



Article

Robust Radar Inertial Odometry in Dynamic 3D Environments

Yang Lyu ^{*}, Lin Hua, Jiaming Wu, Xinkai Liang and Chunhui Zhao 

School of Automation, Northwestern Polytechnical University, Xi'an 710129, China; hlnwpu@163.com (L.H.); wujiaming@mail.nwpu.edu.cn (J.W.); 23b936092@stu.hit.edu.cn (X.L.); zhaochunhui@nwpu.edu.cn (C.Z.)

* Correspondence: lyu.yang@nwpu.edu.cn

Abstract: Millimeter-Wave Radar is one promising sensor to achieve robust perception against challenging observing conditions. In this paper, we propose a Radar Inertial Odometry (RIO) pipeline utilizing a long-range 4D millimeter-wave radar for autonomous vehicle navigation. Initially, we develop a perception frontend based on radar point cloud filtering and registration to estimate the relative transformations between frames reliably. Then an optimization-based backbone is formulated, which fuses IMU data, relative poses, and point cloud velocities from radar Doppler measurements. The proposed method is extensively tested in challenging on-road environments and in-the-air environments. The results indicate that the proposed RIO can provide a reliable localization function for mobile platforms, such as automotive vehicles and Unmanned Aerial Vehicles (UAVs), in various operation conditions.

Keywords: radar inertial odometry; localization; point cloud; UAV



Citation: Lyu, Y.; Hua, L.; Wu, J.; Liang, X.; Zhao, C. Robust Radar Inertial Odometry in Dynamic 3D Environments. *Drones* **2024**, *8*, 197. <https://doi.org/10.3390/drones8050197>

Academic Editor: Oleg Yakimenko

Received: 16 March 2024

Revised: 3 May 2024

Accepted: 7 May 2024

Published: 13 May 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Reliable localization is a prerequisite for most autonomous operations of mobile robot platforms [1]. Localization methods based on various sensory mechanisms have been proposed for different robot operation conditions, especially for GPS-denied environments. Cameras [2,3] and LiDARs [4,5] are the most commonly used exteroceptive sensors for localization functions in GPS-denied environments. The localization reliability and accuracy can be further improved by fusing exteroceptive measurements with interoceptive measurements, such as the Inertial Measurement Units (IMUs). Visual-Inertial-Based Navigation Systems (VINSs) [6], and LiDAR-Inertial-Based Navigation Systems [7] have been extensively investigated and have become essential elements for many robotics applications, such as indoor service robots [8], and outdoor civil and industrial robots [9–11]. When the application field of autonomous robots is further expanded to more complex tasks and environments, cameras and LiDARs become less competitive, as their perception qualities are easily degraded by light condition changes, particles, or weather conditions such as fog, rain, or snow [12].

Millimeter-wave radar is considered a more robust mobile sensor against various perception interference. However, the traditional radar sensor often suffers from low spatial resolution and highly noisy measurement, which makes it hard to utilize to support decent environment perception—and afterward reliable localization—in complex large-scale environments. To improve the perception density, the Frequency-Modulated Continuous Wave (FMCW) technique has been implemented on millimeter-wave radar recently, which enables a more elaborate localization framework such as Simultaneous Localization and Mapping (SLAM) [12]. To further expand the application fields from 2D space to 3D space, millimeter-wave radar that can simultaneously measure the azimuth angle, elevation angle, range, and Doppler speed, named 4D millimeter-wave radar, is further developed, which aids more mobile robots, such as UAVs, to achieve perception tasks in more complex environments.

Radar-based localization has begun to attract researchers' attention in recent years. The MulRan dataset [13] introduces a multimodal range dataset that includes both radar and LiDAR data for environment perception, especially for place recognition. The radar data are defined in an image format for later association convenience. Similar 2D formatting has also been used in [14] to represent 2D point clouds, and the ORB descriptor [2] was used to associate point clouds in consecutive frames. The effects of motion distortion and the Doppler effect on localization have been investigated. To tackle the effect of outliers on radar odometry, a novel outlier-robust method called ORORA was proposed in [15], which is an abbreviation of Outlier-Robust Radar odometry. The work [12] described a complete radar SLAM that demonstrates the power of using 3D radar to achieve reliable localization in challenging perception conditions. Although radar-based localization has been successfully demonstrated in the above works, most of them are based on 3D measurement, which contains an azimuth angle, a range, and a Doppler velocity. The methods can be hardly extended to 3D environments. Until very recently, a 4D millimeter-wave radar with an extra elevation angle has been adopted for localization and navigation purposes. Ref. [16] provided a 4D radar dataset for 3D odometry and mapping. However, the range of the radar was short and could not be used for large-scale scenarios localization and navigation purposes. Ref. [17] presented a real-time imaging radar inertial odometry and mapping method, iRIOM, based on the submap concept. iRIOM was the first to achieve real-time 3D odometry and consistent mapping using 4D radar data. Considering that its experimental data were collected on an unmanned vehicle, its performance on a 3D UAV platform still needs to be verified.

In this paper, we propose a Radar Inertial Odometry (RIO)-based on a long-range 4D radar system and an IMU, which facilitates autonomous mobile systems, such as unmanned vehicles and unmanned aerial vehicles, to localize themselves reliably in complex environments. An illustrative example of the proposed method in a large outdoor scene is shown in Figure 1. The framework is described in Figure 2. First, we design a point cloud preprocessing pipeline, which contains a spatial filter and a velocity filter to eliminate noisy points and outliers from moving objects. Afterward, the point clouds are used to estimate ego-motion velocities based on its Doppler measurement and to obtain an odometry measurement based on the Normal Distribution Transformation (NDT). The IMU data are preintegrated to provide prior information for the velocity filter and the NDT-based point cloud matching. Finally, the three measurements, namely the velocity, the odometry, and the IMU preintegration measurement, are factorized into a graph, and the poses of the robot are solved. The contributions of the paper are as follows:

- We developed a 4D long-range radar-based navigation frontend, which includes odometry and a velocity estimator to estimate the radar ego-motion.
- We built a backend estimator, which factorizes the radar odometry, Doppler velocities, and IMU measurement in a factor graph and solves the robot's poses in a sliding window fashion.
- We extensively tested our framework in complex large-scale 2D/3D environments, including dynamic objects.

The rest of the paper is organized as follows. The related works are reviewed in Section 1. The radar odometry and radar inertial navigation system are given in Sections 2 and 3, respectively. The experimental validations are provided in Section 4. We conclude the paper in Section 5.

In this section, we present the preprocessing steps of the 4D radar point cloud to obtain odometry measurement and ego-motion velocity. Initially, we give a simple but effective process to eliminate outliers; then, the odometry and Doppler velocity are obtained based on filtered point clouds.

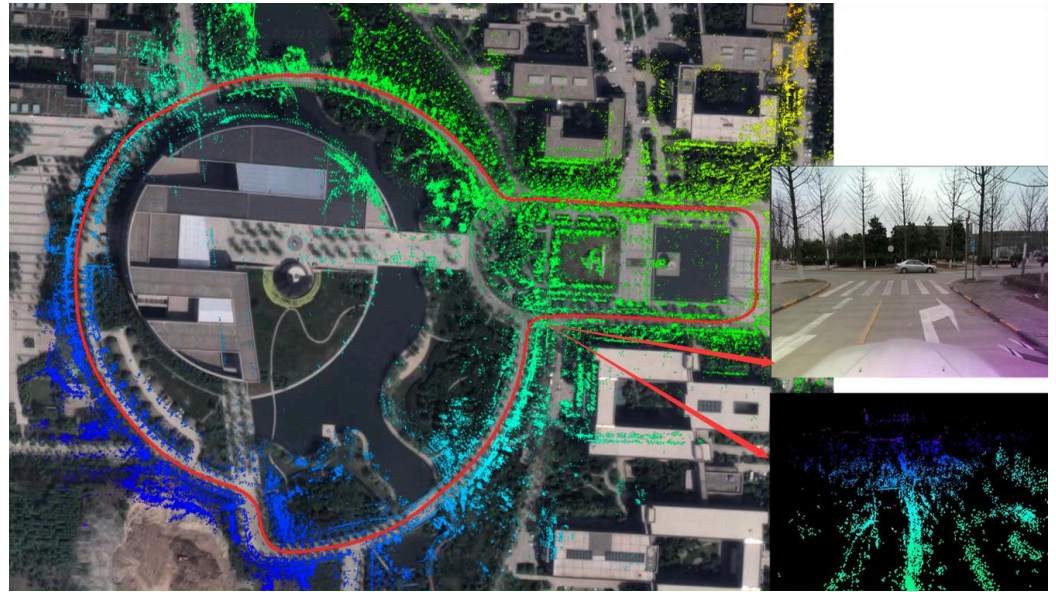


Figure 1. The 3D mapping results of radar SLAM in large outdoor scenes.

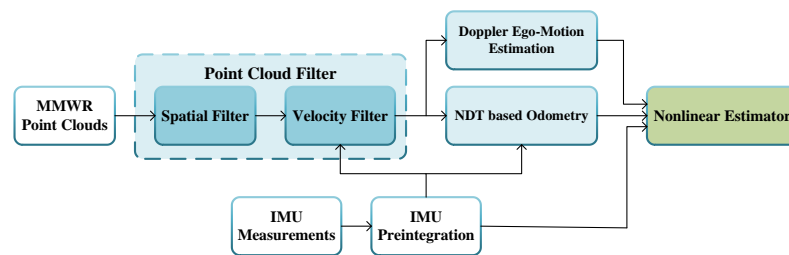


Figure 2. Framework of the proposed RIO system.

2. Radar Odometry

In this section, we present the preprocessing steps of the 4D radar point cloud to obtain the odometry measurement and ego-motion velocity. Initially, we give a simple but effective process to eliminate outliers; then, the odometry and Doppler velocity are obtained based on filtered point clouds.

2.1. Point Cloud Filter

Affected by the sensor's own characteristics and environmental interference, the measured point cloud data contain many outliers. The outliers make the pose estimation inaccurate and destroy the stability of the SLAM system. There are mainly two types of outliers in the 4D millimeter-wave radar data. The first type is clutter points from ground effects and multipath effects. The second type comes from dynamic object perception. We use two filters, namely the spatial filter and the velocity filter, to remove the two types of outliers.

2.1.1. Spatial Filter

Ground effects and multipath effects are the main outlier resources in 4D millimeter radar perception data in a complex environment. Different from most normal radar points that belong to certain patterns, which reflect the structure of surrounding environments, the outliers are usually rather sparse and lack neighbor points. Based on this consideration, we implement a spatial filtering process that contains a pass-through filter and a radius outlier removal filter.

First, the pass-through filter is constructed by defining a pass-through space S originating from the center of radar as $S = \{(\theta_a, \theta_e, d, v) | \theta_a \in [\theta_{a,\min}, \theta_{a,\max}], \theta_e \in [\theta_{e,\min}, \theta_{e,\max}], r \in [r_{\min}, r_{\max}], v \in [v_{\min}, v_{\max}]\} \in \mathbb{R}^4$. A point measurement is an inlier when it belongs to S .

After that, all linear points are converted to 3D Cartesian coordinates for neighbored analysis. Given a point $\mathbf{p}_i \in S$, its d neighborhood points set is defined as $N_d(p_i) = \{\mathbf{p}_j | \|\mathbf{p}_i - \mathbf{p}_j\| \leq d, \mathbf{p}_i, \mathbf{p}_j \in S\}$. An outlier can be defined when

$$i = \begin{cases} \text{an outlier} : & |N_d|(\mathbf{o}_i) \leq n_{th} \\ \text{an inlier} : & \text{otherwise.} \end{cases} \quad (1)$$

The effect of using the above operations on radar point clouds is shown in Figure 3.

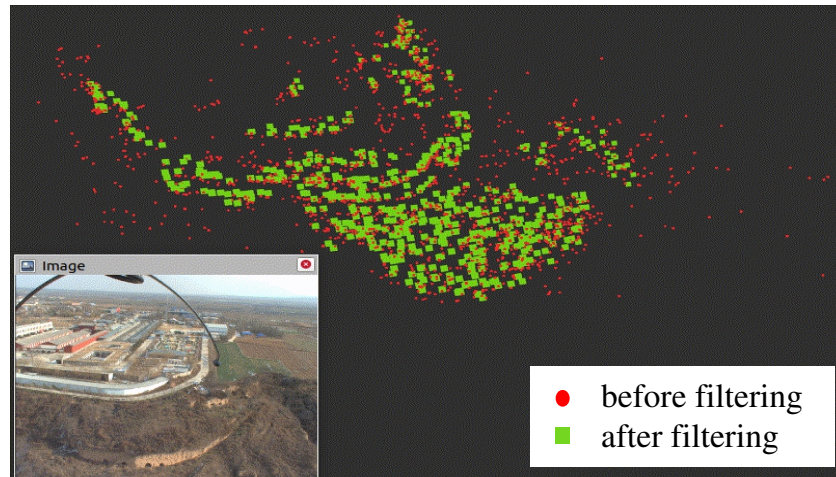


Figure 3. The spatial filter effects on 4D radar measurement.

2.1.2. Point Cloud Filtering from Dynamic Targets

It is well known that moving objects may generate false motion estimation in the SLAM-based method. Removing the point cloud that belongs to the moving object is critical for localization in dynamic environments. Thanks to the Doppler effect, the points that belong to moving objects have obvious velocity differences. For a point i belonging to a static object, the following Equation (2) for the Doppler velocity and the radar ego-motion velocity is defined as follows:

$$\frac{\mathbf{p}_i^\top}{\|\mathbf{p}_i\|} \mathbf{v}_{radar} + v_i = 0. \quad (2)$$

For a dynamic point, the above equation does not hold. Therefore, we use (2) to check points within a scan. As the true value \mathbf{v}_{radar} is not known ahead, we turn to an IMU to acquire a guess of the ego-motion of the radar. We assume there is an IMU fixed to the body frame, and there is a rigid transformation between the radar and the IMU, \mathbf{T}_{imu}^{radar} , as shown in Figure 4. First, an ego-motion velocity of the IMU temporally that is aligned with the radar, which we denoted as \mathbf{v}_{imu}^i , is obtained by preintegrating the IMU measurement data. Subsequently, the prior guess of the radar ego-motion in the radar local frame based on the IMU data, denoted as \mathbf{v}_{imu}^r , is obtained by applying a constant transformation between the radar frame and the IMU frame, which is denoted as on \mathbf{v}_{imu}^i . We replace \mathbf{v}_{radar} with \mathbf{v}_{imu}^i in (2) to check whether a point is static or not. Moreover, considering the uncertainties of the IMU preintegration and Doppler measurement, we can set a threshold v_{thr} to determine whether a point is an outlier as (3).

$$i = \begin{cases} \text{an outlier} : & \left| \frac{\mathbf{p}_i^\top}{\|\mathbf{p}_i\|} \mathbf{v}_{imu}^r + v_i \right| \geq v_{thr} \\ \text{an inlier} : & \text{otherwise.} \end{cases} \quad (3)$$

A demonstrative result of the radar odometry by removing velocity outliers is provided in Figure 5. As shown in Figure 5b, the moving objects (pedestrians) will cause unwanted trajectories in the map and will affect the localization as well. By filtering out the velocity outliers, more clear and more accurate localization can be obtained, as shown in Figure 5c.

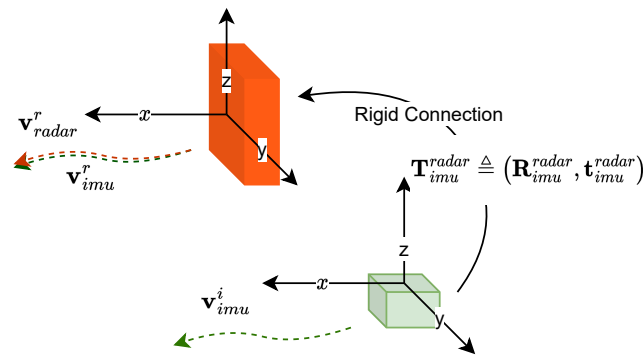


Figure 4. The transformation relationship between \mathbf{v}_{radar}^r and \mathbf{v}_{imu}^i .

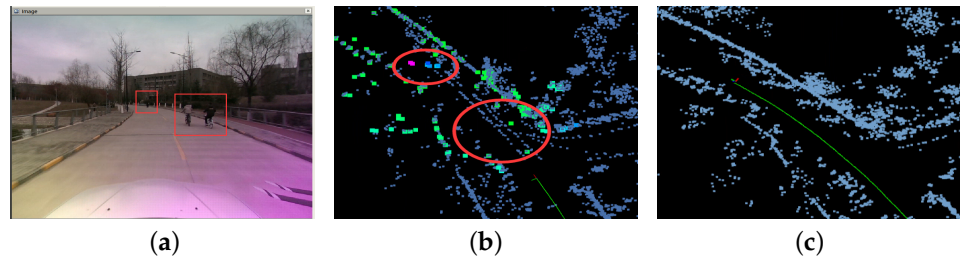


Figure 5. The filtering of moving objects. (a) Visuals captured pedestrians as moving objects. (b) Side-effect by wrongfully taking moving points. (c) Localization and mapping after velocity filtering.

2.2. NDT-Based Odometry

Point cloud registration is the main process for localization tasks. By registering between frames, the pose of the robot can be obtained incrementally. In this paper, we implement the Normal Distribution Transformation (NDT) method, which is known for its efficiency and tolerance to outliers [18]. The basic idea behind NDT is to represent a point cloud as a probability distribution and then to use statistical methods to match the distributions of two point clouds. Our registration goal is to calculate the pose transformation between millimeter-wave radar frames collected at consecutive time instances. Specifically, each point in the point cloud is assigned a Gaussian distribution with a mean and covariance matrix. The mean represents the location of the point, and the covariance matrix describes the uncertainty of the point's location. The NDT algorithm then finds the transformation that minimizes the difference between the two probability distributions.

Given two point clouds P_1 and P_2 , each consisting of n points, the NDT algorithm seeks to find the transformation \mathbf{T} that best aligns P_2 with P_1 . The objective function to be minimized can be written as

$$E(\mathbf{T}) = \sum_{i=1}^n \sum_{j=1}^n w_{ij} \|\mathbf{t} + \mathbf{R}\mathbf{p}_j - \mathbf{p}_i\|^2 \quad (4)$$

where \mathbf{t} and \mathbf{R} are the translation and rotation components of the transformation \mathbf{T} , \mathbf{p}_i and \mathbf{p}_j are the i th and j th points in P_1 and P_2 , respectively, and w_{ij} is a weight function that determines the influence of the j th point in P_2 on the i th point in P_1 . The weight function is typically chosen to be a function of the distance between the mean of the Gaussian distributions assigned to the two points.

The optimization problem can be solved using gradient descent, where the gradient is given by

$$\nabla E(\mathbf{T}) = -2 \sum_{i=1}^n \sum_{j=1}^n w_{ij} \begin{bmatrix} \mathbf{p}_i - (\mathbf{t} + \mathbf{R}\mathbf{p}_j) \\ \mathbf{p}_i - (\mathbf{t} + \mathbf{R}\mathbf{p}_j) \end{bmatrix} \begin{bmatrix} \mathbf{p}_j^T \mathbf{R}^T & 1 \\ \mathbf{0} & 1 \end{bmatrix} \quad (5)$$

where $\mathbf{1}$ and $\mathbf{0}$ are the 3×3 identity and zero matrices, respectively. The NDT algorithm can be used for SLAM by iteratively applying the algorithm to pairs of consecutive point clouds generated by the radar sensor. The transformation between each pair of point clouds can then be used to update the robot's pose estimate and construct a map of the environment. The general flow of the algorithm is outlined in Algorithm 1 below.

Algorithm 1: NDT-based SLAM algorithm

Input: Point clouds from radar scanner
Output: Map of the environment and robot's trajectory
Initialize robot pose estimate \mathbf{T} ;
Initialize empty map;
for each consecutive pair of point clouds **do**
 Apply NDT algorithm to obtain transformation $\mathbf{T}_{i,i+1}$;
 Update robot pose estimate: $\mathbf{T} \leftarrow \mathbf{T} \cdot \mathbf{T}_{i,i+1}$;
 Transform point cloud to global frame using $\mathbf{T}_{i,i+1}$;
 Add transformed points to map;
end
return Map and robot trajectory;

2.3. Doppler Ego-Motion Estimation

As one advantage over other sensors, such as LiDARs and RGBD cameras, radar provides radial velocity over the Doppler effect. The velocity of a 4D millimeter-wave radar can be obtained based on scanned points in each frame [19]. Multiple points with velocity measurement v_i and position \mathbf{p}_i can be used to estimate the 3D velocity of the radar $\mathbf{v}_{\text{radar}}$ according to Equation (6).

$$\frac{\mathbf{p}_i^\top}{\|\mathbf{p}_i\|} \mathbf{v}_{\text{radar}} = -v_i. \quad (6)$$

Given n points with a radar scan, the above equation can be written as

$$\mathbf{v}_{pc} \triangleq \begin{bmatrix} -v_1 \\ -v_2 \\ \vdots \\ -v_n \end{bmatrix} = \begin{bmatrix} \frac{\mathbf{p}_1^\top}{\|\mathbf{p}_1\|} \\ \frac{\mathbf{p}_2^\top}{\|\mathbf{p}_2\|} \\ \vdots \\ \frac{\mathbf{p}_n^\top}{\|\mathbf{p}_n\|} \end{bmatrix} \mathbf{v}_{\text{radar}} \triangleq \mathbf{H} \mathbf{v}_{\text{radar}} \quad (7)$$

The vector \mathbf{v}_{pc} is the stacked $-v_i$ s for all points from a radar point clouds scan. Then the least squares solution of $\mathbf{v}_{\text{radar}}$ can be obtained as

$$\mathbf{v}_{\text{radar}} = (\mathbf{H}^\top \mathbf{H})^{-1} \mathbf{H}^\top \mathbf{v}_{pc} \quad (8)$$

When the radar points contain outliers, using the results directly obtained from (8) may introduce errors. The abnormal values caused by noise can be further removed by a Random Sample Consensus (RANSAC) method, which was first proposed in [20] and has been demonstrated to be an effective method to remove noise and outliers in radar point clouds [21,22].

A test of the velocity estimation based on the above method against the RTK ground truth is provided in Figure 6. It can be seen that the error of ego-velocity estimation is within a reasonable range with RMSE 0.175 m/s. Because our millimeter-wave radar has a long detection distance of 300 meters, and the previous work has included the removal of outliers, we can also achieve more accurate velocity estimation in much more open areas.

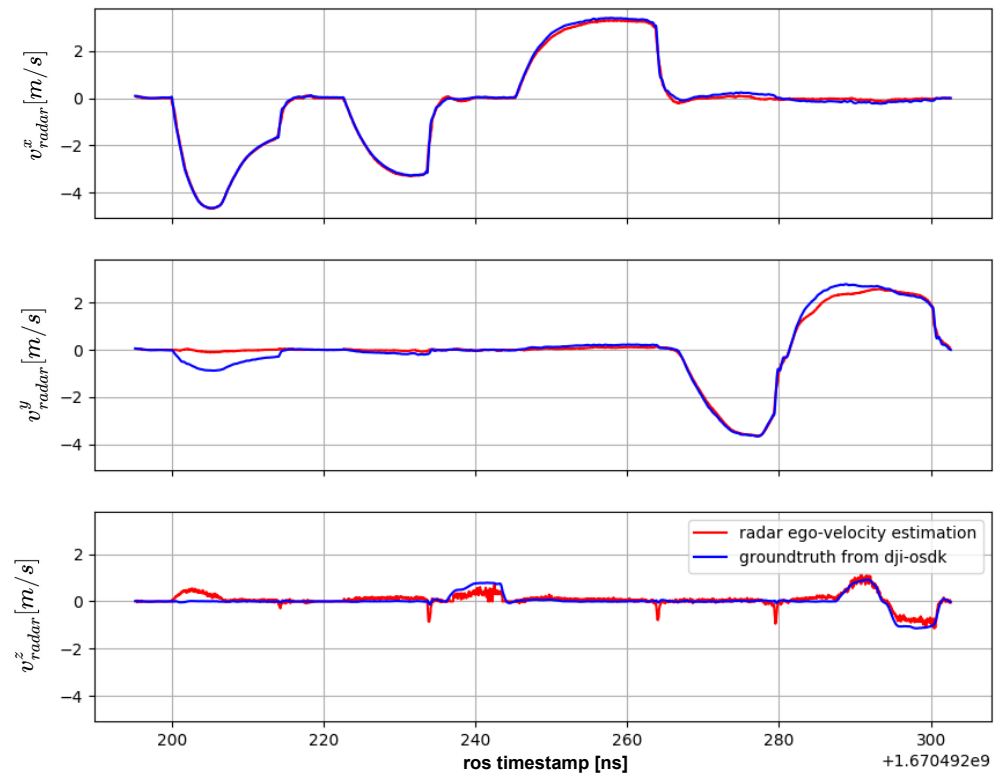


Figure 6. The 3D ego-motion velocity estimation $\mathbf{v}_{radar} \triangleq [v_{radar}^x, v_{radar}^y, v_{radar}^z]^\top$ compared to RTK ground truth from dji-osdk.

3. Nonlinear Estimator

3.1. Estimator Formulation

In this part, we aim at developing a nonlinear optimization-based framework to estimate the pose of a robot. Specially, we consider fusing the IMU measurement, radar odometry measurement, and velocity estimation in a sliding window fashion, as shown in Figure 7. The factor graph of our fusion scheme is shown in Figure 8. The states included in the sliding window at time t are defined as

$$\mathcal{X}_t \triangleq [\{\mathbf{x}_{I,m}^\top\}_{m \in T_t}] \quad (9)$$

where $\{\mathbf{x}_{I,m}\}_{m \in T_t}$ contains the active IMU states within the sliding window at time instance t . T_t denotes the set of IMU measurements at t . The IMU state is

$$\mathbf{x}_I \triangleq [{}^I_w \bar{\mathbf{q}}^\top \quad \mathbf{p}^{w^\top} \quad \mathbf{v}^{w^\top} \quad \mathbf{b}_g^\top \quad \mathbf{b}_a^\top]^\top, \quad (10)$$

where ${}^I_w \bar{\mathbf{q}}$ is a unit quaternion that denotes the rotation from the global frame $\{w\}$ to the IMU frame $\{I\}$. \mathbf{p}^{w^\top} and \mathbf{v}^{w^\top} are the IMU position and velocity, respectively. \mathbf{b}_g , \mathbf{b}_a are the random walk biases for the gyroscope and accelerometer, respectively.

With the state definition (9), the objective is to minimize the cost function of different measurement residuals in (11).

$$\min_{\mathcal{X}_t} \left\{ \sum_{m \in T_t} \|\mathbf{r}_{I,m}\|_{\Sigma_m}^2 + \sum_{i \in O_t} \|\mathbf{r}_i\|_{\Sigma_i}^2 + \sum_{j \in V_t} \|\mathbf{r}_j\|_{\Sigma_j}^2 \right\}. \quad (11)$$

The first term is the cost of IMU-based residual, and $\mathbf{r}_{I,m}$ defines the measurement residual between the active frames m and $m + 1$. The IMU samples consist of all IMU frames between adjacent radar active frames; thus, the preintegration of the IMU samples

is not dependent on a fixed time step but rather determined by the time interval between radar active frames. The second term is the cost of odometry residuals, and the last term is the cost of the velocity residuals. O_t and V_t are the set of measurements within the sliding window at time t . Σ_i and Σ_j are the covariance of the two measurements. The optimization of (11) is usually solved with an iterative least squares solver through a linear approximation. In this paper, we formulate and solve the problem based on the GTSAM package [23].

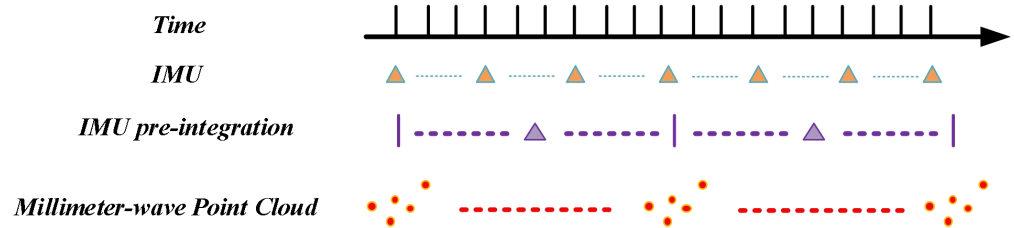


Figure 7. Time alignment of sensor data.

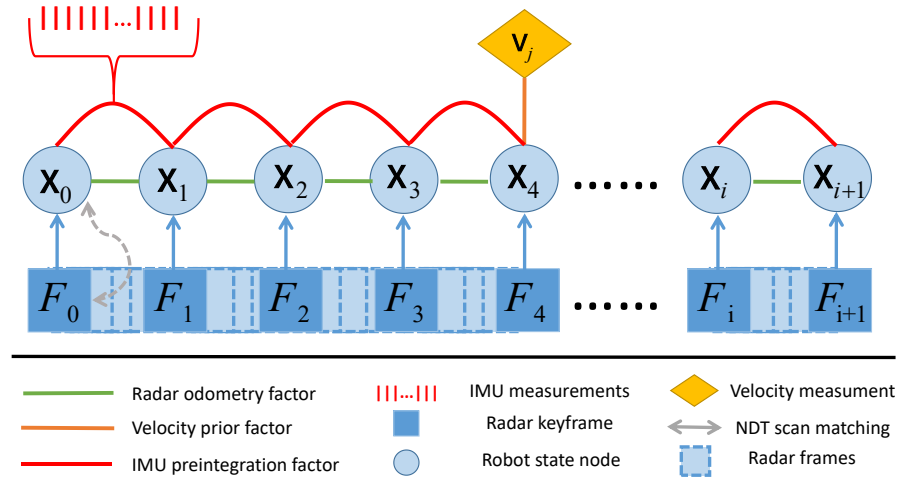


Figure 8. The factor graph of our RIO framework.

3.2. Factorization

In this part, the three information sources, namely the IMU, radar odometry, and radar velocity, are factorized.

3.2.1. IMU Preintegration Factor

The error term $\mathbf{r}_{I,m}$ incorporates the relative motion constraints between radar keyframes. IMU measurements between m and $m + 1$ are preintegrated as in [24], and the residual term at m is defined as

$$\mathbf{r}_{I,m} = \begin{bmatrix} \mathbf{R}_m^{w\top} \left(\mathbf{p}_{m+1}^w - \mathbf{p}_m^w + \frac{1}{2} \mathbf{g}^w \Delta \tau_m^2 - \mathbf{v}_m^w \Delta \tau_m \right) - \hat{\mathbf{a}}_{m+1}^m \\ \mathbf{R}_m^{w\top} \left(\mathbf{v}_{m+1}^w + \mathbf{g}^w \Delta \tau_m - \mathbf{v}_m^w \right) - \hat{\mathbf{b}}_{m+1}^m \\ 2\text{vec} \left[\left(\mathbf{q}_m^w \right)^{-1} \otimes \mathbf{q}_{m+1}^w \otimes \left(\hat{\gamma}_{m+1}^m \right)^{-1} \right] \\ \mathbf{b}_{a,m+1} - \mathbf{b}_{a,m} \\ \mathbf{b}_{g,m+1} - \mathbf{b}_{g,m} \end{bmatrix} \quad (12)$$

where $\hat{\mathbf{z}}_{m+1}^m = \left[\hat{\mathbf{a}}_{m+1}^{m\top}, \hat{\mathbf{b}}_{m+1}^{m\top}, \hat{\gamma}_{m+1}^{m\top} \right]^\top$ is the preintegrated IMU measurements incorporating gyroscope and accelerometer readings from keyframe m to $m + 1$. $\Delta \tau_m$ is the time interval between keyframes. The operator $\text{vec}(\cdot)$ is to extract the vector part of a quaternion. More details on the preintegration can be found in [24].

3.2.2. Odometry Factor

The odometry factor relates the consecutive measurements of millimeter-wave radar data to calculate the robot's motion between two poses. Given the odometry observation between $m, m + 1$ as $\Delta \mathbf{q}_k, \Delta \mathbf{p}_k$, the odometry residual is calculated as

$$\mathbf{r}_i \triangleq \begin{bmatrix} 2 \text{vec}(\Delta \bar{\mathbf{q}}_m^{-1} \circ \Delta \hat{\mathbf{q}}_m) \\ \Delta \hat{\mathbf{p}}_m - \Delta \mathbf{p}_m \end{bmatrix} \quad (13)$$

where \circ denotes the quaternion multiplication operator. The detailed construction of the factor is given in [25] and is omitted here due to space limitation.

3.2.3. Velocity Prior Factor

These factors constrain the estimate of the robot's velocity to improve the robustness of SLAM. Let $\mathbf{v}_{\text{radar}}$ be a prior velocity estimate which comes from the output of the radar ego-velocity estimator described in Section 2.3, and the velocity residual can be simply formulated as

$$\mathbf{r}_j = \hat{\mathbf{v}}_I - \mathbf{v}_{\text{radar}}, \quad (14)$$

where $\hat{\mathbf{v}}_I$ is the estimated velocity.

4. Experiment Results

In this section, we validate our proposed RIO in complex environments. Especially, the experiments are carried out in both 2D environments with a vehicle and 3D environments with a quadrotor UAV.

4.1. Experiment Setups

4.1.1. System Setups

In the experiment, we use a Continental ARS5484D radar as the main perception sensor, which is a long-range millimeter-wave radar that has a 300 m maximum sensing range, and the FOV is a 120° azimuth angle and 30° elevation angle. In addition, we use an Xsense IMU to measure the local motion of the platform. For comparison purposes, we use a D455 camera to carry out vision-based localization algorithms. The RTK positioning information from the DJI M300 UAV navigation system is used as the ground truth. A DJI Manifold-2C is used to record the sensor data and ground truth. The detailed setup is illustrated in Figure 9.



Figure 9. Experimental equipment.

4.1.2. Experimental Environments

As we intended to implement the proposed method in a large-scale environment, we carried out experimental tests in two environment categories, which were (1) an on-road experiment by carrying the UAV on a vehicle, and (2) an in-the-air experiment by flying the UAV in a low-altitude environment with dense objects. The experiment environments are shown in Figure 10.

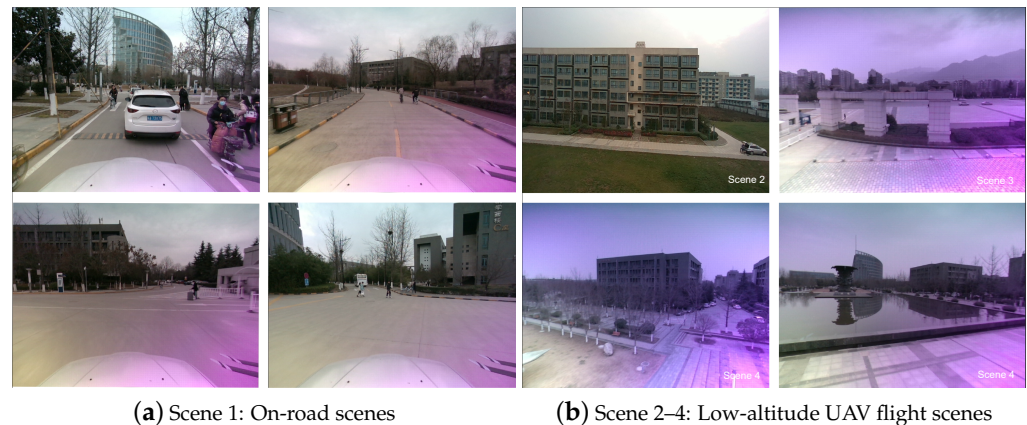


Figure 10. Experimental environment.

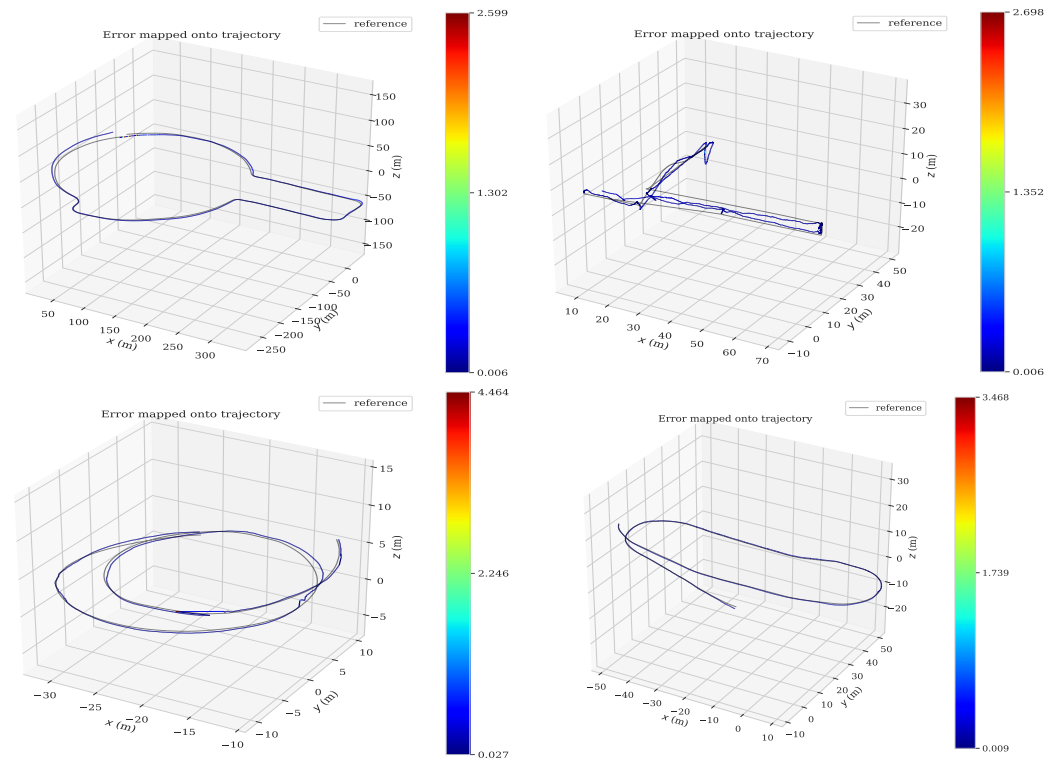
4.1.3. Metrics

The effectiveness of our method is based on two metrics, the Relative Pose Error (RPE) and the Absolute Pose Error (APE), which are carried out based on the EVO toolbox [26]. The RPE metric is used to evaluate the relative error between the estimated and ground truth poses over time. It is calculated by computing the difference between the estimated pose and the ground truth pose at each time step and then computing the Root Mean Squared Error (RMSE) of these differences over the entire trajectory. On the other hand, the APE metric is used to evaluate the absolute error between the estimated and ground truth poses at a specific time step. It is calculated by computing the Euclidean distance between the estimated position and orientation and the ground truth position and orientation at a given time step.

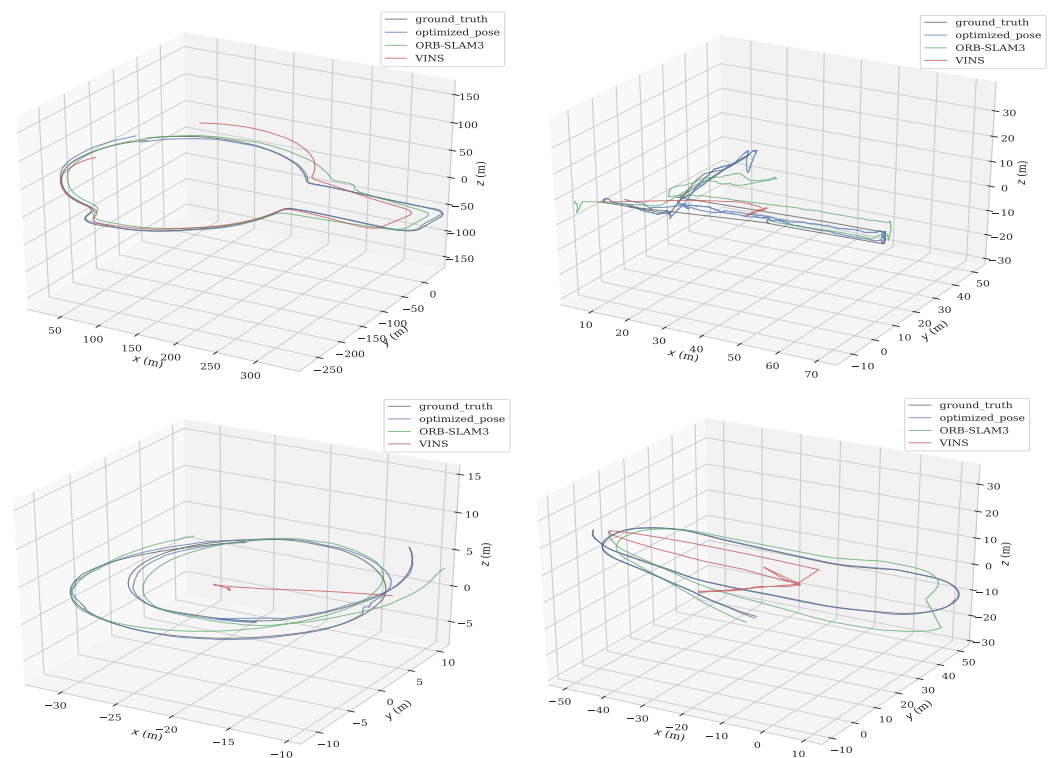
4.2. Experiment Analysis

We evaluated our proposed RIO with four scenes in the above two types of environments. Scene 1 is the on-road test in a large campus environment, and Scenes 2–4 are in-the-air tests. In Figure 11, we stipulate that each subfigure's top left, top right, bottom left, and bottom right correspond to Scene 1, 2, 3, and 4, respectively. The RPE and APE results of our methods are provided in Tables 1 and 2, respectively. Additionally, we provide the RPE of our method in Figure 11a. Apparently, the results indicate that the accuracy of our proposed RIO was within the acceptable level of error, with RPE errors below 0.5.

To further demonstrate the effectiveness of our proposed method, we further carried out the localization using two vision-based methods, namely the VINS-Mono and ORB-SLAM3. The results show that the environment and highly aggressive maneuverability of the platform were quite challenging for vision-based navigation systems. The comparative results of our methods and the two vision-based localization methods are plotted in Figure 11. Specifically, Figure 11b show the estimation trajectories based on different methods. Figure 11c shows the trajectories with respect to time. (b) and (c) indicate that the two vision-based methods failed to achieve reliable localization in in-the-air 3D environments due to the aggressive motion of the UAV. In Scene 1, all methods were able to provide estimation results. Among them, our method could achieve the best localization performance.

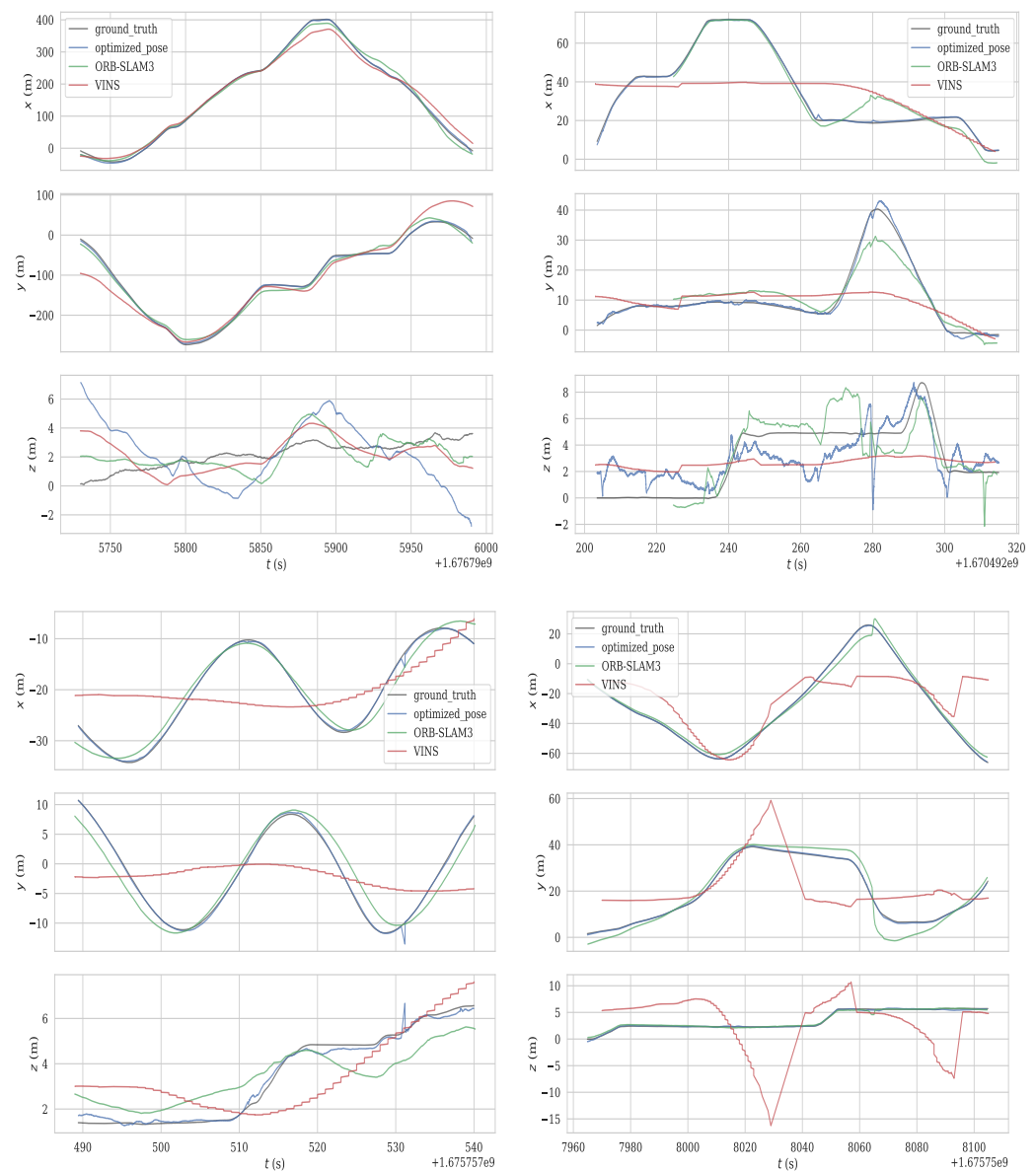


(a) RPE errors



(b) Trajectories vs. ground truth

Figure 11. Cont.



(c) Comparison of positioning results with respect to time

Figure 11. Experiment results of Scene 1–4.**Table 1.** RPE evaluation results [m].

	Max.	Mean	Median	Min.	RMSE	SSE	Std.
Scene 1	2.598	0.134	0.099	0.006	0.220	63.009	0.174
Scene 2	2.697	0.082	0.067	0.006	0.106	62.806	0.066
Scene 3	4.464	0.074	0.068	0.027	0.151	56.348	0.131
Scene 4	3.468	0.078	0.075	0.009	0.091	58.641	0.047

Table 2. APE evaluation results [m].

	Max.	Mean	Median	Min.	RMSE	SSE	Std.
Scene 1	15.863	4.027	2.961	0.255	5.067	33,357.022	3.075
Scene 2	6.417	1.768	1.693	0.244	1.917	20,415.619	0.741
Scene 3	4.902	0.337	0.300	0.016	0.390	375.374	0.197
Scene 4	0.932	0.316	0.326	0.037	0.348	841.778	0.144

5. Conclusions

Millimeter-wave radar provides more robust sensing in harsh conditions. In this paper, we proposed a radar inertial odometry framework using a 4D millimeter-wave radar. First, by analyzing the strengths and weaknesses of the point cloud generated by the 4D millimeter-wave radar, the point cloud was preprocessed to remove noise, thereby allowing for the estimation of the robot's relative transformation through interframe point cloud registration. Then, we proposed a graph-based SLAM framework that fuses the velocity information from the IMU data, millimeter-wave radar odometry, and velocity estimation module. Our proposed method was extensively tested in challenging environments, including changing weather conditions, dynamic targets, and various platforms. The results show that the proposed SLAM framework can provide reliable localization for mobile platforms such as vehicles and drones under various operating conditions. In the future, we will investigate the environmental factors that affect localization performance and also the fusion of multimodality fusion-based methods.

Author Contributions: Conceptualization, Y.L. and C.Z.; methodology, L.H., Y.L. and X.L.; software, L.H.; validation, J.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China under Grant 62203358, Grant 62233014, and Grant 62073264.

Data Availability Statement: The source code and recorded experimental data sets are available at <https://github.com/lincoln1587/radar-inertial-odometry> (accessed on 3 May 2024).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Barfoot, T.D. *State Estimation for Robotics*; Cambridge University Press: Cambridge, UK, 2017.
2. Mur-Artal, R.; Montiel, J.M.M.; Tardos, J.D. ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Trans. Robot.* **2015**, *31*, 1147–1163. [\[CrossRef\]](#)
3. Forster, C.; Pizzoli, M.; Scaramuzza, D. SVO: Fast semi-direct monocular visual odometry. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 15–22.
4. Zhang, J.; Singh, S. LOAM: Lidar odometry and mapping in real-time. In Proceedings of the Robotics: Science and Systems, Berkeley, CA, USA, 12–16 July 2014; Volume 2, pp. 1–9.
5. Yi, S.; Lyu, Y.; Hua, L.; Pan, Q.; Zhao, C. Light-LOAM: A Lightweight LiDAR Odometry and Mapping based on Graph-Matching. *IEEE Robot. Autom. Lett.* **2024**, *9*, 3219–3226. [\[CrossRef\]](#)
6. Qin, T.; Li, P.; Shen, S. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Trans. Robot.* **2018**, *34*, 1004–1020. [\[CrossRef\]](#)
7. Shan, T.; Englot, B.; Meyers, D.; Wang, W.; Ratti, C.; Rus, D. Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 October 2020–24 January 2021; IEEE: Piscataway, NJ, USA, 2020; pp. 5135–5142.
8. Kasper, A.; Xue, Z.; Dillmann, R. The KIT object models database: An object model database for object recognition, localization and manipulation in service robotics. *Int. J. Robot. Res.* **2012**, *31*, 927–934. [\[CrossRef\]](#)
9. Lyu, Y.; Nguyen, T.M.; Liu, L.; Cao, M.; Yuan, S.; Nguyen, T.H.; Xie, L. SPINS: A structure priors aided inertial navigation system. *J. Field Robot.* **2023**, *40*, 879–900. [\[CrossRef\]](#)
10. Lyu, Y.; Yuan, S.; Xie, L. Structure Priors Aided Visual-Inertial Navigation in Building Inspection Tasks with Auxiliary Line Features. *IEEE Trans. Aerosp. Electron. Syst.* **2022**, *58*, 3037–3048. [\[CrossRef\]](#)
11. Lyu, Y.; Cao, M.; Yuan, S.; Xie, L. Vision-Based Plane Estimation and following for Building Inspection with Autonomous UAV. *IEEE Trans. Syst. Man Cybern. Syst.* **2023**, *53*, 7475–7488. [\[CrossRef\]](#)
12. Hong, Z.; Petillot, Y.; Wallace, A.; Wang, S. RadarSLAM: A robust simultaneous localization and mapping system for all weather conditions. *Int. J. Robot. Res.* **2022**, *41*, 519–542. [\[CrossRef\]](#)
13. Kim, G.; Park, Y.S.; Cho, Y.; Jeong, J.; Kim, A. Mulran: Multimodal range dataset for urban place recognition. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 6246–6253.
14. Burnett, K.; Schoellig, A.P.; Barfoot, T.D. Do We Need to Compensate for Motion Distortion and Doppler Effects in Spinning Radar Navigation? *IEEE Robot. Autom. Lett.* **2021**, *6*, 771–778. [\[CrossRef\]](#)
15. Lim, H.; Han, K.; Shin, G.; Kim, G.; Hong, S.; Myung, H. ORORA: Outlier-Robust Radar Odometry. *arXiv* **2023**, arXiv:2303.01876.

16. Kramer, A.; Harlow, K.; Williams, C.; Heckman, C. ColoRadar: The direct 3D millimeter wave radar dataset. *Int. J. Robot. Res.* **2022**, *41*, 351–360. [\[CrossRef\]](#)
17. Zhuang, Y.; Wang, B.; Huai, J.; Li, M. 4D iRIOM: 4D Imaging Radar Inertial Odometry and Mapping. *IEEE Robot. Autom. Lett.* **2023**, *8*, 3246–3253. [\[CrossRef\]](#)
18. Biber, P.; Strasser, W. The normal distributions transform: A new approach to laser scan matching. In Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453), Las Vegas, NV, USA, 27–31 October 2003; Volume 3, pp. 2743–2748. [\[CrossRef\]](#)
19. Doer, C.; Trommer, G.F. An EKF Based Approach to Radar Inertial Odometry. In Proceedings of the 2020 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), Karlsruhe, Germany, 14–16 September 2020; pp. 152–159. [\[CrossRef\]](#)
20. Fischler, M.A.; Bolles, R.C. Random sample consensus. *Commun. ACM* **1981**, *24*, 381–395. [\[CrossRef\]](#)
21. Kellner, D.; Barjenbruch, M.; Klappstein, J.; Dickmann, J.; Dietmayer, K. Instantaneous ego-motion estimation using doppler radar. In Proceedings of the 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013), The Hague, The Netherlands, 6–9 October 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 869–874.
22. Park, Y.S.; Shin, Y.S.; Kim, J.; Kim, A. 3d ego-motion estimation using low-cost mmwave radars via radar velocity factor for pose-graph slam. *IEEE Robot. Autom. Lett.* **2021**, *6*, 7691–7698. [\[CrossRef\]](#)
23. Dellaert, F. *Factor Graphs and GTSAM: A Hands-On Introduction*; Technical Report; Georgia Institute of Technology: Atlanta, Georgia, 2012.
24. Forster, C.; Carlone, L.; Dellaert, F.; Scaramuzza, D. On-manifold preintegration for real-time visual-inertial odometry. *IEEE Trans. Robot.* **2016**, *33*, 1–21. [\[CrossRef\]](#)
25. Nguyen, T.M.; Cao, M.; Yuan, S.; Lyu, Y.; Nguyen, T.H.; Xie, L. Viral-fusion: A visual-inertial-ranging-lidar sensor fusion approach. *IEEE Trans. Robot.* **2021**, *38*, 958–977. [\[CrossRef\]](#)
26. Grupp, M. Evo: Python Package for the Evaluation of Odometry and SLAM. 2017. Available online: <https://github.com/MichaelGrupp/evo> (accessed on 3 May 2024).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.