2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

RoLM: Radar on LiDAR Map Localization

Yukai Ma[1], Xiangrui Zhao[1], Han Li[1], Yaqing Gu[1], Xiaolei Lang[1],Yong Liu[1,*]

***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 of Radar on LiDAR Map (RoLM), which can eliminate the accumulated error of radar odometry in real-time to achieve higher localization accuracy without dependence on loop closures. We embed the two sensor modalities into a density map and calculate the spatial vector similarity with offset to seek the corresponding place index in the candidates and calculate the rotation and translation. We use the ICP to pursue perfect matching on the LiDAR submap based on the coarse alignment. Extensive experiments on Mulran Radar Dataset, Oxford Radar RobotCar Dataset, and our data verify the feasibility and effectiveness of our approach.**

I. Introduction

Localization is an indispensable part of autonomous driving. Sensors such as GPS, camera, Radar, and LiDAR are widely used in current systems. However, there are many limitations to the direct application of certain sensors in specific environments. The accuracy of the visual localization is limited when illumination changes and the GPS becomes unreliable in urban canyons.

LiDAR is more precise than the sensors mentioned above. It can achieve accurate mapping under different illumination through point cloud registration. Nevertheless, its noise will increase significantly in rainy, snowy, and foggy weather. Accurate and stable localization in severe conditions is still a challenge. Moreover, radar has a large-scale observing ability, and strong robustness in extreme weather, such as rain and snow [1, 2]. It has also become a hotspot of research in recent years.

Therefore, radar localization on pre-build lidar maps will complement each other [3–5]. However, there are two challenges in matching and aligning the radar data and the lidar map: 1. FMCW Radar can only get the 2D information of the sweep line plane, which is one dimension less than the LiDAR; 2. Lidar point clouds can clearly outline even tiny objects, while radar polar can only show approximate changes in reflectivity in a scene, i.e., there is no direct correspondence between lidar points and radar points in space.

To align the data from different modalities, we utilize vectors with offset, which contains the idea of projection dimensionality reduction. For eliminating the occlusions and ghost reflections in the radar images, we extract the keypoints [6] of each frame and fuse the feature of multiple frames. The pose estimation of the system can be divided into four steps. First, obtain the initial pose estimation from radar odometry. Second, find a lidar frame similar to the radar keyframe and calculate its external parameters. After that, get the deviation between the current position and the map pose. At last, an optimization method of heterogeneous pose graph is introduced.

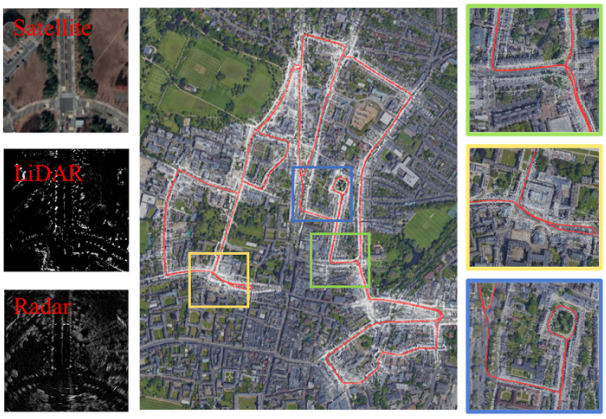


Fig. 1: Radar odometry generated using RoLM in which the colorful box shows some details. The left side of the figure provides the difference between lidar data and radar data in the same scene.

To verify the feasibility and effectiveness, we validate our algorithm on the Mulran Dataset [7], Oxford Radar RobotCar Dataset [8], and our ZJU Radar Datasets (Fig. 6).

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 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 [7] (multiple periods and scenarios), the Oxford Radar RobotCar Dataset [8, 9], and our Zhejiang University (ZJU) Dataset (Fig. 6) validate the effectiveness and feasibility of the proposed system.

The rest of this paper is organized as follows: Section II summarizes the related works in recent years. Our system is

[1]The authors are with the Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou, China. (*Yong Liu is the corresponding author, email: yongliu@iipc.zju.edu.cn)

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

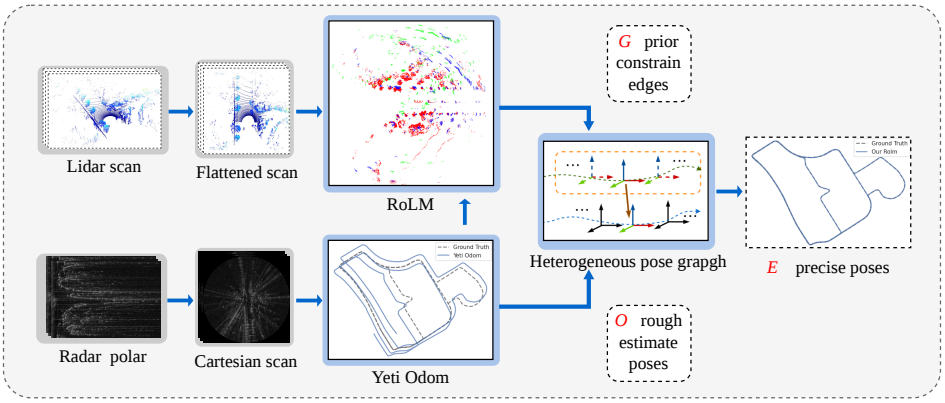


Fig. 2: The overall framework. Given the raw range measurements, RoLM can find the corresponding location index from a set of locations in the map and computes the pose bias to add to pose graph optimization.

introduced in Section III. Section IV includes some details about experimental settings and results on several datasets. Ultimately, we conclude a brief overview of our system and a future outlook in Section V.

II. RELATED WORK

A. 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 [10, 11]. There are generally two routes for radar feature extraction, traditional methods[12, 13] and neural network methods [14, 15]. We focus on the former in this paper. Owing to the redundant keypoints and false positives generated by the Constant False Alarm Rate (CFAR), Cen et al. proposed a feature detection method in 2018. This method scales the radar power spectrum according to its truth probability [12]. One year later, they proposed an updated detector [13], which identifies the continuously scanned region with high intensity and low gradient.

Later, based on Cen's work on radar feature extraction, Burnett et al. proposed Yeti Radar Odometry to eliminate motion distortion and the Doppler effect. The Gaussian filter is used in this algorithm instead of the binomial filter. This method also preserves multipath reflection. After feature extraction, the original data in polar coordinates are converted into Cartesian form. The ORB descriptor used to perform violent matching is then computed for each keypoint. After mismatch removal, the remaining matches are eventually sent to the MC-RANSAC-based estimator [16].

B. Localization on Pre-built Maps

The concept of localization on pre-built maps is relative to SLAM, with high real-time requirements. A pre-built map can avoid repeated online mapping for long-term fixed systems, which improves efficiency. The existing pre-built map localization algorithms include visual localization on visual and lidar maps [17, 18] and lidar localization on lidar maps [19]. These methods are still flawed in robustness.

Compared with LiDAR, radar has a powerful ability to penetrate smoke and dust, with characteristics of all-weather and anti-interference. Therefore, radar is widely used in localization research, covering indoor and outdoor scenes [20, 21]. In recent years, Navtech has provided radars with higher accuracy, less motion distortion, and a 360° range for corresponding research, producing rich results. Famous datasets [7–9, 22] and various algorithms [1, 6] were established.

Radar-on-LiDAR localization algorithm [3] proposed by Yin uses a conditional generative adversarial network (GAN): pix2pix [23], which is trained to transfer the radar data to fake lidar points. Then a Monte Carlo localization (MCL) system is built to achieve accurate localization on pre-built lidar maps. Later, Yin proposed an end-to-end learning system [5], achieving localization by a back-propagating gradient from the pose supervision. This method also utilizes the Kalman Filter to improve accuracy. The recent study [4] of Yin introduces a heterogeneous place recognition method via joint learning. And we further add heterogeneous prior constraints to the factor graph for global optimization.This method utilizes joint training to perform place recognition by extracting radar and lidar shared embeddings. In addition, there are also algorithms solving the problem by inputting satellite maps [24].

III. METHODOLOGY

This section introduces the proposed system. Overall framework of proposed RoLM of the system is shown in Fig. 2. Unlike existing methods for matching heterogeneous sensor information, we use Scan Projection Descriptors (SPD) to describe their similarity without using GPU for acceleration.

A. Motivation for RoLM

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.

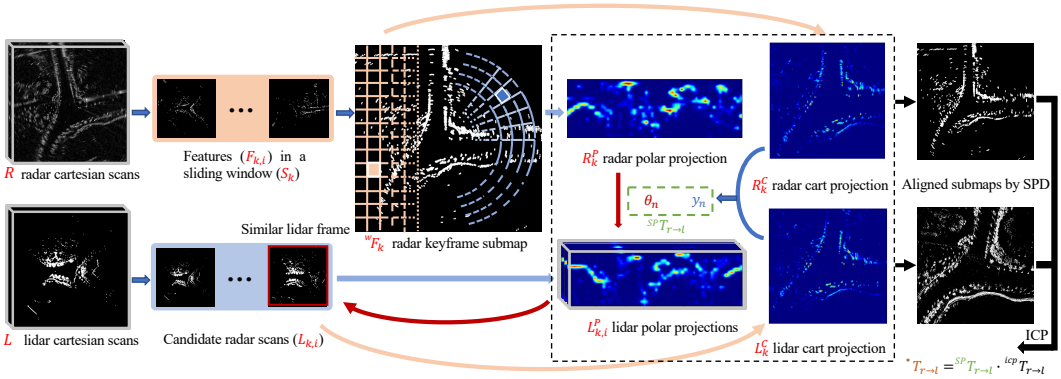


Fig. 3: Scan Projection Based Rough Estimate. Given the initial measurement set R, the feature points $F_{k,i}$ in the sliding window S_K are spliced into a keyframe self-map ${}^wF_k$. The most similar lidar frame is selected from the candidate list using polar and Cartesian projection descriptors, 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 ${}^*T_{r\to l}$.

B. 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 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.

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 [6] :

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

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

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 the feature point cloud ${}^rF_i$ of each frame 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 ${}^wF_k$:

$${}^wF_k = {}^r\boldsymbol{F}_{k-i}^k \dots {}^r\boldsymbol{F}_{k-1}^k \cup {}^r\boldsymbol{F}_k \cup {}^r\boldsymbol{F}_{k+1}^k \dots {}^r\boldsymbol{F}_{k+i}^k, (i \in N_A). \tag{2}$$

where N_A is a set whose size is equal to radar frame numbers in the window, and ${}^r\boldsymbol{F}_j^k$ means the register of ${}^r\boldsymbol{F}_j$ at ${}^wO_k$.

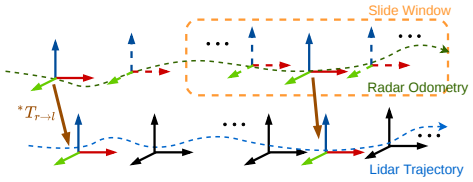


Fig. 4: A diagram illustrating the heterogeneous pose graph optimization procedure.

C. Scan Projection Descriptor for RoLM

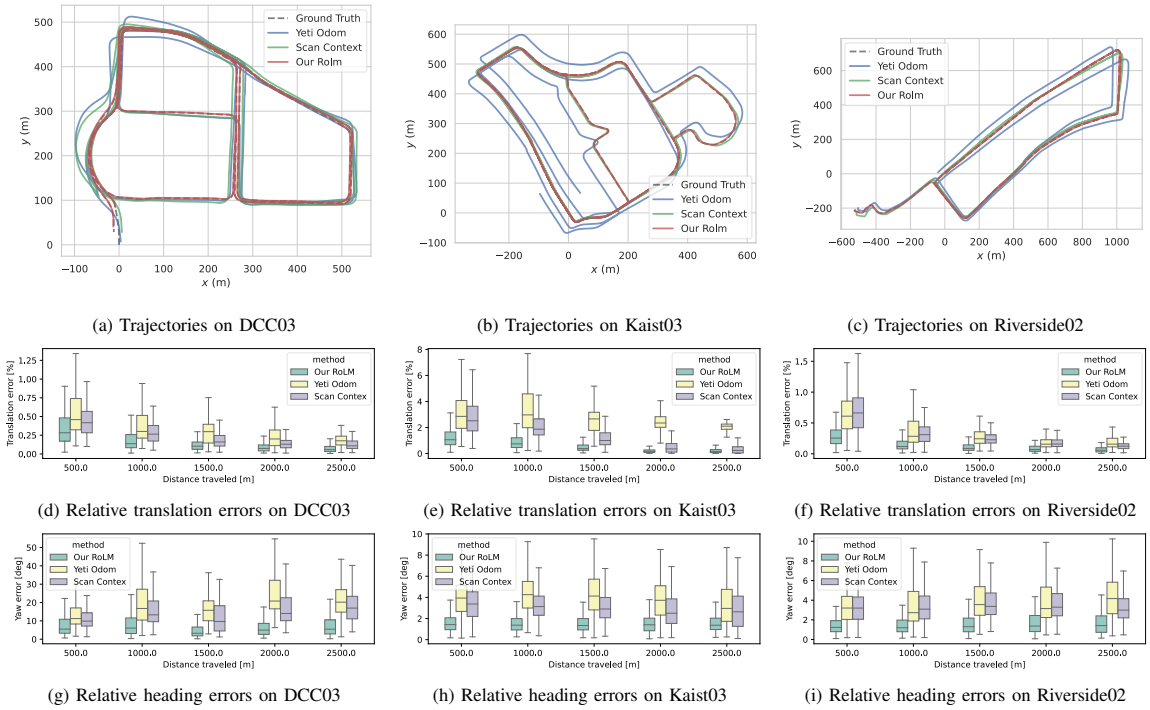
Inspired by [25], we replace the value of each bin with the normalized value of the point density of each patch. We first rasterize the space on the XY plane for a single point cloud frame, then count the number of points in all grids. Finally, we normalize the number of points in all grids to get the point cloud descriptor of this frame.

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

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. 60×20 sectors of $6^\circ \times 2m$ are used in IV). 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.



(a) Trajectories on DCC03 (b) Trajectories on Kaist03 (c) Trajectories on Riverside02

(d) Relative translation errors on DCC03 (e) Relative translation errors on Kaist03 (f) Relative translation errors on Riverside02

(g) Relative heading errors on DCC03 (h) Relative heading errors on Kaist03 (i) Relative heading errors on Riverside02

Fig. 5: Evaluation of three different methods on the MulRan Dataset. Trajectories (a)(b)(c), percentage of relative translation errors (d)(e)(f), and relative heading errors (g)(h)(i). Each column represents a different sequence of results.

- 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.

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.

D. Scan Projection Estimate

Although III-A 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 Eq. 3, and the distance between them can be expressed as:

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

The lidar keyframes used in IV 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 III-C, $d(S_R, S_L)$ also has 1 DOF along the horizontal axis. $S_{R,i}$ is an SPD whose columns are shifted from the original one by an amount i. Traverse the similarity of all frames with different offsets, and obtain the similar lidar with the miniature score. The alignment result is n_{align}.

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

Thus, we can obtain the rotation $\theta_{n_{align}} = n_{align} \times \frac{360^\circ}{N_A}$ and translation $y_{n_{align}} = (n_{align} - N_A) \times \frac{2*R_y}{N_A}$, ($R_y = 100m$ is the farthest distance of the point used to calculate S_R in IV) of any key measurement ${}^wF_k$ 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:

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

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

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

where ${}^*T_{r\to l}$ indicates the corresponding between radar and lidar poses, the entire initial alignment process can refer to the Fig. 3.

E. Heterogeneous Pose Graph Optimization

The optimized estimation (Fig. 4) of the whole system 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 groundtruth.

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.

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 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 ρ 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 IV-B 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.

IV. EXPERIMENTS

A. Implementation Strategy

We tested our RoLM system on the Mulran [7] and the Oxford [8, 9] Datasets. We provide a dataset that was collected using Navtech Radar CIR sensor and 32-rings LiDAR at the Zijingang campus of ZJU (Fig. 6). 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.

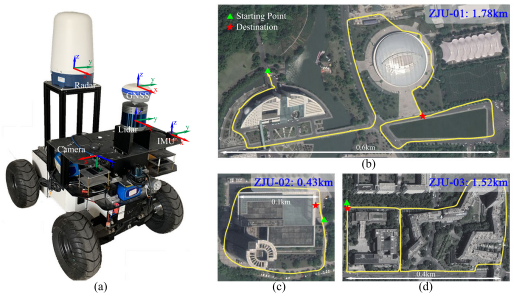


Fig. 6: (a) Our test vehicle with Radar, LiDAR, IMU, and RTK sensors. (b)(c)(d) Three paths contained in our dataset, collected in Zijingang Campus of ZJU.

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 Fig. 6a. It is equipped with Radar, LiDAR, IMU, and RTK sensors. We collected three representative short, medium, and long sequences in the Zijingang Campus of ZJU on January 15, 2022. Each sequence had at least one closed loop. Lane changes and different directions of the same road also existed in ZJU-02 and ZJU-03. There are also a large number of dynamic objects in the dataset. The collection route can refer to the right side of Fig. 6.

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.

B. Experiment Setup and Hypotheses

In all experiments, we set the size of the sliding window to 10 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 τ, 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

TABLE I: RMSE OF GLOBAL TRAJECTORIES

Sequence	Yeti Odom [6]		Scan Context [26]		RaLL [5]		Our Rolm(SPD) [5]		Our RoLM(SPD+ICP)	
	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	2.96	2.11	1.97	0.97	**1.23**	**0.78**	1.27
DCC-02	26.56	4.49	20.14	4.16	4.71	2.01	1.06	1.16	**0.80**	**0.91**
DCC-03	19.94	4.02	12.63	2.53	5.14	2.55	0.73	1.10	**0.60**	**1.06**
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	**0.77**	**1.24**	1.39	1.59
Kaist-03	30.55	3.50	4.94	2.41	1.27	1.50	0.82	**1.25**	**0.81**	1.47
Riverside-01	40.36	5.96	8.10	2.99	4.12	2.84	3.59	1.70	**3.32**	**1.57**
Riverside-02	37.56	3.40	11.47	3.29	**2.52**	1.93	6.10	2.40	2.72	**1.76**
Sejong-02	2893.17	38.14	2847.81	37.40	-	-	8.90	3.02	**5.20**	**1.43**
ZJU-01	51.26	48.02	50.32	47.59	85.61	88.64	8.87	6.98	**1.10**	**6.94**
ZJU-02	171.60	157.14	-	-	6.82	41.55	**1.17**	9.11	2.38	**8.50**
ZJU-03	137.25	178.48	-	-	253.52	96.99	2.46	6.55	**2.36**	**6.53**

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[27] and GPS for the self-built dataset to obtain the lidar groundtruth.

C. Experiments Results Evaluation

We compared the proposed system with the four methods on two public datasets [7, 9] and data from ZJU. These competitive methods include RO [6], RO with loop detection [26] and Rall [5]. We also verified the effectiveness of the proposed descriptor through ablation experiments. The results are presented in Tab. I. We use the open-source tool [28] for error calculation. An overview of the trajectory estimation results on some sequences is shown in Fig. 5. 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 Fig. 5 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.

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 (Tab. I). 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.

V. 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 [25] and cross-sensor [4] global relocalization method in the future.

REFERENCES

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

[2] Z. Hong, Y. Petillot, A. Wallace, and S. Wang. "Radar SLAM: A Robust SLAM System for All Weather Conditions". In: *arXiv preprint arXiv:2104.05347* (2021).

[3] H. Yin, Y. Wang, L. Tang, and R. Xiong. "Radar-on-lidar: metric radar localization on prior lidar maps". In: *2020 IEEE International Conference on Real-time Computing and Robotics (RCAR)*. IEEE. 2020, pp. 1–7.

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

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

[6] K. Burnett, A. P. Schoellig, and T. D. Barfoot. "Do we need to compensate for motion distortion and doppler effects in spinning radar navigation?" In: *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 771–778.

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

[8] W. Maddern, G. Pascoe, C. Linegar, and P. Newman. "1 Year, 1000km: The Oxford RobotCar Dataset". In: *The International Journal of Robotics Research (IJRR)* 36.1 (2017), pp. 3–15.

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

[10] J. W. Marck, A. Mohamoud, E. vd Houwen, and R. van Heijster. "Indoor radar SLAM A radar application for vision and GPS denied environments". In: *2013 European Radar Conference*. IEEE. 2013, pp. 471–474.

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

[12] S. H. Cen and P. Newman. "Precise ego-motion estimation with millimeter-wave radar under diverse and challenging conditions". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 6045–6052.

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

[14] R. Aldera, D. De Martini, M. Gadd, and P. Newman. "Fast radar motion estimation with a learnt focus of attention using weak supervision". In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 1190–1196.

[15] D. Barnes and I. Posner. "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. 2020, pp. 9484–9490.

[16] S. Anderson and T. D. Barfoot. "RANSAC for motion-distorted 3D visual sensors". In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2013, pp. 2093–2099.

[17] X. Ding, Y. Wang, R. Xiong, D. Li, L. Tang, H. Yin, and L. Zhao. "Persistent stereo visual localization on cross-modal invariant map". In: *IEEE Transactions on Intelligent Transportation Systems* 21.11 (2019), pp. 4646–4658.

[18] H. Huang, H. Ye, Y. Sun, and M. Liu. "Gmmloc: Structure consistent visual localization with gaussian mixture models". In: *IEEE Robotics and Automation Letters* 5.4 (2020), pp. 5043–5050.

[19] H. Yin, Y. Wang, X. Ding, L. Tang, S. Huang, and R. Xiong. "3d lidar-based global localization using siamese neural network". In: *IEEE Transactions on Intelligent Transportation Systems* 21.4 (2019), pp. 1380–1392.

[20] S. Clark and G. Dissanayake. "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. 1999, pp. 1316–1321.

[21] E. Jose and M. D. Adams. "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. 2005, pp. 3087–3092.

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

[23] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. "Image-to-image translation with conditional adversarial networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1125–1134.

[24] T. Y. Tang, D. De Martini, D. Barnes, and P. Newman. "Rsl-net: Localising in satellite images from a radar on the ground". In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 1087–1094.

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

[26] G. Kim and A. Kim. "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. 2018, pp. 4802–4809.

[27] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and R. Daniela. "LIO-SAM: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2020, pp. 5135–5142.

[28] M. Grupp. *evo: Python package for the evaluation of odometry and SLAM.* https://github.com/MichaelGrupp/evo. 2017.