IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

VGC-RIO: A Tightly Integrated Radar-Inertial Odometry with Spatial Weighted Doppler Velocity and Local Geometric Constrained RCS Histograms

Jianguang Xiang, Xiaofeng He, Zizhuo Chen, Lilian Zhang, Xincan Luo, and Jun Mao

***Abstract*—Recent advances in 4D radar-inertial odometry have demonstrated promising potential for autonomous localization under adverse conditions. However, effective handling of sparse and noisy radar measurements remains a critical challenge. In this letter, we propose a novel 4D radar-inertial odometry that fuses inertial pre-integration, radar scan matching and radar Doppler velocity in a tight way. Unlike most radar-inertial odometry that fuses the Doppler velocity with equal weights, we integrate each radar point's Doppler reading with an adaptive method that can adjust the weights according to the non-uniform point distribution. We further design a new point descriptor for point-to-point matching by combining the point cloud's local geometric and RCS (Radar Cross Section) information in a histogram. Extensive experiments conducted on multiple datasets demonstrate its localization accuracy improvement and adaptability under different environments and motion conditions.**

***Index Terms*—4D Radar-Inertial Odometry, Radar Cross Section, Histogram.**

I. Introduction

ACCURATE state estimation is crucial for autonomous vehicles and mobile robots navigating in complex and unknown environments, particularly in areas where GNSS is denied. Although LiDAR and camera-based solutions have shown excellent performance, they may fail under adverse weather conditions such as rain, fog, and snow [1]. In contrast, millimeter-wave radar (hereinafter referred to as radar), due to its operating frequency band characteristics, remains unaffected by these conditions [2]–[4], making it an attractive alternative. In particular, the emerging 4D radar provides additional elevation information and a denser point cloud than the conventional 3D radar, enabling three-dimensional imaging capabilities similar to those of LiDAR [5]. However, 4D radar point clouds remain sparser and noisier than those of LiDAR [6], making it challenging to directly adapt Lidar algorithms to radar systems.

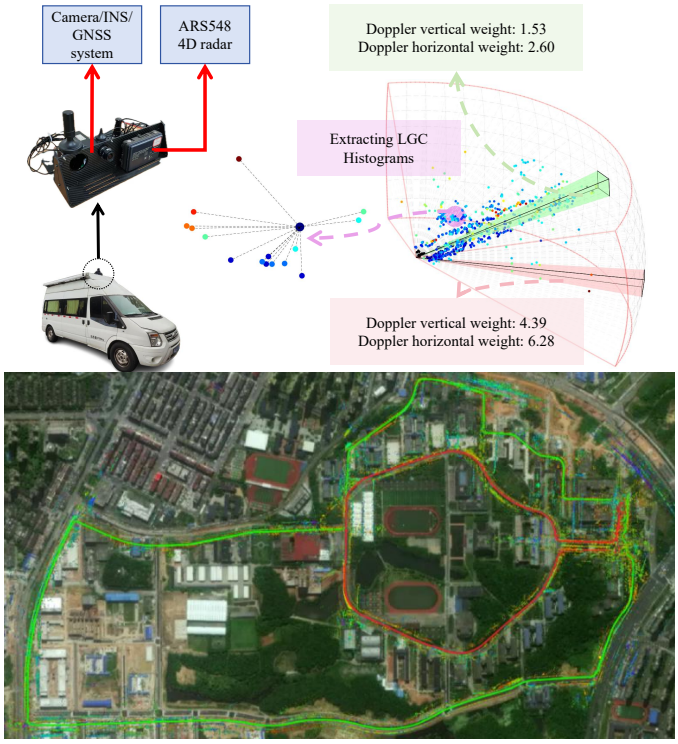


Fig. 1. Upper left: Vehicle platform and sensors. Upper right: Point cloud segmented into a spherical coordinate system-like structure, with Doppler weights and LGC histograms extracted (details in later sections). Dynamic points shown in black. Lower: Trajectory and mapping results visualized on Google Earth.

Recent studies show that the fusion system that combines the 4D radar and Inertial Measurement Unit (IMU) displays significant advantages for localization [7]–[9]. The Inertial System (INS) can predict radar inter-frame motions and help to filter out dynamic objects in radar scans, meanwhile, the Doppler velocity from radar directly provides motion constraints for INS. However, most existing Radar-Inertial Odometry (RIO) approaches equally weight the Doppler velocity of each radar point; this may overweight the velocity from points concentrated in a subregion and underweight the constraints from other parts, as the 4D radar points distributes unevenly. Radar points also provide implicit motion constraints by registering current scan with previous frames or prior map. As radar points are sparse and depend on environments, most radar odometry and SLAM adopt the point-to-point registra-

Manuscript received: May 13, 2025; Revised Month: August 15, 2025; Accepted: September 7, 2025. This paper was recommended for publication by Editor Javier Civera upon evaluation of the Associate Editor and Reviewers'comments. This research was funded by the National Nature Science Foundation of China, grant number: 62103430, 62103427, and 62073331 and Major Project of Natural Science Foundation of Hunan Province (No. 2021JC0004)

All authors are with the College of Intelligence Science and Technology, National University of Defense Technology, and National Key Laboratory of Equipment State Sensing and Smart Supporty, Changsha 410073, China (e-mail: xiangjianguang.22@nudt.edu.cn; hexiaofeng@nudt.edu.cn; 1316676920@qq.com; lilianzhang@nudt.edu.cn; 394013122@qq.com; maojun12@nudt.edu.cn).

Co-first authors: Jianguang Xiang and Xiaofeng He

Corresponding author: Jun Mao

Digital Object Identifier (DOI): see top of this page.

©2026 IEEE

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

tion strategy as it can generalize to different environments [10]. Recent study [8] shows using point-to-plane approach has better performance, but its accuracy may decrease in unstructured environments [11].

This work proposes a tightly integrated 4D radar-inertial odometry named VGC-RIO, which aims to make full use of the radar measurements (including point cloud geometric information, Doppler velocity and RCS) while also enabling it to adapt to different environments and motion conditions. The main contributions of our work are as follows:

- Instead of treating the Doppler velocity of each point equally, we propose a weighting method to assign different weights on each point's vertical and horizontal Doppler velocity according to their spatial distribution.
- We propose a novel point-description histogram named LGC, which integrates the spatial distribution of neighboring points and their RCS into a histogram. The LGC histograms help to register noisy radar points under vigorous motion conditions.
- We conducted ablation studies and comparative experiments on diverse and extensive datasets to comprehensively evaluate the performance of our proposed VGC-RIO. Specifically, we evaluated the robustness of the algorithm against sensor jitter on sequences exhibiting different levels of sensor jitter.

The remainder of this letter is organized as follows. Section II reviews the related work. Section III describes the proposed VGC-RIO method. Section IV presents ablation study and comparative experiments on publicly and self-constructed datasets, with detailed result analysis. Finally, Section V concludes the paper and discusses future work.

II. RELATED WORK

We briefly review related work on 3D and 4D radar odometry.

A. 3D Radar Odometry

Traditional 3D radars represent data as 2D point clouds or planar images due to lacking height information. Odometry methods can be categorized into feature-based and direct approaches.

Feature-based methods typically extract and match keypoints with distinctive descriptors. Barnes et al. [12] proposed a self-supervised framework that learns key point positions and descriptors by minimizing localization errors. Cen et al. [13] introduced a rotation-invariant descriptor based on angular histograms and annular regions, which enhances matching via graph optimization. Adolfsson et al. [14] leveraged the k-strongest radar echoes to reconstruct surface normals and employed point-to-line matching with multi-keyframe registration for precise odometry. Their follow-up work [15] further improved robustness using weighted residuals and joint multi-keyframe optimization.

Direct methods process raw radar measurements without explicit feature extraction. Park et al. [16] proposed a Fourier-Mellin transform-based approach that maximizes correlation in log-polar space for decoupled rotation and translation estimation. Kung et al. [17] addressed sparsity by constructing probabilistic radar submaps and applying a weighted normal distributions transform for scan matching. Haggag et al. [18] developed a probabilistic ego-motion estimation framework tailored for automotive radar data.

B. 4D Radar Odometry

Compared with 3D radar, 4D radar provides additional height information, enhanced point cloud quantity, and resolution. However, the point cloud density remains low (usually under 1000 points per frame), with significant noise. Current research mainly focuses on these issues.

In 4D iRIOM [19], they fuse radar and imu, adopt a distribution-to-multi-distribution matching strategy, and leverage a robust Graduated Non-Convexity (GNC) algorithm to eliminate dynamic-point interference. Wang et al. [20] present a tightly-coupled 4D radar-inertial SLAM using hybrid scan-to-scan/scan-to-submap registration and factor graph optimization for a robust pose estimation. In their subsequent work [8], they propose a Doppler-aided radar-inertial SLAM framework, tightly coupling Doppler velocity and point cloud in a frontend and backend optimization for robust localization and mapping. Yang et al. [21] present a ground-optimized 4D radar-inertial odometry using uncertainty-aware filtering and Gaussian Process based velocity pre-integration to enhance vertical accuracy and address radar-IMU asynchrony. Michalczyk et al. [22] propose a tightly-coupled Extended Kalman Filter with stochastic cloning for 3D point matching robustness. In their subsequent work [23], persistent surfaces are introduced to enhance localization accuracy by considering stable radar points as landmarks.

Some other works enhance registration with RCS or intensity. Huang et al. [7] propose a radar-inertial odometry system using RCS for improved point-to-point correspondences. In Milli-RIO [24], intensity information is used to eliminate erroneous matches. Kim et al. [25] propose a feature extraction algorithm based on a polar coordinate network using RCS for robust odometry. DGRO [26] integrates 4D radar and gyroscope data with RCS for point cloud filtering and weighted registration.

Current research on 4D radar odometry primarily focuses on adapting 3D LiDAR odometry methods with minor modifications to fit radar odometry [8] [9] [19], such as the ICP [27], NDT [28], and their variants. Methods based on feature matching, such as LeGO-LOAM [29] and Fast-LIO [30], which rely on features like planes and edges, are still challenging due to the sparsity of 4D radar point clouds compared to LiDAR. Additionally, methods using handcrafted feature descriptors, such as Fast Point Feature Histograms (FPFH) [31], and Signature of Histogram of Orientation (SHOT) [32], are computationally intensive and thus difficult to apply in real-time odometry applications.

The VGC-RIO we propose draws inspiration from previous works [7], [14], [19], [23], [31].

III. METHODOLOGY

A. Notations

The notations and frame definitions used in this paper are defined in this part. The world frame is denoted as $(\cdot)^w$. The

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

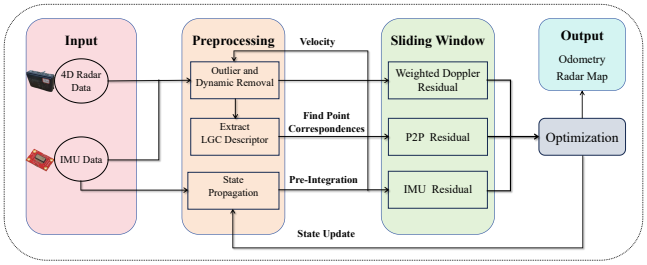


Fig. 2. System overview of VGC-RIO

body frame, which is consistent with the IMU frame, are both denoted as $(\cdot)^b$, while the radar frame are represented as $(\cdot)^r$. The rotation matrix $\boldsymbol{R}$ and Hamilton quaternions $\boldsymbol{q}$ are used to represent the rotations. b_k and r_k respectively represent the body frame and the radar frame at the time when the k-th frame of radar data is obtained. $\boldsymbol{R}_r^b$ and $\boldsymbol{t}_r^b$ denote the rotation matrix and translation from the radar frame to the body frame, respectively. $\boldsymbol{q}_b^w$ and $\boldsymbol{p}_b^w$ denote the rotation and translation from the body frame to the world frame, respectively. $\otimes$ denotes the multiplication of two quaternions. $\boldsymbol{g}^w = [0, 0, g]^T$ is the gravity vector in the world frame. Finally, $\hat{(\cdot)}$ indicates the estimation of a certain quantity.

B. Problem Formulation

The proposed VGC-RIO is shown in Fig. 2. The system's inputs consist of radar point clouds and IMU data. Firstly, preprocessings are implemented to remove dynamic radar points and to divide the points into different sub-regions. Then three residual terms are constructed, which are the IMU pre-integration residual, the weighted Doppler residual using static points, and the P2P (Point-to-Point) residual based on LGC histograms for point-to-point correspondences. Finally, the three residuals in the sliding window are optimized as a whole with nonlinear solvers. The complete state of the system is defined as follows:

$$\begin{aligned} \mathcal{X} &= [\boldsymbol{x}_0, \boldsymbol{x}_1, \cdots, \boldsymbol{x}_k, \boldsymbol{l}_0^w, \boldsymbol{l}_1^w, \cdots, \boldsymbol{l}_{m-1}^w] \\ \boldsymbol{x}_k &= \left[\boldsymbol{p}_{b_k}^w, \boldsymbol{v}_{b_k}^w, \boldsymbol{q}_{b_k}^w, \boldsymbol{b}_{a_k}, \boldsymbol{b}_{g_k}\right]^T, \quad k \in [0, n-1] \end{aligned} \tag{1}$$

where $\boldsymbol{x}_k$ represents the IMU state at the time of obtaining the k-th radar frame; $\boldsymbol{p}_{b_k}^w$, $\boldsymbol{v}_{b_k}^w$, $\boldsymbol{q}_{b_k}^w$, $\boldsymbol{b}_{a_k}$ and $\boldsymbol{b}_{g_k}$ denote the position, rotation (represented by a quaternion), velocity, accelerometer bias, and gyroscope bias of the k-th frame respectively. $\boldsymbol{l}_j^w \in \mathbb{R}^3$ represents the j-th landmark. n denotes the number of radar frames within the sliding window and m is the total number of landmarks. The optimization problem we construct is formulated as follows:

$$\begin{aligned} \min_{\mathcal{X}} \Bigg\{ & \sum_{k \in I} \left\| r_I(\hat{\boldsymbol{z}}_{b_{k+1}}^{b_k}, \mathcal{X}) \right\|_{\boldsymbol{P}_k^{k+1}}^2 \\ & + \sum_{i \in D} \rho\Big(\left\| \boldsymbol{w}_{D_h} r_D(\boldsymbol{p}_i^r, v_i^r, \mathcal{X}) \right\|^2 \Big) \\ & + \sum_{j \in P} w_P \rho\Big(\left\| r_P(\boldsymbol{l}_j^w, \boldsymbol{p}_j^r, \mathcal{X}) \right\|^2 \Big) \Bigg\} \end{aligned} \tag{2}$$

where $\rho(\cdot)$ is the Huber norm [33]. $\boldsymbol{w}_{D_h}$ and w_P are residual weights. $\boldsymbol{I}$ is the set of IMU measurements, $\boldsymbol{D}$ is the set of static points, and $\boldsymbol{P}$ is the set of static points that have been observed at least thrice in the current sliding window. Detailed explanations of $\boldsymbol{w}_{D_h}$, w_P and the three types of residuals r_I, r_D, and r_P are provided in Section III-C, Section III-E, and Section III-G, respectively.

C. IMU Pre-integration

To avoid the redundancy of repeated integration, we employ the IMU pre-integration. According to [34], the residual of pre-integrated IMU measurements between two consecutive radar frames b_k and b_{k+1} within the sliding window can be defined as:

$$\begin{aligned} & r_I\left(\hat{\boldsymbol{z}}_{b_{k+1}}^{b_k}, \mathcal{X}\right) \\ & = \begin{bmatrix} \boldsymbol{R}_w^{b_k}\left(\boldsymbol{p}_{b_{k+1}}^w - \boldsymbol{p}_{b_k}^w + \frac{1}{2}\boldsymbol{g}^w \Delta t_k^2 - \boldsymbol{v}_{b_k}^w \Delta t_k\right) - \hat{\boldsymbol{\alpha}}_{b_{k+1}}^{b_k} \\ \boldsymbol{R}_w^{b_k}\left(\boldsymbol{v}_{b_{k+1}}^w + \boldsymbol{g}^w \Delta t_k - \boldsymbol{v}_{b_k}^w\right) - \hat{\boldsymbol{\beta}}_{b_{k+1}}^{b_k} \\ 2\left[\boldsymbol{q}_{b_k}^{w-1} \otimes \boldsymbol{q}_{b_{k+1}}^w \otimes \left(\hat{\boldsymbol{\gamma}}_{b_{k+1}}^{b_k}\right)^{-1}\right]_{xyz} \\ \boldsymbol{b}_{a_{k+1}} - \boldsymbol{b}_{a_k} \\ \boldsymbol{b}_{g_{k+1}} - \boldsymbol{b}_{g_k} \end{bmatrix} \end{aligned} \tag{3}$$

where $\hat{\boldsymbol{z}}_{b_{k+1}}^{b_k} = \begin{bmatrix} \hat{\boldsymbol{\alpha}}_{b_{k+1}}^{b_k} & \hat{\boldsymbol{\beta}}_{b_{k+1}}^{b_k} & \hat{\boldsymbol{\gamma}}_{b_{k+1}}^{b_k} \end{bmatrix}$ represents the pre-integrated IMU measurements between two radar frames. Δt_k is the time interval between the two radar frames. The operator $[\cdot]_{xyz}$ is used to extract the vector part of a quaternion. Due to space limitations, more details about this formula can be found in [34].

D. Radar Data Preprocessing

Raw radar data requires preprocessing to remove invalid and dynamic points, as they can severely impact state estimation accuracy if unprocessed.

1) Outlier Removal: We adopt the methods used in [35] to filter out invalid points. First, we transform the point cloud in the k-th radar frame $\boldsymbol{P}_k^{r_k}$ into frame $\boldsymbol{r}_{k+1}$ to obtain $\boldsymbol{P}_k^{r_{k+1}}$. Then, for each point in $\boldsymbol{P}_{k+1}^{r_{k+1}}$, perform a fixed-radius (δ_p) nearest-neighbor search; if no neighbor from $\boldsymbol{P}_k^{r_{k+1}}$ is found, mark and remove it as an outlier.

2) Dynamic Removal: Similar to the method in [7], we use the IMU to assist in filtering out dynamic points. $\hat{\boldsymbol{v}}_{b_{k+1}}^w$ is predicted from $\hat{\boldsymbol{v}}_{b_k}^w$ and IMU data. If $\boldsymbol{p}_i^r$ is a static point in the current frame, it should satisfy the following:

$$\begin{aligned} \boldsymbol{v}_{\text{error}_i} = & \left(\frac{\boldsymbol{p}_i^r}{\|\boldsymbol{p}_i^r\|}\right)^T \boldsymbol{R}_b^r \Bigg(\hat{\boldsymbol{R}}_w^{b_{k+1}} \hat{\boldsymbol{v}}_{b_{k+1}}^w \\ & + \left(\hat{\boldsymbol{\omega}}_{k+1} - \boldsymbol{b}_{g_{k+1}}\right) \times \boldsymbol{t}_r^{b^T} \Bigg) + v_i^r \approx 0 \end{aligned} \tag{4}$$

where $\hat{\boldsymbol{R}}_w^{b_{k+1}}$ is the estimated rotation matrix from the world frame to the current body frame. v_i^r is the Doppler velocity measured for the point $\boldsymbol{p}_i^r$ (The Doppler velocity in radar is a radial velocity and is a scalar quantity, with negative values indicating that the target is moving towards the radar). $\hat{\boldsymbol{\omega}}_{k+1}$ is the angular velocity measured by the IMU, and $\boldsymbol{b}_{g_{k+1}}$ denotes the gyroscope bias of IMU. Firstly, we preliminarily remove points with large velocity errors, as shown below:

$$\mathcal{P}_{\text{static}}^{r_{k+1}} = \left\{ \boldsymbol{p}_i^r \;\middle|\; |v_{\text{error}_i}| < v_{\text{thr}} \text{ and } \left\| \frac{v_{\text{error}_i}}{v_i^r} \right\| < p_{\text{thr}} \right\} \tag{5}$$

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

where $\mathcal{P}_{\text{static}}^{r_{k+1}}$ represents the set of static points in the current frame. v_{thr} and p_{thr} are the velocity error threshold and the percentage threshold of velocity error, respectively.

3) Point Cloud Division: In order to extract key points and count the number of points in different directional intervals subsequently, we divided the entire point cloud in a frame into different intervals. Concretely, we divide the point cloud into several equal length intervals based on azimuth and elevation. Taking the azimuth as an example, it is shown as follows:

$$\mathcal{P}_k = \left\{ p_i \mid k \leq \frac{\theta_i - \theta_{start}}{\theta_{res}} < k+1 \right\}, \quad k \in [0, s-1] \quad (6)$$

where $\mathcal{P}_k$ represents the set of points belonging to the k-th interval. s is the number of intervals. θ_{start} is the starting azimuth, θ_{res} is the interval spacing, and θ_i is the azimuth of p_i. The aforementioned operation is analogous to establishing a discrete spherical coordinate system, which involves only two angular parameters and does not pertain to the radial distance. As shown in Fig. 1. The purpose of this approach will be demonstrated in Sections III-E and Sections III-F.

E. Weighted Doppler Velocity Residual

Ideally, the Doppler velocity measured by radar should correspond to the projection of the target-radar relative velocity vector along the radar's line-of-sight direction. In the data preprocessing step, the dynamic points have already been filtered out. Therefore, for the retained static points, they should satisfy Eq. 4. The unweighted Doppler velocity residual of $\boldsymbol{p}_i^r$ is defined as:

$$\begin{aligned} r_D(\boldsymbol{p}_i^r, v_i^r, \mathcal{X}) = \left(\frac{\boldsymbol{p}_i^r}{\|\boldsymbol{p}_i^r\|} \right)^T \boldsymbol{R}_b^r \Big(\hat{\boldsymbol{R}}_w^{b_k} \hat{\boldsymbol{v}}_{b_k}^w \\ + (\hat{\boldsymbol{\omega}}_k - \boldsymbol{b}_{g_k}) \times \boldsymbol{t}_r^{b^T} \Big) + v_i^r \end{aligned} \quad (7)$$

Unlike LiDAR, radar provides sparse point clouds and tends to concentrate reflection points. Therefore, treating all points equally [7] [22] [35] may result in an excessive concentration of constraints in directions with dense point clouds, while neglecting constraints in directions with sparse point clouds. This is because points in the same direction provide similar constraints for velocity estimation. To make better use of all the points in the scan, we propose weighting the points according to their spatial distribution. Based on the point cloud division in the preprocessing, s azimuth intervals $\{\Omega_h\}_{h=1}^s$ and t elevation intervals $\{\Theta_h\}_{h=1}^t$ are obtained. Specifically, considering the azimuth intervals as an example, the total constraint of the h-th azimuth interval is:

$$C_h = \sum_{i \in \Omega_h} (w_h \cdot r_D(\boldsymbol{p}_i^r, v_i^r, \mathcal{X}))^2 \quad (8)$$

Under the assumption that all static points in a frame have approximately equal Doppler residuals ($r_D(\boldsymbol{p}_i^r, v_i^r, \mathcal{X}) \approx \overline{r}$), the above equation can be expressed as:

$$C_h = \sum_{i \in \Omega_h} (w_h \cdot \overline{r})^2 \quad (9)$$

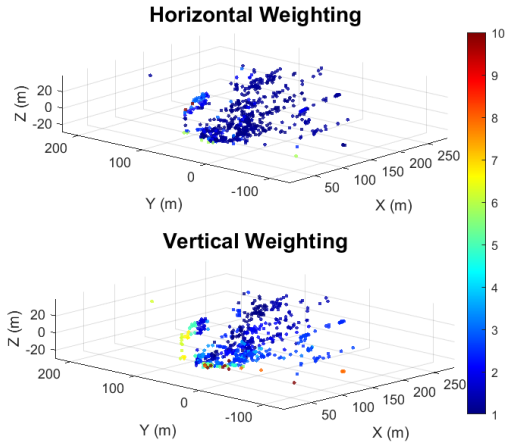


Fig. 3. Visualization of weights at different points and in different directions. The point cloud is colored according to the weights.

The guiding principle of weight calculation is that the constraints in all directional intervals should be kept approximately the same. Then it satisfies the following:

$$w_h^2 \cdot n_h \cdot \overline{r}^2 = C \quad (10)$$

where C is a constant, and n_h is the number of points in the h-th directional interval. By eliminating the influence of the constant term and $\overline{r}$, the Doppler weight for the h-th azimuth interval is:

$$w_h = \frac{1}{\sqrt{n_h}} \quad (11)$$

The visualization of the weights is shown in Fig. 3. The weighted Doppler residual of $\boldsymbol{p}_i^r$ is then defined as follows:

$$\begin{aligned} \boldsymbol{w}_{D_h} r_D(\boldsymbol{p}_i^r, v_i^r, \mathcal{X}) = \begin{bmatrix} w_{D_{h_p}} \sin\theta \\ w_{D_{h_q}} \cos\theta \end{bmatrix} \cdot \Bigg(\left(\frac{\boldsymbol{p}_i^r}{\|\boldsymbol{p}_i^r\|} \right)^T \boldsymbol{R}_b^r \\ \left(\hat{\boldsymbol{R}}_w^{b_k} \hat{\boldsymbol{v}}_{b_k}^w + (\hat{\boldsymbol{\omega}}_k - \boldsymbol{b}_{g_k}) \times \boldsymbol{t}_r^{b^T} \right) + v_i^r \Bigg) \end{aligned} \quad (12)$$

where

$$\begin{cases} w_{D_{h_p}} = \frac{1}{\sqrt{n_p}} & \text{for } p \in [0, s-1] \\ w_{D_{h_q}} = \frac{1}{\sqrt{n_q}} & \text{for } q \in [0, t-1] \end{cases} \quad (13)$$

where p and n_p represent the sequence number and the number of points in the azimuth interval to which $\boldsymbol{p}_i^r$ belongs, respectively. q and n_q represent the sequence number and the number of points in the elevation interval to which $\boldsymbol{p}_i^r$ belongs, respectively. θ represents the elevation.

Weights are assigned lower for dense directions and higher for sparse directions to achieve more comprehensive constraints. To enhance the robustness of the estimation, the weights are normalized to the interval [1, 10].

F. Local Geometric Constrained RCS Histograms

A single feature alone is typically not sufficient to provide a comprehensive description. Therefore, we propose a feature histogram descriptor that combines local geometric feature with the RCS feature. The detail of RCS can be found in [7]. Here, we directly present the assumptions followed by

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

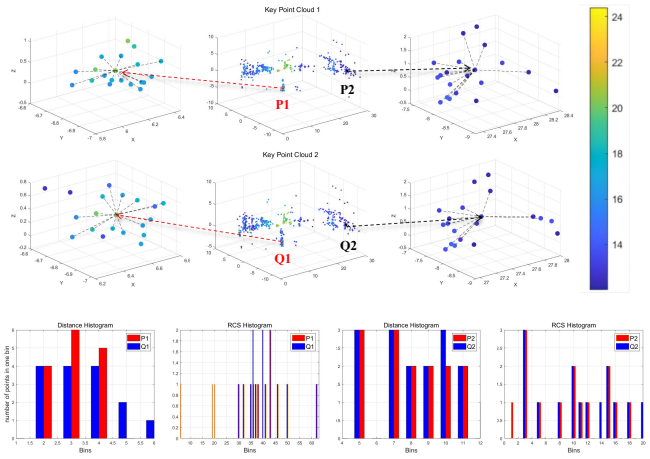


Fig. 4. The LGC histograms of corresponding points between consecutive radar frames. For the sake of visualization, the distance dimension and the RCS dimension of the 2D histogram are compared separately.

the proposed method: (1) The RCS of the same point remains nearly invariant between two consecutive radar scan frames. (2) Points with higher RCS are usually stable and persistent.

Subsequently, we provide a detailed exposition on the process of histogram construction and matching:

1) LGC Histogram Construction: In the radar data preprocessing step, we obtain several intervals through cloud division. We then extend the k-strong filtering algorithm, originally proposed in [14], to a three-dimensional approach. To elaborate, we identify the k points (k is set to 30 in the subsequent experiments) with the highest RCS values in each interval as key points, which collectively form the key point cloud $\boldsymbol{P}_{\text{key}}$. This processing ensures that the retained points are spatially uniformly distributed and guarantees their stable existence (based on assumption (2)). Subsequently, we extract the nearest neighbor distance and RCS features for each point in $\boldsymbol{P}_{\text{key}}$ and encode them into a 2D histogram. Specifically, we first construct a kd tree for $\boldsymbol{P}_{\text{key}}$, and then traverse each point $\boldsymbol{p}_i$ in $\boldsymbol{P}_{\text{key}}$ ($i \in [0, l-1]$). For each point, we find its k_{nn} nearest neighbors $\boldsymbol{p}_{n_j}$ ($j \in [0, k_{nn}-1]$).

The distance and RCS features are defined as follows:

$$f_{dis} = \left\| \boldsymbol{p}_i - \boldsymbol{p}_{n_j} \right\| \tag{14}$$

$$f_{rcs} = \mathrm{RCS}_{n_j} \tag{15}$$

where $\|\cdot\|$ represents the Euclidean distance and RCS_{n_j} represents the RCS of $\boldsymbol{p}_{n_j}$.

The construction of histogram for each point $\boldsymbol{p}_i$ can be represented by the following equation:

$$\begin{aligned} \boldsymbol{H}_i(n,m) = \sum_{j=0}^{k_{nn}-1} & \mathbb{I}\left(f_{\text{dis}}(\boldsymbol{p}_i, \boldsymbol{p}_{n_j}) \in \text{bin}_n\right) \cdot \\ & \mathbb{I}\left(f_{\text{rcs}}(\boldsymbol{p}_{n_j}) \in \text{bin}_m\right), \\ n \in [0, l_{dis}-1], \quad & m \in [0, l_{rcs}-1] \end{aligned} \tag{16}$$

where $\mathbb{I}(\cdot)$ is the indicator function (1 if true, 0 otherwise). Where l_{dis} and l_{rcs} represent the total number of bins in the distance dimension and the RCS dimension, respectively. In histogram, the bin width for distance is denoted as res_{dis}, and for RCS as res_{rcs}. The construction of LGC histogram is depicted in Fig. 4.

2) LGC Histogram Matching: After obtaining the feature histograms of all keypoints in two consecutive frames, we can proceed to identify matching point pairs. To enhance the efficiency of matching, the RCS is employed as an initial screening criterion. Concretely, point pairs with RCS differences exceeding a predefined threshold are deemed non-matching (based on assumption (1)).

Considering the effects of discretization binning and radar measurement noise, the mapped positions of targets in histograms may exhibit slight deviations. Therefore, we propose a 2D histogram similarity measure called Neighborhood-Expanded Histogram Intersection (NHI), which effectively enhances the robustness of similarity computation by introducing a neighborhood search strategy. For histograms $\boldsymbol{A}$ and $\boldsymbol{B}$, the NHI similarity S_{NHI} is defined as:

$$S_{NHI} = \sum_{(i,j)\in\boldsymbol{\Omega}_A} \max_{\substack{(x,y)\in\Omega_B \\ x\in[i-r,i+r] \\ y\in[j-r,j+r]}} \left(\min(\boldsymbol{A}_{i,j}, \boldsymbol{B}_{x,y}) \cdot W(i,j,x,y)\right) \tag{17}$$

where $\boldsymbol{\Omega}_A = \{(i,j) \mid \boldsymbol{A}_{i,j} \neq 0\}$, represents the set of coordinates of non-zero elements in histogram $\boldsymbol{A}$. $\boldsymbol{\Omega}_B = \{(i,j) \mid \boldsymbol{B}_{i,j} \neq 0\}$, represents the set of coordinates of non-zero elements in histogram $\boldsymbol{B}$. r represents the radius of the neighborhood search, which is set to 1 by default. Among them, $W(i,j,x,y)$ is the distance weight, expressed as:

$$W(i,j,x,y) = \frac{1}{1+|x-i|+|y-j|} \tag{18}$$

where $|\cdot|$ denotes the Manhattan distance. If a point pair exhibits the highest similarity and surpasses a predefined threshold ($thres_{NHI}$), it is considered a successful match. Considering that there may be a few incorrect matches, we employ the Random Sample Consensus (RANSAC) to remove outliers.

G. Point-to-Point Residual

For a robust estimation, when constructing residual, we only consider the matched point pairs in the key point cloud and points in the non-key point cloud that have been successfully matched more than a certain number of times (set to 3 in the subsequent experiments). The P2P residual is defined as follows:

$$r_P(\boldsymbol{l}_j^w, \boldsymbol{p}_j^{r_k}, \mathcal{X}) = \boldsymbol{l}_j^w - \left(\boldsymbol{R}_{b_k}^w \left(\hat{\boldsymbol{R}}_{\boldsymbol{r}}^{\boldsymbol{b}} \boldsymbol{p}_j^{r_k} + \boldsymbol{t}_r^b\right) + \boldsymbol{p}_{b_k}^w\right) \tag{19}$$

For continuously observed points, which are considered more stable, higher weights (w_P) are assigned when constructing P2P constraints. We use the Ceres-Solver [36] to solve the entire optimization problem (Eq. 2).

IV. EXPERIMENTS

A. Datasets and Setup

We tested our proposed VGC-RIO system on handheld and vehicle-mounted platforms in the real world, as shown in Fig. 1. The platform is equipped with Continental's ARS548 4D

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

radar [1] and a SCHA63T IMU [2]. The ground truth was provided by RTK (Real-Time Kinematic) positioning.

We constructed seven sequences, covering a variety of scenarios, as detailed in Tab. I. Motion sequences with three different levels of jitter were collected on basketball courts. For court1, we used handheld devices under normal operating conditions, while for court2 and court3, we deliberately shook the devices vertically and horizontally to acquire the data. In addition, we evaluate our approach using the ARS548 radar on three publicly available datasets: SNAIL [37], HeRCULES [38] and RIO [7]. For a fair comparison, we evaluated trajectories on the plane in both SNAIL and HeRCULES datasets. The key parameters of the VGC-RIO are detailed in Tab. II, consistent throughout the paper.

TABLE I
SUMMARY OF SEVEN SELF-CONSTRUCTED DATASETS

Platform	Name	Length	Speed	Shake	Dynamic
Handheld	Court 1	≈ 300m	≈ 1m/s	Mid	Low
	Court 2	≈ 150m	≈ 1m/s	High	Low
	Court 3	≈ 200m	≈ 1m/s	Extreme	Low
	Garden	≈ 300m	≈ 1m/s	Mid	Mid
Car	Loop 1	2635m	25km/h	Low	Mid
	Loop 2	5468m	25km/h	Low	Mid
	Overpass	7422m	50km/h	Mid	High

[1] "Shake" characterizes the degree of sensor jitter, while "Dynamic" indicates the environmental dynamism. Both are categorized into Low, Mid, High, and Extreme levels.

TABLE II
ALGORITHM PARAMETERS SETUP

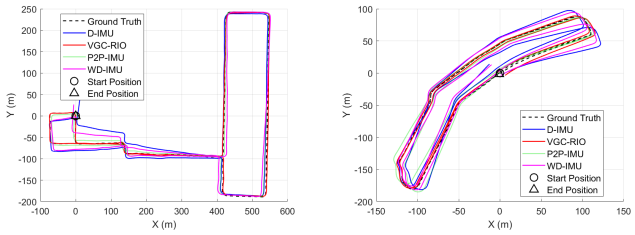
Parameters	value
Histogram Distance Bin Width (res_{dis})	0.2
Histogram RCS Bin Width (res_{rcs})	1
Histogram Distance Number of Bins (l_{dis})	100
Histogram RCS Number of Bins (l_{rcs})	50
Neighbor count for histogram (k_{nn})	30
Threshold of NHI ($thres_{NHI}$)	5

B. Ablation Study

To evaluate the contribution of each component to the system, we conduct ablation studies on sequence "20240115/2" from the SNAIL dataset and "Library Day 1" from the HeRCULES dataset, evaluating four variants: (1) Unweighted Doppler residuals with IMU residuals (D-IMU); (2) Weighted Doppler residuals with IMU residuals (WD-IMU); (3) P2P residuals with IMU residuals (P2P-IMU); (4) Full System (VGC-RIO). The results are shown in Tab. III.

TABLE III
ABLATION STUDY ON PUBLICLY DATASETS: RMSE OF ABSOLUTE TRAJECTORY ERROR (M)

Method	20240123_3	Library Day 1
D-IMU	17.58	10.88
WD-IMU	14.25	8.06
P2P-IMU	6.23	5.82
VGC-RIO	**1.85**	**4.02**



(a) **20240123_3** (b) **Library Day 1**

Fig. 5. Trajectory Comparison of Four Variants on the SNAIL and HeRCULES Datasets

Firstly, Full System (VGC-RIO) achieved the optimal localization accuracy across all sequences, which indicates that each component contributes to the system. The combination of the weighted Doppler residuals and the P2P residuals can further enhance the localization accuracy. In comparison with WD-IMU and D-IMU, it is observed that the localization accuracy has been improved across all sequences. This confirms the effectiveness of the weighting strategy. As shown in Fig. 5, comparing WD-IMU with the Full System, we find that when the P2P residual constraint is disabled, relying solely on velocity constraints leads to significant drift in the heading angle estimation. This indicates that the P2P residual constraint is crucial for enhancing the stability and accuracy of the heading angle estimation.

C. Comparison Experiment

We compared the proposed method with the state-of-the-art radar odometry/SLAM algorithms using the results reported in Doppler-SLAM [8], RaI-SLAM [9], and RIO [7]. For reference, we also present the results from Fast-LIO2 [39]. Given the critical role of point cloud processing in radar odometry performance, we classify methods into two paradigms: **Point-to-Point** and **Point-to-Plane** registration.

As presented in Tab. IV. Doppler-Odometry shows a significant localization accuracy advantage over other methods on SNAIL dataset (urban environment), where the radar point clouds satisfy the local-planar constraint. However, as presented in Tab. V, as the environment becomes more challenging in the HeRCULES dataset, Doppler-Odometry's precision advantage becomes less obvious. When the planar constraint becomes weak, its accuracy degrades severely even with loop closure, as seen in the "Mountain Day 1" sequence (mountain environment). Similarly, FAST-LIO2, which relies on point-to-plane registration, also exhibits substantial localization errors in this sequence. This indicates limitations in the environmental adaptability of point-to-plane registration method.

Both RIO and our method utilize RCS for point registration. However, RIO establishes correspondences by simply finding nearest points with similar RCS values, whereas our approach treats the neighboring points' spatial and RCS distribution as an integrated whole. Therefore, RIO performs worse on SNAIL, HeRCULES and self-constructed datasets. It is noteworthy that the self-constructed dataset contains sequences

[1] https://conti-engineering.com/components/ars-548-rdi/
[2] https://www.murata.com.cn/products/sensor/gyro/overview/lineup/scha63t

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

TABLE IV
COMPARISON OF RADAR/LIDAR ODOMETRY METHODS ON SNAIL DATASET: RMSE OF ABSOLUTE TRAJECTORY ERROR (M)

Method	Registration Method	0113_3 (4.6km)	0113_1 (0.5km)	0115_2 (6.6km)	0123_2 (8.5km)	0123_3 (2.2km)
VGC-RIO	Point-to-Point	16.53	0.64	16.74	46.38	**1.85**
RIO		62.87	2.76	53.18	85.70	16.01
Radar-ICP		18.2	3.9	31.6	37.5	7.9
RaI-SLAM-LI		-	-	39.90	-	-
Doppler-Odom	Point-to-Plane	**3.38**	0.29	**9.39**	**10.59**	2.61
FAST-LIO2		25.61	**0.16**	52.66	62.43	5.80

[1] **Bold**: best results of all, underline: best results of Point-to-Point based Odometer.
[2] Abbreviations: Sequence names omit year 2024 (e.g., **"0113_3"** stands for **"20240113_3"**). Doppler-Odom for Doppler-Odometry.

TABLE V
COMPARISON OF ODOMETRY/SLAM METHODS ON HERCULES DATASET: RMSE OF ABSOLUTE TRAJECTORY ERROR (M)

Type	Method	Mou 1 (4km)	Lib 1 (1.6km)	Spo 1 (1.4km)	Par 3 (0.5km)	Str 1 (1km)
Odometry	FAST-LIO2	23.81	5.23	4.11	1.83	**2.97**
	VGC-RIO	14.46	4.02	3.93	2.59	5.31
	RIO	14.98	11.24	5.25	4.52	15.36
	Radar-ICP	118.6	10.2	7.1	3.4	11.7
	RaI-SLAM-LI	-	16.99	-	-	-
SLAM	Doppler-SLAM	43.05	13.24	**2.72**	**0.72**	7.45
	RaI-SLAM	**11.2**	**2.1**	3.9	2.2	11.5

[1] **Bold**: best results of all, underline: best results of radar based method, red: best results of radar odometry.
[2] Abbreviations: **Mou 1** for **Mountain Day 1**, **Lib 1** for **Library Day 1**, **Spo 1** for **Sports Complex Day 1**, **Par 3** for **Parking Lot Night 3**, **Str 1** for **Street Day 1**.

with varying levels of motion jitter. As illustrated in Fig. 6 and Tab. VI, under these conditions, RIO shows larger localization errors and even fails in the "court3" sequence, since its registration method relies heavily on the accuracy of inertial prediction. By contrast, VGC-RIO can adapt to different motion conditions by benefiting from the LGC histogram. As the radar points get sparser in the dataset used in RIO, the local constraints get weaker, and RIO and VGC-RIO have similar performance, as shown in Tab. VII.

Radar-ICP is a radar-only odometry that lacks motion constraints; consequently, it is prone to erroneous point-cloud registration, as seen in the Mountain Day 1 sequence. RaI-SLAM extends Radar-ICP and fuses radar with IMU, yet assumes zero vertical and lateral motion during scan matching [40]; this assumption causes a larger drift in its odometry component RaI-SLAM-LI (RaI-SLAM local graph with imu), evident in the "20240115/2" and "Library Day 1" sequences. Our method shows superior accuracy compared to other point-to-point odometry in the used datasets, as it uses not only the geometric distribution but also RCS of the neighboring points in scan matching. It also demonstrates consistent performance under different environments, since it employs a point-to-point registration strategy and does not need to rely on local plane constraints as in Doppler-SLAM.

Jitter severely destabilizes VINS's frontend tracking, leading to complete failure in both "court2" and "court3" sequences. Furthermore, intense dynamic objects in "overpass" sequence critically mislead its data association. In contrast, the dynamic point removal modules of VGC-RIO and RIO render them unaffected.

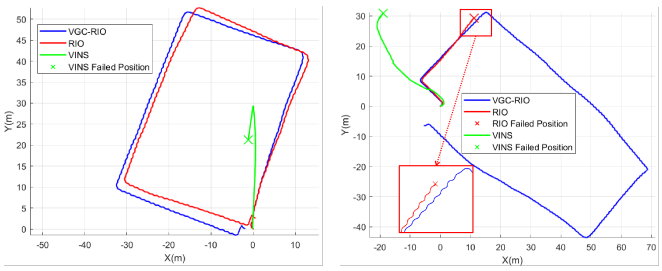
Overall, VGC-RIO achieves comparable accuracy to FAST-LIO2. However, on the four longer sequences of the SNAIL dataset, VGC-RIO outperforms FAST-LIO2, as shown in Tab. IV and Tab. V. This improvement stems from the fact that LiDAR-based odometry relies solely on geometric feature matching, whereas VGC-RIO integrates geometric features, RCS features, and weighted Doppler velocity constraints, enabling more robust and accurate localization.

TABLE VI
QUANTITATIVE ANALYSIS: RMSE OF ABSOLUTE TRAJECTORY ERRORS (M) ON SELF-CONSTRUCTED DATASETS

Method	court1	court2	court3	garden	loop1	loop2	overpass
VGC-RIO	**2.49**	**3.65**	**8.21**	**1.79**	**15.62**	**44.01**	**99.59**
RIO	3.36	12.33	Failed	6.58	53.36	110.33	124.23
VINS	6.54	Failed	Failed	2.04	26.83	95.75	Failed

TABLE VII
COMPARISON ON RIO DATASET: RMSE OF ABSOLUTE TRAJECTORY ERROR (M)

Method	Sequence 1	Sequence 2	Sequence 3
VGC-RIO	**0.198**	0.245	**1.18**
RIO	0.202	**0.235**	1.31



(a) **court2** (b) **court3**

Fig. 6. Comparative Of VGC-RIO, RIO and VINS on Self-Constructed Dataset

V. CONCLUSIONS

In this letter, we propose a accurate and robust tightly-coupled 4D radar-inertial odometry system. We introduce a histogram descriptor that combines RCS features and local geometric features. After removing dynamic points, accurate point-to-point correspondences can be obtained through histogram matching. We also propose a spatial-distribution-based weighted Doppler residual, which enables a more robust velocity estimation. The proposed VGC-RIO has been validated through ablation studies and comparative analyses on both publicly datasets and our self-constructed datasets. Compared with current methods, it demonstrates greater robustness and environmental adaptability. In the future, we will explore the application of hybrid features in point cloud registration as well as the application of RCS features in loop closure detection.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

REFERENCES

[1] J. Zhang, R. Xiao, Li *et al.*, "4drt-slam: Robust slam in smoke environments using 4d radar and thermal camera based on dense deep learnt features," in *2023 IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, 2023, pp. 19–24.

[2] E. Bayramov and Z. Istenes, "Weather-informed vision enhancement for autonomous vehicles in adverse conditions," *Pollack Periodica*, May 2025.

[3] J. Dickmann, Klappstein, Hahn *et al.*, "Automotive radar the key technology for autonomous driving: From detection and ranging to environmental understanding," in *2016 IEEE Radar Conference (RadarConf)*. IEEE, May 2016, p. 1–6.

[4] C. Doer and G. F. Trommer, "An ekf based approach to radar inertial odometry," in *2020 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, 2020, pp. 152–159.

[5] J. Zhang, H. Zhuge, Z. Wu *et al.*, "4dradarslam: A 4d imaging radar slam system for large-scale environments based on pose graph optimization," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 8333–8340.

[6] Y. Cheng, J. Su, M. Jiang *et al.*, "A novel radar point cloud generation method for robot environment perception," *IEEE Transactions on Robotics*, vol. 38, no. 6, pp. 3754–3773, 2022.

[7] Q. Huang, Y. Liang, Z. Qiao *et al.*, "Less is more: Physical-enhanced radar-inertial odometry," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 15 966–15 972. [Online]. Available: https://github.com/HKUST-Aerial-Robotics/RIO

[8] D. Wang, H. Haag, D. C. Herraez *et al.*, "Doppler-slam: Doppler-aided radar-inertial and lidar-inertial simultaneous localization and mapping," *IEEE Robotics and Automation Letters*, vol. 10, no. 9, pp. 9438–9445, 2025.

[9] D. C. Herraez, M. Zeller, D. Wang *et al.*, "Rai-slam: Radar-inertial slam for autonomous vehicles," *IEEE Robotics and Automation Letters*, vol. 10, no. 6, pp. 5257–5264, 2025.

[10] I. Vizzo, T. Guadagnino, B. Mersch *et al.*, "Kiss-icp: In defense of point-to-point icp – simple, accurate, and robust registration if done the right way," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 1029–1036, 2023.

[11] D. Lee, H. Lim, and S. Han, "Genz-icp: Generalizable and degeneracy-robust lidar odometry using an adaptive weighting," *IEEE Robotics and Automation Letters*, vol. 10, no. 1, pp. 152–159, 2025.

[12] 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)*, 2020, pp. 9484–9490.

[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)*, 2019, pp. 298–304.

[14] D. Adolfsson, M. Magnusson, Alhashimi *et al.*, "Cfear radarodometry - conservative filtering for efficient and accurate radar odometry," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 5462–5469.

[15] D. Adolfsson, M. Magnusson *et al.*, "Lidar-level localization with radar? the cfear approach to accurate, fast, and robust large-scale radar odometry in diverse environments," *IEEE Transactions on Robotics*, vol. 39, no. 2, pp. 1476–1495, 2023.

[16] Y. S. Park, Y.-S. Shin, and A. Kim, "Pharao: Direct radar odometry using phase correlation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 2617–2623.

[17] P.-C. Kung, C.-C. Wang, and W.-C. Lin, "A normal distribution transform-based radar odometry designed for scanning and automotive radars," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2021.

[18] K. Haggag, S. Lange, T. Pfeifer *et al.*, "A credible and robust approach to ego-motion estimation using an automotive radar," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6020–6027, 2022.

[19] Y. Zhuang, B. Wang, J. Huai *et al.*, "4d iriom: 4d imaging radar inertial odometry and mapping," *IEEE Robotics and Automation Letters*, vol. 8, no. 6, pp. 3246–3253, 2023.

[20] D. Wang, S. May, and A. Nuechter, "Riv-slam: Radar-inertial-velocity optimization based graph slam," in *2024 IEEE 20th International Conference on Automation Science and Engineering (CASE)*, 2024, pp. 774–781.

[21] W. Yang, H. Jang, and A. Kim, "Ground-optimized 4d radar-inertial odometry via continuous velocity integration using gaussian process," *arXiv preprint arXiv:2502.08093*, 2025.

[22] J. Michalczyk, R. Jung, and S. Weiss, "Tightly-coupled ekf-based radar-inertial odometry," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 12 336–12 343.

[23] J. Michalczyk, R. Jung, Brommer *et al.*, "Multi-state tightly-coupled ekf-based radar-inertial odometry with persistent landmarks," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 4011–4017.

[24] Y. Almalioglu, M. Turan, C. X. Lu *et al.*, "Milli-rio: Ego-motion estimation with low-cost millimetre-wave radar," *IEEE Sensors Journal*, vol. 21, no. 3, pp. 3314–3323, 2021.

[25] S. Kim, J. Seok, J. Lee *et al.*, "Radar4motion: Imu-free 4d radar odometry with robust dynamic filtering and rcs-weighted matching," *IEEE Transactions on Intelligent Vehicles*, pp. 1–11, 2024.

[26] C. Guo, B. Wei, B. Lan *et al.*, "Dgro: Doppler velocity and gyroscope-aided radar odometry," *Sensors*, vol. 24, no. 20, p. 6559, 2024.

[27] P. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.

[28] M. Magnusson, "The three-dimensional normal-distributions transform : an efficient representation for registration, surface analysis, and loop detection," *renewable energy*, 2009.

[29] T. Shan and B. Englot, "Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 4758–4765.

[30] W. Xu and F. Zhang, "Fast-lio: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3317–3324, 2021.

[31] R. B. Rusu, Blodow *et al.*, "Fast point feature histograms (fpfh) for 3d registration," in *2009 IEEE International Conference on Robotics and Automation*, 2009, pp. 3212–3217.

[32] S. Salti, F. Tombari, and L. Di Stefano, "Shot: Unique signatures of histograms for surface and texture description," *Computer Vision and Image Understanding*, vol. 125, pp. 251–264, 2014.

[33] L. Catania and A. Luati, "Robust estimation of a location parameter with the integrated hogg function," *Statistics & Probability Letters*, vol. 164, p. 108812, 2020.

[34] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[35] X. Li, H. Zhang, and W. Chen, "4d radar-based pose graph slam with ego-velocity pre-integration factor," *IEEE Robotics and Automation Letters*, vol. 8, no. 8, pp. 5124–5131, 2023.

[36] S. Agarwal, K. Mierle, and T. C. S. Team, "Ceres Solver," 10 2023. [Online]. Available: https://github.com/ceres-solver/ceres-solver

[37] J. Huai, B. Wang, Y. Zhuang *et al.*, "Snail-radar: A large-scale diverse dataset for the evaluation of 4d-radar-based slam systems," *arXiv preprint arXiv:2407.11705*, 2024.

[38] H. Kim, M. Jung, C. Noh *et al.*, "Hercules: Heterogeneous radar dataset in complex urban environment for multi-session radar slam," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 2025.

[39] W. Xu, Y. Cai, D. He *et al.*, "Fast-lio2: Fast direct lidar-inertial odometry," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2053–2073, 2022.

[40] D. Kellner, M. Barjenbruch, J. Klappstein *et al.*, "Instantaneous ego-motion estimation using doppler radar," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, 2013, pp. 869–874.