene.

can retrieve the corresponding position index from historical LiDAR observations and estimate the coarse transformation.

- We first add the association of heterogeneous sensors to the sliding window pose graph optimization, which effectively improves localization accuracy.

- A new mobile cart Radar dataset is available[1]. Extensive experiments on the Mulran Radar Dataset [Kim et al., 2020] (multiple periods and scenarios), the Oxford Radar RobotCar Dataset [Maddern et al., 2017, Barnes et al., 2020], and our Zhejiang University (ZJU) Dataset (Fig. 5) 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. As for its front-end, many Radar algorithms are migrated from visual, or LiDAR platforms [Marck et al., 2013, Săftescu et al., 2020]. There are generally two routes for Radar feature extraction, traditional methods [Cen and Newman, 2018, Cen and Newman, 2019] and neural network methods [Aldera et al., 2019, Barnes and Posner, 2020].

In recent years, Radar SLAM has gained significant attention as a research topic. Radar sensors can provide multiple levels of data, including signals, images, or point clouds. For the front-end processing of Radar data, many algorithms have been adapted from vision or LiDAR platforms, with the goal of processing millimeter wave data as images or point clouds.

[1] https://github.com/HR-zju/ZJU-Radar-Dataset.git

There are two general approaches for Radar feature extraction: traditional methods and neural network 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. 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.

Both traditional and neural network methods have their advantages and limitations in Radar feature extraction for SLAM. Traditional methods may be computationally efficient and interpretable, but they may struggle to capture complex and abstract features in Radar data. Neural network methods, on the other hand, may offer higher accuracy and flexibility in feature extraction, but they may require larger amounts of training data and computational resources. The choice of approach depends on the specific requirements and constraints of the Radar SLAM system 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 and 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 and Newman, 2019].

Based on Cen's work, Burnett et al. introduced the Yeti Radar Odometry algorithm to eliminate motion distortion and Doppler effect in Radar data using a Gaussian filter instead of a binomial filter. This method also mitigated the effect of multipath reflections. The researchers found that Cen2018, combined with their RANSAC-based matching method, has outstanding performance. 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 [Rublee et al., 2011, Lowe, 2004]. The remaining matches are sent to an MC-RANSAC-based estimator to exclude outliers while correcting motion distortion [Anderson and 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 and 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++ that uses two descriptor representations of polar and Cartesian contexts, respectively, to robustly handle rotation and translation, and sub-descriptors for efficient information compression [Kim et al., 2021].

### 2.2 Localization on Pre-built Maps

The concept of localization on pre-built maps is closely related to SLAM, with high real-time requirements. Utilizing a pre-built map can eliminate the need for repeated online mapping in long-term fixed systems, thereby improving efficiency. Existing localization algorithms on pre-built 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 have limitations 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 and Dissanayake, 1999, Jose and Adams, 2005]. In recent years, Navtech has provided Radar sensors with higher accuracy, less motion distortion, and a 360-degree range for research, resulting in rich datasets [Kim et al., 2020, Barnes et al., 2020, Maddern et al., 2017, Sheeny et al., 2021] and various algorithms [Hong et al., 2020, Burnett et al., 2021].

Nevertheless, Radar also has limitations, such as being susceptible to the Doppler effect and noise, and its accuracy may not be as high as LiDAR. As a result, Radar-based localization algorithms often require the use of graph optimization [Schuster et al., 2016, Holder et al., 2019] 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 pre-built LiDAR maps.

Later, Yin also proposed an end-to-end learning system for localization in their work [Yin et al., 2021a]. This system used back-propagation of gradients from pose supervision to achieve localization and also incorporated a Kalman Filter to improve accuracy.

In a recent study by Yin [Yin et al., 2021b], a heterogeneous place recognition method via joint learning was introduced. 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 [Yin et al., 2021b] introduced a heterogeneous place recognition method via joint learning. 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 pre-built 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

This section introduces the proposed system. Figure 2 shows the system's proposed RoLM framework. 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, we preprocess them separately. Then, we process the LiDAR data into a Teach sequence, while the Radar data is processed into a Repeat sequence, as described in Section 3.2. During the execution of Radar odometry [Burnett et al., 2021], we utilize our proposed RoLM to extract scene descriptions from Radar and LiDAR data (Section 3.4). Subsequently, we align these descriptions using a coarse-to-fine method (Section 3.6.2). This alignment produces a heterogeneous constrained edge incorporated into the pose graph optimization (Section 3.6). The main steps of the system are illustrated in Algorithm 1.

## 3.1 Motivation for RoLM

Scan Context descriptor [Kim and Kim, 2018], designed especially for closed-loop detection of LiDAR odometry, performs well in the urban environment. It uses the highest point of the point cloud block in the area as a bin in the descriptor, and the distance between each column of ray vectors measures the scene's similar-

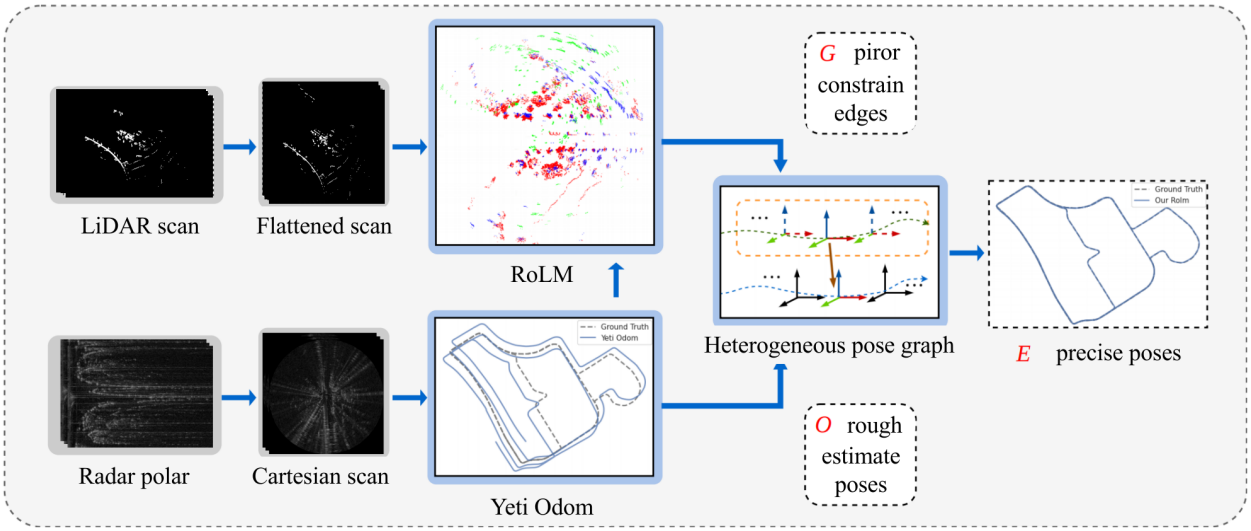



Figure 2: The overall framework. Given the raw range measurements, RoLM can find the corresponding location index from a set of locations in the map and computes the pose bias to add to pose graph optimization.

ity. Instead of height information, Kim [Kim et al., 2021] encodes intensity on the Cartesian Radar image. Unfortunately, there is no connection between the Radar point's power and the LiDAR point's height.

Assuming a non-transparent object in 3D space, it appears with a clear outline and geometric structure in the LiDAR point cloud, while its edge appears to be blurred in the Radar point cloud. The correspondence between LiDAR and Radar points is vague but relatively explicit between their point cloud clusters. The density of a point cloud can indicate an object's size, thickness, and hollowness. For example, the point cloud density of a tree will be sparser than that of a wall, which makes it more conspicuous from a top-down perspective.

### 3.2 Teach and Repeat

Teach and Repeat (T&R) is an autonomous path-tracking framework that manually instructs a robot to navigate a predefined path network [Furgale and Barfoot, 2010, Krüsi et al., 2015, Paton et al., 2018]. The Teach path is initially established by constructing a topological map based on LiDAR scans, as depicted in Figure 3. The sensor data obtained along a formal path is processed to create a sequence of local maps (vertices) connected through relative positional estimates (edges). The Repeat path is a new branch derived from the same path using a similar approach. Unlike the Teach sequence, the Repeat sequence is positioned at the fixed location of the previous sequence to compensate for accumulated odometer deviation. This feature is handy for linear and straightforward driving paths, where the robot can accurately localize and track the ground path without relying on precise global reconstruction. Examples of such paths include city roads or patrol vehicle routes. In this study, we employ LiDAR for constructing the teach topology map and utilize Radar for precise positioning on the created map.

#### 3.2.1 Teaching and Map-Building on LiDAR

The pre-built map is created based on LiDAR scans. The map is constructed using a series of LiDAR keyframes with ground truth information. If the translation or rotation between two scans, we add a new vertex $\mathcal{M}_m$ connected with new edge $T_{\mathcal{M},m-1\to m}$. The true transformation relationship $T_{\mathcal{M},m-1\to m}$ between two key frames $\mathcal{M}_{m-1}$ and $\mathcal{M}_m$ can be calculated using the ground truth provided by dataset.

**Algorithm 1:** Multi-modal Localization

**Input:** A set of Radar Cartesian images $\mathcal{R}$ and initial pose $\boldsymbol{O}_0^w$
**Output:** 3DOF precise pose at every imaging time $\boldsymbol{E}_i$

**Thread 1:** `// Pre-processing & Radar Odometry`
**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 ${}^{w}F_k$ generated by Equation 2
    $\mathcal{W} \leftarrow \mathcal{W} + \{[{}^{w}F_k, \boldsymbol{O}_k]\}$

**Thread 2:** `// Coarse-to-Fine Multi-model alignment`
**for** $w({}^{w}F_k, \boldsymbol{O}_k) \in \mathcal{W}$ **do**
  Candidated LiDAR scans $L_k \leftarrow \text{radiusSearch}(\boldsymbol{O}_k, R_{research})$
  $S_R \leftarrow$ Equation 3 $({}^{w}F_k)$; $\mathcal{S}_L \leftarrow$ Equation 3 $(\{L_{k,i}\})$
  Calculate the smallest $D(S_R, \mathcal{S}_L)$ by Equation 5 to find the most similar LiDAR scan $L_{k,s}$
  Calculate coarse transformation ${}^{SP}T_{r\to l}$ by Equation 6
  ${}^{w}F'_k \leftarrow \text{transformPointCloud}({}^{w}F_k, {}^{SP}T_{r\to l})$
  ${}^{icp}T_{r\to l} \leftarrow ICP({}^{w}F'_k, L_{k,s})$; ${}^{*}T_{r\to l} \leftarrow {}^{SP}T_{r\to l} \cdot {}^{icp}T_{r\to l}$
  **if** *gtFactorUseful()* **then**
    $\mathcal{G} \leftarrow \mathcal{G} + \{[\boldsymbol{g}_k^w, {}^{*}T_{r\to l}]\}$

**Thread 3:** `// Heterogeneous Pose Graph Optimization`
Create Optimizer
Add odometer node $\boldsymbol{O}_i$ and edge $\boldsymbol{O}_{ij}$
**for** $\boldsymbol{g}_k^w \in \mathcal{G}$ **do**
  Add prior node $\boldsymbol{g}_k^w$ and edge $T_{r\to l,k}$
$\boldsymbol{E} \leftarrow \text{updataPoses}(\boldsymbol{O})$

In our collected data sequence, this step includes localization and mapping using LiDAR, IMU, and GNSS data, and the resulting ground truth is used 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 repeating sequence $\mathcal{R}_{k+1}$ obtained from Radar scans is processed. First, an a prior transformation $T_{k-1\to k}$ is estimated between successive Radar scans $\mathcal{R}_{k-1}$ and $\mathcal{R}_k$ using Radar odometry. Then, the most similar LiDAR frame $\mathcal{M}_{m-1}$ corresponding to $\mathcal{R}_k$ is found using the proposed method. The prior relationship ${}^{*}T_{r\to l}$ between the two frames is calculated using the SPD and Iterative Closest Point (ICP) algorithm. Finally, the posterior $T_k$ is optimized by constructing a pose graph, which refines the transformation estimation for accurate localization using the pre-built LiDAR map as a reference.

### 3.3 Radar Keyframe Generation

The Radar image has noise and ghost reflections due to multipath return. The key to aligning the Radar point cloud with the LiDAR point cloud is to extract an accurate description of the environment from the Radar. The typical practice is to filter out noise in a single frame. However, we cannot remove the ghost reflections with this single-frame information, and the peak part of the white noise will also be regarded as a

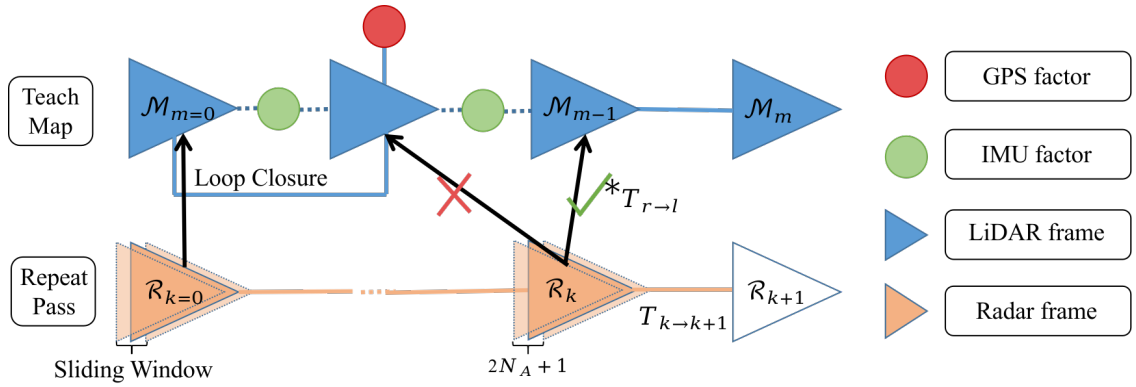



Figure 3: The structure of T&R. $\mathcal{R}$ represents the robot's frame equipped with a Radar. Subscript $k$ indicates the keyframes during millimeter-wave Radar odometry. $\mathcal{M}$ denotes the LiDAR fixation frames on the Teach sequence, which consist of $m$ vertices connected by edges representing relative positional transformations. Once all the Radar scan frame rates $R_{k+1,i} \in \mathcal{R}_{k+1}$ are collected, we estimate the transformation from $\mathcal{R}_{k+1}$ to the map frame $\mathcal{M}$. This estimation combines the odometry-estimated transformation ${}^*T_{r\to l}$ from $\mathcal{R}_{k+1}$ to $\mathcal{R}_k$ with the edge $T_{k\to k+1}$, 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 collected sequences, we combine LIO-SAM [Shan et al., 2020] and Fast-LOAM [Wang et al., 2020], which fuse GPS and IMU information, to construct an accurate Teach Map.

tiny object, affecting the final result. We choose to extract the keypoints of each Radar image frame directly and fuse the feature points of multiple frames to avoid occlusion and ghost reflections.

### 3.3.1 Keypoint extraction

The image is divided into high gradient regions of interest (ROI) and low gradient regions (using the Prewitt operator) according to the gradient of the image, and the areas mask ${}^rM$ is generated to remove redundant points. For each serial region ${}^rm_i$, the range bin $r$ with the highest value is taken as the keypoint ${}^rf_i$ after motion correction and Doppler removal [Burnett et al., 2021] :

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

where the velocity $v_x, v_y$ comes from their motion estimator. Doppler removal can remove radial distortion in millimeter-wave Radar rays.

### 3.3.2 Keyframe Submap

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 multi-frame features to the sensor coordinates of the intermediate position. A sliding window is a collection of $n$ Radar frames, including each frame's feature point cloud ${}^rF_i$ and the odometry estimate ${}^wO_i$. The middle position is the base coordinate of the submap. The critical point clouds of all frames in the window are registered to this coordinate system to form the Radar local feature point cloud map under the sliding window denoted as ${}^wF_k$:

$${}^wF_k = {}^r\boldsymbol{F}^k_{k-N_A} \ldots {}^r\boldsymbol{F}^k_{k-1} \cup {}^r\boldsymbol{F}_k \cup {}^r\boldsymbol{F}^k_{k+1} \ldots {}^r\boldsymbol{F}^k_{k+N_A} \tag{2}$$

where $2N_A + 1$ is equal to Radar frame numbers in the window, and ${}^r\boldsymbol{F}^k_j$ means the register of ${}^r\boldsymbol{F}_j$ at ${}^wO_k$.

### 3.4 Scan Projection Descriptor for RoLM

Inspired by [Kim et al., 2021], we replace the value of each bin with the normalized value of the point density of each patch. We first rasterize the space on the XY plane for a single point cloud frame, then count the number of points in all grids. Finally, we normalize the number of points in all grids to get the point cloud descriptor of this frame.

$$\begin{cases} {}^{d}R_{i,j}^{rec} = \dfrac{D_{i,j}^{rec}}{\max D_{rec}}, (i,j = 1,2,\ldots,n) \\ {}^{d}R_{i,j}^{arc} = \dfrac{D_{i,j}^{arc}}{\max D_{arc}}, (i = 1,2,\ldots,m, j = 1,2,\ldots,n) \end{cases} \tag{3}$$

where $D^{rec}, D^{arc}$ represent the density of point cloud blocks distinguished by rectangles and arcs, $i, j$ are the indices of the grid (e.g. $60 \times 20$ sectors of $6^\circ \times 2m$ are used in Section 4). We denote the point cloud descriptor obtained by projection as $S$.

The resolution of the descriptor depends on the size and number of rasters with a single-degree-of-freedom (DOF) in the row vector direction between them. Descriptors can be divided into two categories according to the DOF:

- Polar Projection (PP): The PP leverages polar coordinates, with the angle as the horizontal axis and the radius $r$ as the vertical axis. Count the number of points that fall into each arc to fill the descriptor. It stores 1 DOF in the heading direction.
- Cart Projection (CP): Take the $x$ axis of the sensor coordinates as the vertical axis and the $y$ axis as the horizontal axis. Count the number of points that fall into the rectangular box. It contains 1 DOF in the y-direction (usually to the left of the car when the front of the car is facing forward).

The above two descriptors lack the x-axis for Radar odometry. However, in a large-scale scene such as an urban road, the lane-level translation has little effect on the calculation results of PP. We can complete the alignment of the two frames of point clouds on the x axis by evaluating the score of PP.

### 3.5 Scan Projection Estimate

Although Section 3.1 clarifies that there is an apparent correspondence between the dense part of the LiDAR point cloud and the bright spot of the Radar point cloud, they do not have an accurate numerical relationship. The similarity between the descriptor column vectors is first compared. Adding the distances of each column vector gives an equal representation between the two full descriptors. We have known Radar descriptor $S_R$ and LiDAR descriptor $S_L$ from Equation 3, and the distance between them can be expressed as:

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

The LiDAR keyframes used in Section 4 for comparison are obtained by taking one frame every 0.5m in all scans. All of them are used to constitute the complete LiDAR map.

As described in Section 3.4, $d(S_R, S_L)$ also has 1 DOF along the horizontal axis. $S_{R,i}$ is an SPD whose columns are shifted by an amount $i$ from the original one. Traverse the similarity of all frames with different

offsets, and obtain a similar LiDAR with the miniature score. The alignment result is $n_{align}$.

$$\begin{aligned} D\left(S_R, S_L\right) &= \min_{i \in [N]} d_i\left(S_{R,i}, S_L\right) \\ n_{\text{align}} &= \arg\min_{i \in [N]} d_i\left(S_{R,i}, S_L\right). \end{aligned} \tag{5}$$

Thus, we can obtain the rotation $\theta_{n_{align}} = n_{align} \times \frac{360^\circ}{N}$ and translation $y_{n_{align}} = (n_{align} - N) \times \frac{2 * R_y}{N}$, ($R_y = 100m$ is the farthest distance of the point used to calculate $S_R$ in Section 4) of any key measurement ${}^{w}F_k$ relative to similar LiDAR frames based on their PP and CP scores. Moreover, the translation on the $x$ axis can also be roughly estimated. We use the transformation matrix ${}^{SP}T_{r \to l}$ to express it:

$${}^{SP}T_{r \to l} = \begin{bmatrix} \cos\theta_{n_{align}} & -\sin\theta_{n_{align}} & 0 & 0 \\ \sin\theta_{n_{align}} & -\cos\theta_{n_{align}} & 0 & y_{n_{align}} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{6}$$

#### 3.5.1 Precise alignment using ICP

The accuracy of the initial rotation matrix depends on the choice of parameters $n, m$ in the Equation 3, which may bring a significant mistake to the final estimate. Based on the above alignment results in Equation 5, we use ICP with the RANSAC to adjust it in a small range. Record the result as:

$${}^{*}T_{r \to l} = {}^{SP}T_{r \to l} \cdot {}^{icp}T_{r \to l}, \tag{7}$$

where ${}^{*}T_{r \to l}$ indicates the corresponding between Radar and LiDAR poses. The entire initial alignment process can refer to the Figure 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 external parameters of the two to get the deviation between the current position and ground truth.

#### 3.6.1 Radar Odometry Edge

Given a new Radar scan, we first perform a coordinate transformation. Key connected regions are divided according to gradient transformation, and intensity peaks in a small range are extracted as feature points. All Radar frames ${}^{r}F_i$ in the sliding window are registered at keyframe coordinates according to the estimated pose $\boldsymbol{O}_i$, forming a local keymap $\mathbb{F}_k$. 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 of edge between Radar odometery frame $i$ and $j$ minimally as

$$\begin{aligned} &\boldsymbol{r}_{i,j}({\boldsymbol{o}_i^w}, {}^{y}\phi_i, {\boldsymbol{o}_j^w}, {}^{y}\phi_j) \\ &= \begin{bmatrix} \boldsymbol{R}(0, 0, {}^{y}\phi_i)^{-1}(\boldsymbol{o}_j^w - \boldsymbol{o}_i^w) - \hat{\boldsymbol{o}}_{ij}^i \\ {}^{y}\phi_j - {}^{y}\phi_i - {}^{y}\hat{\phi}_{ij} \end{bmatrix}, \end{aligned} \tag{8}$$

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

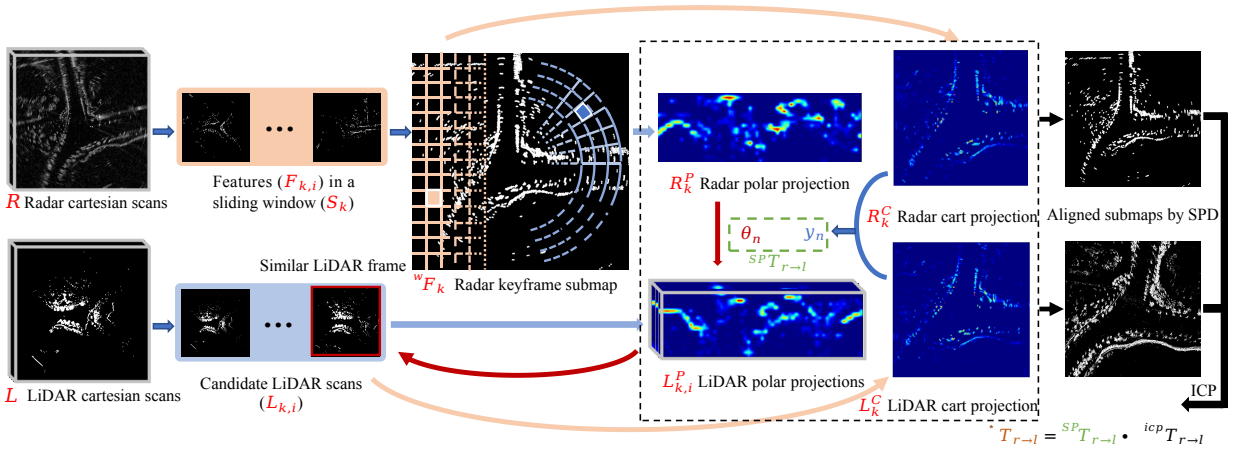



Figure 4: Scan Projection Based Rough Estimate. Given the initial measurement set $R$, the feature points $F_{k,i}$ in the sliding window $S_K$ are spliced into a keyframe self-map ${}^{w}F_k$. The most similar LiDAR frame is selected from the candidate list using polar and Cartesian projection descriptors, respectively, and the rotation $\theta_n$ and translation $y_n$ are calculated. On this basis, ICP is used to complete the alignment to obtain the primary edge constrain edges ${}^{*}T_{r\to l}$.

### 3.6.2 Isomerous Edge

Isomerous Factor is like a high-frequency loop closure factor. It is different from the odometry factor because it provides a prior constraints on the SE3 6 DOF.

$$\boldsymbol{e}_{k,k}(\boldsymbol{o}_k^w, \boldsymbol{g}_k^w) = T_{r\to l,k}\, {}^{*}T_{r\to l,k}^{\top}, \tag{9}$$

where ${}^{*}T_{r\to l,k}^{\top}$ is the relative estimates of transformation, which is obtained from Equation 7, and $T_{r\to l,k}$ is the actual transformation between the current Radar and LiDAR frame.

The whole graph is optimized by minimizing sequential edges and isomerous edges:

$$\min_{\boldsymbol{o},\boldsymbol{\Phi}} = \left\{ \sum_{(i,j)\in S} \|\boldsymbol{r}_{i,j}^2\| + \sum_{k\in H} \rho_k b_k \|\boldsymbol{e}_{k,k}^2\| \right\}, \tag{10}$$

where $S$ is the set of all sequential edges, and $H$ is the set of all isomerous edges. The scale coefficient $\rho$ is used to adjust the weight of heterogeneous prior constraints. Usually, we use $\rho > 1$. The flag variable $b_k \in \{0, 1\}$ comes from the judgment of some driving experience (see Section 4.3 for details) and indicates that the edge is valid or ignored. Their assignment strategy will be detailed in the experiments. By contrast, we do not add any constraints on sequential edges because these edges are extracted from RO, where some outliers have been removed.

# 4 Experimental Validation

## 4.1 Implementation Strategy

We tested our RoLM system on the Mulran [Kim et al., 2020] and the Oxford [Barnes et al., 2020, Maddern et al., 2017] Datasets. We provide a dataset that was collected using Navtech Radar CIR sensor and 32-rings LiDAR at the Zijingang campus of ZJU (Figure 5). We also conducted experiments on the same route sequences collected at different times in the Oxford Radar Dataset. It is distinct from the multiple

sequential Mulran datasets that collected different routes within the same area. Besides, Sejong-02 tests the performance of our Rolm over 23km.

In order to further verify the effectiveness of the algorithm in different types of sensors and onboard platforms, we built a test vehicle, as shown in Figure 5a. The platform incorporates various sensors, including the Navtech CTS350-X millimeter wave Radar, RoboSense RS-LiDAR-32 LIDAR, CHCNAV X6 RTK, Xsense MTi-680G IMU, and FLIR Blackfly BFS-U3-16S2C-CS Camera.

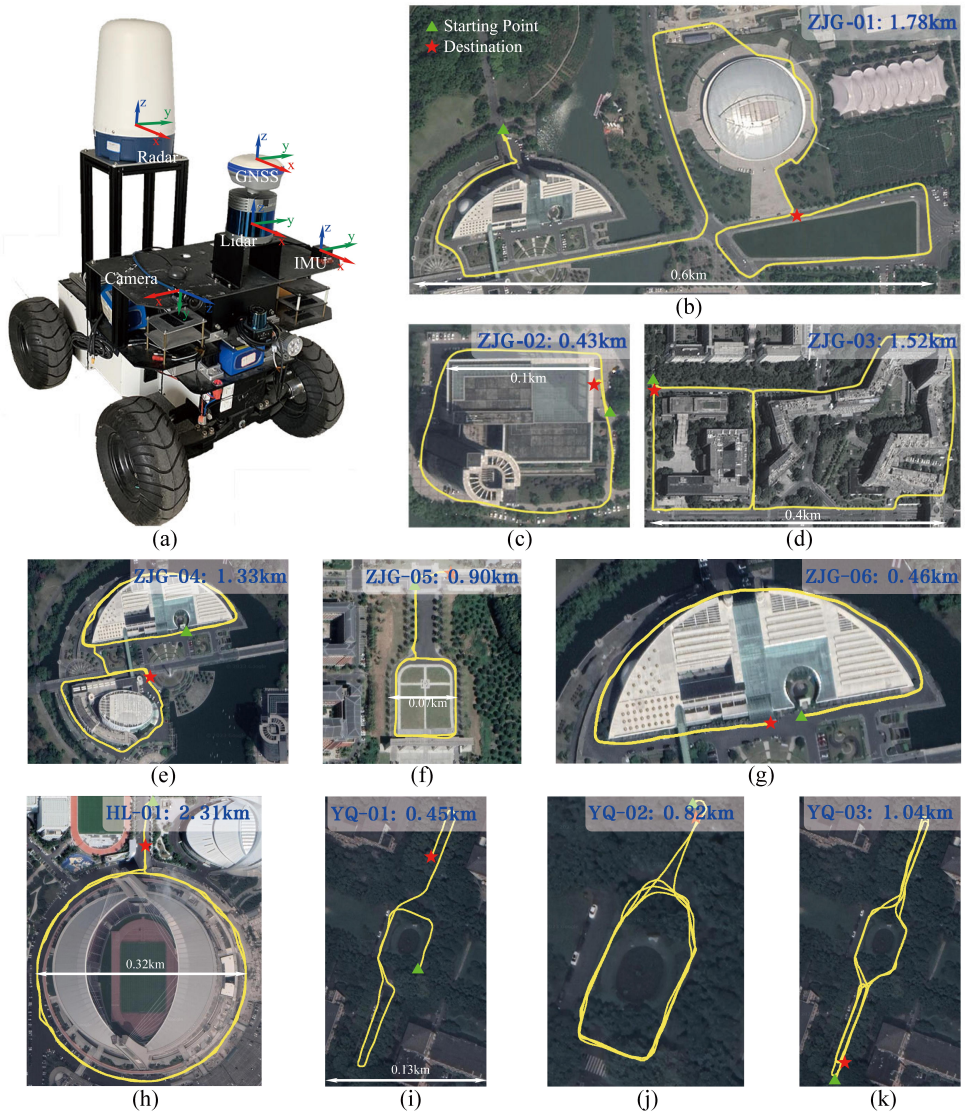



Figure 5: (a) Our test vehicle with Radar, LiDAR, IMU, and RTK sensors. (b) to (k) Then paths contained in our dataset, collected in Zhejiang University and Huanglong Sports Center.

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 0.2m to 200m. The Navtech Radar operates on a frequency-modulated continuous wave (FMCW) principle and offers a horizontal angle resolution of 0.9° and a distance resolution of 5.96 cm. It can measure distances of up to 200 meters at a rate of 4 Hz.

The datasets used in our experiments were collected at the Zhejiang University campus and Huanglong

Sports Center. These sequences encompass diverse environments, including urban buildings, flat grasslands, and dense forests. We employed post-processing 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 the right side of Figure 5. Table 1 shows some details about the data series.

It is worth mentioning that the sensors types and locations on vehicles are different in each dataset, and all experiments are done on the same system with an Intel® Core™ i7-9700 CPU @ 3.00GHz × 8.

Table 1: Information on the ZJU Dataset

| 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 |

### 4.2 SPD Performances

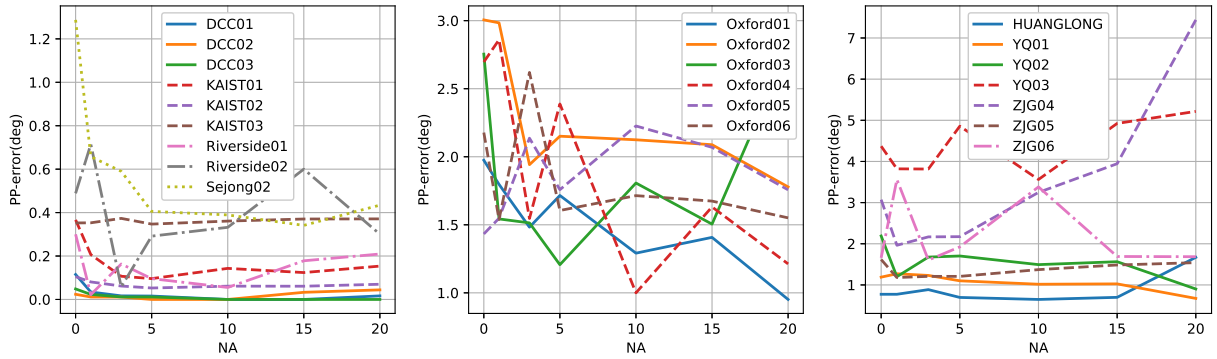
Before officially starting the experiment, we conducted some simple experiments to demonstrate the effectiveness of SPD. First, we use the Radar local point cloud image with different frame numbers to match the LiDAR to determine the frame numbers $2N_A + 1$ in the window.

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 5 frames Radar ${}^{r}\mathbf{F}_k$ as the central point and stitch together $N_A = 1, 3, 5, 10, 15, 20$ into a point cloud denoted as ${}^{w}\mathbf{F}_k$, using the Equation 2. We randomly apply uniform lateral translations $\Delta y \in U(0, 8)$ ranging from 0 to 8m to simulate lane changes. Next, we search for the LiDAR scan that is closest in time and calculate the transformations $\theta_{n_{align}}$ and $y_{align}$ using Equations 3 and 4 (theoretical values being $\theta_{n_{align}} = \hat{\theta} = 0°$ 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 15m when calculating the CP error $y_{align} - \hat{y}$. The experimental results are presented in Figure 6.

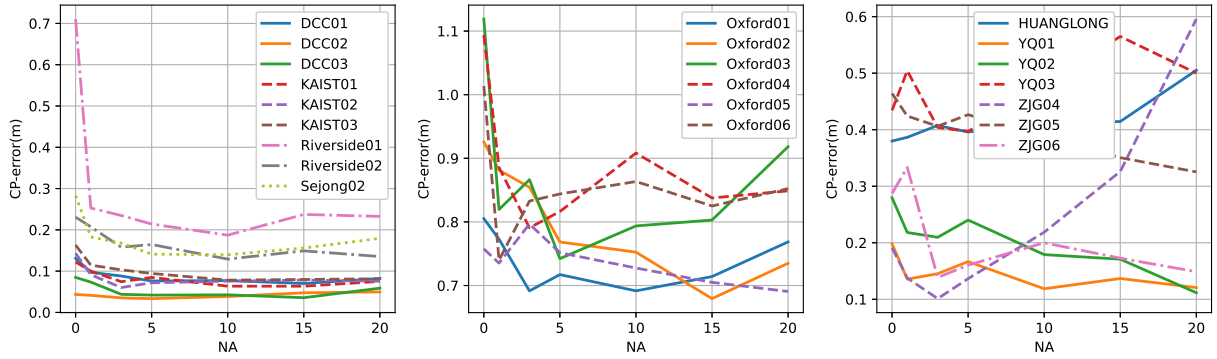
The results show that our proposed SPD can work effectively in most ideal urban autonomous driving scenarios. Moreover, a small number of stitched Radar point clouds effectively improves the accuracy of coarse alignment, with $N_A = 5, 10$ being the best stitching, and going forward, it is gainful but not worth the revenue. Therefore, in the experiments in Section 4.4, we consider using $N_A = 5$.

To validate the effectiveness of SPD on multiple sequence data, we employed different search ranges. The search localization strategy of RolM involves traversing all laser scans in the teaching sequence $\{\mathbf{T}_{teach,i}(\mathbf{R}_{teach,i}, \mathbf{t}_{teach,i}) | \|\mathbf{t}_{teach,i} - \mathbf{t}_{repeat,k}\| < R_{search}\}$ 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 $\mathbf{T}_{teach,k}(\mathbf{R}_{teach,k}, \mathbf{t}_{teach,k})$. During the experiment, a pair of matches with a distance of less than 10m, i.e., $\|\{\mathbf{t}_{repeat,k} - \mathbf{t}_{teach,k}\| < 10\text{m}$, were considered as true positives. Specifically, we evaluated the performance using recall@1, which is calculated as follows:

$$\text{recall@1} = \frac{\text{true positive samples}}{\text{number of query scans}} \tag{11}$$




(a) Mulran, PP (b) Oxford, PP (c) ZJU, PP



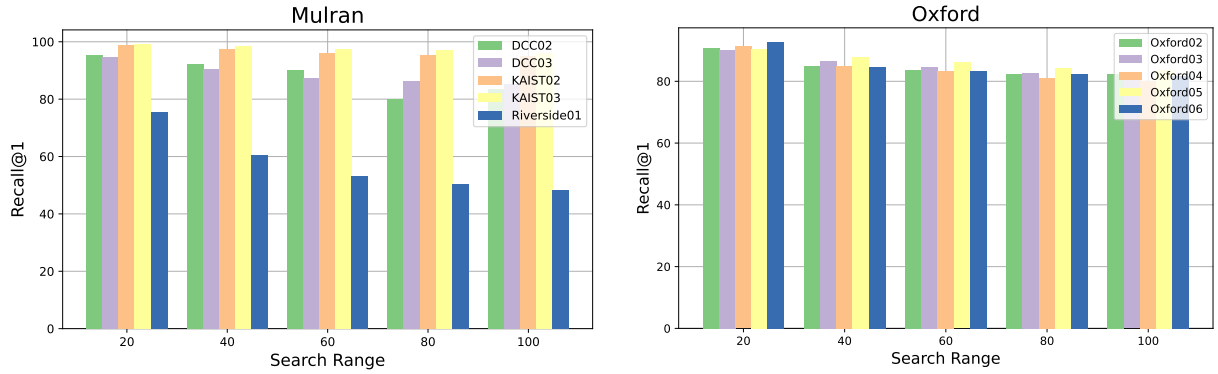

(d) Mulran, CP (e) Oxford, CP (f) ZJU, CP

Figure 6: Illustration of Frame-Error. We show the effect of multi-frame splicing on PP (a)(b)(c) and CP (d)(e)(f). We find that the best splicing occurs around $N_A = 5$ under the experimental conditions of this paper.

Figure 7 demonstrate the adequate performance of SPD on multiple sequence data, yielding high recall@1 values. Moreover, we observe that as the value of $R_{search}$ increases, the recall@1 decreases. 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.

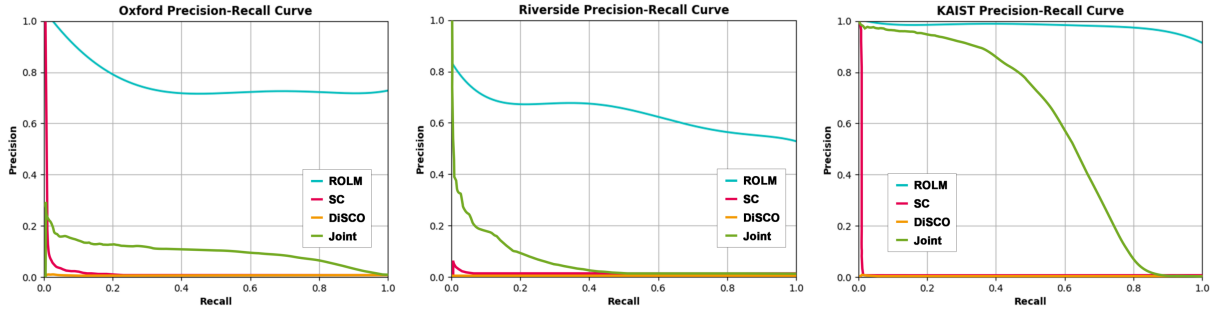
We employ Yin's experimental framework [Yin et al., 2021b] to examine the match scores between candidate LiDAR and Radar frames. Following a coarse-to-fine strategy, we select a subset of LiDAR frames that accounted for 1% of the total frames. We consider the top-1 positive match as true positive when the inter-frame distance is less than 3 meters.We generate precision-recall curves by adjusting the score thresholds, as depicted in Figure 8. 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 and Kim, 2018, Kim et al., 2021], DiSCO [Xu et al., 2021], and Joint Learning (JL) [Yin et al., 2021b]. 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. Attempting to directly apply these methods to LiDAR and Radar data independently yielded unsatisfactory results. So we test R2L using the signatures from L2L and R2R models we train separately. While designed for Radar on LiDAR localization, the Joint Learning method still exhibited suboptimal performance on the test sequences.




(a) Mulran (b) Oxford

Figure 7: 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.




(a) Oxford (b) Riverside (c) KAIST

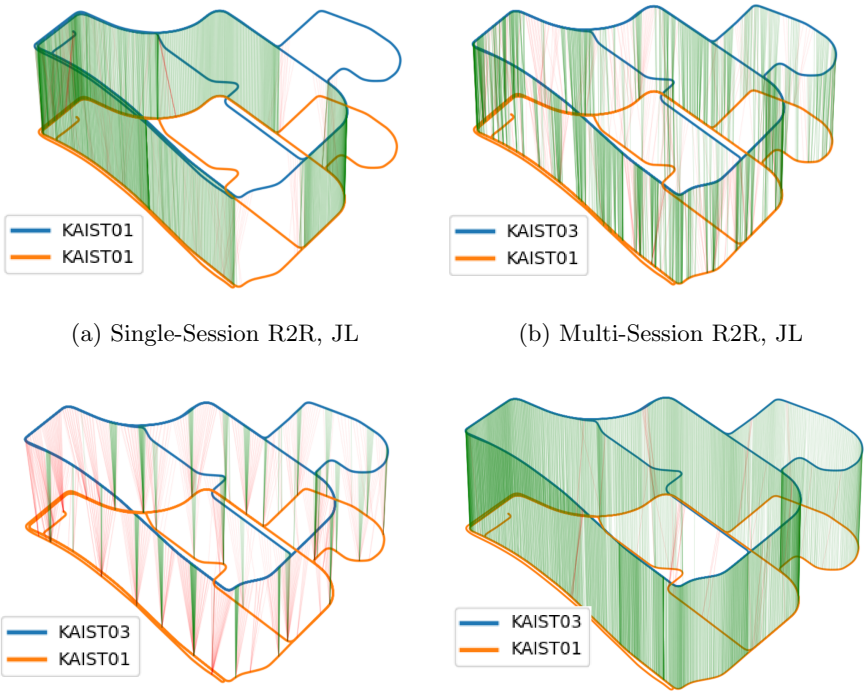
Figure 8: Precision-Recall Curve. We compare the SPD with the three methods [Xu et al., 2021, Yin et al., 2021b, Kim et al., 2021]. We consider the top-1 positive match from 1% coarse candidates of the database as true positive when the inter-frame distance is less than 3 meters. The closer the curve is to the upper right indicates, the better method.

To compare the approach proposed in this paper, we also conduct a series of place recognition tests using Yin's Joint Learning [Yin et al., 2021b], which includes both the single-session loop detection test mentioned in his paper and multi-session localization validation. These tests utilize pre-trained models that have been made available as open source. Figure 9a 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 evaluation is expanded to encompass multi-session scenarios, specifically KAIST03 on KAIST01. We conduct Radar-to-Radar tests (Figure 9b) and Radar-to-LiDAR tests (Figure 9c). As evident from the figures, the proposed method yields 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 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 9d).

Table 2: Maximum F1 Score of Precision-Recall Curves

| Sequence | Scan Context [Kim et al., 2021] | DiSCO [Xu et al., 2021] | JL [Yin et al., 2021b] | 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** |




(a) Single-Session R2R, JL

(b) Multi-Session R2R, JL

(c) Multi-Session R2L, JL

(d) Multi-Session R2L, Ours

Figure 9: Visualization of localization results of different methods on multi-sequence data. We set the search radius $R_{research} = 50m$, connecting the positive pair with the green line and the negative pair with the red line with a threshold $thr_{pos} = 10m$. (a)(b)(c) are the tests using Joint Learning [Yin et al., 2021b] proposed by Yin. (d) is the results of our RoLM method.

### 4.3 Experiment Setup and Hypotheses

In all experiments, we set the size of the sliding window to 11 and perform window acquisition every ten frames of Radar. 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 lower than the threshold $\tau$, what we get is thought to be a false match.
- During daily driving, U-turns rarely occur. In order to eliminate the resulting 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 is driving 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 does not include an initial positioning of the first frame. Therefore, it is required that the starting point of the Radar odometry is not too far from the map origin; otherwise, the initial offset needs to be given manually. We leveraged the k-d 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 instance, 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.

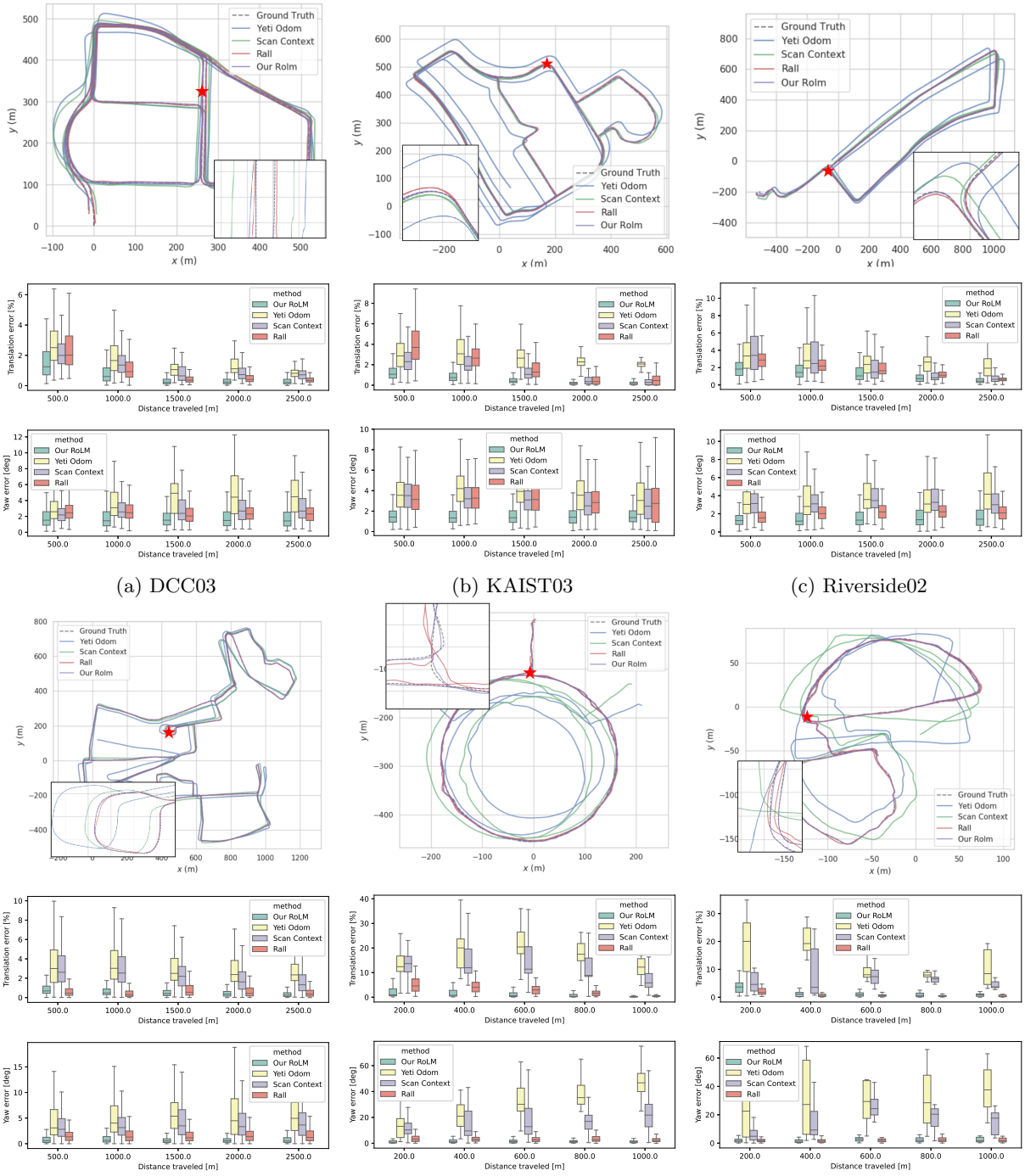
As for the LiDAR groundtruth of the public dataset referenced in the experiment, we transform the groundtruth and laser point cloud provided by the dataset into the Radar coordinate system with the provided extrinsics. Moreover, we used a loosely-coupled scheme based on LIO-SAM [Shan et al., 2020] and GPS for the self-built dataset to obtain the LiDAR groundtruth.

### 4.4 Experiments Results Evaluation

We compared the proposed system with the three methods on two public datasets [Kim et al., 2020, Barnes et al., 2020] and data from ZJU. These competitive methods include RO [Burnett et al., 2021], RO with loop detection [Kim and Kim, 2018] and Rall [Yin et al., 2021a]. We added the data from ZJU1-3 to fine-tune the weight of Rall's pre-trained 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 10. The Scan Context method is significantly better than Yeti Odom in repeated road sections because of including closed-loop detection. Our RoLM has the slightest trajectory error. In addition, the second and third rows of Figure 10 show the translational and rotational relative errors of our trajectory. Specifically, the relative errors equal the mean translation and rotation errors from 500m to 2500m with the incremental distance traveled.

Table 3: RMSE of Global Trajectories

| Sequence | Yeti Odom [Burnett et al., 2021] | | Scan Context [Kim et al., 2021] | | RaLL [Yin et al., 2021a] | | 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 |
| 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** |
| 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** |




(a) DCC03 (b) KAIST03 (c) Riverside02

(d) Oxford02 (e) HUANGLONG (f) ZJG04

Figure 10: Evaluation of four different methods on the MulRan(a)(b)(c), Oxford(d), and ZJU(e)(f) Dataset. Each subplot has three rows of images. The first row compares the estimated trajectories of the four methods and the ground truth, the second row shows the percentage of relative translation errors, and the third row shows the relative heading errors. We show details of the trajectory of the marked part of the pentagram.

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

Our system performs better in structured urban scenes (i.e. Oxford, DDC, Kasit) when it has a more significant error in degraded scenes (i.e. Riverside, Sejong). Among them, there 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 datasets 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 non-smooth trajectory, which also causes the Scan algorithm to crash.

Finally, we also designed a set of ablation experiments. We tested RoLM (SPD) and RoLM (SPD+ICP), respectively, which shows that SPD has significantly improved the system, and adding ICP can make it more stable. In a nutshell, 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

A heterogeneous localization system RoLM is proposed in this paper, which can correct the cumulative error of Radar odometry in real-time without closed loops:

- Point clouds are transformed into density maps of polar and Cartesian coordinates.
- We use the SPDs to get their rough external parameter estimates. After that, we perform a small-scale accurate alignment of the ICP based on the initial rough alignment.
- The obtained primary constraints are added to the overall pose graph optimization.

We demonstrate the reliability of the proposed localization system and its advantages over other methods in multi-session multi-scenario and our collected datasets.

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., 2021] and cross-sensor [Yin et al., 2021b] global relocalization method in the future.

### References


Aldera, R., De Martini, D., Gadd, M., and 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)*, pages 1190–1196. IEEE.

Anderson, S. and Barfoot, T. D. (2013). Ransac for motion-distorted 3d visual sensors. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2093–2099. IEEE.

Barnes, D., Gadd, M., Murcutt, P., Newman, P., and Posner, I. (2020). The oxford radar robotcar dataset: A radar extension to the oxford robotcar dataset. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6433–6438. IEEE.

Barnes, D. and Posner, I. (2020). Under the radar: Learning to predict robust keypoints for odometry estimation and metric localisation in radar. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9484–9490. IEEE.

Burnett, K., Schoellig, A. P., and 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. and Newman, P. (2018). Precise ego-motion estimation with millimeter-wave radar under diverse and challenging conditions. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6045–6052. IEEE.

Cen, S. H. and Newman, P. (2019). Radar-only ego-motion estimation in difficult settings via graph matching. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 298–304. IEEE.

Clark, S. and 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)*, volume 2, pages 1316–1321. IEEE.

Ding, X., Wang, Y., Xiong, R., Li, D., Tang, L., Yin, H., and Zhao, L. (2019). Persistent stereo visual localization on cross-modal invariant map. *IEEE Transactions on Intelligent Transportation Systems*, 21(11):4646–4658.

Furgale, P. and Barfoot, T. D. (2010). Visual teach and repeat for long-range rover autonomy. *Journal of field robotics*, 27(5):534–560.

Grupp, M. (2017). evo: Python package for the evaluation of odometry and slam. https://github.com/MichaelGrupp/evo.

Holder, M., Hellwig, S., and Winner, H. (2019). Real-time pose graph slam based on radar. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 1145–1151. IEEE.

Hong, Z., Petillot, Y., Wallace, A., and Wang, S. (2021). Radar slam: A robust slam system for all weather conditions. *arXiv preprint arXiv:2104.05347*.

Hong, Z., Petillot, Y., and Wang, S. (2020). Radarslam: Radar based large-scale slam in all weathers. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5164–5170. IEEE.

Huang, H., Ye, H., Sun, Y., and 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., and Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134.

Jose, E. and 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*, pages 3087–3092. IEEE.

Kim, G., Choi, S., and Kim, A. (2021). Scan context++: Structural place recognition robust to rotation and lateral variations in urban environments. *IEEE Transactions on Robotics*.

Kim, G. and 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)*, pages 4802–4809. IEEE.

Kim, G., Park, Y. S., Cho, Y., Jeong, J., and Kim, A. (2020). Mulran: Multimodal range dataset for urban place recognition. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6246–6253. IEEE.

Krüsi, P., Bücheler, B., Pomerleau, F., Schwesinger, U., Siegwart, R., and Furgale, P. (2015). Lighting-invariant adaptive route following using iterative closest point matching. *Journal of Field Robotics*, 32(4):534–564.

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., and Newman, P. (2017). 1 year, 1000km: The oxford robotcar dataset. *The International Journal of Robotics Research (IJRR)*, 36(1):3–15.

Marck, J. W., Mohamoud, A., vd Houwen, E., and van Heijster, R. (2013). Indoor radar slam a radar application for vision and gps denied environments. In *2013 European Radar Conference*, pages 471–474. IEEE.

Paton, M., MacTavish, K., Berczi, L.-P., van Es, S. K., and Barfoot, T. D. (2018). I can see for miles and miles: An extended field test of visual teach and repeat 2.0. In *Field and Service Robotics: Results of the 11th International Conference*, pages 415–431. Springer.

Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision*, pages 2564–2571. Ieee.

Săftescu, Ş., Gadd, M., De Martini, D., Barnes, D., and Newman, P. (2020). Kidnapped radar: Topological radar localisation using rotationally-invariant metric learning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4358–4364. IEEE.

Schuster, F., Keller, C. G., Rapp, M., Haueis, M., and Curio, C. (2016). Landmark based radar slam using graph optimization. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 2559–2564. IEEE.

Shan, T., Englot, B., Meyers, D., Wang, W., Ratti, C., and 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)*, pages 5135–5142. IEEE.

Sheeny, M., De Pellegrin, E., Mukherjee, S., Ahrabian, A., Wang, S., and Wallace, A. (2021). Radiate: A radar dataset for automotive perception in bad weather. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–7. IEEE.

Wang, H., Wang, C., Chen, C., and 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., and Xiong, R. (2021). Disco: Differentiable scan context with orientation. *IEEE Robotics and Automation Letters*, 6(2):2791–2798.

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

Yin, H., Wang, Y., Ding, X., Tang, L., Huang, S., and 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., and 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)*, pages 1–7. IEEE.

Yin, H., Xu, X., Wang, Y., and Xiong, R. (2021b). Radar-to-lidar: Heterogeneous place recognition via joint learning. *Frontiers in Robotics and AI*, 8:101.