Indoor Localization and Mapping with 4D mmWave Imaging Radar

Jui-Te Huang

CMU-RI-TR-24-19

May, 2024

The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee:
Prof. Michael Kaess, *chair*
Prof. Sebastian Scherer
Srinivasan Vijayarangan

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Robotics.

To my family and friends,
your unwavering support has been the guiding force of my academic journey.

Abstract

State estimation is a crucial component for the successful implementation of robotic systems, relying on sensors such as cameras, LiDAR, and IMUs. However, in real-world scenarios, the performance of these sensors is degraded by challenging environments, e.g. adverse weather conditions and low-light scenarios. The emerging 4D imaging radar technology is capable of providing robust perception in adverse conditions. Despite its potential, challenges remain for indoor settings where noisy radar data does not present clear geometric features. Moreover, disparities in radar data resolution and field of view (FOV) can lead to inaccurate measurements. While prior research has explored radar-inertial odometry based on Doppler velocity information, challenges remain for the estimation of 3D motion because of the discrepancy in the FOV and resolution of the radar sensor. In this thesis, we address Doppler velocity measurement uncertainties. We present a method to optimize body frame velocity while managing Doppler velocity uncertainty. Based on our observations, we propose a dual imaging radar configuration to mitigate the challenge of discrepancy in radar data. To attain high-precision 3D state estimation, we introduce a strategy that seamlessly integrates radar data with a consumer-grade IMU sensor using fixed-lag smoothing optimization. Finally, we evaluate our approach using real-world 3D motion data and demonstrate down stream tasks of localization and mapping.

Acknowledgments

I would like to thank my advisor, Professor Michael Kaess, for recognizing my interest and experience in radar technology and providing me with the wonderful opportunity to work on this project. I have gained so much knowledge from Michael as a robotics researcher and practitioner. Over the past two years, Michael has offered me significant guidance and insights into this project while also granting me the freedom to explore different ideas. His support will always be remembered.

I would like to express my gratitude to the sponsors from Amazon Lab126, Dr. Adam Fineberg and Mr. Pratik Patel. Their ongoing support and patience made this project successful. Our discussions every two months have pushed the project forward and taught me a great deal from their industry experience.

I would also like to thank our former lab member, Rouyang, from whom I inherited this project. He provided me with a solid foundation to build upon and generously shared many ideas and experiences. A special thanks to Dan and Mohammad for offering me valuable insights on optimization problems. Thank you to Akash and Montiel, whom I frequently turn to for support on machine learning issues. Thanks to Akshay and Sam for building the prototype of our sensor rig and assisting me in designing the new sensor rig. Additionally, thanks to Taylor and Easton for sharing their real-world engineering experience with me. I am also grateful to Lihong, Andrew, Chris, Tianxiang, Joe, and everyone in the lab for sharing their campus life with me.

Finally, I would like to thank my family. Your unwavering support has been the guiding force in my life.

Contents

When this dissertation is viewed as a PDF, the page header is a link to this Table of Contents.

List of Figures

List of Tables

Chapter 1

Introduction

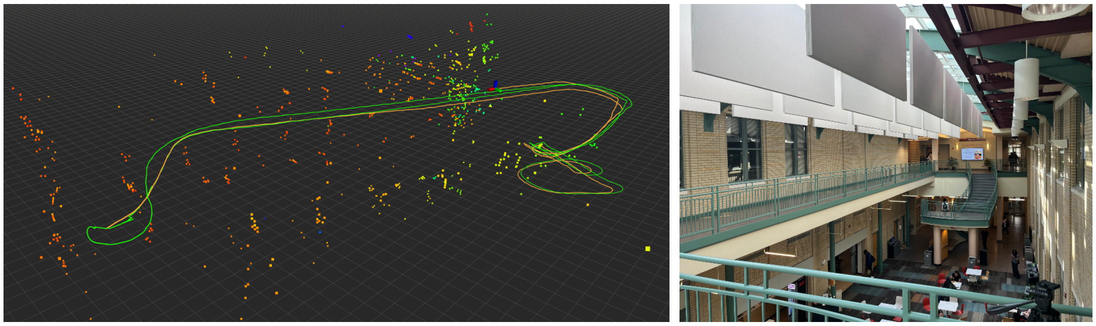


Figure 1.1: A demonstration of multi-radar inertial odometry (orange) walked through different levels of an atrium compared to visual-inertial odometry (green). The colors of the radar point cloud indicate Doppler velocity from high (red) to low (purple).

1.1 Robust State Estimation

State estimation serves as a fundamental component in the majority of robotics applications. Commonly used sensors for state estimation include cameras, LiDAR, and IMUs [42]. However, deploying robots can be challenging in adverse environments. For instance, LiDAR sensors can experience substantial performance degradation when exposed to adverse factors such as smoke and fog, as well as in environments lacking distinct geometric features. Similarly, cameras encounter the same challenges with additional difficulties in environments with low-light conditions or lacking distinct visual features.

1.2 4D mmWave Imaging Radar

The emerging 4D mmWave imaging radar sensor [16] can provide robust perception in demanding environments. The imaging radar sensor utilizes electromagnetic waves with wavelengths at the millimeter level that can function in adverse environmental conditions and do not rely on external lighting. Using frequency-modulated continuous wave technology, the imaging radar can provide Doppler velocity measurements for each detected 3D point. However, radar point clouds are known to be noisy and sparse and can be severely discretized due to post-processing procedures, thereby providing limited geometry information, especially in confined indoor environments.

Despite the challenges presented by the mmWave imaging radar, several recent studies have found success using this technology for object detection [38, 39], navigation [14], and state estimation [8, 21, 24, 32]. Further applications are discussed in the survey paper [36].

1.3 State Estimation using mmWave Radar

There are several challenges in developing a state estimation algorithm using mmWave radar. Previous works demonstrated the capability of building trajectories based on geometry information and scan-to-scan matching in automotive settings [3, 12, 22]. However, these methods are limited to planar localization and often exhibit degraded performance in indoor environments where there are limited key point features and clear geometry patterns in addition to the noise introduced by multi-path reflections of the signals.

Another popular approach involves leveraging Doppler speed information to optimize body frame velocity and then fusing it with an inertial measurement sensor [7, 20, 32]. However, inaccuracies in body frame velocity and its associated uncertainty can lead to trajectory drift along the elevation direction. This issue arises because the antenna of imaging radar is designed to provide significantly lower data resolution and FOV in the elevation dimension compared to the azimuth dimension [35].

1.4 Contribution

To achieve high-precision 3D motion state estimation using a Doppler velocity-based radar-inertial odometry approach, we tackle the above-mentioned challenges. We analyze radar measurement uncertainty and develop an algorithm for seamless radar-IMU fusion. We showcase the effectiveness of this approach by using dual mmWave cascade imaging radars coupled with a consumer-grade IMU sensor to obtain accurate 3D state estimates. Our contributions can be summarized as follows:

- **Doppler and Velocity Uncertainty:** We assess the uncertainty of radar Doppler measurements and the resulting uncertainty on estimated linear body frame velocity for fusion with IMU sensor data.
- **Multi-Radar Inertial Odometry:** We employ a fixed-lag smoother optimization strategy capable of fusing IMU and multiple radar data to compensate for measurement uncertainty from each other and ensure robustness to outlier measurements.
- **Evaluate with 3D Motion:** We evaluate our method with hand-held collected data while traversing through different levels of indoor buildings.

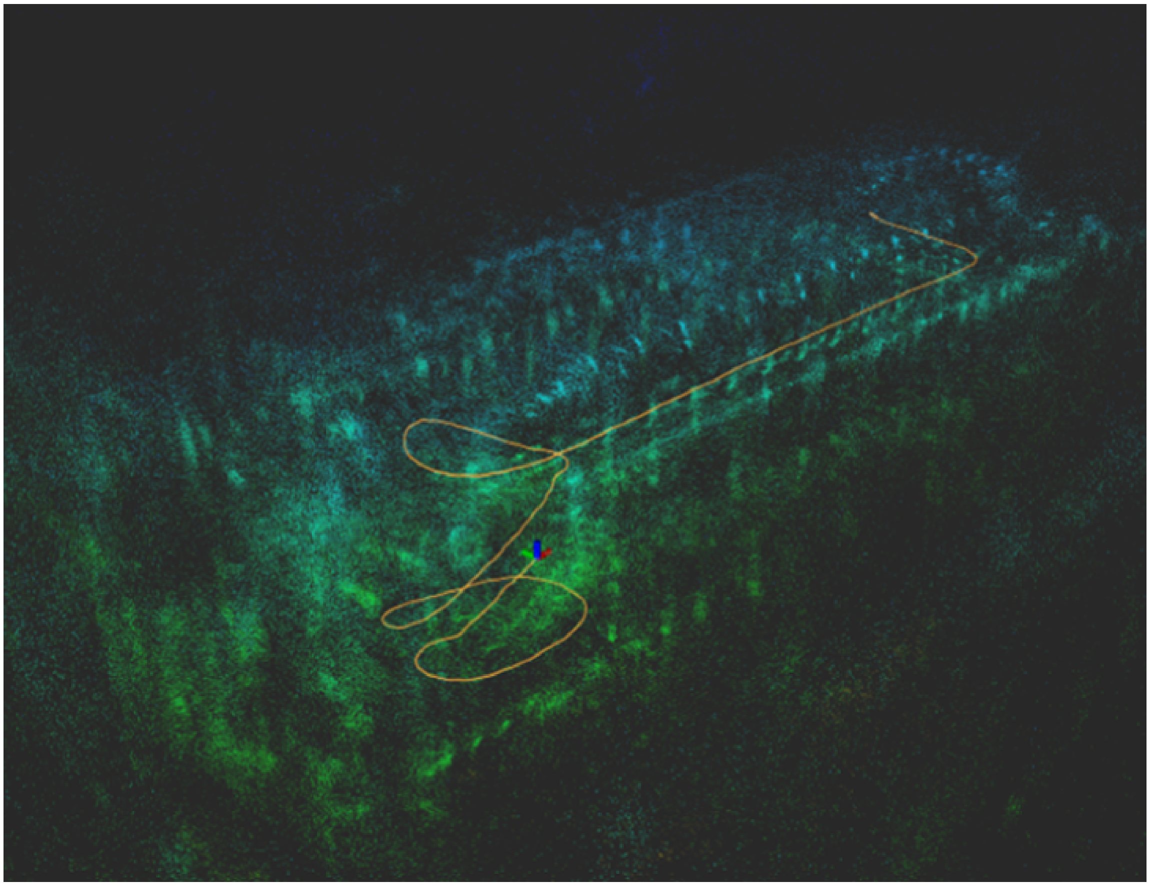


Figure 1.2: Radar points mapping of an atrium.

Chapter 2

Background

2.1 SLAM and Nonlinear Least Square

Simultaneous Localization and Mapping (SLAM) problems can be viewed as the maximum a posteriori estimation of robot states $\mathbf{X}$ and sensor measurements $\mathbf{Z}$. Given the measurements, we would like to find the most likely states $\hat{\mathbf{X}}$.

$$\hat{\mathbf{X}} = \underset{\mathbf{X}}{\operatorname{argmax}}\, p(\mathbf{X}|\mathbf{Z}) \tag{2.1}$$

Assume a prior distribution $p(\mathbf{X})$ exists. We can write the posterior distribution $p(\mathbf{X}|\mathbf{Z})$ using the Bayes rule.

$$\hat{\mathbf{X}} = \underset{\mathbf{X}}{\operatorname{argmax}}\, \frac{p(\mathbf{Z}|\mathbf{X})p(\mathbf{X})}{p(\mathbf{Z})} = \underset{\mathbf{X}}{\operatorname{argmax}}\, p(\mathbf{Z}|\mathbf{X})p(\mathbf{X}) \tag{2.2}$$

We can make the Markov assumption with conditional independence between the associated variables. Assume that each probability distribution here is a Gaussian distribution $\phi_i(\mathbf{X}_i)$.

$$\hat{\mathbf{X}} = \underset{\mathbf{X}}{\operatorname{argmax}} \prod \phi_i(\mathbf{X}_i) = \underset{\mathbf{X}}{\operatorname{argmax}} \prod \exp\left(\frac{-1}{2}||h_i(\mathbf{X}_i) - \mathbf{Z}_i||^2_{\Sigma_i}\right) \tag{2.3}$$

Where $h_i(\mathbf{X}_i)$ is the measurement prediction function given the state. We can further reduce this to a non-linear least-squares optimization problem.

$$\hat{\mathbf{X}} = \operatorname*{argmin}_{\mathbf{X}} \sum ||h_i(\mathbf{X}_i) - \mathbf{Z}_i||^2_{\Sigma_i} \tag{2.4}$$

To solve this problem, we can linearize the measurement function by using first-order Taylor expansion.

$$h_i(\mathbf{X}_i) = h_i(\mathbf{X}_i^0) + H_i\Delta_i \tag{2.5}$$

$$H_i = \left.\frac{\partial h_i(\mathbf{X}_i)}{\partial \mathbf{X}_i}\right|_{\mathbf{X}_i^0} \tag{2.6}$$

We are solving for state update vector Δ_i at each iteration.

$$\Delta^* = \operatorname*{argmin}_{\Delta} \sum ||A\Delta - b||^2 \tag{2.7}$$

Where residuals are whitened by the measurement covariance Σ_i

$$A = \Sigma_i^{1/2} H_i \tag{2.8}$$

$$b = \Sigma_i^{1/2}(\mathbf{Z}_i - h_i(\mathbf{X}_i^0)) \tag{2.9}$$

Then, the states are refined iteratively by applying the state update vector Δ_i, that is, the values are obtained by successive approximation until convergence.

$$\mathbf{X}^{t+1} = \mathbf{X}^t + \Delta \tag{2.10}$$

A variety of algorithms are available to solve for the optimal $\hat{\mathbf{X}}$, including methods such as steepest descent, Gauss-Newton, Levenberg-Marquardt, and Powell's Dogleg approach [6]. Kaess et al. introduced an advanced method, detailed in [18], using incremental update techniques coupled with matrix solvers to efficiently solve the system in real time.

2.2 Robust Error Model

In real-world SLAM problems, there may be completely bogus sensor measurements. By default, the optimization process will treat these outlier measurements as equally important as other measurements. Therefore, the optimization process will attempt to fit those outlier measurements, sacrificing the accuracy of the inlier measurements. Mathematically, since all measurement residuals carry the same weight, a standard nonlinear least-squares approach seeks to minimize the squared sum of all residuals. Consequently, this can distort the estimated parameters.

A group of models created to address this issue by reducing the impact of outlier residuals is called the robust error model, also known as the M-estimator in GTSAM [5], or the robust loss function in Ceres-Solver [1]. These robust error models aim to modify the magnitude of residuals using a function ρ. To replicate the squared cost for minor residuals, certain design options include the following.

$$\rho(0) = 0 \tag{2.11}$$

$$\rho'(0) = 1 \tag{2.12}$$

$$\rho'(r) < 1 \text{ in the outlier region} \tag{2.13}$$

$$\rho''(r) < 0 \text{ in the outlier region} \tag{2.14}$$

Some common robust error models can be found in Table 2.1 and Figure 2.1.

Huber	Cauchy	Tukey
$\rho(r) = \begin{cases} r & r \leq 1 \\ 2\sqrt{r} - 1 & r > 1 \end{cases}$	$\rho(r) = \log(1 + r)$	$\rho(r) = \begin{cases} r & r \leq 1 \\ 2\sqrt{r} - 1 & r > 1 \end{cases}$

Table 2.1: Common robust error models

The advanced technique Graduated Non-Convexity (GNC) [2] is designed for outlier elimination through the utilization of a flexible model. GNC suggests softening non-convexity in a truncated least-square problem by substituting the residual with an alternative function regulated by a single parameter.

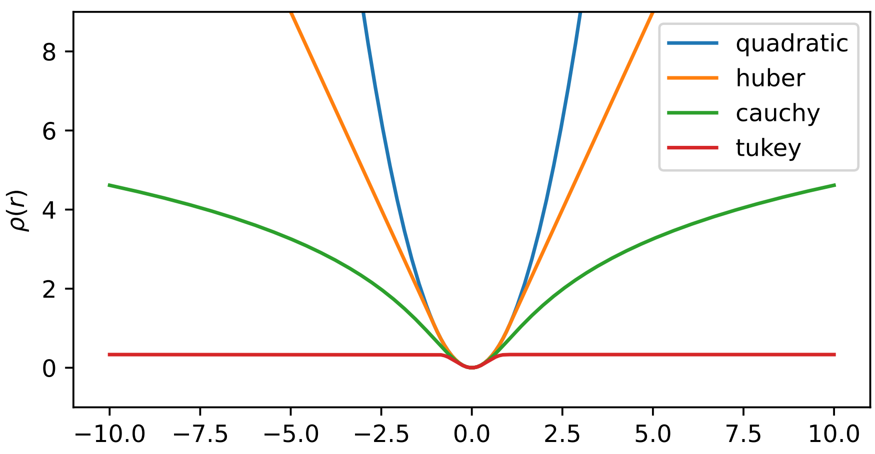


Figure 2.1: Robust error models

2.3 mmWave Imaging Radar

Millimeter wave radar, also known as mmWave radar, belongs to a unique category of radar technology that uses short-wavelength electromagnetic waves. The mmWave radar system emits electromagnetic waves that are reflected by objects in their field of view. When detecting the reflected signal, a radar system can determine the distance, speed, and direction of objects. Moreover, the mmWave radar system from Texas Instruments incorporates a unique type of mmWave technology, known as frequency-modulated continuous wave (FMCW). Unlike the conventional pulsed-radar system, which transmits periodic pulses, the FMCW radar system sends out a signal with frequency modulation in a continuous manner.

2.3.1 Radar Chirp Configuration

The frequency-modulated signal sent by the FMCW radar system is called a ”chirp”. An illustration of a chirp can be found in Figure 2.2, where the frequency of the sinusoidal wave increases linearly over time. The characteristics of a chirp signal can be modified and impact the balance between the highest achievable value and the precision of the measurements.

In a multi-input multi-output (MIMO) radar system, there are multiple transmit

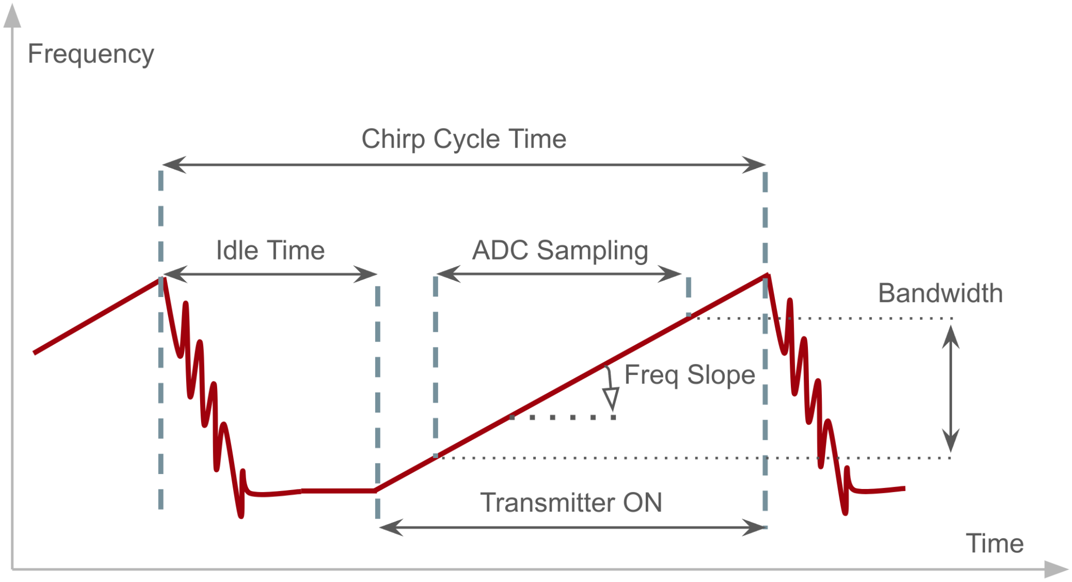


Figure 2.2: Illustration of a radar chirp.

(Tx) and receive (Rx) antennas. It is crucial to note that Rx antennas must be able to distinguish signals associated with various Tx antennas. One possible technique to achieve this is time division multiplexing (TDM). In each measurement frame, each Tx sends their chirp signal in a loop and repeats until the frame time $\mathbf{T}_f$ is reached. Note that the time to complete a loop is $\mathbf{T}_c$ as the Figure 2.3 shows.

In determining the range, our focus lies on the maximum range and the resolution of the range. While the maximum range $\mathbf{d}_{max}$ is influenced by both the ADC sampling rate $\mathbf{F}_s$ and the chirp slope $\mathbf{S}$, the range resolution $\mathbf{d}_{res}$ varies inversely with the chirp bandwidth $\mathbf{B}$.

$$\mathbf{d}_{res} = \frac{\mathbf{C}}{2\mathbf{B}} \qquad \mathbf{d}_{max} = \frac{\mathbf{F}_s\mathbf{C}}{2\mathbf{S}} \tag{2.15}$$

For measuring the velocity, we are concerned with the maximum velocity $\mathbf{v}_{max}$ and the velocity resolution $\mathbf{v}_{res}$. Where $\mathbf{v}_{max}$ is inversely proportional to $\mathbf{T}_c$, and $\mathbf{v}_{res}$ is inversely proportional to $\mathbf{T}_f$.

$$\mathbf{v}_{res} = \frac{\lambda}{2\mathbf{T}_f} \qquad \mathbf{v}_{max} = \frac{\lambda}{4\mathbf{T}_c} \tag{2.16}$$

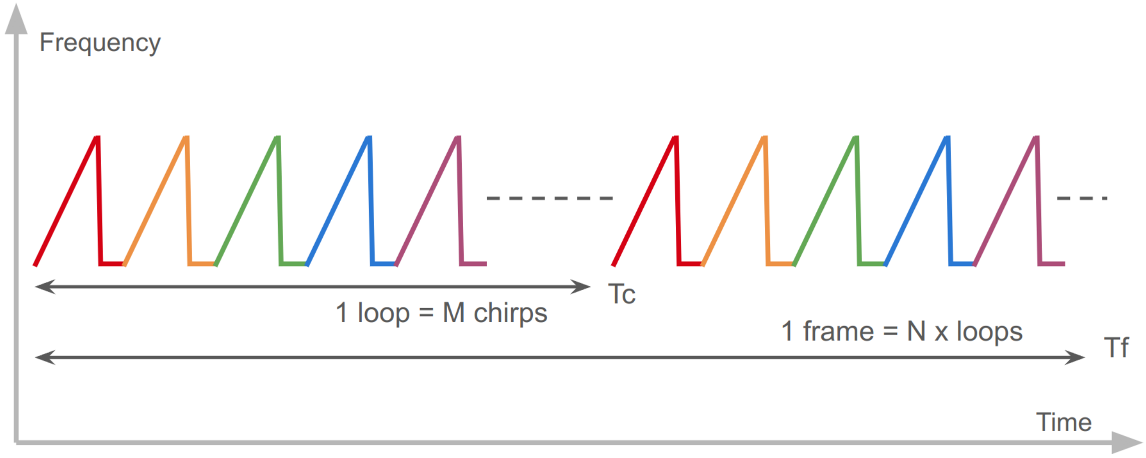


Figure 2.3: A frame of transmits signals from a TDM-MIMO radar, where M is the number of transmit antenna and N is a design choice.

Where **C** is the light speed and λ is the average wavelength. Finally, for measuring the angle of arrival, increasing the angular resolution requires increasing the number of Rx antennas. Details about the derivation of the above properties can be found in TI's documents[35]. The radar chirp configuration employed in this study is detailed in Chapter 4.

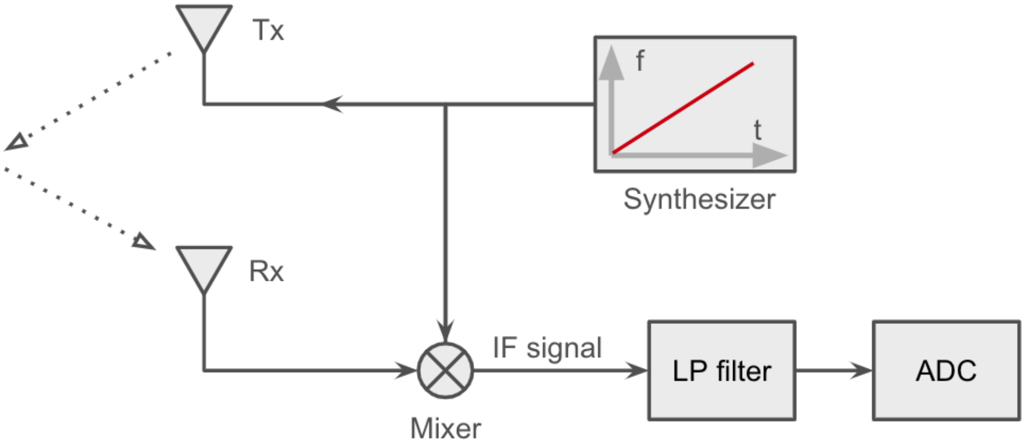


Figure 2.4: A radar system with single Tx and Rx.

2.3.2 Signal Processing

In this part, we will present some fundamental principles for processing radar signals to obtain the desired measurements. As depicted in Figure 2.4, by considering a scenario with a Tx antenna and an Rx antenna, the chirp signals produced by the synthesizer will be directed to a mixer where they will be merged with the signals received from the Rx, generating the IF signal. Here, the frequency and phase of the IF signal are simply the difference between the transmitted and received signals, as Figure 2.5 shows.

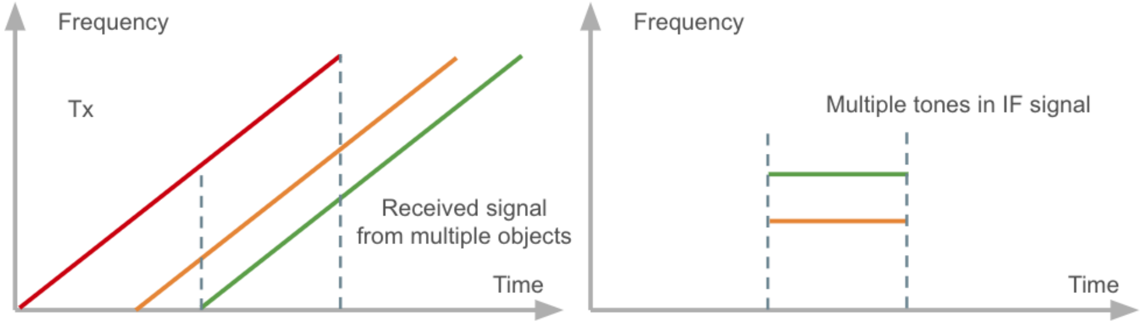


Figure 2.5: IF signal from a Tx antenna and an Rx antenna.

In a TDM-MIMO radar setup, there are several virtual antennas, each comprising a pair of Rx signals received from one of the Tx antennas. The first step to process data from the TDM-MIMO FMCW radar is to organize each digitized IF signal processed from each virtual antenna into an ADC data cube where each dimension is shown in Figure 2.6.

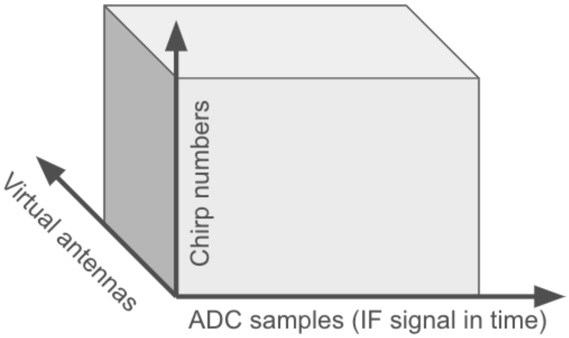


Figure 2.6: ADC data

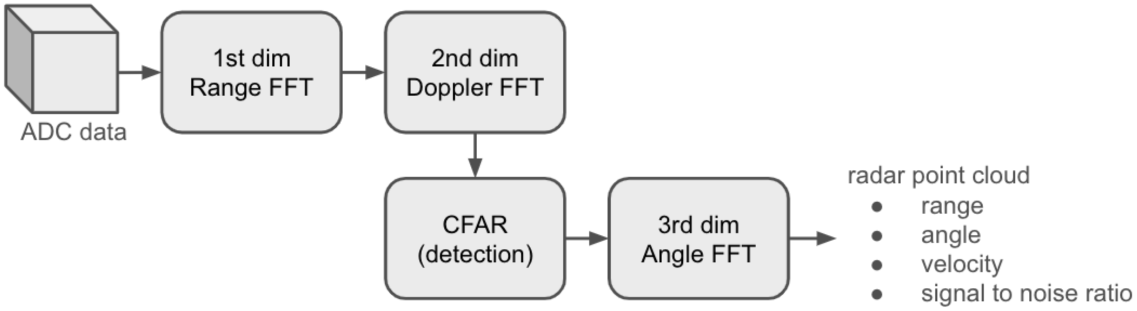


Figure 2.7: Signal processing pipeline to get point cloud from radar ADC data

With the ADC data from each frame. The first step is to apply the Fast Fourier Transform (FFT) in the ADC sample dimension to transform the IF signal into frequency domain, whose frequencies correspond to the distance of the detected objects.

To distinguish objects that could be equidistant from the radar, the data cube undergoes a secondary Doppler FFT along the chirp dimension on the sequence of phasors corresponding to the range FFT. This step is used to identify objects with the same distance, but with a different radial velocity from the sensor. The combination of the above two steps is called 2D-FFT.

The following step involves selecting potential objects from the data cube through the utilization of the constant false alarm rate (CFAR) algorithm. The last stage involves determining the angle of arrival (AoA) of various objects. This process can be achieved by implementing angle FFT across the virtual antenna dimension on the phasors linked to the 2D-FFT. By employing antennas positioned horizontally and vertically, it becomes possible to approximate the azimuth and elevation angles of arrival. The physics behind these signal processing steps are discussed in [16].

The processing procedure outlined above involves estimating objects in polar coordinates, which can then be converted into Cartesian space. The ultimate result of each radar frame is a point cloud, where each point contains information about its 3D position, velocity, and signal-to-noise ratio of a possible object in the scene.

Chapter 3

Related Work

3.1 Radar-Inertial Odometry

Doppler velocity information from mmWave radar point clouds has been used to estimate trajectory in previous research. Kramer et al. [20] proposed a sliding window optimization approach with a robust loss kernel to fuse IMU and radar Doppler velocity measurements. They demonstrate that the estimated body frame velocity is close to a visual-inertial solution. Doer et al. [7] separately estimated the body frame velocity from radar points using a RANSAC method to remove outlier measurements introduced by noise. Then they proposed an EKF-based approach to fuse radar body frame velocity with IMU and barometer measurements to estimate drone poses in indoor environments. However, their results show difficulty in correctly estimating height without the help of barometer data.

Zhuang et al. extended the previous EKF-based approach by integrating the complete radar SLAM system, known as 4D iRIOM [43]. The authors proposed the utilization of a Graduated Non-Convexity technique to eliminate outlier Doppler velocity measurements and estimate velocity based on radar data. Additionally, they introduced scan-to-map matching using radar points to enhance trajectory estimation within the system. Loop closure was achieved by employing Scan Context [19] to identify similar geometric patterns in the radar points map. The effectiveness of their methods was demonstrated on a small ground robot operating in a planar environment.

Recent research by Michalczyk et al. [26, 27] proposed building 3D landmarks from radar point clouds and fuse them with IMU using multi-state EKF. However, this method required a controlled environment where persistent landmarks with high radar cross sections are required.

A learning-based approach [24] by Lu et al. was proposed to fuse IMU and radar measurements to estimate trajectory. The authors use convolutional neural networks to build feature encoders for radar and recurrent neural networks to process IMU and the fused feature embeddings. However, their method was compared in [32] and was found to have difficulty generalizing to new environments.

3.2 Multi-Radar for State Estimation

To further improve state estimation using the mmWave radar sensor, adding multiple radar sensors in an attempt to increase the FOV has appeared in some research. Doer et al. based on their previous work EKF-RIO [7] proposed the x-RIO [8] using a triple radar setting to increase the horizontal FOV of the robot. The setting enables them to use the Manhattan World Assumption [4] on the radar point clouds to improve the trajectory estimation. In [31] Ng et al. use a sliding window optimization approach to fuse 4 radar measurements with an IMU to jointly estimate body frame velocity and odometry in an autonomous car application. However, both of the above multi-radar methods were applied only in the planar setting and did not address the elevation drift problem caused by relatively lower resolution on the elevation dimension.

Park et al. [32] proposed a solution to improve 3D ego-motion estimation by adding a second ground-facing radar that operates in synchronization with the primary radar. They only extract 2D velocity on the high-resolution azimuth dimension from each radar to fuse with the IMU. They proposed a radar velocity factor that fuses radar velocity with the IMU gyroscope. The authors evaluated their method by traversing different levels of a construction site to prove their performance with 3D motion. In comparison to their method, we present a solution to fuse multiple radars and IMU data without discarding high-frequency accelerometer measurements and without the need to trigger the radars synchronously.

The publicly available dataset for radar state estimation Coloradar dataset [21] provides a cascade imaging radar and a single-chip imaging radar both placed horizontally. The dataset was collected with mostly planar motion. Therefore, it's necessary to design our own radar sensors rig in order to collect 3D motion radar data for the research of radar-inertial state estimation.

3.3 Improve on Radar Mapping

To improve radar map quality, researchers have experimented with learning-based reconstruction methods supervised by other sources of accurate range information. Xu et al. [41] developed a learning pipeline to train a 3D convolutional network using radar intensity volume as input, reconstructing a depth image and a valid distance classification image. This approach can produce LiDAR-like depth images and was trained with a real-world LiDAR-radar pair dataset. Another study by Narashimha et al. [29] employed a custom generative transformer architecture, UpPoinTr, which upsamples, denoises, and fills incomplete radar maps to resemble LiDAR maps.

Drawing inspiration from Neural Radiance Fields [28], Huang et al. [15] developed a method to train a neural field for reflectance and transmittance using radar range-Doppler images. Instead of rendering a ray, they render the reflectance and transmittance values along a Doppler cone. This trained neural field can function as a novel view synthesizer or be processed as 3D tomography.

However, these works rely on other sources of sensors to obtain accurate poses for their systems or obtain accurate depth as labels for learning neural networks. Further research should focus on improving mapping solely using radar sensors.

Chapter 4

Sensor System Design

4.1 Sensor Rig

We designed our handheld sensor platform to collect time-synchronized radar, IMU, and image data for evaluating multi-radar inertial state estimation. The platform includes a Velodyne VLP-16 LiDAR, two PointGrey Cameras running at 20Hz, an Epson G-364 IMU running at 200Hz, an ICM-20948 IMU running at 100Hz, and two Texas Instrument Cascade Imaging Radars (MMWCAS-RF-EVM) running at 10Hz. The sensor platform layout is depicted in Fig. 1.2. Timing is synchronized using an external Teensy MCU with pseudo GPS/PPS signals and trigger signals for the cameras.

4.2 Dual Cascade Imaging Radar Dataset

The commercially available TI mmWave cascade imaging radar has a theoretical azimuth/elevation angular resolution of 1.4°/18° [35]. These resolutions represent the minimum angle between two equally large targets at the same range that the radar can distinguish and separate from each other. To compensate for this limitation, we employ two radars placed horizontally and vertically in this research. Our radar operates in Multi-Input Multi-Output (MIMO) mode. To avoid frequency conflicts, we set the start frequency of the horizontal and vertical radars as 77GHz and 79GHz, with

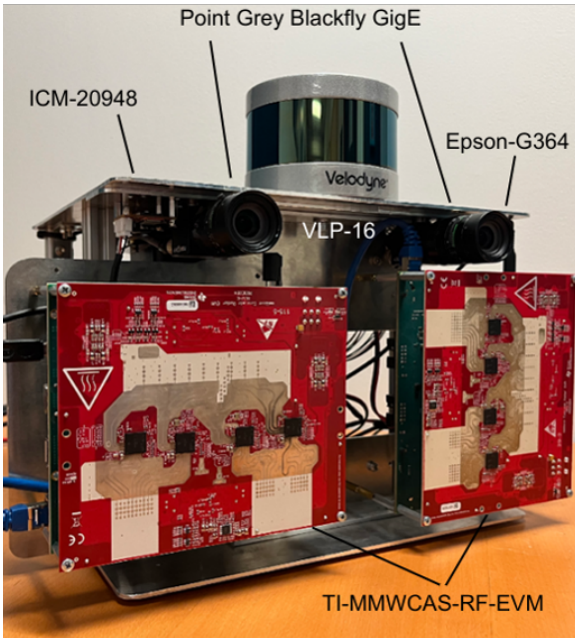


Figure 4.1: Sensor rig.

both radars having a chirp slope of $40\mathrm{MHz}\,\mu\mathrm{s}^{-1}$ and a bandwidth of approximately 1.5GHz. With these radar signal settings, we achieve a maximum range of 30m, a Doppler velocity resolution of $0.055\mathrm{m\,s}^{-1}$, and a maximum Doppler velocity range of $\pm 1.76\mathrm{m\,s}^{-1}$. We employed a Constant False Alarm Rate (CFAR) algorithm provided by the TI mmWave studio to process the raw radar data into point clouds, including signal-to-noise ratio and Doppler velocity information for each point.

Unlike previous work [32], where one radar was directed downward to the ground, we designed both radars to face forward. This decision was made to accommodate robots with limited space for installing downward-facing sensors. Additionally, our state estimation results suggest potential future enhancements in imaging radar designs to improve resolution in both azimuth and elevation dimensions so that a single sensor would be sufficient.

We collected three sequences with mostly planar motion and three sequences involving 3D motion across various levels within the building. The data collection was conducted at a normal human walking pace and motion. The traverse length of each trajectory is detailed in TABLE 6.1.

Chapter 5

Radar-Inertial Odometry

5.1 Velocity Estimation using Radar Doppler Velocity

One special property of the Imaging Radar is the Doppler velocity of the detected points. The physical significance of the Doppler velocity is the projection of the sensor's linear velocity in the direction of the point, as the Figure 5.1 shows. Assuming that most of the points in one frame are static objects, we can establish the relationship between the Doppler velocity $\mathbf{d}_n \in \mathbb{R}^1$ of the point n, its 3D position $\mathbf{r}_n \in \mathbb{R}^3$, and the sensor's linear velocity $\mathbf{v}_j^s \in \mathbb{R}^3$ in frame j with the following equation:

$$-\mathbf{d}_n = (\mathbf{v}_j^s)^\top \frac{\mathbf{r}_n}{||\mathbf{r}_n||} \tag{5.1}$$

To obtain an accurate linear velocity of the sensor that aligns with the most measured Doppler velocities, we can use a least-squares approach to solve for $\mathbf{v}^s$. The noise of Doppler measurement can be modeled by a Gaussian distribution with variance $\Sigma_d \in \mathbb{R}^1$. The optimized sensor velocity will be:

$$\mathbf{v}^s = \underset{\mathbf{v}^s}{\operatorname{argmin}} \left(\sum_n ||(\mathbf{v}^s)^\top \frac{\mathbf{r}_n}{||\mathbf{r}_n||} + \mathbf{d}_n||^2_{\Sigma_d} \right) \tag{5.2}$$

We can find the marginalized covariance $\Sigma_{\mathbf{v}^s} \in \mathbb{R}^{3\times3}$ on variable $\mathbf{v}^s$ by inverting the information matrix in the system.

$$\Sigma_{\mathbf{v}^s} = \left(A^\top \Sigma_d^{-1} A\right)^{-1} \tag{5.3}$$

Where $A \in \mathbb{R}^{n\times3}$ and each row of A is a unit vector directing to the radar point $\mathbf{r}_n/||\mathbf{r}_n||$. Notice that due to the radar antenna design, the radar points distribution is uneven in azimuth dimension and elevation dimension. As a result, the estimated velocity uncertainty exhibits variations along the XYZ axes. Further details regarding the evaluation of Doppler measurement uncertainty Σ_d and the estimated velocity uncertainty will be discussed in 6.1.

In reality, radar data often includes noisy measurements and non-stationary objects. To remove these outliers, previous works have explored the RANSAC approach [8], non-linear optimization with Cauchy robust loss [20], and a Graduated Non-Convexity (GNC) method [43]. While RANSAC is non-deterministic, GNC requires extra iterations to adjust the kernel, and the Cauchy robust loss is sensitive to the initial value setting. Considering computation efficiency and accuracy, we use a Cauchy robust loss kernel to remove outliers and use the Levenberg–Marquardt optimizer to solve the system. The initial value of the variable $\mathbf{v}^s$ was set using the preintegrated body frame velocity from IMU measurements and rotated to the sensor's coordinate. Please refer to the next section for IMU integration.

Finally, to fuse the sensor velocity from different radars with IMU, we treat the IMU frame as the body frame. The linear velocity and the corresponding covariance on the body frame are as follows:

$$\mathbf{v}^b = \mathbf{R}_r \mathbf{v}^s \tag{5.4}$$

$$\Sigma_{\mathbf{v}^b} = \mathbf{R}_r \Sigma_{\mathbf{v}^s} \mathbf{R}_r^\top \tag{5.5}$$

Where $\mathbf{R}_r \in SO(3)$ is the rotation from the radar sensor coordinate to the IMU coordinate.

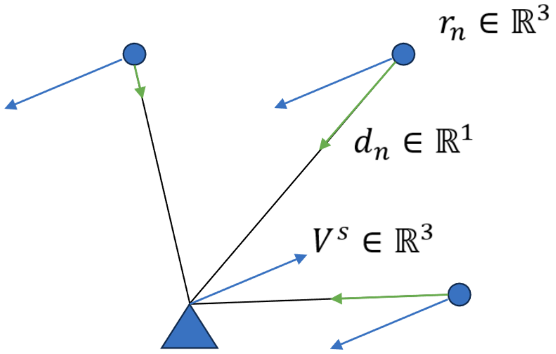


Figure 5.1: An illustration of Doppler speed projection.

5.2 Preintegrated IMU Factor

As introduced in [9], we can efficiently fuse the IMU sensor with other low frame rate measurements using preintegration on manifold to avoid the relinearization procedure when the keyframe linearization point changes. With sensor measured linear acceleration $\tilde{\mathbf{a}}_k$ and angular velocity $\tilde{\omega}_k$ between keyframe i and the next keyframe j, the preintegrated measurement of relative position $\Delta\tilde{\mathbf{p}}_{ij}$, orientation $\Delta\tilde{\mathbf{R}}_{ij}$, and velocity $\Delta\tilde{\mathbf{v}}_{ij}$ are:

$$\Delta\tilde{\mathbf{R}}_{ij} = \prod_{k=i}^{j-1} Exp((\tilde{\omega}_k - \mathbf{b}_i^g)\Delta t) \tag{5.6}$$

$$\Delta\tilde{\mathbf{v}}_{ij} = \sum_{k=i}^{j-1} \Delta\tilde{\mathbf{R}}_{ik}(\tilde{\mathbf{a}}_k - \mathbf{b}_i^a)\Delta t \tag{5.7}$$

$$\Delta\tilde{\mathbf{p}}_{ij} = \sum_{k=i}^{j-1} \left(\Delta\tilde{\mathbf{v}}_{ik}\Delta t + \frac{1}{2}\tilde{\mathbf{R}}_{ik}(\tilde{\mathbf{a}}_k - \mathbf{b}_i^a)\Delta t^2 \right) \tag{5.8}$$

Where $\mathbf{b}_i^a$ and $\mathbf{b}_i^g$ are the slow varying linear acceleration bias and angular velocity bias. The measurement constraint of the IMU between 2 keyframes has the residual in the following form:

$$\mathbf{r}_{\Delta \mathbf{R}_{ij}} = \mathit{Log}(\Delta \tilde{\mathbf{R}}_{ij}) \mathbf{R}_i^\top \mathbf{R}_j \tag{5.9}$$

$$\mathbf{r}_{\Delta \mathbf{v}_{ij}} = \mathbf{R}_i^\top \left(\mathbf{v}_j - (\mathbf{v}_i + \mathbf{g} \Delta t_{ij}) \right) - \Delta \tilde{\mathbf{v}}_{ij} \tag{5.10}$$

$$\mathbf{r}_{\Delta \mathbf{p}_{ij}} = \mathbf{R}_i^\top \left(\mathbf{p}_j - \left(\mathbf{p}_i + \mathbf{v} \Delta t_{ij} + \frac{1}{2} \mathbf{g} \Delta t_{ij}^2 \right) \right) - \Delta \tilde{\mathbf{p}}_{ij} \tag{5.11}$$

$$\mathbf{r}_{\Delta \mathbf{b}^a_{ij}} = \mathbf{b}^a_j - \mathbf{b}^a_i \tag{5.12}$$

$$\mathbf{r}_{\Delta \mathbf{b}^g_{ij}} = \mathbf{b}^g_j - \mathbf{b}^g_i \tag{5.13}$$

The combined preintegrated IMU residual is written as:

$$\mathbf{r}_{\Delta \mathbf{I}_{ij}} = \left[\mathbf{r}_{\Delta \mathbf{R}_{ij}}, \mathbf{r}_{\Delta \mathbf{v}_{ij}}, \mathbf{r}_{\Delta \mathbf{p}_{ij}}, \mathbf{r}_{\Delta \mathbf{b}^a_{ij}}, \mathbf{r}_{\Delta \mathbf{b}^g_{ij}} \right] \in \mathbb{R}^{15} \tag{5.14}$$

With covariance $\Sigma_{\Delta \mathbf{I}_{ij}} \in \mathbb{R}^{15 \times 15}$ which takes into account the noise in the estimated bias used for integration, it also preserves the correlation between the bias uncertainty and the preintegrated measurements' uncertainty. Part of the covariance matrix related to the IMU bias variables is used to describe slow-varying bias evolution, the magnitude of which is proportional to the preintegration time Δt_{ij}. More details about IMU preintegration on manifold can be found in [9].

5.3 Body Frame Velocity Factor

Upon receiving any body frame velocity $\mathbf{v}^b_j$ estimated from the radar. We build a keyframe out of the preintegrated IMU measurements. This body frame velocity constraint will be added between the integrated global velocity variable $\mathbf{v}_j$ and the rotation part of the pose variable $\mathbf{R}_j \in SO(3)$. By the definition of body frame velocity, the residual of this factor at frame j is:

$$\mathbf{r}_{\mathbf{V}^b_j} = \mathbf{R}_j^\top \mathbf{v}_j - \mathbf{v}^b_j \tag{5.15}$$

Occasionally, inaccurate estimates of the body frame velocity lead to outlier residuals. To counteract this issue, we incorporate a Huber loss kernel on this body frame velocity residual when solving the system.

To linearize the system, the Jacobian matrix with respect to rotation $\mathbf{J}_R$ and global velocity $\mathbf{J}_\mathbf{v}$ are:

$$\mathbf{J}_R = [\mathbf{R}^\top \mathbf{v}]_\times \tag{5.16}$$

$$\mathbf{J}_\mathbf{v} = \mathbf{R}^\top \tag{5.17}$$

Where $[\mathbf{R}^\top \mathbf{v}]_\times$ is a skew symmetric matrix

$$[\mathbf{R}^T \mathbf{v}]_\times = [\mathbf{q}]_\times = \begin{bmatrix} 0 & -\mathbf{q}_z & \mathbf{q}_y \\ \mathbf{q}_z & 0 & -\mathbf{q}_x \\ -\mathbf{q}_y & \mathbf{q}_x & 0 \end{bmatrix} \tag{5.18}$$

5.4 IMU Static Initialization

Since our preintegrated IMU factor is only constrained by the integrated velocity without a direct constraint within or among poses, an incorrect initial bias and rotation can easily cause the system to fail at the beginning. As IMU biases are modeled as slow-varying variables, it requires more steps to converge to the correct scale. Therefore, we have adapted and deployed a static initialization strategy only using IMU from the visual-inertial state estimation system [10].

The initialization process involves two consecutive sliding windows collecting IMU measurements. The second window, which includes the latest acceleration measurements, is used to detect any sensor movement by measuring the acceleration variance. Once motion is detected, we average the acceleration measurements in the first window to determine the gravity vector and its orthonormal basis using the Gram-Schmidt process. The initial rotation is determined by taking the inverse of the $SO(3)$ rotation formed by the orthonormal basis. The initial accelerometer bias is calculated by rotating the gravity constant using the initial rotation and then subtracting it from the measured gravity vector. The initial gyroscope bias is simply the mean of the angular velocities in the first measurement window. We have found this initialization procedure to be particularly crucial for uncalibrated IMUs with larger biases.

5.5 Factor Graph Formulation

As IMU and radar measurements are inherently asynchronously triggered in time, IMU measurements were interpolated to be temporally aligned with radar measurements. This procedure guarantees that IMU measurements will always be available to construct a preintegrated IMU factor, even when two radar measurements are close and no IMU measurement exists in between.

Given all the estimated measurements and their covariance. The full system optimizes states in a sliding window $\mathcal{S}$ and minimizes their Mahalanobis distance:

$$\mathcal{X}_{\mathcal{S}}^{*} = \underset{\mathcal{X}_{\mathcal{S}}}{\operatorname{argmin}} \left[\sum_{i,j\in\mathcal{S}} \left(||\mathbf{r}_{\Delta\mathbf{I}_{ij}}||^2_{\Sigma_{\Delta\mathbf{I}_{ij}}} + ||\mathbf{r}_{\mathbf{V}_j^b}||^2_{\Sigma_{\mathbf{v}^b}} \right) \right] \tag{5.19}$$

Where body frame velocity covariance $\Sigma_{\mathbf{v}^b}$ was used for proper fusion with preintegrated IMU factor.

An illustration of the system as a factor graph is shown in Fig. 5.2. The aforementioned system was constructed using the GTSAM library [5]. The optimization problem is solved by a fixed-lag smoother with iSAM2 [18]. We set a 5-second optimization window where variables passed the window will be marginalized as prior factors.

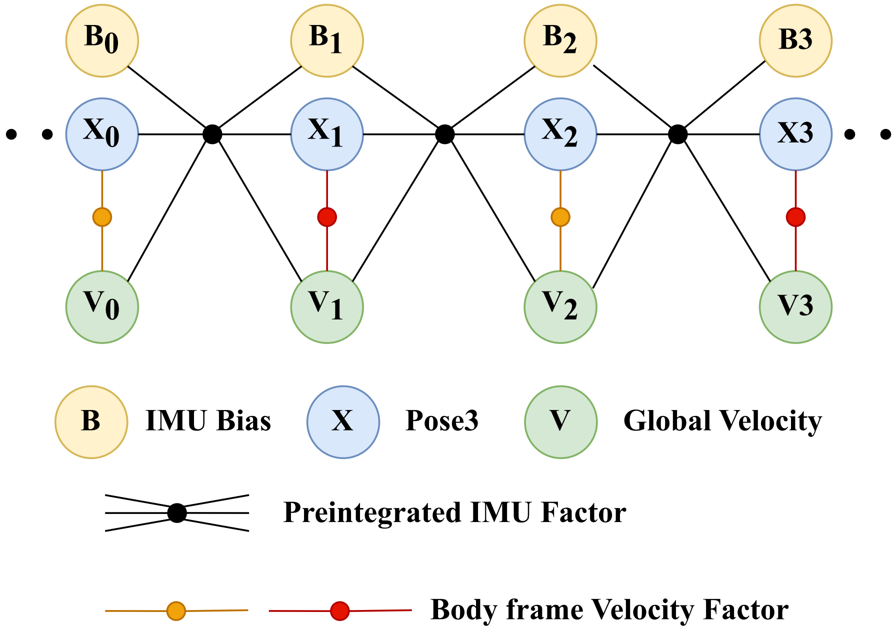


Figure 5.2: An illustration of our multi-radar inertial state estimation system in the form of a factor graph. The body frame velocity factor can be built with either our horizontal or vertical imaging radar.

Chapter 6

Experiments

6.1 Doppler and Velocity Uncertainty Evaluation

To properly fuse all radar measurements with IMU measurements, we need to understand the noise levels of both sensors. The IMU measurement uncertainty can be found in manufacturers' manuals or determined by analyzing hours of static IMU data [40]. However, current research and the radar sensor manufacturer do not provide Doppler velocity measurement uncertainty, and the Doppler velocity resolution does not accurately represent it.

To assess radar Doppler velocity errors, we utilize body-frame velocity data obtained from visual-inertial odometry pseudo ground truth. By projecting this velocity onto the direction of all radar points during a two-minute sequence (NSH_atrium), we can analyze the Doppler velocity error distribution, as depicted in Fig. 6.1. The smaller peak in this bimodal distribution is the result of Doppler velocity values exceeding sensor limits. After excluding such outliers, we approximate the noise as a Gaussian distribution with variance $\Sigma_d \approx (0.124\ m/s)^2$. It's important to note that this uncertainty may vary with different radar signal settings, depending on factors such as Doppler velocity resolution and elevation/azimuth angle resolution.

We use this Doppler velocity noise model to compute the marginalized covariance on the optimized sensor frame velocity, and then the corresponding uncertainty on body frame velocity follows equations (5.3) and (5.5). Fig. 6.2 displays the standard deviation of the body frame linear velocity estimated from two different radars in

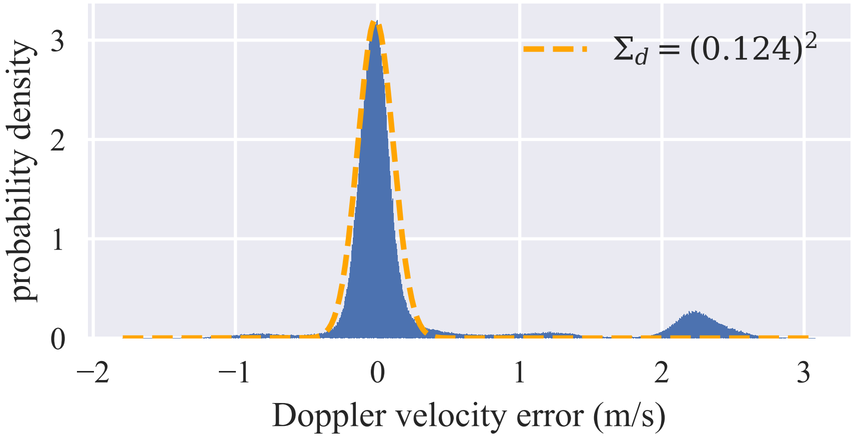


Figure 6.1: The probability density of the Doppler velocity error distribution is calculated by comparing the velocities of 1,092,224 radar points with the VIO body-frame velocity projected in their respective directions.

a two-minute sequence. For each radar, the axis with high uncertainty aligns with the elevation direction. This emphasizes the importance of using multiple radars to compensate for inaccurate measurements from each other and to fuse them with the correct covariance scale. We also noticed that uncertainty increases when there are fewer radar measurements, such as at the sequence's start and end when sensor motion is nearly stationary, and around the 70-second mark when the radar faces an open area for a short period.

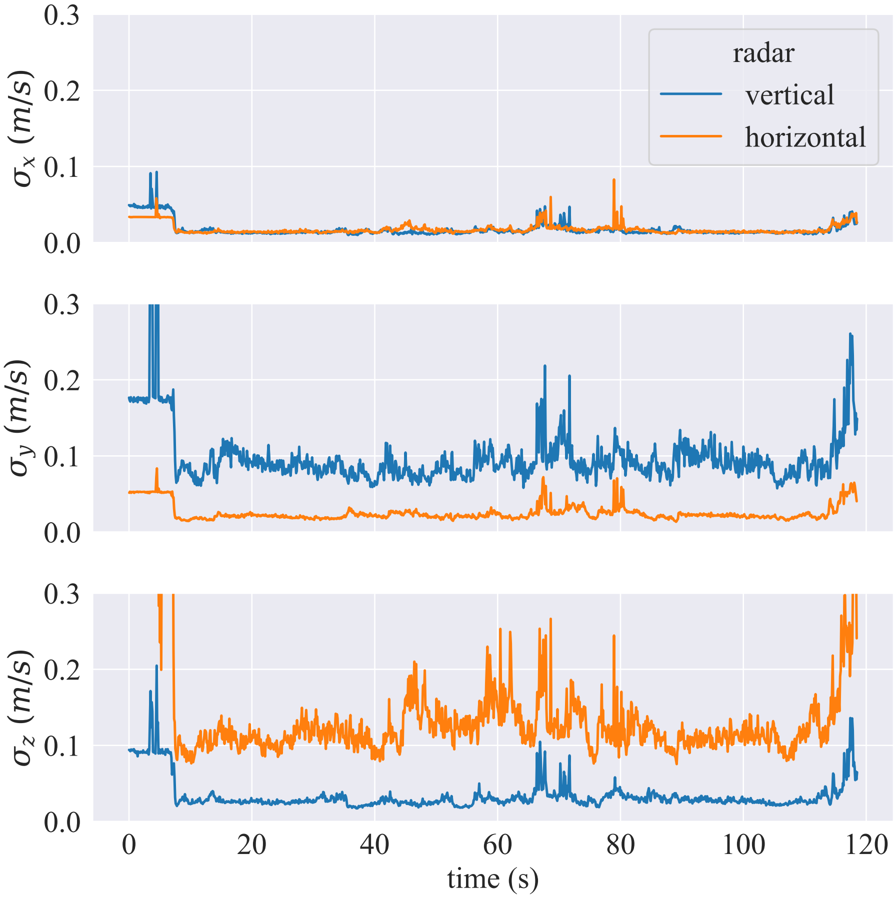


Figure 6.2: Standard deviation on XYZ axes of the estimated body frame velocity from two radars.

6.2 Quantitative Evaluation

Given that our evaluation environment is indoors and includes long corridors, LiDAR point cloud registration becomes challenging in environments with such geometric features. Therefore we evaluate our method using pseudo ground truth generated by the stereo visual-inertial odometry system OpenVINS [10], with two PointGrey cameras and the Epson IMU. Our radar-inertial system is running with ICM-20948 IMU. The final trajectory was assessed using the EVO library [11], without the use of any alignment algorithms. The results for absolute and relative pose errors are presented in TABLE 6.1. From the results, it's apparent that using both radars leads to a slight decrease in rotation performance. However, we are able to significantly improve translational drift compared to using a single radar setting.

We had considered comparing our method with, to the best of our knowledge, the only open-sourced EKF-based radar-inertial odometry algorithm, X-RIO [8]. However, their algorithm was primarily designed for single-chip mmWave radar and requires handcrafted radar velocity uncertainty, such as adjustments for offset and maximum limit of the radar velocity covariance, defining the threshold for rejecting radar velocity updates. We chose not to include this comparison in our evaluation, as we were unable to generate meaningful results.

Table 6.1: RMSE of APE and RPE in 6 different sequences

		APE		RPE	
sequence & info	method	trans	rot	trans	rot
2D FRC_1st 113 m	dual	**1.48**	2.73	**0.08**	0.66
	horizontal	3.46	**2.33**	0.15	**0.58**
	vertical	2.16	2.95	0.21	0.72
2D FRC_2nd 158 m	dual	**1.31**	3.76	**0.09**	0.65
	horizontal	2.88	**3.40**	0.12	0.69
	vertical	2.92	5.16	0.13	**0.63**
2D NSH_4th 167 m	dual	**1.29**	2.27	**0.08**	0.75
	horizontal	4.07	**1.52**	0.12	**0.66**
	vertical	1.77	2.74	0.12	0.68
3D NSH_atrium 139 m	dual	**1.09**	4.92	**0.10**	0.89
	horizontal	7.22	**2.33**	0.18	0.88
	vertical	1.93	5.38	0.22	**0.78**
3D GHC_stair 146 m	dual	**1.16**	8.13	**0.11**	0.63
	horizontal	6.25	**5.02**	0.14	**0.57**
	vertical	1.58	7.50	0.16	0.71
3D GHC_ramp 219 m	dual	**1.36**	7.23	**0.06**	**0.53**
	horizontal	4.09	**6.43**	0.09	0.60
	vertical	1.78	7.30	0.08	**0.53**

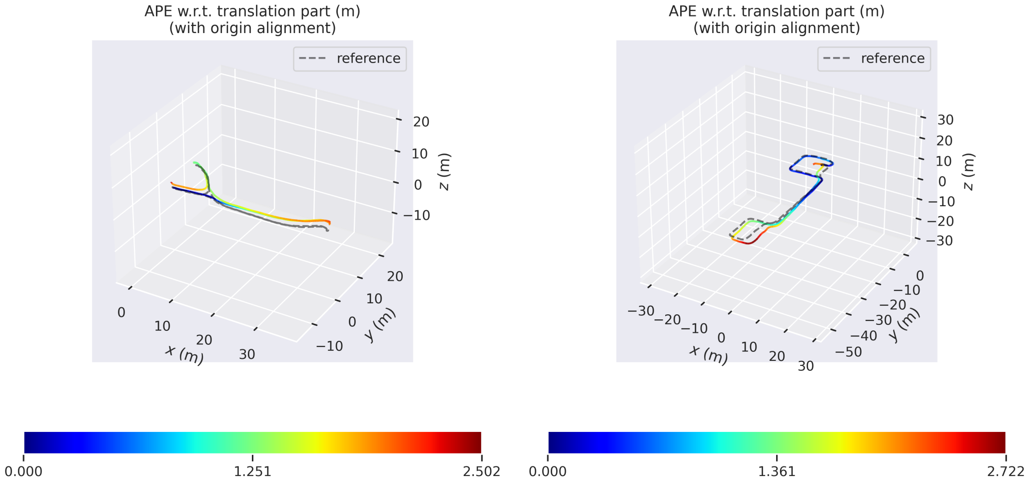
The units for APE: translation / rotation are meter and degree.
The units for RPE: translation / rotation are percentage and degree per meter.

6.3 Qualitative Evaluation

Fig. 6.5 displays the estimated trajectory of our method compared to visual-inertial odometry. From the top view, it's clear that using only vertical radar results in increased translational drift compared to other settings, mainly due to its limited capacity to estimate lateral velocity. On the other hand, employing only a horizontal radar yields excellent performance in the XY plane but leads to substantial drift along the z-axis due to its reduced capacity to estimate vertical velocity effectively.

Given that human walking motion involves minimal lateral movement, the introduction of more vertical motion when traversing different levels of a building can result in significant drift when exclusively using horizontally placed radar sensors. This highlights the importance of considering motion primitives when deciding where to install radars for Doppler velocity-based radar-inertial odometry.

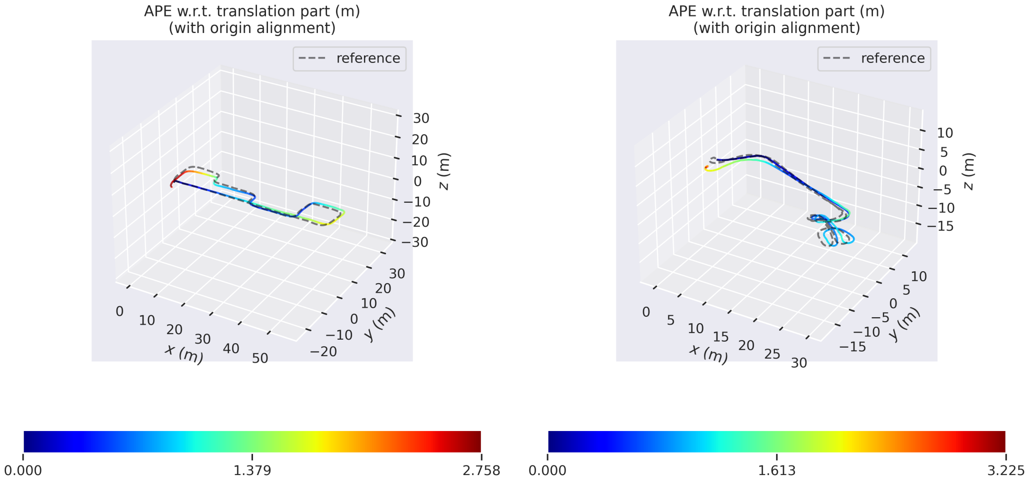
The attached video shows mapping results using estimated poses and radar points that have Doppler velocity projection error under the threshold. In the video, the geometry of the environment emerges even with noisy radar point clouds.



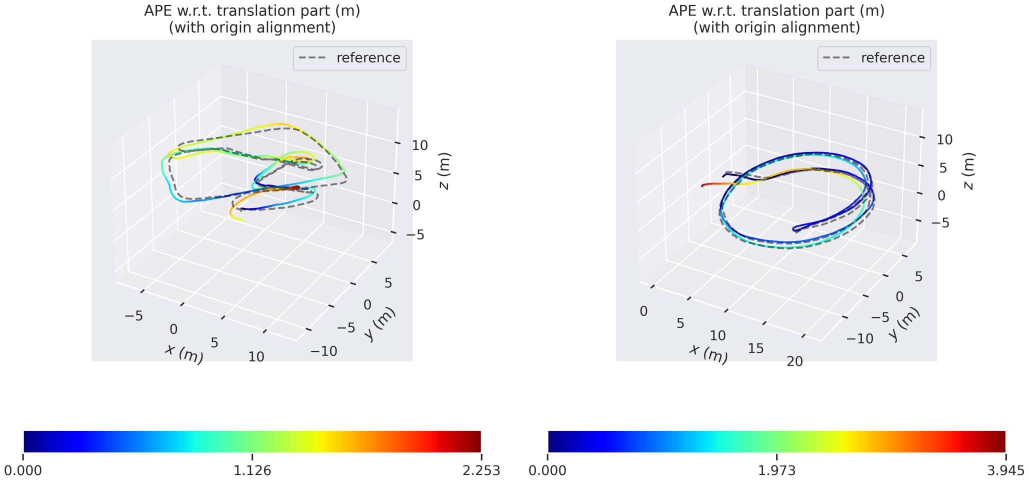
(a) FRC_1st

(b) FRC_2nd

Figure 6.3: Comparing the trajectories of visual-inertial odometry to radar-inertial odometry using dual radars.



(a) NSH_4th (b) NSH_atrium



(c) GHC_stair (d) GHC_ramp

Figure 6.4: Comparing the trajectories of visual-inertial odometry to radar-inertial odometry using dual radars.

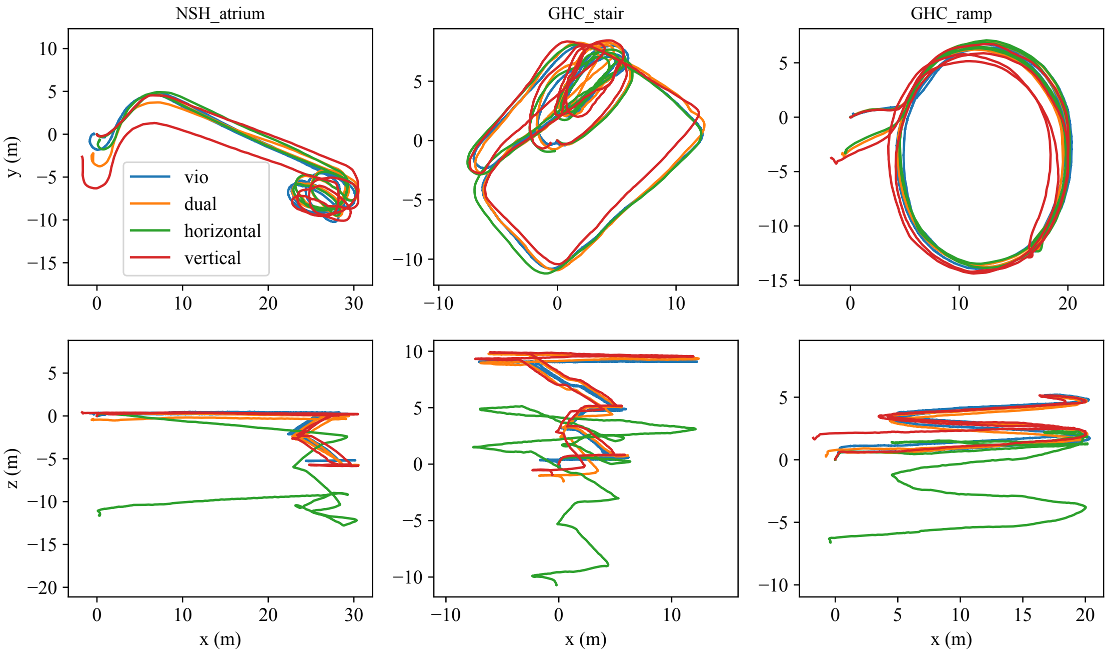


Figure 6.5: Comparing the trajectories of visual-inertial odometry to radar-inertial odometry using either single horizontal/vertical radar or dual radars. Each column represents our 3D motion sequences, with the top row displaying the top-down view and the bottom row presenting the side view.

Chapter 7

Downstream Tasks

In this part, we showcase the effectiveness of our radar-inertial localization system beyond just assessing the trajectory. Additional downstream tasks encompass mapping, relocalization, and integrating sensor data from visual-inertial odometry to enhance the overall performance of localization and mapping using mmWave Imaging Radar.

7.1 Mapping

To build a clear radar point cloud map, we transform the radar point clouds in each frame into the global frame according to their estimated poses. For each frame of radar points, we remove the noisy points whose Doppler velocity residual is greater than one standard deviation.

To build a geometry-consistent map, we took inspiration from the LiDAR-Inertial odometry system [34] by building scan-to-map constraints. However, because of the sparsity in radar point cloud, we need to accumulate multiple frames as a sub-key frame for enforcing scan-to-map constraint in our fixed-lag smoother optimization frame work. For each sub-key frame, the accumulated frames will be transformed to the last frame position and voxel-down sampled before being stored in a KD-Tree. Whenever we build a sub-key frame, we will search the KD-Tree for sub-key frames within 6 meters to build a local map. With this local map, we can find the transformation between our sub-key frame and the local map using a general ICP algorithm. We build this transformation as a prior factor for the pose variable

$\mathbf{X}_n$ to constraint the pose on the global map as shown in Figure 7.1. Considering computation efficiency and our traveling speed, we set the number of frames to accumulate to 40.

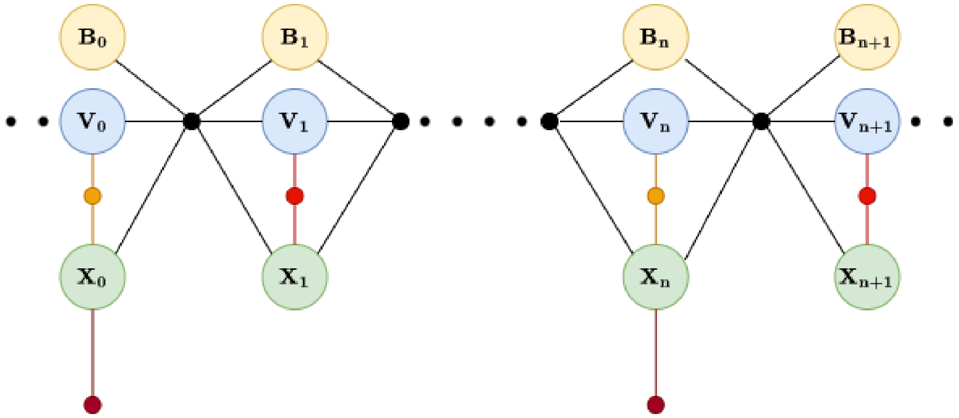


Figure 7.1: Factor Graph of Radar-Inertial odometry with scan-to-map constraint.

With this map constraint, we were able to build a more consistent radar point map, as Figure 7.2 shows. Even with the sparse and noisy radar point cloud, certain geometries such as column structures and an open staircase emerge. In addition, we can build 3D and 2D occupancy maps with the processed radar point cloud using OctoMap [13] as Figure 7.3 shows.

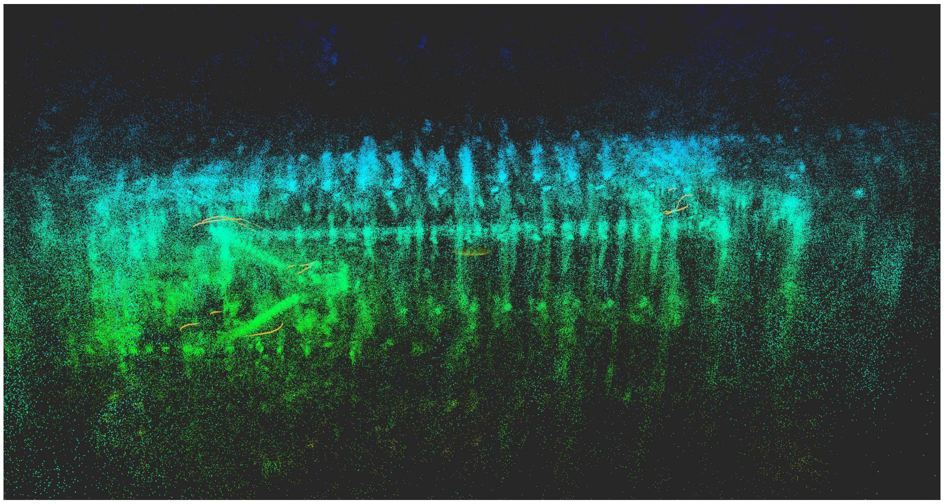


Figure 7.2: Radar point 3D map of an atrium with open staircase.

7.2 Relocalization

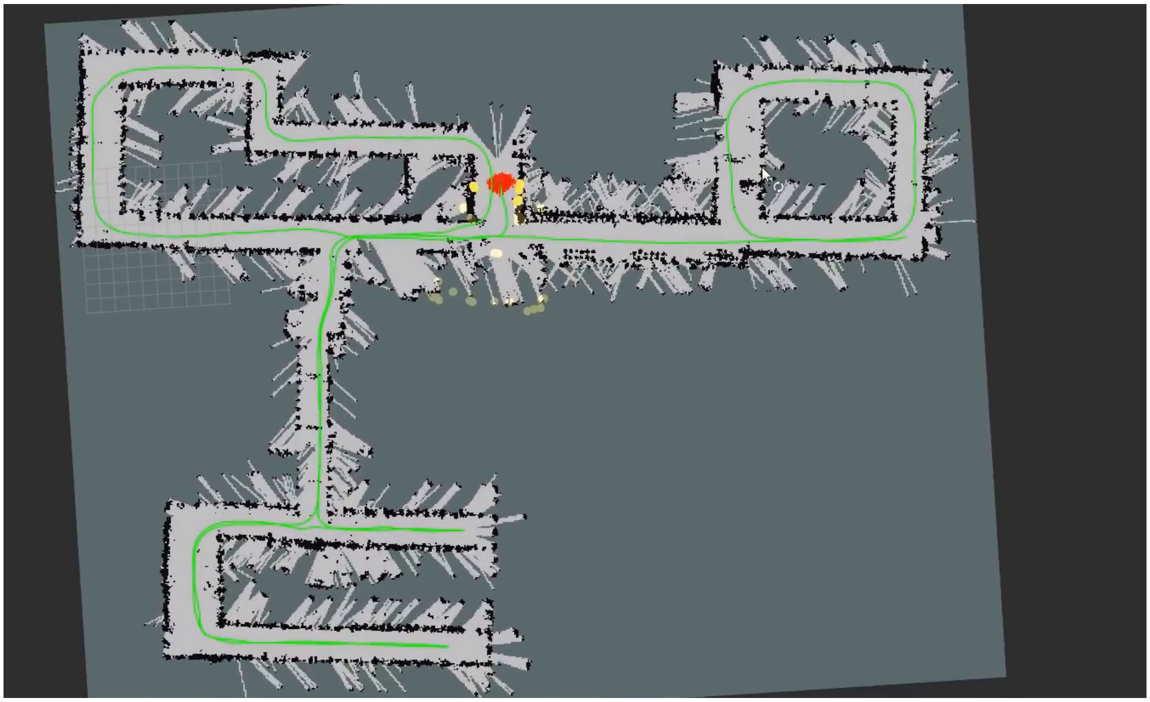


Figure 7.3: A 2D occupancy map of a indoor office corridor built by radar. Using particle filter to relocate robot on the map with radar range measurement (yellow points). The red dots are states (particles).

With the occupancy map built from our radar-inertial pipeline, we can localize our robot using a particle filter when revisiting the same area. For a standard particle filtering approach, we need a motion model to update the particles and a sensor model to measure the likelihood of particles. For the motion model, we used our radar-inertial odometry results to update the states of the particles. To accurately determine the weight distribution of the particles on a 2D occupancy map, we need to project the 3D radar points onto a 2D plane. Additionally, to overcome the sparsity of the radar points, we accumulate the past five consecutive frames of 3D radar points and transform them to the current position before converting them to a 2D laser scan.

However, due to the symmetry of the environment, the sparsity of our radar range measurement, and a lack of details in our radar map, it is difficult to locate the robot without any hint of the initial location. In this demonstration, we assume the initial location is given. As Figure 7.4 shows, localization on the 2D occupancy radar map was successful. Even when encountering a symmetric area on the map, our radar-inertial motion model and radar range measurement were able to recover the location on the map.


(a) Robot at Initialization

(b) Encountering Symmetry

Figure 7.4: Localizing our radar sensor rig in a predefined 2D occupancy radar map.

7.3 Radar-Visual-Inertial Odometry

Visual-Inertial Odometry (VIO) is a staple in modern SLAM systems. This type of odometry is used in conjunction with a factor graph, utilizing fixed-lag smoothing to provide a well-constrained and temporally consistent pose estimate for the robot of interest. However, the VIO system still suffers from performance degradation in low-light situations. To combat this challenge, we demonstrate the use of radar modality to offer more robust state estimates in degraded environments.

7.3.1 Stereo Visual Measurements

For estimating visual landmarks, we use the onboard calibrated and stereo rectified cameras to perform feature extraction and tracking. We follow the front-end pipeline from OpenVINS [10]. Corner features from both cameras will be tracked over time using an optical flow approach. With visual features identified and tracked, we can build a stereo factor between the landmark variable and the corresponding pose variable to minimize repojection error. The k-th landmark location in the left and right camera frame can be represented as follows:

$$\mathbf{L}_k^L = \left[l_x^k, l_y^k, l_z^k\right]^T \quad \mathbf{L}_k^R = \left[l_x^k - b, l_y^k, l_z^k\right]^T \in \mathbb{R}^3 \tag{7.1}$$

When formulating the camera matrix P that projects the landmark position in 3D to the image, we have:

$$P_{\text{world}} = KR\left[\begin{array}{c|c} I & -C \end{array}\right] \qquad P_{\text{camera}} = K\left[\begin{array}{c|c} I & 0 \end{array}\right] \tag{7.2}$$

Since the landmarks are defined to exist in the camera frame, we can transform the landmark from world frame to camera frame.

Then we project the landmark in camera frame to both images:

$$\begin{bmatrix} u_L \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} l_x^k \\ l_y^k \\ l_z^k \end{bmatrix} \qquad \begin{bmatrix} u_R \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} l_x^k - b \\ l_y^k \\ l_z^k \end{bmatrix} \tag{7.3}$$

From the preceding two equations, we can derive the landmark given stereo feature measurement:

$$l_x^k = \frac{l_z^k}{f_x}(u_L - c_x), \quad l_y^k = \frac{l_z^k}{f_y}(v - c_y), \quad l_z^k = \frac{bf_x}{u_L - u_R} \tag{7.4}$$

Here, the left landmarks $\mathbf{L}_k^L$ are estimated as a variable in our system. Then the stereo reprojection error is defined as:

$$\mathbf{r}_{\mathbf{L}_{ik}} = \begin{bmatrix} u_L^{'} - u_L \\ u_R^{'} - u_R \\ v^{'} - v \end{bmatrix} \tag{7.5}$$

where $(u_L^{'}, u_R^{'}, v^{'})$ are the reprojection measurements given estimated landmark and pose, and (u_L, u_R, v) is the landmark measured in pixel space. More details on Jacobian and uncertainty modeling for nonlinear optimization and fixed-lag smoothing can be found in [5].

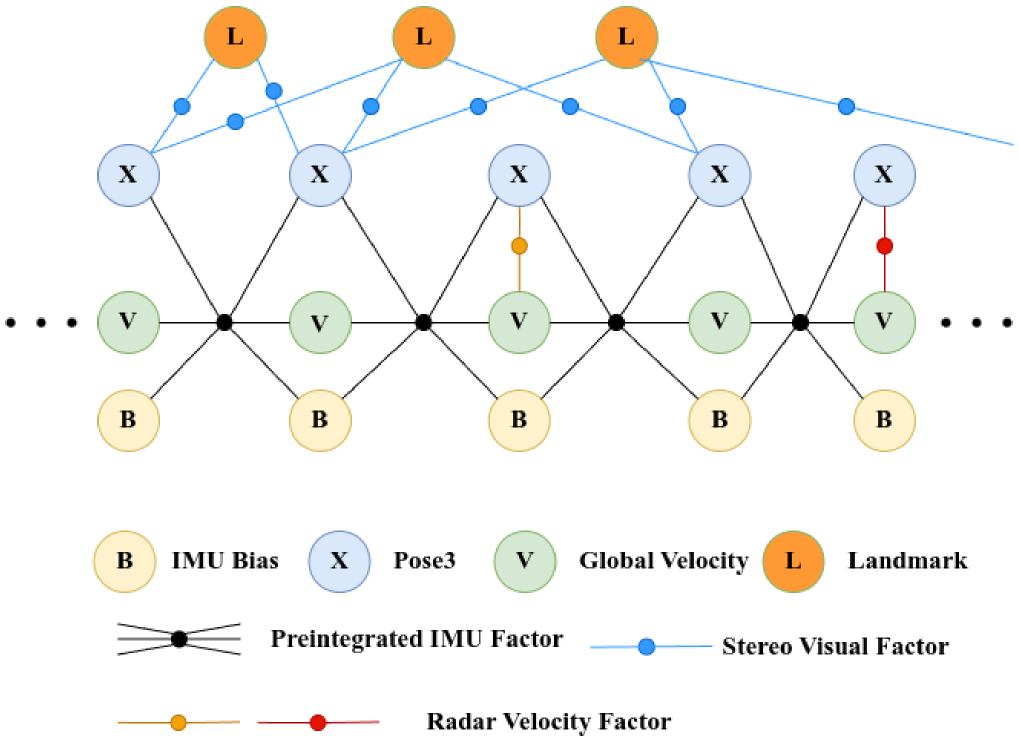


Figure 7.5: Radar-Visual-Inertial Odometry system as a Factor Graph.

7.3.2 Graph Optimization

As all the sensor measurements are inherently not synchronized in time. We interpolate the IMU measurements whenever visual or radar measurements are received to build a keyframe. And we make sure that visual features are always connected to at least two keyframes to better constrain the landmarks and pose variables. A visualization of our state optimization system in a factor graph is shown in Fig. 7.5

Given all the estimated measurements and their uncertainties, the complete system optimizes states and landmarks to minimize their Mahalanobis distance:

$$\mathcal{X}^*, \mathcal{L}^* = \underset{\mathcal{X},\mathcal{L}}{\operatorname{argmin}} \left[\sum_{i,j} \left(||\mathbf{r}_{\Delta \mathbf{I}_{ij}}||^2_{\Sigma_{\Delta \mathbf{I}_{ij}}} + \sum_{k} ||\mathbf{r}_{\mathbf{L}_{ik}}||^2_{\Sigma_{\mathbf{L}_{ik}}} + ||\mathbf{r}_{\mathbf{V}_j^b}||^2_{\Sigma_{\mathbf{v}^b}} \right) \right] \tag{7.6}$$

The optimization problem is solved incrementally using iSAM2 [18] to maintain efficiency.

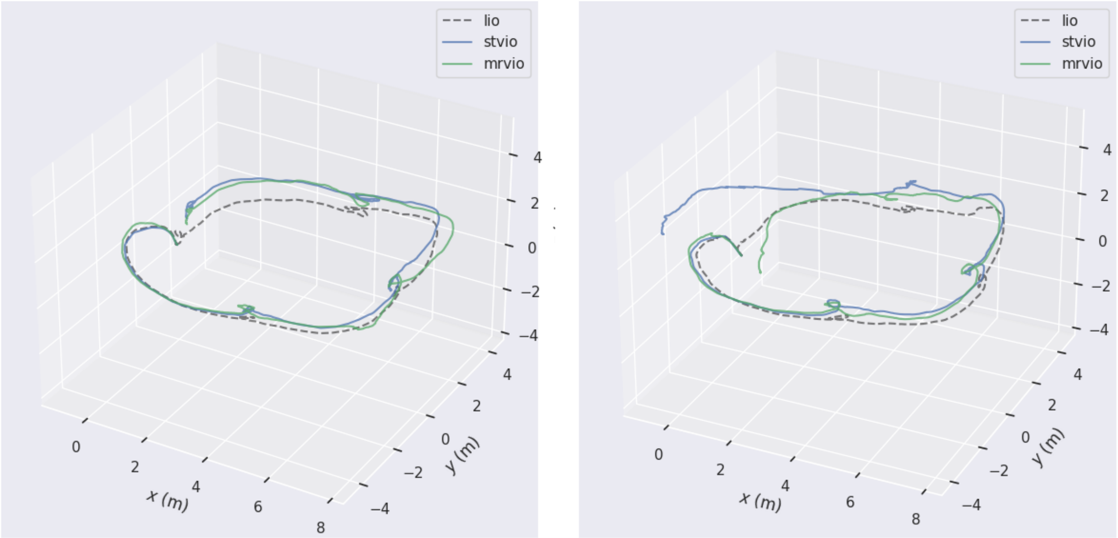
7.3.3 Experiment

To demonstrate the capability of our radar-visual-inertial state estimation system, we ran our system under various lighting conditions, including normal, low, and no lighting conditions. To compare the performance with the visual-inertial-only system [10], we used the odometry from the LiDAR-inertial system [34] as the ground truth. We conducted this experiment in an auditorium classroom, where we took the sensor rig and walked around the room. For each test, we had normal lighting for the first quarter of the trajectory. Then, we switched to low lighting and no lighting conditions accordingly.

In this experiment, our radar-visual-inertial system showed similar performance compared to a state-of-the-art visual-inertial system under normal lighting conditions. When the lighting conditions degraded, the failure in visual feature detection and tracking severely affected the estimation of visual landmarks. While our system is supported by the radar velocity constraint, which is not affected by lighting conditions. Nevertheless, a slight performance degradation also occurred for our radar-visual-inertial system when the lighting conditions changed, introducing some bogus visual landmark measurements.

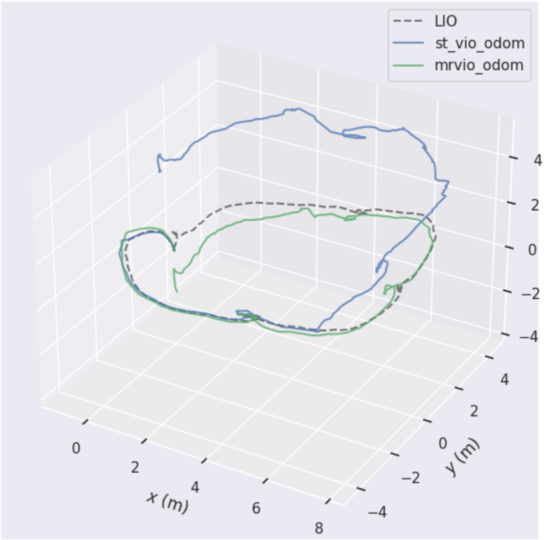
Table 7.1: Absolute and relative translation error in variate lighting condition.

	APE		RPE	
	OpenVINS	RVIO	OpenVINS	RVIO
No Light	1.408	**0.376**	0.163	**0.078**
Low Light	0.859	**0.489**	0.114	**0.068**
Normal Light	0.358	**0.324**	0.083	**0.062**



(a) Normal Light

(b) Low Light



(c) No Light

Figure 7.6: Trajectories in variate lighting conditions

Chapter 8

Conclusions and Future Work

8.1 Conclusion

This thesis presents a system using dual mmWave cascaded imaging radars fused with an IMU sensor capable of achieving high-precision 3D motion state estimation. We provide insights into the limitations of radar measurements, emphasizing the resulting estimated uncertainty and the imperative need to compensate for inaccuracies through the use of multiple radars. We present our radar configurations and fixed-lag optimization solutions, which effectively integrate the radar and IMU measurements. Our method is demonstrated using a real-world 3D motion dataset and downstream tasks including mapping, relocalization, and fusion with visual features.

8.2 Future Work

8.2.1 Radar Noise Prediction

There are several components required to make our radar SLAM system function effectively. For our radar velocity estimation process, robust kernel methods or RANSAC for outlier rejection can effectively filter out radar noise in most cases. However, setting heuristic parameters is necessary to achieve ideal performance. An alternative approach is to train a neural network capable of predicting the variance of radar Doppler velocity.

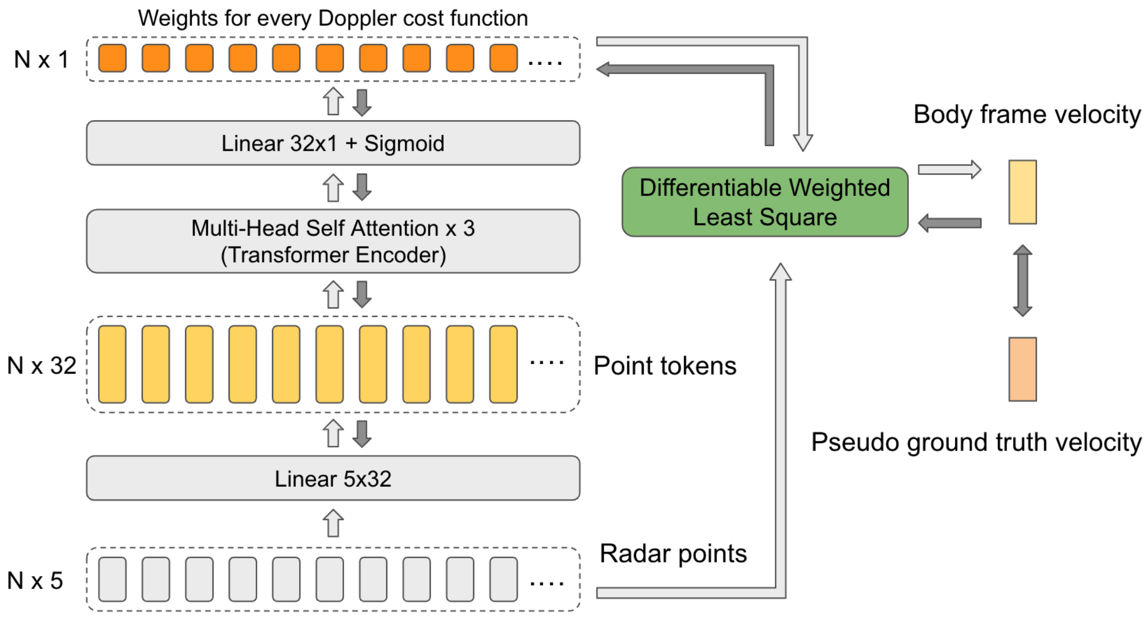
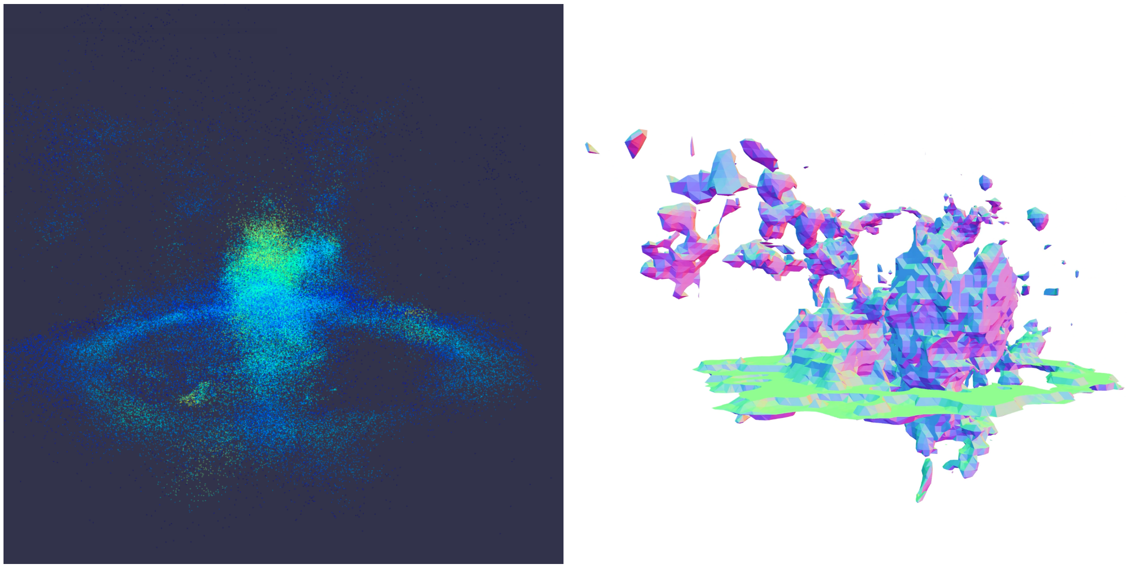


Figure 8.1: Predicting radar noise for body frame velocity optimization. The self-attention-based neural network takes each radar point as a token. Each radar point contained information on 3D position, Doppler velocity, and signal-to-noise ratio.

To predict the variance of Doppler velocity per radar point, which is essential for weighting and whitening the system for body frame velocity optimization, we have devised a pipeline using self-supervised learning with ground truth body frame velocity. Given an arbitrary number of radar points in one frame, we can employ a self-attention-based neural network to output the weighting of each point. With the predicted weightings, we can simply re-weight the Doppler velocity residuals and utilize the least squares approach to optimize a body frame velocity. This process is fully differentiable, enabling the use of a body frame velocity generated from SLAM pipelines using other sensors as pseudo-ground truth to supervise the training. The complete pipeline is illustrated in Figure 8.1.

However, this method can only estimate the relative uncertainty between radar points in the same frame. We were not able to recover the true variance of Doppler velocity measurements or the relative covariance of optimized body frame velocity with what the IMU sensor measured. To incorporate this noise prediction method for our state estimation pipeline, we need a solution to jointly learn the IMU measurement covariance.

8.2.2 Implicit Surface Reconstruction using Imaging Radar



(a) Radar points colored with SNR.

(b) Reconstructed mesh.

Figure 8.2: Radar maps of a metal statue.

3D surface dense reconstruction aims to build a detailed surface of the scene structures by reasoning and merging sensor information from multiple viewpoints. This structural information will be very useful for many downstream tasks, including robot navigation and augmented and virtual reality applications. So far, many sensor modalities have been discussed to achieve this task, including LiDAR [17, 25], camera [23, 30, 37], and recently imaging sonar [33]. In future work, we would like to explore the possibility of using mmWave Imaging Radar to achieve 3D surface dense reconstruction through multi-view volume rendering techniques.

The primary challenge in employing volume rendering lies in developing an accurate mathematical model to translate surface geometry into mmWave Radar signal intensity. This task is notably more intricate than the well-understood and straightforward processes of LiDAR and camera light ray rendering. The complexity of mmWave Radar signal rendering arises from factors such as signal destructive and constructive interference, as well as the behavior of radar signals through multiple surfaces and materials.

An alternative method is to directly learn a neural signed distance function to generate the radar point map. However, due to the level of noise in the radar points, optimizing the zero-level set of this neural signed distance function is very difficult. Figure 8.2 shows a reconstructed mesh generated from a trained neural SDF using the radar point map. In future work, our aim is to develop a solution for dense surface reconstruction using imaging radar information.

Bibliography

[1] Sameer Agarwal, Keir Mierle, and The Ceres Solver Team. Ceres Solver, 10 2023. URL `https://github.com/ceres-solver/ceres-solver`. 2.2

[2] Pasquale Antonante, Vasileios Tzoumas, Heng Yang, and Luca Carlone. Outlier-robust estimation: Hardness, minimally tuned algorithms, and applications. *IEEE Trans. on Robotics (TRO)*, 38(1):281–301, 2021. 2.2

[3] Sarah H Cen and Paul Newman. Radar-only ego-motion estimation in difficult settings via graph matching. In *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 298–304, Montreal, CA, 2019. 1.3

[4] James Coughlan and Alan L Yuille. The Manhattan world assumption: Regularities in scene statistics which enable Bayesian inference. *Advances in Neural Information Processing Systems*, 13, 2000. 3.2

[5] Frank Dellaert and GTSAM Contributors. borglab/gtsam, May 2022. URL `https://github.com/borglab/gtsam)`. 2.2, 5.5, 7.3.1

[6] Frank Dellaert and Michael Kaess. *Factor Graphs for Robot Perception*. 2017. doi: 10.1561/2300000043. 2.1

[7] Christopher Doer and Gert F. Trommer. An EKF based approach to radar inertial odometry. In *Proc. Intl. Conf. on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 152–159, Karlsruhe, DE, September 2020. 1.3, 3.1, 3.2

[8] Christopher Doer and Gert F. Trommer. x-RIO: Radar inertial odometry with multiple radar sensors and yaw aiding. *Gyroscopy and Navigation*, 12:329–339, 02 2022. 1.2, 3.2, 5.1, 6.2

[9] Christian Forster, Luca Carlone, Frank Dellaert, and Davide Scaramuzza. On-manifold preintegration for real-time visual–inertial odometry. *IEEE Trans. on Robotics (TRO)*, 33(1):1–21, 2016. 5.2, 5.2

[10] Patrick Geneva, Kevin Eckenhoff, Woosik Lee, Yulin Yang, and Guoquan Huang. OpenVINS: A research platform for visual-inertial estimation. In *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 4666–4672, Paris, FR,

May 2020. 5.4, 6.2, 7.3.1, 7.3.3

[11] Michael Grupp. evo: Python package for the evaluation of odometry and SLAM. `https://github.com/MichaelGrupp/evo`, 2017. 6.2

[12] Ziyang Hong, Yvan Petillot, and Sen Wang. RadarSLAM: Radar based large-scale SLAM in all weathers. In *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 5164–5170, Las Vegas, USA, September 2020. 1.3

[13] Armin Hornung, Kai M. Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 2013. doi: 10.1007/s10514-012-9321-0. URL `https://octomap.github.io`. Software available at `https://octomap.github.io`. 7.1

[14] Jui-Te Huang, Chen-Lung Lu, Po-Kai Chang, Ching-I Huang, Chao-Chun Hsu, Po-Jui Huang, Hsueh-Cheng Wang, et al. Cross-modal contrastive learning of representations for navigation using lightweight, low-cost millimeter wave radar for adverse environmental conditions. *IEEE Robotics and Automation Letters (RA-L)*, 6(2):3333–3340, 2021. 1.2

[15] Tianshu Huang, John Miller, Akarsh Prabhakara, Tao Jin, Tarana Laroia, Zico Kolter, and Anthony Rowe. Dart: Implicit doppler tomography for radar novel view synthesis. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024. 3.3

[16] Cesar Iovescu and Sandeep Rao. The fundamentals of millimeter wave sensors. *Texas Instruments*, pages 1–8, 2017. 1.2, 2.3.2

[17] Seth Isaacson, Pou-Chun Kung, Mani Ramanagopal, Ram Vasudevan, and Katherine A. Skinner. Loner: Lidar only neural representations for real-time slam. *IEEE Robotics and Automation Letters (RA-L)*, 8(12):8042–8049, 2023. doi: 10.1109/LRA.2023.3324521. 8.2.2

[18] Michael Kaess, Hordur Johannsson, Richard Roberts, Viorela Ila, John J Leonard, and Frank Dellaert. iSAM2: Incremental smoothing and mapping using the bayes tree. *Intl. J. of Robotics Research (IJRR)*, 31(2):216–235, 2012. 2.1, 5.5, 7.3.2

[19] Giseop Kim and Ayoung Kim. Scan context: Egocentric spatial descriptor for place recognition within 3D point cloud map. In *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 4802–4809, Madrid, ES, October 2018. 3.1

[20] Andrew Kramer, Carl Stahoviak, Angel Santamaria-Navarro, Ali-Akbar Agha-Mohammadi, and Christoffer Heckman. Radar-inertial ego-velocity estimation for visually degraded environments. In *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 5739–5746, Paris, FR, May 2020. 1.3, 3.1, 5.1

[21] Andrew Kramer, Kyle Harlow, Christopher Williams, and Christoffer Heckman. ColoRadar: The direct 3d millimeter wave radar dataset. *Intl. J. of Robotics Research (IJRR)*, 41(4):351–360, 2022. 1.2, 3.2

[22] Pou-Chun Kung, Chieh-Chih Wang, and Wen-Chieh Lin. A normal distribution transform-based radar odometry designed for scanning and automotive radars. In *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 14417–14423, Xi'an, CN, May 2021. 1.3

[23] Zhaoshuo Li, Thomas Müller, Alex Evans, Russell H Taylor, Mathias Unberath, Ming-Yu Liu, and Chen-Hsuan Lin. Neuralangelo: High-fidelity neural surface reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 8.2.2

[24] Chris Xiaoxuan Lu, Muhamad Risqi U Saputra, Peijun Zhao, Yasin Almalioglu, Pedro PB De Gusmao, Changhao Chen, Ke Sun, Niki Trigoni, and Andrew Markham. milliEgo: single-chip mmWave radar aided egomotion estimation via deep sensor fusion. In *Proc. ACM Conf. on Embedded Networked Sensor Systems*, pages 109–122, Yokohama, JP, November 2020. 1.2, 3.1

[25] Baorui Ma, Yu-Shen Liu, and Zhizhong Han. Learning signed distance functions from noisy 3d point clouds via noise to noise mapping. In *International Conference on Machine Learning (ICML)*, 2023. 8.2.2

[26] Jan Michalczyk, Roland Jung, and Stephan Weiss. Tightly-coupled EKF-based radar-inertial odometry. In *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 12336–12343, Kyoto, JP, October 2022. 3.1

[27] Jan Michalczyk, Roland Jung, Christian Brommer, and Stephan Weiss. Multi-state tightly-coupled EKF-based radar-inertial odometry with persistent landmarks. In *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 4011–4017, London, UK, May 2023. 3.1

[28] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *Proc. Eur. Conf. on Computer Vision (ECCV)*, 2020. 3.3

[29] Ajay Narasimha Mopidevi, Kyle Harlow, and Christoffer Heckman. Rmap: Millimeter-wave radar mapping through volumetric upsampling. *arXiv preprint arXiv:2310.13188*, 2023. 3.3

[30] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, July 2022. doi: 10.1145/3528223.3530127. URL `https://doi.org/10.1145/3528223.3530127`. 8.2.2

[31] Yin Zhi Ng, Benjamin Choi, Robby Tan, and Lionel Heng. Continuous-time

radar-inertial odometry for automotive radars. In *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 323–330, Prague, CZ, September 2021. 3.2

[32] Yeong Sang Park, Young-Sik Shin, Joowan Kim, and Ayoung Kim. 3D ego-motion estimation using low-cost mmWave radars via radar velocity factor for pose-graph SLAM. *IEEE Robotics and Automation Letters (RA-L)*, 6(4): 7691–7698, 2021. 1.2, 1.3, 3.1, 3.2, 4.2

[33] Mohamad Qadri, Michael Kaess, and Ioannis Gkioulekas. Neural implicit surface reconstruction using imaging sonar. In *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 1040–1047. IEEE, 2023. 8.2.2

[34] Tixiao Shan, Brendan Englot, Drew Meyers, Wei Wang, Carlo Ratti, and Rus Daniela. LIO-SAM: Tightly-coupled lidar inertial odometry via smoothing and mapping. In *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 5135–5142. IEEE, 2020. 7.1, 7.3.3

[35] Inc Texas Instruments. Imaging radar using cascaded mmwave sensor reference design. 2020. 1.3, 2.3.1, 4.2

[36] Arthur Venon, Yohan Dupuis, Pascal Vasseur, and Pierre Merriaux. Millimeter wave FMCW radars for perception, recognition and localization in automotive applications: A survey. *IEEE Trans. on Intelligent Vehicles (TIV)*, 7(3):533–555, 2022. 1.2

[37] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021. 8.2.2

[38] Yizhou Wang, Jenq-Neng Hwang, Gaoang Wang, Hui Liu, Kwang-Ju Kim, Hung-Min Hsu, Jiarui Cai, Haotian Zhang, Zhongyu Jiang, and Renshu Gu. ROD2021 challenge: A summary for radar object detection challenge for autonomous driving applications. In *Proc. ACM Intl. Conf. on Multimedia Retrieval (ICMR)*, pages 553–559, Taipei, TW, November 2021. 1.2

[39] Yizhou Wang, Zhongyu Jiang, Xiangyu Gao, Jenq-Neng Hwang, Guanbin Xing, and Hui Liu. RODNet: Radar object detection using cross-modal supervision. In *Proc. IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 504–513, Waikoloa, USA, January 2021. 1.2

[40] Oliver J Woodman. An introduction to inertial navigation. Technical report, University of Cambridge, Computer Laboratory, 2007. 6.1

[41] Ruoyang Xu, Wei Dong, Akash Sharma, and Michael Kaess. Learned depth estimation of 3d imaging radar for indoor mapping. In *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 13260–13267. IEEE, 2022. 3.3

[42] Shibo Zhao, Hengrui Zhang, Peng Wang, Lucas Nogueira, and Sebastian Scherer. Super odometry: IMU-centric LiDAR-visual-inertial estimator for challenging environments. In *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 8729–8736, Prague, CZ, September 2021. 1.1

[43] Yuan Zhuang, Binliang Wang, Jianzhu Huai, and Miao Li. 4D iRIOM: 4D imaging radar inertial odometry and mapping. *IEEE Robotics and Automation Letters (RA-L)*, 8(6):3246–3253, 2023. 3.1, 5.1