Push the Limit of Single-Chip mmWave Radar-Based Egomotion Estimation with Moving Objects in FoV

Rong Ding
Shanghai Jiao Tong University
dingrong@sjtu.edu.cn

Haiming Jin*
Shanghai Jiao Tong University
jinhaiming@sjtu.edu.cn

Jianrong Ding
Shanghai Jiao Tong University
rafaelding@sjtu.edu.cn

Xiaocheng Wang
Shanghai Jiao Tong University
curryjam_cg@sjtu.edu.cn

Guiyun Fan
Shanghai Jiao Tong University
fgy726@sjtu.edu.cn

Fengyuan Zhu
Shanghai Jiao Tong University
jsqdzhufengyuan@sjtu.edu.cn

Xiaohua Tian
Shanghai Jiao Tong University
xtian@sjtu.edu.cn

Linghe Kong
Shanghai Jiao Tong University
linghe.kong@sjtu.edu.cn

ABSTRACT

This paper presents EmoRI, a novel single-chip mmWave radar-based egomotion estimation approach that works in challenging scenarios where moving objects exist in radar's Field of View (FoV). Essentially, estimating a mobile platform's egomotion using an on-board mmWave radar requires inferring the relative motion between radar and the points of the stationary objects (PSOs) in the radar point cloud. However, in practice, there could be no PSOs because of the blockage of moving objects. Even if PSOs exist, precisely identifying them is still challenging due to (i) the large quantity of points generated by the moving objects, and (ii) the huge angle estimation errors of the conventional point cloud generation algorithm. We empower EmoRI to overcome the above challenges incurred by moving objects with three core techniques, which include (i) a hybrid FFT-MUSIC algorithm that improves the angle estimation accuracy of single-chip mmWave radar, (ii) a multiple stationary target consensus algorithm that precisely selects the PSOs from the radar point cloud, and (iii) a simultaneous fusion and calibration mechanism that introduces an IMU as the auxiliary sensor, meticulously calibrates IMU accelerations with radar measurements, and complimentarily fuses these two modalities to obtain the 6-DoF egomotion. Our extensive experiments validate that EmoRI pushes the limit of single-chip mmWave radar-based egomotion estimation with moving objects in radar's FoV by **reducing the per-meter destination error from decimeter to centimeter level**.

CCS CONCEPTS

• **Computer systems organization → Robotic autonomy**; • **Networks → Location based services**.

*Haiming Jin is the corresponding author.

SenSys '23, November 12–17, 2023, Istanbul, Turkiye

ACM ISBN 979-8-4007-0414-7/23/11…$15.00
https://doi.org/10.1145/3625687.3625795

KEYWORDS

Wireless Sensing; Egomotion Estimation; mmWave Radar

ACM Reference Format:
Rong Ding, Haiming Jin, Jianrong Ding, Xiaocheng Wang, Guiyun Fan, Fengyuan Zhu, Xiaohua Tian, and Linghe Kong. 2023. Push the Limit of Single-Chip mmWave Radar-Based Egomotion Estimation with Moving Objects in FoV. In *The 21st ACM Conference on Embedded Networked Sensor Systems (SenSys '23), November 12–17, 2023, Istanbul, Turkiye.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3625687.3625795

1 INTRODUCTION

Recent years have witnessed the prevalence of a wide spectrum of intelligent mobile platforms that pervade almost every corner of our daily lives, such as delivery robots that provide last-mile delivery services for food, groceries, and packages, automobiles that offer enhanced driving safety and experience with advanced driver-assistance systems, as well as UAVs that cruise across various altitudes for power line inspection, crop monitoring, damage identification after natural disasters, and many other tasks.

Among the many technologies empowering a mobile platform, one can never ignore the *6-degree-of-freedom (6-DoF) egomotion estimation*, which enables a mobile platform to estimate its own 3-DoF translation and 3-DoF rotation. Delivery robots rely on estimated egomotion to navigate to designated indoor places without GPS [1]. Even in outdoor environments, automobiles have to rely on egomotion estimation for localization in urban canyons and under elevated bridges where GPS signals are usually blocked [2]. Vehicular AR devices (e.g., AR HUD [3–5]) use estimated egomotion of the automobile to properly project navigation information to the driver's view. UAVs use estimated egomotion as an input to the pose adjustment and path planning modules [6].

Thus far, a series of methods have been proposed to estimate the egomotion of a mobile platform. Among them, some methods utilize external peripherals (e.g., WiFi APs [6]), which unfortunately only work in the restricted regions where the external peripherals locate. In contrast, the methods that only leverage the sensors on the mobile platforms (e.g., 3D LiDAR, camera, mmWave radar) are free of this problem. However, the 3D LiDAR-based methods [7–11] usually fail in adverse weathers (e.g., rain, fog, snow), and 3D LiDARs (around 1.1kg, $1600 each [12]) are usually prohibitively

heavy and expensive to be deployed on lightweight and low-cost mobile platforms. The camera-based methods [13–17] oftentimes fail in overly weak or strong illumination, as well as in environments with insufficient landmarks.

Among all the sensors commonly equipped on mobile platforms, single-chip mmWave radars (around 0.02kg, $200 each [18]) enjoy the key advantages of light weight, low cost, and robustness against weather and illumination, which facilitates their increasing deployment [19–21] on mobile platforms. We thus leverage a single-chip mmWave radar as the primary sensor for egomotion estimation.

Intuitively, egomotion estimation with a mmWave radar essentially seeks to obtain the radar's relative motion to the *points of the stationary objects (PSOs)* in the radar point cloud. In practice, there oftentimes exist a wide spectrum of *moving objects* (e.g., human, bicycle, automobile) in radar's FoV, which makes the radar point cloud a mixture of points of both the stationary and moving objects. Thus, correctly selecting the PSOs from the radar point cloud is a fundamental step towards radar-based egomotion estimation in scenarios with moving objects in radar's FoV.

However, the dominant PSO selection algorithm, namely point-level RANSAC (PL-RANSAC) [22], usually fails to perform satisfactorily in such scenarios. The reason is that PL-RANSAC only utilizes the point-level consensus of the PSOs, and simply chooses the largest set of points with the same relative velocity to the radar as the PSOs. However, we discover that those points could oftentimes come from the moving objects, especially when the moving objects are (i) nearby the radar, (ii) in the center of radar's FoV, or (iii) occupying the majority of radar's FoV. Thus, we devise the *multiple stationary target consensus (MSTC)* algorithm, which properly selects the PSOs by *identifying the object-level consensus of the stationary objects*. Specifically, MSTC selects one representative point from each object in the environment, and uses the largest set of representative points with the same relative velocity to the radar as the PSOs. Such design makes MSTC utilize only one point from each object for PSO selection, and thus avoids the negative effect of the large quantity of points of moving objects.

As MSTC leverages the radar point cloud as its input, its performance will inevitably be affected by the quality of the input point cloud. Unfortunately, the points generated by a single-chip mmWave radar that utilizes the conventional angle-FFT algorithm usually have large angle estimation errors. To address this problem, we propose the *hybrid FFT-MUSIC (HFM)* point cloud generation algorithm, whose rationale is to leverage the super-resolution MUSIC algorithm [23] to estimate the angles of the points in the point cloud. Specifically, as MUSIC achieves lower angle estimation errors when taking in longer signal vectors as its input, HFM *constructs the longest possible signal vectors by meticulously leveraging all the virtual antennas of the radar.*

Based on the PSOs selected from the HFM-augmented radar point cloud, MSTC goes one step further and estimates the radar egovelocity, which can be integrated over time to estimate the 3-DoF translation. In practice, however, the blockage of moving objects would make the stationary objects in radar's FoV undesirably scarce. In such cases, MSTC cannot perform PSO selection, as well as radar egovelocity estimation. Thus, we propose to utilize an inertial measurement unit (IMU) as the auxiliary sensor, and utilize its acceleration measurements to estimate the translation when the radar egovelocity is unavailable. However, the IMU acceleration measurements are highly drifted due to the notorious gravity pollution problem [24], and directly utilizing them for translation estimation would incur large errors.

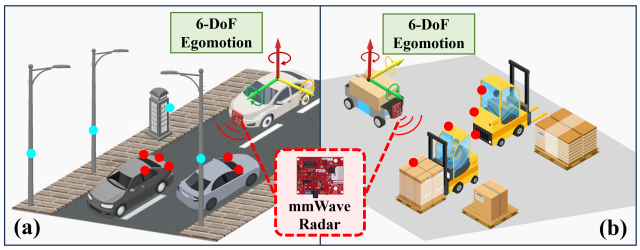


Figure 1: EmoRI is a single-chip mmWave radar-based egomotion estimation paradigm that works in challenging scenarios where (a) the large quantity of points of moving objects make it hard to identify PSOs, and (b) there are even no PSOs in the radar point cloud. The blue and red points indicate the PSOs and the points of moving objects, respectively.

To solve this problem, we propose the *simultaneous fusion and calibration (SFC)* mechanism, which is the first radar-IMU fusion mechanism that *leverages the reliable radar egovelocity estimations to continuously calibrate the highly-drifted IMU acceleration measurements.* SFC utilizes the calibrated IMU acceleration measurements to estimate the 3-DoF translation when the radar egovelocity is unavailable. Besides, SFC also leverages the IMU angular velocity measurements to estimate the 3-DoF rotation. As such, SFC generates the reliable and full 6-DoF egomotion estimation.

Overall, we fuse the above components including HFM, MSTC, and SFC into an integrated framework, namely *EmoRI*[1], which generates reliable 6-DoF egomotion estimation of a mobile platform with moving objects in radar's FoV. In summary, the contributions of this paper are as follows.

- We propose EmoRI, a novel 6-DoF egomotion estimation method that pushes the limit of single-chip mmWave radar-based egomotion estimation with moving objects in radar's FoV by **reducing the per-meter destination error from decimeter level to centimeter level**.
- The proposed method EmoRI consists of a series of novel techniques, including (i) the HFM algorithm that distinctly improves the angle estimation accuracy of single-chip mmWave radar through MUSIC-augmented estimation with dedicated signal vector construction, (ii) the MSTC algorithm that properly selects only the PSOs from the radar point cloud though object-level consensus identification, and (iii) the SFC mechanism that generates reliable and full 6-DoF egomotion estimation through radar-aided IMU calibration and complementary radar-IMU fusion.
- Extensive experiments are conducted to evaluate EmoRI with three types of mobile platforms, including a mobile robot, an automobile, and a UAV. Specifically, we test EmoRI (i) with the mobile robot platform moving along 64 trajectories (15m to 30m each) in both indoor and outdoor environments (i.e, corridor, hall, parking lot) with different numbers and types of moving objects (i.e., human, bicycle, and automobile), (ii) with the automobile

[1]The name EmoRI comes from Egomotion estimation with moving objects in radar's FoV utilizing mmWave Radar and IMU.

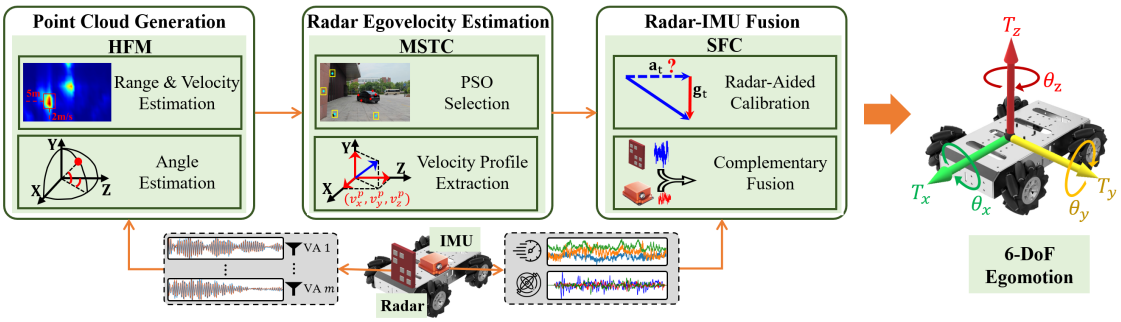


Figure 2: Overview of EmoRI.

platform moving along 20 trajectories (270m to 500m each) in real-world road environments, and (iii) with the UAV platform moving along 14 trajectories (20 minutes in total) with diverse shapes. The experimental results show that EmoRI achieves an average of over 59% reduction on the absolute trajectory error and over 74% reduction on the destination error compared with state-of-the-art methods [25–27].

2 PROBLEM STATEMENT

In this paper, we aim to address the problem of *estimating the 6-DoF egomotion of a mobile platform*, namely enabling a mobile platform to estimate its own 6-DoF motion. Suppose the time period of the problem is from t to $t+H\Delta t$, where t is the start time, Δt is the time interval between two consecutive egomotion estimation outputs[2], and H is an integer that determines the problem's time horizon. The output of the problem is a sequence of real-time egomotion estimation[3] $\{\mathbf{y}_i\}_{i\in[H]}$. Specifically, $\mathbf{y}_i = (T_i^x, T_i^y, T_i^z, \theta_i^x, \theta_i^y, \theta_i^z)$ is the estimated egomotion from time $t+(i-1)\Delta t$ to $t+i\Delta t$, where T_i^x, T_i^y, and T_i^z denote the mobile platform's 3-DoF translation along the X, Y, and Z axis, and θ_i^x, θ_i^y, and θ_i^z denote the mobile platform's 3-DoF rotation around the X, Y, and Z axis, in the mobile platform's local reference frame at time $t+(i-1)\Delta t$. Note that one can obtain the global odometry pose by stitching such egomotion sequence.

As discussed in Sec. 1, the moving objects in radar's FoV pose great challenges to egomotion estimation with single-chip mmWave radar. As a result, we seek to break such curse of moving objects for reliable egomotion estimation.

3 SYSTEM OVERVIEW

In this section, we provide an overview of *EmoRI*. The workflow of EmoRI is illustrated in Fig. 2.

- **Point Cloud Generation.** Upon obtaining the beat frequency signals from the mmWave radar, EmoRI generates the radar point cloud through our proposed *hybrid FFT-MUSIC (HFM)* algorithm in the point cloud generation module. Compared with the conventional point cloud generation approach adopted in many existing works [28–30], HFM has a better angle estimation ability, and thus yields more accurate point cloud.
- **Radar Egovelocity Estimation.** After that, the PSOs are selected from the radar point cloud by our proposed *multiple stationary target consensus (MSTC)* algorithm in the radar egovelocity estimation module. Intuitively, the radar egomotion is inferred by its relative motion against the PSOs. MSTC is designed to select only the PSOs from the point cloud, which we find the conventional PL-RANSAC algorithm [31] usually fails to accomplish. The common velocity profile of the PSOs is then extracted and utilized to estimate the radar egovelocity, which is a vital input for the final 3-DoF translation estimation.
- **Radar-IMU Fusion.** Finally, the radar-IMU fusion module introduces an IMU as the auxiliary sensor, meticulously calibrates the IMU accelerations with radar egovelocity, and complementarily fuses radar egovelocity with the measurements of IMU through the *simultaneous fusion and calibration (SFC)* mechanism. SFC enables EmoRI to yield reliable and full 6-DoF egomotion estimation in scenarios when even no PSOs exist in the point cloud.

By integrating the three modules, EmoRI achieves the fundamental objective of robust egomotion estimation against moving objects in radar's FoV.

4 DESIGN DETAILS

4.1 Point Cloud Generation

4.1.1 Challenge: Large Angle Estimation Error. The numbers of TXs and RXs of single-chip mmWave radars [18, 32, 33] are usually both no larger than 4, which makes their angle estimation errors range from roughly 7.5° to 30° when the conventional angle-FFT algorithm is applied. As we will demonstrate in Sec. 5.6.1, such angle estimation errors can incur large egomotion estimation errors. Thus, we propose the HFM algorithm introduced in Sec. 4.1.2 to improve the angle estimation accuracy of single-chip mmWave radar.

4.1.2 Hybrid FFT-MUSIC. The HFM algorithm utilizes FFT for range and radial velocity estimation, and utilizes the super-resolution MUSIC algorithm [23] for azimuth and elevation angles estimation. The workflow of HFM is as follows.

HFM first generates the range-doppler matrices (RDMs) and estimates the range and the radial velocity of each point in the radar point cloud by applying the range- and doppler-FFT on the beat frequency signals output by the radar. After that, HFM estimates the azimuth and elevation angles of each point, where resides the core innovations of HFM compared with the conventional point cloud generation approach. Specifically, the innovations include (i) adopting MUSIC to estimate the azimuth and elevation angles, and

[2]In our implementation, Δt = 100ms, which is comparable to the time granularity of 3D LiDAR-based egomotion estimation approaches [7–11].
[3]We let $[A]$ denote the set $\{1, 2, \cdots, A\}$ for any positive integer A.

(ii) leveraging the peaks in RDMs to construct the longest possible signal vectors as the input of MUSIC, which enhances its angle estimation accuracy. HFM's signal vector construction procedure consists of the following two steps.

- **Step 1: Peak Extraction.** Given the virtual antenna (VA) layout of the radar consisting of m VAs, HFM first generates one RDM for each VA, and then extracts all the peaks in the m RDMs via the CFAR algorithm [34]. Each group of the m peaks located at the same position of each RDM are utilized to estimate the azimuth and elevation angles of one point in the point cloud. We denote Φ_j as the set of the peaks to estimate the angles of the point P_j.
- **Step 2: Signal Vector Synthesis.** HFM synthesizes the azimuth signal vector ϕ_a^j and elevation signal vector ϕ_e^j from Φ_j, and uses ϕ_a^j and ϕ_e^j as inputs to MUSIC to estimate the azimuth and elevation angles of P_j, respectively. To synthesize ϕ_a^j, HFM first records the distance between each pair of horizontally aligned VAs, and then extracts the mode of these distances, which we denote as d. Any two elements in Φ_j generated from two horizontally aligned VAs with distance d have the same theoretical phase offset, which we denote as ψ_j. As we illustrate in Fig. 3, the elements of ϕ_a^j are synthesized to be the linear combinations of the elements in Φ_j, such that every two consecutive elements in ϕ_a^j have the same theoretical phase offset ψ_j. The synthesis of ϕ_e^j is similar to that of ϕ_a^j by replacing the horizontally aligned VAs with the vertically aligned VAs. Since MUSIC can achieve lower angle estimation errors when taking in longer azimuth and elevation signal vectors, HFM synthesizes the longest possible ϕ_a^j and ϕ_e^j.

Take the TI IWR6843AOP mmWave radar [18] used in our experiments as an example. The layout of its VAs is shown in Fig. 4. Let $\Delta\phi_{i,k}^j = \phi_i^j - \phi_k^j$. The azimuth signal vector ϕ_a^j and elevation signal vector ϕ_e^j synthesized by HFM are

$$\phi_a^j = (\phi_{11}^j, \phi_9^j, \phi_7^j, \phi_5^j, \phi_5^j + \Delta\phi_{10,12}^j, \phi_5^j + \Delta\phi_{8,12}^j, \phi_5^j + \Delta\phi_{6,12}^j, \phi_5^j + \Delta\phi_{6,12}^j + \Delta\phi_{1,3}^j, \phi_5^j + \Delta\phi_{6,12}^j + \Delta\phi_{1,3}^j + \Delta\phi_{2,4}^j),$$
$$\phi_e^j = (\phi_9^j, \phi_{10}^j, \phi_1^j, \phi_2^j, \phi_2^j + \Delta\phi_{12,11}^j, \phi_2^j + \Delta\phi_{3,11}^j, \phi_2^j + \Delta\phi_{4,11}^j, \phi_2^j + \Delta\phi_{4,11}^j + \Delta\phi_{8,7}^j, \phi_2^j + \Delta\phi_{4,11}^j + \Delta\phi_{8,7}^j + \Delta\phi_{6,5}^j).$$

Note that the calculation of the RDMs in HFM naturally separates multipath signals into different peaks in different range bins. Consequently, every time MUSIC is called by HFM, it is leveraged to estimate the azimuth or elevation angles of only one signal propagation path.

4.1.3 Evaluation of HFM. We evaluate the performance of the angle estimation accuracy of HFM by measuring the azimuth and elevation angles of a corner reflector, as illustrated in Fig. 4. We set both the azimuth and elevation angles of the corner reflector to span evenly from 0° to 50° in a 5° increment. We compare the angle estimation absolute errors of HFM with the conventional angle-FFT algorithm, and a variant of HFM that constructs the signal vectors following the conventional approach that only utilizes horizontally or vertically aligned VAs[4] (HFM-Conv). As illustrated in Fig. 5, the angle estimation errors of HFM are smaller than those of HFM-Conv and angle-FFT in most of the azimuth and elevation angles. Specifically, the angle estimation mean absolute error (MAE) of HFM for azimuth and elevation angles are respectively 1.18° and 1.91°, while those of HFM-Conv are 2.01° and 2.75°, and those of angle-FFT are 11.2° and 8.9°. The respective over 30% and 78% reduction in MAE compared with HFM-Conv and angle-FFT indicate that HFM enables more accurate angle estimation and consequently generates more accurate point cloud.

[4]For the TI IWR6843AOP mmWave radar in our experiments, this method constructs the azimuth signal vector with (VA11, VA9, VA7, VA5), and the elevation signal vector with (VA9, VA10, VA1, VA2).

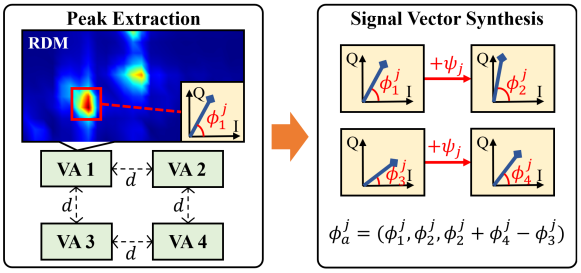


Figure 3: HFM's azimuth signal vector construction procedure for a mmWave radar with 4 VAs in a rectangular layout.

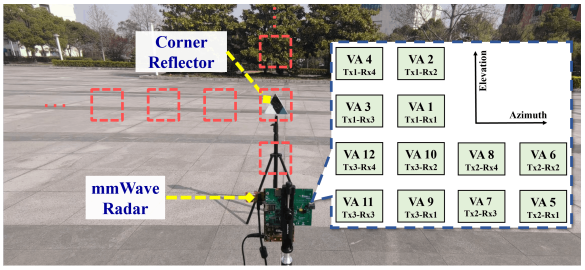


Figure 4: Experiments for HFM with TI IWR6843AOP mmWave radar and a corner reflector. The red dashed rectangles illustrate the positions of the corner reflector.

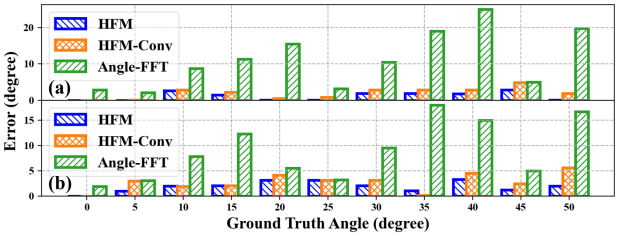


Figure 5: Angle estimation absolute errors of HFM, HFM-Conv, and the angle-FFT-based approach, measured under different (a) azimuth and (b) elevation angles.

4.2 Radar Egovelocity Estimation

4.2.1 Challenge: PSO Selection with Moving Objects in Radar's FoV. As EmoRI leverages the relative motion between the radar and the PSOs to estimate the radar egovelocity, PSO selection is thus a critical step for egomotion estimation. However, properly selecting PSOs from the radar point cloud is anything but trivial. The intuitive PSO selection method that directly rejects the points whose doppler velocities are not zero could be problematic, because the doppler velocities of the points are jointly influenced by the motion of the detected objects and the radar. Consequently, dedicated algorithm design is indispensable for PSO detection. The most popular

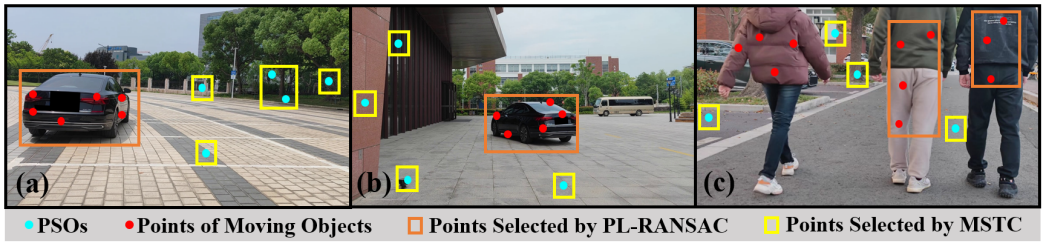


Figure 6: Experiments for PSO selection with moving objects (a) nearby the radar, (b) in the center of radar's FoV, and (c) occupying the majority of radar's FoV.

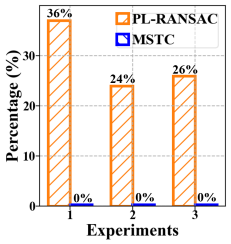


Figure 7: Percentage of incorrect PSO selections.

PSO selection method is PL-RANSAC [22], which selects PSOs by utilizing their point-level consensus. Specifically, PL-RANSAC selects from the point cloud the largest set of points with a common *velocity profile* as the PSOs. Each tuple (v_x, v_y, v_z) that satisfies Eq. (1) is a velocity profile of the point P_i, i.e.,

$$v_i = \cos\theta_{xi} v_x + \cos\theta_{yi} v_y + \cos\theta_{zi} v_z, \tag{1}$$

where v_i is the radial velocity of P_i, and θ_{xi}, θ_{yi}, and θ_{zi} are the angles between $\overrightarrow{OP_i}$ and the X, Y, and Z axis of the radar reference frame with O denoting its origin. Intuitively, each velocity profile of the point P_i is a tuple of the possible 3D velocity of P_i relative to the radar, given the constraint that its projection on $\overrightarrow{OP_i}$ equals to the doppler velocity measured by the radar.

Though widely applied, we discover that PL-RANSAC oftentimes could oftentimes fail, especially when the moving objects are (i) nearby the radar, (ii) in the center of radar's FoV, or (iii) occupying the majority of radar's FoV. In these cases, the strong mmWave signals reflected by moving objects could generate a large quantity of points with the same velocity profile. These points could even be more than the PSOs, which eventually leads to the failure of PL-RANSAC.

We illustrate the failure of PL-RANSAC by conducting the following three experiments, where we attach a mmWave radar on a moving robot. As illustrated in Fig. 6, in the first and second experiments, an automobile is driving nearby the radar and in the center of radar's FoV, respectively. In the third experiment, three humans are walking side by side and occupy the majority of radar's FoV. Apart from the automobile and the humans, other objects in the environments are all stationary. As illustrated in Fig. 6, PL-RANSAC falsely selects the points of the moving object as the PSOs in all the three experiments. In the first experiment, all the 6 points of the automobile have a common velocity profile[5] and are selected by PL-RANSAC, whereas there are only 5 PSOs. Similarly, in the second and third experiments, the moving automobile and walking humans contribute 5 points and 7 points with common velocity profile, while there are both only 4 PSOs, respectively.

4.2.2 Multiple Stationary Target Consensus. To address the challenge of PSO selection with moving objects in radar's FoV, we propose the MSTC algorithm elaborated in Algorithm 1. The key intuition of MSTC is to *identify the object-level consensus of stationary objects* for precise PSO selection. Specifically, MSTC selects one representative point for each object in the environment, and then uses the largest set of representative points with a common velocity profile as the PSOs. Such design utilizes the fact that all the representative points of stationary objects have the same relative velocity to the radar, and hence they have a common velocity profile. Besides, our design of utilizing only one point from each object for PSO selection prevents MSTC from being affected by the large quantity of points of moving objects.

Algorithm 1: Multiple Stationary Target Consensus

Input: Radar point cloud $\mathcal{P}$, number of candidate sets R;
Output: Estimated radar egovelocity $\mathbf{v}^r$;
/* PSO selection */
1 $\mathcal{P}^s \leftarrow \emptyset$, $\mathcal{Q} \leftarrow \emptyset$;
2 $\mathcal{P}_1, \mathcal{P}_2, \cdots, \mathcal{P}_L \leftarrow$ DBSCAN($\mathcal{P}$);
3 **foreach** $l \in [L]$ **do**
4 Randomly select one point P_l from $\mathcal{P}_l$;
5 $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{P_l\}$;
6 **foreach** *iteration* $r \in [R]$ **do**
7 Randomly select three points P_i, P_j, and P_k from $\mathcal{Q}$;
8 Calculate the common velocity profile (v'_x, v'_y, v'_z) of P_i, P_j, and P_k according to Eq. (2);
9 $\mathcal{Q}_r \leftarrow \emptyset$;
10 **foreach** $P_m \in \mathcal{Q}$ **do**
11 **if** $\cos\theta_{xm} v'_x + \cos\theta_{ym} v'_y + \cos\theta_{zm} v'_z = v_m$ **then**
12 $\mathcal{Q}_r \leftarrow \mathcal{Q}_r \cup \{P_m\}$;
13 $\mathcal{P}^s \leftarrow$ The largest set in $\{\mathcal{Q}_r\}_{r\in[R]}$;
/* Velocity profile extraction */
14 Randomly select three points P_i, P_j, and P_k from $\mathcal{P}^s$;
15 Calculate the common velocity profile (v^p_x, v^p_y, v^p_z) of P_i, P_j, and P_k according to Eq. (2);
16 $\mathbf{v}^r \leftarrow (-v^p_x, -v^p_y, -v^p_z)$;

As described in Algorithm 1, the detailed procedure of MSTC is as follows. MSTC takes the radar point cloud $\mathcal{P}$ generated by HFM, and the number of candidate sets R as the inputs, and outputs the radar egovelocity $\mathbf{v}^r$. The algorithm contains two parts including PSO selection (line 1-13) and velocity profile extraction (line 14-16). After initializing the set of PSOs $\mathcal{P}^s$ and the set of representative points $\mathcal{Q}$ as empty sets (line 1), the algorithm divides $\mathcal{P}$ into multiple sets by the DBSCAN algorithm [35] (line 2). As each set of points output by DBSCAN have similar positions and velocities, MSTC treats them as coming from the same object in the environment. Then, the algorithm randomly selects one point from each set generated by DBSCAN as the representative point, and adds the representative point into $\mathcal{Q}$ (line 4-5).

Next, the algorithm searches for the largest set of representative points with a common velocity profile (line 6-13) in an iterative process. Specifically, in the r-th iteration, the algorithm randomly

[5]The elements of such velocity profile are actually the components of the moving object's velocity on the X, Y, and Z axis of the radar reference frame.

selects three representative points P_i, P_j, and P_k from $\mathcal{Q}$, and calculates their common velocity profile (v'_x, v'_y, v'_z), according to the following Eq. (2) (line 7-8), i.e.,

$$\begin{bmatrix} v'_x \\ v'_y \\ v'_z \end{bmatrix} = \begin{bmatrix} \cos\theta_{xi} & \cos\theta_{yi} & \cos\theta_{zi} \\ \cos\theta_{xj} & \cos\theta_{yj} & \cos\theta_{zj} \\ \cos\theta_{xk} & \cos\theta_{yk} & \cos\theta_{zk} \end{bmatrix}^{-1} \begin{bmatrix} v_i \\ v_j \\ v_k \end{bmatrix}. \quad (2)$$

After that, the algorithm collects the representative points that have a common velocity profile (v'_x, v'_y, v'_z) in set $\mathcal{Q}_r$ (line 9-12). The algorithm outputs the largest set in $\{\mathcal{Q}_r\}_{r\in[R]}$ as the set of PSOs $\mathcal{P}^s$ (line 13). After that, the algorithm extracts the common velocity profile (v^p_x, v^p_y, v^p_z) of the points in $\mathcal{P}^s$ (line 14-15). Finally, the algorithm sets the estimated radar egovelocity in the radar reference frame as $\mathbf{v}^r = (-v^p_x, -v^p_y, -v^p_z)$ (line 16).

4.2.3 Evaluation of MSTC. We plot in Fig. 6 the PSOs selected by MSTC in one radar frame for the three experiments described in Sec. 4.2.1. Clearly, MSTC correctly outputs the PSOs. We also compare the percentage of radar frames in the whole process of the three experiments where PL-RANSAC and MSTC incorrectly select PSOs. As illustrated in Fig. 7, PL-RANSAC makes incorrect PSO selection in 36%, 24%, and 26% frames in the first, second, and third experiment, respectively, where each experiment contains 300 radar frames. However, those numbers for MSTC are all 0%.

4.3 Radar-IMU Fusion

4.3.1 Motivation for Radar-IMU Fusion. By our design in Sec. 4.2, MSTC enjoys a notable merit of avoiding to select points of moving objects. However, it is possible that MSTC fails to yield any output when the blockage of moving objects makes the detected stationary objects undesirably scarce. MSTC can detect such case if all the sets of representative points with a common velocity profile ($\mathcal{Q}_r$ in Algorithm 1) are of size three. When that happens, MSTC cannot utilize the velocity profile consensus of PSOs to estimate the radar egovelocity, because actually any three representative points have a common velocity profile.

Therefore, in this paper, we leverage an IMU as the auxiliary sensor, and utilize its acceleration measurements to obtain the 3-DoF translation when the radar egovelocity is unavailable. However, as we will discuss in Sec. 4.3.2, translation estimation solely by IMU is highly drifted because of gravity pollution.

4.3.2 Challenge: Gravity Pollution. The IMU acceleration measurements suffer from *gravity pollution* in that they are in fact the summation of the motion-incurred acceleration and the gravity. Clearly, without careful calibration which removes the gravity from the IMU acceleration measurements, gravity pollution could easily introduce large drifts to the translation estimated via integrating the acceleration twice.

However, accurately estimating the gravity is anything but trivial. The conventional calibration approach [24] estimates the gravity in the following two cases accordingly.

- At any time instance when the IMU is stationary or rotating without translating, the IMU has zero motion-incurred acceleration, and thus the conventional calibration approach directly adopts the IMU acceleration measurement as the estimated gravity.

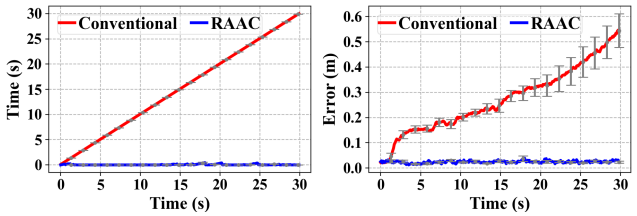


Figure 8: The post-anchor time (left) and translation error (right) in every 0.1s interval. The results are averaged over 10 trajectories.

- For any other time instance t, the conventional calibration approach traces back to the last time instance t' when the gravity $\mathbf{g}_{t'}$ is directly measured by the IMU accelerometer, and uses $\mathbf{g}_{t'}$ to estimate the gravity $\mathbf{g}_t$ at time instance t by the following Eq. (3), i.e.,

$$\mathbf{g}_t = \mathbf{R}_{t':t}\mathbf{g}_{t'}, \quad (3)$$

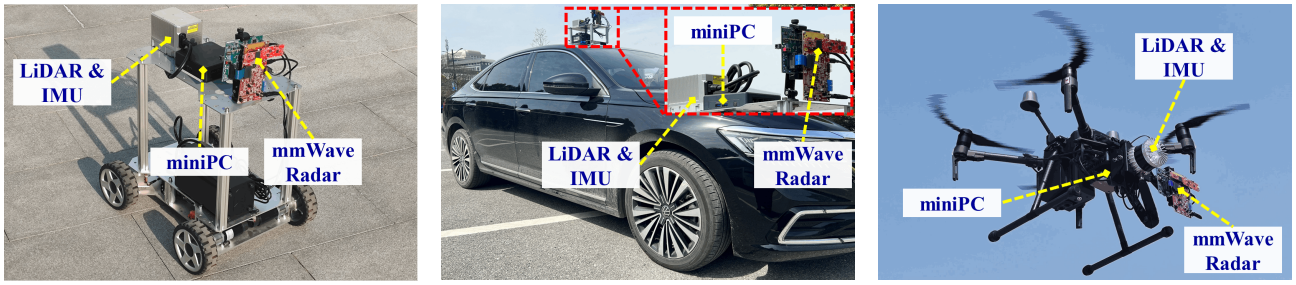
where $\mathbf{R}_{t':t}$ is the rotation matrix [36] from t' to t, which is converted from the rotation angles estimated by integrating the IMU angular velocity measurements over time.

Clearly, the gravity directly measured by the IMU accelerometer serves as the *anchor gravity* for estimating the gravities in the subsequent time instances until the next directly measured gravity. However, such anchor gravities are rare in practice, because the mobile platform typically has non-zero motion-incurred acceleration when it is moving. As a result, the time interval from t' to t in Eq. (3) could be as long as several minutes, during which a certain amount of angular velocity error could be accumulated. Although such error is negligible for estimating $\mathbf{R}_{t':t}$, it will be significantly amplified during the process of twice integration over time and cause enormous translation estimation error.

We study the impact of scarce anchor gravities to the conventional calibration approach by conducting the following experiments, in which we equip a mobile robot with an IMU, and set it to cruise following 10 random trajectories with each trajectory lasting for 30s. In our experiments, the anchor gravities are only available at the beginning of each trajectory when the robot is stationary. We define the length of the time interval utilized to estimate the gravity (i.e., $t - t'$ in Eq. (3)) as the post-anchor time. The red curves in Fig. 8 show that the post-anchor time of the conventional calibration approach grows over time and finally reaches around 30s, and the translation estimation error with the conventional calibration approach for gravity elimination drastically increases and reaches around 0.5m within 30s.

4.3.3 Simultaneous Fusion and Calibration. To address the issue of gravity pollution and embrace the advantages of both the mmWave radar and the IMU for egomotion estimation, we propose the SFC mechanism, which jointly estimates the mobile platform's 3-DoF rotation and 3-DoF translation by the following complementary fusion procedure.

- **3-DoF Rotation Estimation.** SFC estimates the 3-DoF rotation by integrating the angular velocity measured by the IMU gyroscope over the time interval between two consecutive egomotion estimation outputs.
- **3-DoF Translation Estimation.** SFC estimates the 3-DoF translation by integrating the mobile platform's egovelocity over the time interval between two consecutive egomotion estimation outputs. When the radar egovelocity is available, SFC adopts



(a) The mobile robot platform. (b) The automobile platform. (c) The UAV platform.

Figure 9: The mobile platforms for evaluating EmoRI.

it as the mobile platform's egovelocity. Otherwise, SFC obtains it from the IMU egovelocity. The IMU egovelocity is estimated by removing the gravity from the accelerometer measurements by our designed *Radar-Aided Acceleration Calibration (RAAC)* algorithm, and integrating the calibrated acceleration over time.

4.3.4 Radar-Aided Acceleration Calibration. The core of SFC's 3-DoF translation estimation is RAAC, whose key intuition is to *leverage the radar egovelocity to generate anchor gravities* as long as the radar egovelocity is available regardless of the mobile platform's motion status. Compared with the conventional calibration approach that only generates anchor gravities when the mobile platform has zero motion-incurred acceleration, RAAC has access to more anchor gravities for gravity estimation. Specifically, RAAC estimates the gravity at each time t in the following two cases accordingly.

- Case 1: radar egovelocity is available at time t. Suppose the radar egovelocity in the radar reference frame is $\mathbf{v}_t^R$, the IMU angular velocity in the IMU reference frame is ω_t, and the IMU egovelocity in the IMU reference frame is $\mathbf{v}_t^I$. The function $f(\cdot,\cdot)$ bridges $\mathbf{v}_t^I$, $\mathbf{v}_t^R$, and ω_t by the following Eq. (4), i.e.,
$$\mathbf{v}_t^I = f(\mathbf{v}_t^R, \omega_t) = \mathbf{R}_{RI}(\mathbf{v}_t^R + \omega_t \times \mathbf{r}_{RI}), \tag{4}$$
where $\mathbf{R}_{RI}$ is the rotation matrix that transforms the IMU egoveloity in the radar reference frame into that in the IMU reference frame, $\mathbf{r}_{RI}$ is the vector from the origin of the radar reference frame to the origin of the IMU reference frame, and $\times$ represents the cross product of two vectors. Suppose the IMU acceleration in the IMU reference frame is $\mathbf{a}_t$. Eq. (5) bridges the radar egovelocity and the gravity,
$$\begin{aligned} f(\mathbf{v}_t^R, \omega_t) - f(\mathbf{v}_{t-\tau}^R, \omega_{t-\tau}) &= \int_{t-\tau}^{t} (\mathbf{a}_u - \mathbf{g}_u) du \\ &\approx \frac{1}{K}\sum_{k=1}^{K\tau} \left(\mathbf{a}_{t-k/K} - \mathbf{R}_{t:t-k/K}\mathbf{g}_t\right), \end{aligned} \tag{5}$$
where τ is the time interval between t and the latest time instance when the radar egovelocity is available, and $\mathbf{R}_{t:t-k/K}$ is the IMU rotation matrix from time t to $t-k/K$ with K denoting the IMU data generation frequency. Eq. (5) depicts the fact that the difference between the IMU egovelocity at time t and $t-\tau$ in the IMU reference frame equals to the integration of the motion-incurred acceleration over time from $t-\tau$ to t, which can be approximated by multiplying the summation of the motion-incurred acceleration in a series of small time intervals from $t-\tau$ to t with the length of each time interval equal to $1/K$. We rearrange Eq. (5) and get the anchor gravity $\mathbf{g}_t$ in the IMU reference frame at time t given by Eq. (6),
$$\mathbf{g}_t \approx \mathbf{C}_t^{-1}\left(\mathbf{B}_t - K\mathbf{A}_t\right), \tag{6}$$
where $\mathbf{A}_t = f(\mathbf{v}_t^R, \omega_t) - f(\mathbf{v}_{t-\tau}^R, \omega_{t-\tau})$, $\mathbf{B}_t = \sum_{k=1}^{K\tau} \mathbf{a}_{t-k/K}$, and $\mathbf{C}_t = \sum_{k=1}^{K\tau} \mathbf{R}_{t:t-k/K}$. RAAC estimates $\mathbf{A}_t$, $\mathbf{B}_t$, and $\mathbf{C}_t$, and substitutes them into Eq. (6) to estimate the anchor gravity $\mathbf{g}_t$. Specifically, $f(\cdot,\cdot)$ in $\mathbf{A}_t$ is a fixed function obtained by measuring the orientations and locations of the radar and IMU, $\mathbf{v}_t^R$ and $\mathbf{v}_{t-\tau}^R$ in $\mathbf{A}_t$ are estimated by MSTC, ω_t and $\omega_{t-\tau}$ in $\mathbf{A}_t$ are measured by the IMU gyroscope, the accelerations in $\{\mathbf{a}_{t-k/K}\}_{k\in[K\tau]}$ to calculate $\mathbf{B}_t$ are measured by the IMU accelerometer, and the rotation matrices in $\{\mathbf{R}_{t:t-k/K}\}_{k\in[K\tau]}$ to calculate $\mathbf{C}_t$ are calculated from the angular velocity measured by the IMU gyroscope.
- Case 2: radar egovelocity is unavailable at time t. RAAC finds the latest time instance t' when the radar egovelocity is available and obtains the anchor gravity $\mathbf{g}_{t'}$. After that, RAAC obtains the rotation matrix $\mathbf{R}_{t':t}$ by integrating the angular velocity measured by the gyroscope over time, and estimates $\mathbf{g}_t$ by Eq. (3).

4.3.5 Evaluation of RAAC. We evaluate the effectiveness of RAAC in SFC using the experiments described in Sec. 4.3.2. Fig. 8 shows that the post-anchor time of RAAC is significantly shorter than that of the conventional calibration approach. Specifically, the average of the former and the latter is respectively 0.19s and 15s. We also illustrate the translation estimation with RAAC for gravity elimination in Fig. 8, which does not accumulate over time, and is persistently lower than that with the conventional calibration approach for gravity elimination.

5 IMPLEMENTATION AND EVALUATION

5.1 System Implementation

To thoroughly evaluate the performance of EmoRI on different types of mobile platforms, we implement EmoRI on three representative mobile platforms, including a WHEELTec mobile robot [37], a Volkswagen Passat Sedan automobile [38], and a DJI Matrice 210 UAV [39], as shown in Fig. 9.

We equip all the three platforms with a single-chip TI IWR6843AOP mmWave radar [18], and a DCA1000 data acquisition board [40] for collecting the beat frequency signals of the mmWave radar. We equip one Livox Horizon LiDAR [41] on both the mobile robot and the automobile platforms, and equip an Ouster OS1-64 LiDAR [42] on the UAV platform. We choose to equip the UAV with the OS1-64 LiDAR because the Livox Horizon LiDAR is

(a) Corridor with 4 humans. (b) Hall with 4 humans. (c) Parking lot with 2 bicycles. (d) Parking lot with automobile.

Figure 10: Example snapshots of trace collection with the mobile robot platform.

(a) No moving object. (b) One moving automobile. (c) Multiple moving automobiles. (d) Multiple walking humans.

Figure 11: Example snapshots of trace collection with the automobile platform (a-c) and the UAV platform (d).

Table 1: Trace collection settings with walking humans.

Human #	Traj. Shape	Motion Mode	Trace #/Mode
1	Straight	1, 2, 3	1
2	Straight	1, 2, 3	1
3	Straight	1, 2, 3	1
4	Straight	Hybrid	2
5	Straight	Hybrid	2
6	Straight	Hybrid	2
6	S&C	Hybrid	5

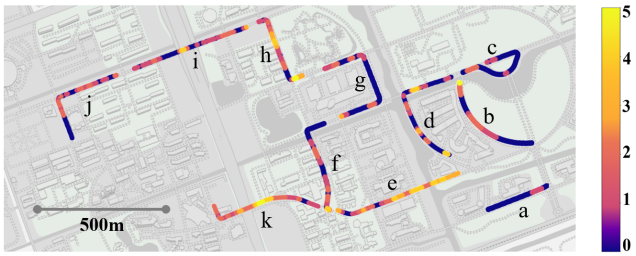


Figure 12: The trajectories of the automobile platform, and the number of moving objects through each trajectory.

too heavy for the UAV to carry. We utilize the BMI088 IMU [43] (around $2) integrated in the Livox Horizon LiDAR, and the IAM-20680HT IMU [44] (around $10) integrated in the Ouster OS1-64 LiDAR as the IMU choices for the corresponding mobile platforms. Note that the LiDAR on each mobile platform is to acquire the LiDAR point cloud, which serves as the input of the LiDAR SLAM algorithm to obtain the egomotion ground truths. Compared with GPS that works poorly in indoor environments and can only yield 3-DoF translation estimation, LiDAR SLAM is a reliable approach for 6-DoF egomotion estimation in both indoor and outdoor environments. In fact, the results of LiDAR SLAM have been adopted as the ground truths for many egomotion estimation approaches [25, 26, 45]. We also equip an Intel NUC 11 miniPC [46] on each mobile platform to control the the data collection process of the mmWave radar and the LiDAR, and run the codes of EmoRI to generate the 6-DoF egomotion estimation.

5.2 Experimental Methodology

5.2.1 Baselines. We compare EmoRI with the following three representative methods that utilize single-chip mmWave radar and IMU for egomotion estimation. *milliEgo* [25]: It is a learning-based method that utilizes the attention networks to fuse the measurements of radar and IMU. *ERI* [27]: It utilizes radar measurements to obtain the translation estimations and utilizes IMU measurements to reduce noise. We augment this method to enable rotation estimation by integrating the angular velocity measurements of IMU over time. *Milli-RIO* [26]: It leverages RNN and Kalman filter to fuse the measurements of radar and IMU for egomotion estimation.

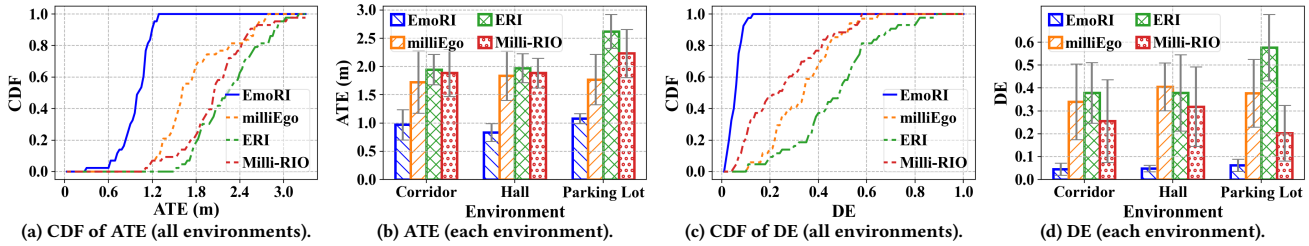
5.2.2 Evaluation Metrics. To evaluate the performance of each method, we first stitch its estimated egomotion sequence into the global odometry trajectory, and leverage the *Absolute Trajectory Error (ATE)* and *Destination Error (DE)* as the evaluation metrics. ATE measures the average distance between each of the two corresponding locations in the estimated and the ground truth trajectory, which depicts the average egomotion estimation accuracy. DE is the ratio of the distance between the destinations of the estimated and ground truth trajectory to the length of the ground truth trajectory, which gives an intuitive description about the cumulative estimation error of the trajectory.

5.3 Trace Collection

A trace is defined as the data of the mmWave radar, IMU, and LiDAR collected while the mobile platform moves along a specific trajectory, and meets specific numbers and types of moving objects with their specific motion modes. Each trace is an atomic unit to validate the egomotion estimation performance. To extensively evaluate EmoRI and the baselines, we collect traces on diverse mobile platforms in diverse representative scenes with different numbers and kinds of moving objects. All experiments have strictly followed the standard procedures of IRB of our institute.

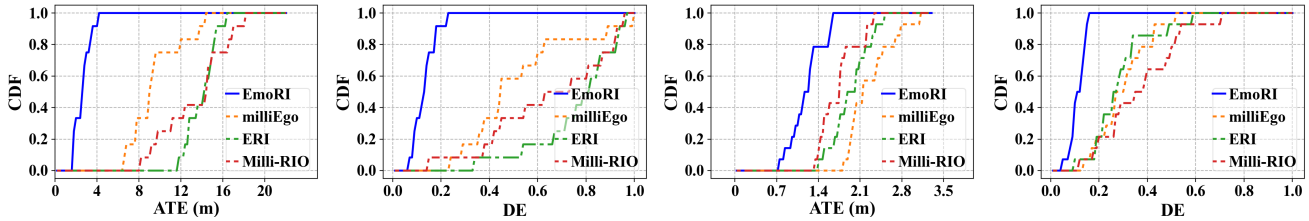
5.3.1 Traces Collected by the Mobile Robot Platform. The experiments with the mobile robot platform are conducted in two indoor environments (i.e., a corridor and a hall), and one outdoor environment (i.e., a parking lot).

We collect traces in both the corridor and the hall according to Tab. 1. Specifically, we let walking humans serve as the moving objects, because humans are the most common moving objects in indoor environments. We vary the number of walking humans from 1 to 6, and designate them to walk in three motion modes, including parallel to (mode 1), diagonally intersecting with (mode 2), and



(a) CDF of ATE (all environments). (b) ATE (each environment). (c) CDF of DE (all environments). (d) DE (each environment).

Figure 13: The overall performance of EmoRI and the baselines with the mobile robot platform.



(a) CDF of ATE (automobile). (b) CDF of DE (automobile). (c) CDF of ATE (UAV). (d) CDF of DE (UAV).

Figure 14: The overall performance of EmoRI and the baselines with the automobile and UAV platforms.

vertically intersecting with (mode 3) the heading direction of the robot. For the convenience of keeping the moving objects in radar's FoV and designating their motion modes, we collect 15 traces with the robot moving in straight lines in either environment. We also collect 5 traces with 6 walking humans in either environment with the robot moving in both straight lines and trajectories with curves (S&C). For the traces with more than 3 humans, we set their motion mode to be hybrid, where each human moves according to a motion mode randomly chosen from the above three. Thus, we collect $15 + 5 = 20$ traces in either environment. It takes around 30s to collect each trace. The length of each trajectory varies from around 15m to 30m. The snapshots of trace collection in both environments are illustrated in Fig. 10a and 10b.

We collect traces in the parking lot with walking humans as moving objects according to Tab. 1. We also collect 2 traces respectively with 2 bicycles and an automobile as the moving objects as in Fig. 10c and 10d. The motions of the bicycles and the antomobile are in mode 1. Totally, we collect $20 + 2 \times 2 = 24$ traces in the parking lot. It takes around 30s to collect each trace. The length of the trajectory of each trace varies from 15m to 30m.

5.3.2 Traces Collected by the Automobile Platform. We collect 20 traces using the automobile platform on the roads of a campus with the humans, bicycles, and automobiles encountered during trace collection as the moving objects. It takes around 120s to collect each trace. We plot in Fig. 12 the trajectories of 11 out of the 20 collected traces with the number of encountered moving objects of each sub-piece of the trajectories marked. Note that the trajectories of the other 9 traces are not plotted, because they overlap with those shown in this figure. The shapes of the trajectories contain the primary road shapes, including straight lines (trajectory a, e, and i), corners (trajectory f, g, h, and j), and curves (trajectory b, c, d, and k). Three snapshots captured by a front-view camera placed next to the radar that correspond to three representative cases are illustrated in Fig. 11a to 11c.

5.3.3 Traces Collected by the UAV Platform. We collect 14 traces using the UAV platform with 2 humans as the moving objects. The

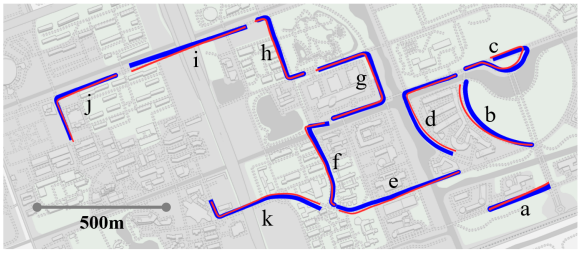


Figure 15: The trajectories estimated by EmoRI (blue) and ground truths (red) in automobile platform experiments.

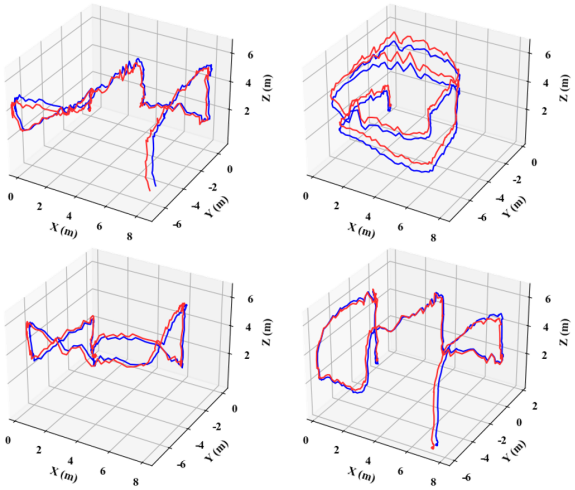
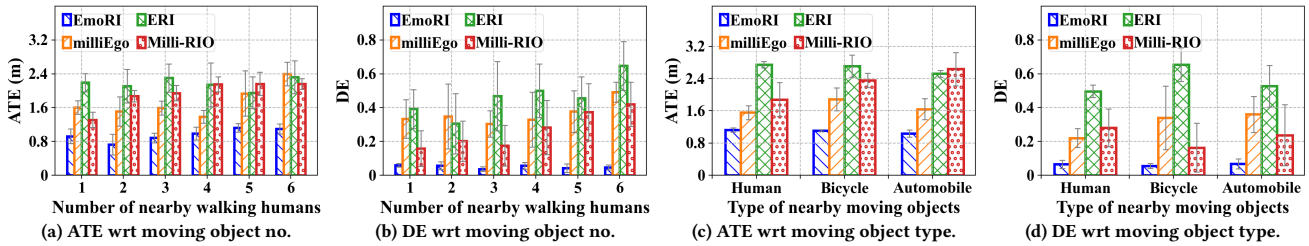


Figure 16: The trajectories estimated by EmoRI (blue) and ground truths (red) in UAV platform experiments.

flight time of each trace is around 90s. A snapshot of trace collection with the UAV platform is shown in Fig. 11d.

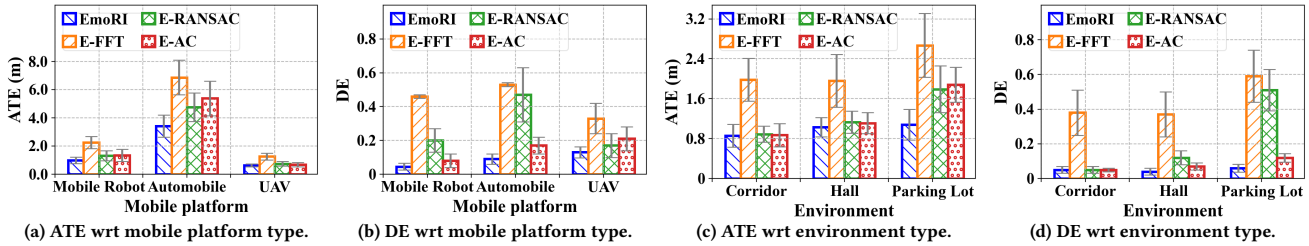
5.4 Overall Performance

We compare the overall 6-DoF egmotion estimation performance of EmoRI with the three baselines introduced in Sec. 5.2. The overall performance is evaluated using all the collected traces, where EmoRI reduces the average ATE for all the experiments from 3.1m,



(a) ATE wrt moving object no. (b) DE wrt moving object no. (c) ATE wrt moving object type. (d) DE wrt moving object type.

Figure 17: The performance of EmoRI and the baselines wrt different numbers and types of nearby moving objects.



(a) ATE wrt mobile platform type. (b) DE wrt mobile platform type. (c) ATE wrt environment type. (d) DE wrt environment type.

Figure 18: The performance of EmoRI and its variants.

4.7m, and 3.5m to 1.27m, and reduces the average DE for all the experiments from 0.27, 0.43, and 0.31 to only 0.07 compared with milliEgo, ERI, and milli-RIO, respectively. Although the three baselines perform reasonably well in environments with all stationary objects as described in their original papers [25–27], their performance drastically degrade in our experiments with moving objects in radar's FoV.

As shown in Fig. 13a and 13c, in the experiments with the mobile robot platform, EmoRI achieves an average of 38%, 57%, and 52% reduction on ATE, and 74%, 84%, and 69% reduction on DE, compared with milliEgo, ERI, and milli-RIO, respectively. As shown in Fig. 13b and 13d, such performance improvement is consistent in different environments. Similarly, as illustrated in Fig. 14, in the experiments with the automobile platform, EmoRI achieves at least an average of 67% reduction on ATE, and at least 62% reduction on DE, compared with the three baselines. In the experiments with the UAV platform, EmoRI achieves an average of at least 32% reduction on ATE, and at least 53% reduction on DE, compared with the three baselines.

To intuitively understand the performance of EmoRI, we plot in Fig. 15 and 16 respectively several representative global odometry trajectories of the automobile and the UAV platform estimated by EmoRI, as well as the corresponding ground truths. We notice that the trajectories estimated by EmoRI are tightly aligned with the ground truth trajectories.

5.5 Impact of Moving Objects in FoV

5.5.1 Impact of the Moving Object Number. We evaluate EmoRI against different numbers of walking humans in radar's FoV with the mobile robot platform. As shown in Fig. 17a and 17b, EmoRI achieves an average of 44%, 56%, and 49% reduction on ATE, and an average of 86%, 88%, and 79% reduction on DE, compared with milliEgo, ERI, and Milli-RIO, respectively. Besides, the ATE and DE of all the baselines show an increasing trend with more walking humans, whereas those of EmoRI do not show such trends. Such results validate the robustness of EmoRI against different numbers of moving objects in radar's FoV.

5.5.2 Impact of the Moving Object Type. We evaluate EmoRI against different types (i.e., human, bicycle, automobile) of moving objects with the mobile robot platform in the parking lot. As shown in Fig. 17c and 17d, EmoRI achieves an average of 35%, 59%, and 51% reduction on ATE, and an average of 78%, 89%, and 72% reduction on DE, compared with milliEgo, ERI, and Milli-RIO, respectively. Besides, the ATE and DE of EmoRI keep low with different types of moving objects. Such results validate the robustness of EmoRI against the types of moving objects in radar's FoV.

5.6 Evaluation of Each Component

We evaluate the necessity of each component of EmoRI, i.e., HFM, MSTC, and SFC, as well as their contribution to EmoRI's superior performance over the baselines. To this end, we build three variants of EmoRI, namely E-FFT, E-RANSAC, and E-AC. E-FFT replaces HFM with the conventional FFT-based algorithm, E-RANSAC replaces MSTC with PL-RANSAC, and E-AC replaces RAAC of SFC with the conventional A^3 algorithm [24]. We also build six variants of the three baselines [25–27] by enhancing them with the components of EmoRI in two ways. Specifically, we change the input point cloud of the three baselines as the point cloud generated by HFM or generated by HFM+MSTC. We evaluate EmoRI, the baselines, and their variants with all the collected traces.

5.6.1 Necessity of HFM. As shown in Fig. 18a and 18b, compared with E-FFT, EmoRI achieves 56%, 50%, and 50% reduction on average ATE, and 90%, 83%, and 61% reduction on average DE for experiments with the mobile robot, the automobile and the UAV platform, respectively. Meanwhile, as shown in Fig. 18c and 18d, EmoRI achieves 56%, 47%, and 59% reduction on average ATE, and 87%, 88%, and 88% reduction on average DE for experiments in the corridor, hall, and parking lot on the mobile robot platform, respectively. Such results indicate that HFM's improvement against the

conventional FFT-based approach brings substantial improvement for the accuracy of egomotion estimation. In fact, accurate point cloud generation with HFM is the premise for MSTC and SFC to take effect. The relatively low-quality point cloud generated by FFT also influences the performance of PSO selection with MSTC and radar-IMU fusion with SFC, and finally severely degrades the accuracy of egomotion estimation.

5.6.2 Necessity of MSTC. As shown in Fig. 18a and 18b, compared with E-RANSAC, EmoRI achieves 25%, 28%, and 15% reduction on ATE, and 78%, 87%, and 23% reduction on DE for experiments with the mobile robot, the automobile, and the UAV platform, respectively. As shown in Fig. 18c and 18d, EmoRI achieves 3%, 28%, and 44% reduction on ATE, and 6%, 64%, and 88% reduction on DE for experiments in the corridor, hall, and parking lot with the mobile robot platform, respectively. The rationale for EmoRI's increasing improvements against E-RANSAC in corridor, hall, and parking lot are as follows. The number of the PSOs is decreasing in the corridor, the hall, and the parking lot. E-RANSAC gets easier to confuse the points of moving objects as the PSOs with less PSOs, while EmoRI still keeps the ATE and DE relatively low thanks to MSTC's reasonable PSO selection.

5.6.3 Necessity of SFC. As shown in Fig. 18a and 18b, compared with E-AC, EmoRI achieves 27%, 36%, and 7% reduction on ATE, and 36%, 47%, and 38% reduction on DE for experiments with the mobile robot, the automobile, and the UAV platform, respectively. Meanwhile, as shown in Fig. 18c and 18d, EmoRI achieves 3%, 6%, and 40% reduction on ATE, and 4%, 36%, and 50% reduction on DE for experiments in corridor, hall, and parking lot with the robot platform, respectively. We observe that the performance gap between EmoRI and E-AC is larger in the parking lot than in the corridor and hall. The reason is that the case where no PSOs can be selected happens more often in the parking lot. Such results validate the necessity of SFC when radar egovelocity is unavailable.

5.6.4 Contribution of Each Component to EmoRI's Superior Performance. We show the average ATE and DE reduction of EmoRI compared with the original and EmoRI-enhanced baselines in Table 2. HFM-enhanced baselines outperform the original baselines by having lower ATE and DE, which indicates that HFM's improvement on angle estimation takes a positive effect on egomotion estimation and contributes to EmoRI's superior performance over the original baselines. Besides, we observe that the baselines enhanced by HFM+MSTC outperform the original and HFM-enhanced baselines. Such results indicate that even if the input point clouds of the baselines are generated by HFM, the baselines still suffer from unsatisfactory performance because they lack the capacity to accurately select the PSOs from the point cloud. We also observe that EmoRI shows at least a 43% reduction on ATE and a 47% reduction on DE compared with the HFM+MSTC-enhanced baselines. Such results indicate that compared with the baselines, EmoRI could more properly fuse the radar and IMU measurements for accurate egomotion estimation. In all, we conclude that the three modules of EmoRI are all indispensable and their joint design contributes to EmoRI's superior performance over the baselines.

Table 2: The ATE and DE reduction of EmoRI compared with the original and EmoRI-enhanced baselines.

	Original			HFM-Enhanced			HFM+MSTC-Enhanced		
	milliEgo	ERI	Milli-RIO	milliEgo	ERI	Milli-RIO	milliEgo	ERI	Milli-RIO
ATE	59%	73%	64%	58%	66%	60%	43%	46%	47%
DE	74%	84%	77%	72%	83%	77%	47%	55%	49%

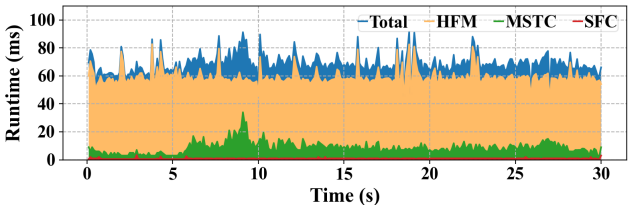


Figure 19: The runtime of EmoRI and its three components.

5.7 Runtime Analysis

We evaluate the runtime of the three components of EmoRI, i.e., HFM, MSTC, and SFC. For each radar frame, the average runtime of HFM, MSTC, and SFC is 59.7ms, 8.7ms, and 0.9ms, respectively. The end-to-end runtime of EmoRI is 69.3ms and is smaller than the time interval of one radar frame (100ms). In Fig. 19, we illustrate the runtime of EmoRI and its three components in a typical trace lasting for 30s. We notice that the runtime of EmoRI remains stably lower than 100ms over time. Such results indicate that EmoRI supports running at a frame rate of 10 frames per second, which is comparable to the 3D LiDAR-based egomotion estimation approaches [7–11].

6 RELATED WORK

6-DoF Egomotion Estimation. A number of methods have been proposed to estimate the egomotion of a mobile platform. One line of works [6, 47] utilize the WiFi NIC to capture the WiFi signal transmitted by WiFi APs, and infer the mobile platform's egomotion from WiFi CSI. These methods only work in the restricted areas where the WiFi APs locate. In contrast, other methods [7–11, 13–17, 25–27, 48–56, 59–62] that leverage on-board sensors (i.e., IMU, LiDAR, camera, mmWave radar) are free from such restriction. Next, we discuss these works with respect to the type of utilized sensors.

IMU-based methods [48, 49] infer the egomotion of a mobile platform directly from the IMU acceleration and angular velocity measurements. These methods usually suffer from severe cumulative drifts. LiDAR-based methods [7–9] utilize LiDAR SLAM to simultaneously localize the mobile platform and estimate its orientation. The performance of these methods drastically degrades in adverse weathers (e.g., rain, fog, snow). Camera-based methods [13, 14, 16] infer the egomotion of the mobile platform through matching the landmarks between consecutive frames of images. These methods usually fail in overly weak or strong illumination. Radar-based methods [50–56] estimate the mobile platform's egomotion by inferring the relative motion of the mobile platform against the PSOs in the radar point cloud. However, either the unbearable weight of the mechanical scanning radar (around 6kg each

[57]) or the prohibitive cost of the high-end cascading mmWave radar (around $1500 each [58]) prevents these methods [50–56] from being applied on low-cost and lightweight mobile platforms.

Recently, a series of fusion-based methods [10, 11, 15, 17, 25–27, 59–62] propose to utilize an IMU as the auxiliary sensor to augment radar/LiDAR/camera for better egomotion estimation. These works can be categorized as radar-IMU fusion [25–27, 59–62], LiDAR-IMU fusion [10, 11], and camera-IMU fusion [15, 17] methods. Among the radar-IMU fusion methods, the works [27, 60] assume the objects in the environment are static and do not consider the scenarios where moving objects exist in radar's FoV. Although the works [25, 26, 59–62] all include PSO selection mechanisms to alleviate the negative influence of moving objects, there exist non-negligible defects in these PSO selection mechanisms. Specifically, the work [59] resorts to RANSAC for PSO selection, but it has been proved in our paper that RANSAC often fails. Besides, this work also uses an additional barometer sensor to overcome the radar's high vertical motion estimation error. The work [61] exploits the points of the ground as the PSOs, which would fail when the radar is on the UAV and cannot receive the reflections from the ground. The methods in [62] are designed based on the assumption that the objects in the environment are located on a single 2D plane, which almost only holds in scenarios of remote sensing, where the objects on the Earth can be viewed as located on the flat Earth plane for the aircraft. Obviously, such an assumption does not generalize well in common egomotion estimation tasks. The learning-based methods [25, 26] resort to deep learning models to intelligently select the PSOs and match them for egomotion estimation. However, the experimental results in Sec. 5.4 indicate that these learning-based approaches could not reliably estimate the egomotion as EmoRI does when moving objects exist in radar's FoV.

Clearly, EmoRI is a radar-IMU fusion method. The core advantage of EmoRI over other radar-IMU fusion methods [25–27, 59–62] is that EmoRI could work robustly with moving objects in radar's FoV, which is attributed to its effective PSO selection and radar-IMU fusion mechanisms. We emphasize that the existing fusion techniques in camera-IMU fusion and LiDAR-IMU fusion methods are not suitable to be applied for radar-IMU fusion, because mmWave radar is fundamentally different from camera and LiDAR in that mmWave radar cannot capture the landmarks required for camera-IMU fusion, and has around 100 times sparser points than LiDAR. Besides, compared with the LiDAR-IMU fusion and camera-IMU fusion methods, EmoRI enjoys the desirable properties of resistance to adverse weather and illumination by adopting mmWave radar as the primary sensor. We comprehensively compare EmoRI with the aforementioned egomotion estimation methods in Tab. 3.

6-DoF Passive Motion Tracking. Another thread of works [63–67] highly related with EmoRI are those that aim at tracking the motion of a mobile platform with devices installed in the environment. Among them, [63] infers the motion of the mobile platform by analyzing the motion incurred features in the acoustic signals generated by the mobile platform, [65] uses high-speed cameras to record the variations of the key feature points of the mobile platform to infers its motion, and [67] fuses mmWave radar and camera to estimate the mobile platform's full 6-DoF motion. Clearly, this line of works treat the mobile platform as a passive object to be tracked in the environment, and are thus orthogonal to EmoRI that enables a mobile platform to actively estimate its own egomotion.

Table 3: Comparison with existing methods.

Method	Primary Sensor	Drifts	Area	Weather/ Illumination	Moving Obj in FoV
[6, 47]	WiFi NIC	Low	Limited	Robust	Robust
[48, 49]	IMU	High	Free	Robust	Robust
[7–11]	LiDAR	Low	Free	Susceptible	Robust
[13–17]	Camera	Low	Free	Susceptible	Robust
[50, 51, 53–55]	Scanning Radar	Low	Free	Robust	Susceptible
[52, 56]	High-end Radar	Low	Free	Robust	Susceptible
[25–27, 59–62]	Single-chip Radar	Low	Free	Robust	Susceptible
EmoRI	**Single-chip Radar**	**Low**	**Free**	**Robust**	**Robust**

Localization. The egomotion estimation results of EmoRI can be naturally utilized for localization. Given the initial position and orientation of a mobile platform, EmoRI can obtain its odometry by stitching the egomotion sequence, and infer its location from the estimated odometry. However, although there exist a wide spectrum of approaches that utilize WiFi [68], acoustic [64], RFID [69], LoRa [70], or mmWave signals [66, 71] for localization, they usually fail for egomotion estimation. The reason is that despite these approaches could know the location variations of the mobile platform, they cannot estimate a mobile platform's orientation, which prevents them from estimating the rotation and transition in the mobile platform's local reference frame.

7 CONCLUSION AND FUTURE WORK

This paper proposes EmoRI, a novel egomotion estimation method consisting of the HFM algorithm that distinctly improves the angle estimation accuracy of single-chip mmWave radar, the MSTC algorithm that properly selects only the PSOs from the radar point cloud, and the SFC mechanism that outputs reliable and full 6-DoF egomotion estimation. Extensive experiments validate that EmoRI pushes the limit of single-chip mmWave radar-based egomotion estimation with moving objects in radar's FoV by reducing the permeter destination error from decimeter level to centimeter level.

A number of techniques could be explored in future to further improve EmoRI. First, the estimated egovelocities of multiple radars that could possibly exist on some platforms (e.g., automobile) can be jointly utilized to calibrate the IMU and yield more accurate acceleration measurements, which promotes the egomotion estimation accuracy. Besides, IMU drifts are still inevitable and may accumulate in scenarios where the radar egovelocity estimation is unavailable for a long time. Although such scenarios are rare in practice, techniques such as loop closure detections [72] could be utilized to enhance EmoRI to further alleviate the IMU drifts. Lastly, since cameras could provide rich scene information for egomotion estimation in environments with good visibility, properly augmenting EmoRI with cameras in such environments would be promising to improve its performance.

ACKNOWLEDGMENTS

This work was supported by NSF China (No. U21A20519, U20A20181, 62372288, 62202298, 61922055, 92167205, 62141220, 61972253).

REFERENCES

[1] M. Ouyang, X. Shi, Y. Wang, Y. Tian, Y. Shen, D. Wang, P. Wang, and Z. Cao, "A collaborative visual slam framework for service robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.

[2] Elektrobit, "Vehicle position and ego-motion estimation for automotive applications," https://www.elektrobit.com/tech-corner/vehicle-position-and-ego-motion-estimation-for-automotive-applications/.

[3] "Ar-hud," https://www.fic.com.tw/automotive/ar-hud/.

[4] "Ar-hud: A heads up on the road ahead," https://blog.huawei.com/2022/01/04/ar-hud-road-ahead-transportation/.

[5] MBUX Augmented Reality Head-up Display, "Mercedodes-benz," https://www.mercedes-benz.com.hk/en/passengercars/mercedes-benz-cars/models/eqs/saloon-v297/equipment.pi.html/mercedes-benz-cars/models/eqs/saloon-v297/equipment/individualization/mbux-ar-hud.

[6] G. Chi, Z. Yang, J. Xu, C. Wu, J. Zhang, J. Liang, and Y. Liu, "Wi-drone: wi-fi-based 6-dof tracking for indoor drone flight control," in *The 20th Annual International Conference on Mobile Systems, Applications and Services (Mobisys)*, 2022.

[7] M. Oelsch, M. Karimi, and E. Steinbach, "R-loam: Improving lidar odometry and mapping with point-to-mesh features of a known 3d reference object," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2068–2075, 2021.

[8] J. Zhang and S. Singh, "LOAM: lidar odometry and mapping in real-time," in *Robotics: Science and Systems X*, 2014.

[9] J. Lin and F. Zhang, "Loam livox: A fast, robust, high-precision lidar odometry and mapping package for lidars of small fov," in *IEEE International Conference on Robotics and Automation, ICRA*, 2020.

[10] A. Tagliabue, J. Tordesillas, X. Cai, A. Santamaria-Navarro, J. P. How, L. Carlone, and A.-a. Agha-mohammadi, "Lion: Lidar-inertial observability-aware navigator for vision-denied environments," in *Experimental Robotics: The 17th International Symposium*. Springer, 2021, pp. 380–390.

[11] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "Fast-lio2: Fast direct lidar-inertial odometry," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2053–2073, 2022.

[12] Livox, "Livox hap," https://www.livoxtech.com/hap.

[13] A. J. Davison, I. D. Reid, N. Molton, and O. Stasse, "Monoslam: Real-time single camera SLAM," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 1052–1067, 2007.

[14] C. Wang, C. E. Thorpe, S. Thrun, M. Hebert, and H. F. Durrant-Whyte, "Simultaneous localization, mapping and moving object tracking," *Int. J. Robotics Res.*, vol. 26, no. 9, pp. 889–916, 2007.

[15] R. Clark, S. Wang, H. Wen, A. Markham, and N. Trigoni, "Vinet: Visual-inertial odometry as a sequence-to-sequence learning problem," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, S. Singh and S. Markovitch, Eds., 2017.

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

[17] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration theory for fast and accurate visual-inertial navigation," *IEEE Transactions on Robotics*, pp. 1–18, 2015.

[18] Texas Instruments Incorporated, "Single-chip 60-ghz to 64-ghz automotive radar sensor integrating antenna on package, dsp and mcu," https://www.ti.com/product/IWR6843AOP.

[19] Texas Instruments, "mmwave radar sensors in robotics applications," https://www.ti.com/lit/pdf/spry311.

[20] Junko Yoshida, "Audi a8: A look under the hood," https://www.eetindia.co.in/audi-a8-a-look-under-the-hood/.

[21] Autel, "Autel evo max 4t millimeter-wave radar obstacle avoidance," https://www.autelpilot.com/blogs/news/autel-max-4t-millimeter-wave-radar-obstacle-avoidance.

[22] D. Kellner, M. Barjenbruch, J. Klappstein, J. Dickmann, and K. C. J. Dietmayer, "Instantaneous ego-motion estimation using multiple doppler radars," in *IEEE International Conference on Robotics and Automation, ICRA*, 2014.

[23] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE transactions on antennas and propagation*, vol. 34, no. 3, pp. 276–280, 1986.

[24] P. Zhou, M. Li, and G. Shen, "Use it free: instantly knowing your phone attitude," in *The 20th Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2014.

[25] C. X. Lu, M. R. U. Saputra, P. Zhao, Y. Almalioglu, P. P. B. de Gusmao, C. Chen, K. Sun, N. Trigoni, and A. Markham, "milliego: single-chip mmwave radar aided egomotion estimation via deep sensor fusion," in *The 18th ACM Conference on Embedded Networked Sensor Systems (Sensys)*, 2020.

[26] Y. Almalioglu, M. Turan, C. X. Lu, N. Trigoni, and A. Markham, "Milli-rio: Ego-motion estimation with low-cost millimetre-wave radar," *IEEE Sensors Journal.*, vol. 21, no. 3.

[27] A. Kramer, C. Stahoviak, A. Santamaria-Navarro, A. Agha-mohammadi, and C. Heckman, "Radar-inertial ego-velocity estimation for visually degraded environments," in *IEEE International Conference on Robotics and Automation, ICRA*, 2020.

[28] S. Palipana, D. Salami, L. A. Leiva, and S. Sigg, "Pantomime: Mid-air gesture recognition with sparse millimeter-wave radar point clouds," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 5, no. 1, pp. 27:1–27:27, 2021.

[29] K. Bansal, K. Rungta, S. Zhu, and D. Bharadia, "Pointillism: accurate 3d bounding box estimation with multi-radars," in *The 18th ACM Conference on Embedded Networked Sensor Systems (Sensys)*, 2020.

[30] X. Shuai, Y. Shen, Y. Tang, S. Shi, L. Ji, and G. Xing, "millieye: A lightweight mmwave radar and camera fusion system for robust object detection," in *International Conference on Internet-of-Things Design and Implementation (IoTDI)*, 2021.

[31] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[32] Texas Instruments Incorporated, "Single-chip 76-ghz to 81-ghz mmwave sensor integrating dsp and mcu," https://www.ti.com/product/IWR1642/.

[33] Texas Instruments Incorporated, "Single-chip 76-ghz to 81-ghz mmwave sensor integrating mcu and hardware accelerator," https://www.ti.com/product/IWR1443/.

[34] M. A. Richards, *Fundamentals of radar signal processing*. McGraw-Hill Education, 2014.

[35] M. Ester, H. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD)*, 1996.

[36] J. Diebel *et al.*, "Representing attitude: Euler angles, unit quaternions, and rotation vectors," *Matrix*, vol. 58, no. 15-16, pp. 1–35, 2006.

[37] WheelTac, "Ros robot," https://wheeltec.jd.com/.

[38] Volkswagen, "Volkswagen passat sedan," https://www.volkswagen.com.au/en/models/passat.html.

[39] "Matrice 210 user manual," https://dl.djicdn.com/downloads/M200/20201120/M200_User_Manual_EN_20201120.pdf.

[40] Texas Instruments Incorporated, "Real-time data-capture adapter for radar sensing evaluation module," http://www.ti.com/tool/DCA1000EVM/.

[41] Livox, "Automotive-grade lidar for serial production," https://www.livoxtech.com/.

[42] Ouster, "Os1," https://ouster.com/products/scanning-lidar/os1-sensor/.

[43] B. Sensortec, "High-performance inertial measurement unit (bmi088)," https://www.bosch-sensortec.com/media/boschsensortec/downloads/product_flyer/bst-bmi088-fl000.pdf.

[44] TDK InvenSence, "Iam-20680ht," https://invensense.tdk.com/products/iam-20680ht/.

[45] Y. S. Park, Y. Shin, J. Kim, and A. Kim, "3d ego-motion estimation using low-cost mmwave radars via radar velocity factor for pose-graph SLAM," *IEEE Robotics Autom. Lett. (RA-L)*, vol. 6, no. 4, pp. 7691–7698, 2021.

[46] Intel, "Intel nuc 11 pro kit nuc11tnhi30z," https://ark.intel.com/content/www/us/en/ark/products/121572/intel-nuc-11-pro-kit-nuc11tnhi30z.html.

[47] C. Wu, F. Zhang, Y. Fan, and K. J. R. Liu, "Rf-based inertial measurement," in *The 33th ACM Special Interest Group on Data Communication (SIGCOMM)*, 2019.

[48] S. Shen, M. Gowda, and R. R. Choudhury, "Closing the gaps in inertial motion tracking," in *The 24th Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2018.

[49] J. Gong, X. Zhang, Y. Huang, J. Ren, and Y. Zhang, "Robust inertial motion tracking through deep sensor fusion across smart earbuds and smartphone," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. (IMWUT)*, vol. 5, no. 2, pp. 62:1–62:26, 2021.

[50] S. H. Cen and P. Newman, "Precise ego-motion estimation with millimeter-wave radar under diverse and challenging conditions," in *International Conference on Robotics and Automation, ICRA*, 2018.

[51] S. H. Cen and P. Newman, "Radar-only ego-motion estimation in difficult settings via graph matching," in *International Conference on Robotics and Automation, ICRA*, 2019.

[52] A. Kingery and D. Song, "Improving ego-velocity estimation of low-cost doppler radars for vehicles," *IEEE Robotics Autom. Lett. (RA-L)*, vol. 7, no. 4, pp. 9445–9452, 2022.

[53] D. Adolfsson, M. Magnusson, A. W. Alhashimi, A. J. Lilienthal, and H. Andreasson, "Lidar-level localization with radar? the CFEAR approach to accurate, fast, and robust large-scale radar odometry in diverse environments," *IEEE Trans. Robotics*, vol. 39, no. 2, pp. 1476–1495, 2023.

[54] R. Zhang, Y. Zhang, D. Fu, and K. Liu, "Scan denoising and normal distribution transform for accurate radar odometry and positioning," *IEEE Robotics Autom. Lett.*, vol. 8, no. 3, pp. 1199–1206, 2023.

[55] R. Weston, M. Gadd, D. D. Martini, P. Newman, and I. Posner, "Fast-mbym: Leveraging translational invariance of the fourier transform for efficient and accurate radar odometry," in *2022 International Conference on Robotics and Automation, (ICRA)*, 2022.

[56] S.-Y. Kwon, S. Kwak, J. Kim, and S. Lee, "Radar sensor-based ego-motion estimation and indoor environment mapping," *IEEE Sensors Journal*, 2023.

[57] NAVTECH, "Cts350-x," https://navtechradar.com/clearway-technical-specifications/.

[58] Texas Instruments Incorporated, "mmwave cascade imaging radar rf evaluation module," https://www.ti.com/tool/MMWCAS-RF-EVM.
[59] C. Doer and G. F. Trommer, "An EKF based approach to radar inertial odometry," in *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, (MFI)*, 2020.
[60] J. Michalczyk, R. Jung, and S. Weiss, "Tightly-coupled ekf-based radar-inertial odometry," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, (IROS)*, 2022.
[61] H. Chen, Y. Liu, and Y. Cheng, "DRIO: robust radar-inertial odometry in dynamic environments," *IEEE Robotics Autom. Lett.*, vol. 8, no. 9, pp. 5918–5925, 2023.
[62] E. B. Quist, P. C. Niedfeldt, and R. W. Beard, "Radar odometry with recursive-ransac," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 52, no. 4, pp. 1618–1630, 2016.
[63] Y. Sun, W. Wang, L. Mottola, R. Wang, and Y. He, "Aim: Acoustic inertial measurement for indoor drone localization and tracking," in *The 20th ACM Conference on Embedded Networked Sensor Systems (Sensys)*, 2022.
[64] W. Wang, L. Mottola, Y. He, J. Li, Y. Sun, S. Li, H. Jing, and Y. Wang, "Micnest: Long-range instant acoustic localization of drones in precise landing," in *The 20th ACM Conference on Embedded Networked Sensor Systems (Sensys)*, 2022.
[65] "Optitrack," https://optitrack.com/.
[66] P. Zhao, C. X. Lu, B. Wang, N. Trigoni, and A. Markham, "3d motion capture of an unmodified drone with single-chip millimeter wave radar," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.
[67] J. Xu, G. Chi, Z. Yang, D. Li, Q. Zhang, Q. Ma, and X. Miao, "Followupar: Enabling follow-up effects in mobile ar applications," in *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services (Mobisys)*, 2021.
[68] R. S. Ayyalasomayajula, A. Arun, C. Wu, S. Rajagopalan, S. Ganesaraman, A. Seetharaman, I. K. Jain, and D. Bharadia, "Locap: Autonomous millimeter accurate mapping of wifi infrastructure," in *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, R. Bhagwan and G. Porter, Eds., 2020.
[69] B. Liang, P. Wang, R. Zhao, H. Guo, P. Zhang, J. Guo, S. Zhu, H. H. Liu, X. Zhang, and C. Xu, "Rf-chord: Towards deployable RFID localization system for logistic networks," in *20th USENIX Symposium on Networked Systems Design and Implementation, (NSDI)*, 2023.
[70] J. Liu, J. Gao, S. K. Jha, and W. Hu, "Seirios: leveraging multiple channels for lorawan indoor and outdoor localization," in *The 27th Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2021.
[71] I. Pefkianakis and K. Kim, "Accurate 3d localization for 60 ghz networks," in *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems (Sensys)*, 2018.
[72] A. Angeli, S. Doncieux, J. Meyer, and D. Filliat, "Real-time visual loop-closure detection," in *IEEE International Conference on Robotics and Automation, (ICRA)*, 2008.