2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Enhancing mmWave Radar Point Cloud via Visual-inertial Supervision

Cong Fan[1], Shengkai Zhang[1]*, Kezhong Liu[1], Shuai Wang[2], Zheng Yang[3], Wei Wang[4]

***Abstract*— Complementary to prevalent LiDAR and camera systems, millimeter-wave (mmWave) radar is robust to adverse weather conditions like fog, rainstorms, and blizzards but offers sparse point clouds. Current techniques enhance the point cloud by the supervision of LiDAR's data. However, high-performance LiDAR is notably expensive and is not commonly available on vehicles. This paper presents mmEMP, a supervised learning approach that enhances radar point clouds using a low-cost camera and an inertial measurement unit (IMU), enabling crowd-sourcing training data from commercial vehicles. Bringing the visual-inertial (VI) supervision is challenging due to the spatial agnostic of dynamic objects. Moreover, spurious radar points from the curse of RF multipath make robots misunderstand the scene. mmEMP first devises a dynamic 3D reconstruction algorithm that restores the 3D positions of dynamic features. Then, we design a neural network that densifies radar data and eliminates spurious radar points. We build a new dataset in the real world. Extensive experiments show that mmEMP achieves competitive performance compared with the SOTA approach training by LiDAR's data. In addition, we use the enhanced point cloud to perform object detection, localization, and mapping to demonstrate mmEMP's effectiveness.**

I. Introduction

The utilization of millimeter-wave (mmWave) radar has achieved significant prevalence in automotive and robotic applications. As the mmWave radio frequency (RF) is robust to perceive environments through small particles, *e.g.*, smoke, fog, and snow, mmWave radar is complementary to optical sensors such as cameras and LiDAR [1]–[6]. Recently, single-chip 4D mmWave radar [7]–[9] extends the sensing capability from 2D to 3D space. Although both 4D radar and 3D LiDAR provide point clouds to describe 3D environments, 4D radar is more favorable for autonomous vehicles as the manufacturing cost of 4D radar is one order of magnitude lower than that of 3D LiDAR ($550 for Continental 77 GHz ARS408 radar vs. $4000 for Velodyne's 16-line LiDAR).

The downside of mmWave radar is that its point cloud is two orders of magnitude sparser than a LiDAR's data due to the larger wavelength of RF [6], [10] (compared with the nanometer-level wavelength of optical signals). The low spatial resolution comes from the RF's specular reflection and the low angular resolution. Current approaches adopt multi-sensor fusion to enhance the radar spatial resolution and enable various robot applications, *e.g.*, object detection, localization, and mapping, in vision/LiDAR-crippled environments [6], [8], [11]–[13]. However, these approaches must collect training data from expensive 3D LiDARs, typically equipped on a few specialized vehicles. The limited number of specialized vehicles will limit the scale of training data, being prone to the long tail of autonomous vehicle perception.

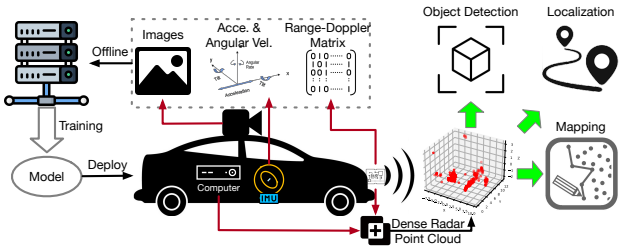


Fig. 1. mmEMP takes images, inertial measurements, and range-Doppler matrices (RDMs) to train a model for enhancing radar point clouds. In the test, the vehicle uses the enhanced point clouds to improve various applications, *e.g.*, object detection, localization, and mapping.

We propose mmEMP, a supervised learning approach that enhances the point cloud of a mmWave radar using camera images and inertial measurements, as shown in Fig. 1. Since inertial measurement units (IMUs) and cameras are low-cost and commonly installed on commercial vehicles like Tesla, mmEMP goes towards crowdsourcing the training data from intelligent vehicles. Typically, a LiDAR's dense point cloud directly provides 3D masks of objects in the scene. It is a solid supervision to guide the spatial information of real objects *w.r.t.* the radar data. In contrast, a visual image only provides 2D pixels, lacking depth information. Nevertheless, the state-of-the-art (SOTA) visual-inertial (VI) simultaneous localization and mapping (SLAM) techniques [14], [15] can reconstruct the 3D scene by dense visual features, holding the opportunity that provides LiDAR-like supervision.

In realizing mmEMP, we encounter two challenges. First, the fundamental of VI 3D reconstruction is epipolar geometry, assuming a fixed point in the world projects on multiple camera frames. A moving point breaks this assumption in that the projections on different frames correspond to points in its moving trajectory. The reconstructed result is the intersection of the moving point *w.r.t.* the camera center across multiple frames (refer to Fig. 3). Second, there exist spurious points that represent no real object due to the multipath reflection of RF signals. Densifying these spurious points will make robots misunderstand the world.

mmEMP addresses the above challenges by two modules. *First*, we devise a dynamic 3D reconstruction algorithm that restores the 3D positions of moving features. The movement breaks the epipolar constraint. Our algorithm builds upon the critical observation that all points on a rigid object share the

Authors[1] are with Wuhan University of Technology, Wuhan, China. {congfan, shengkai, kzliu}@whut.edu.cn
Author[2] is with Southeast University, Nanjing, China. shuaiwang@seu.edu.cn
Author[3] is with Tsinghua University, Beijing, China. hmilyyz@gmail.com
Author[4] is with Huazhong University of Science and Technology, Wuhan, China. weiwangw@hust.edu.cn
*Corresponding author: Shengkai Zhang (shengkai@whut.edu.cn).

identical translation between camera frames, allowing us to formulate the dynamic 3D reconstruction into a non-linear least square problem.

Second, we design a neural network pipeline that takes the VI 3D reconstruction to densify radar point clouds and eliminate spurious points. We eliminate spurious points by checking the spatial stability across consecutive frames. It requires the corresponding rigid transformations to transform coordinate frames. Thus, the neural network first estimates rigid transformation. Then we refine the point cloud using a spatial stability checking algorithm.

Contributions. mmEMP makes three contributions:

- We propose a dynamic VI 3D reconstruction algorithm that restores the 3D positions of dynamic visual features.
- We design a neural network pipeline that takes VI data and radar RDMs to enhance radar point clouds, estimate the vehicle's pose, and eliminate spurious radar points.
- We build a large dataset of radar RDMs, camera images, and IMU sequences along with open-source code[1].

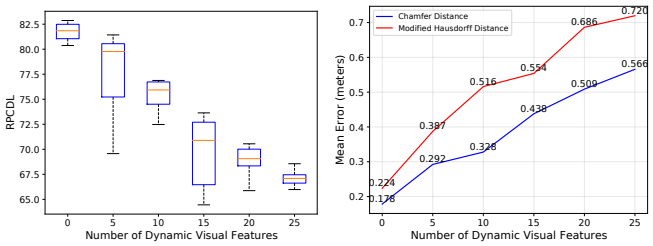
II. Related Work

Radar point cloud enhancement. Currently, point cloud enhancement approaches for mmWave radar either leverage advanced signal processing techniques, *e.g.*, Synthetic Aperture Radar (SAR) imaging [16]–[21], or fuse with other sensors, *e.g.*, camera and LiDAR [6], [8], [11]–[13], [22]–[30]. SAR imaging requires a vehicle to move along a specific trajectory precisely. For example, Qian *et al.* [16] exploited the natural linear motion of vehicle radar to improve sensing resolution. Thus, the radar perception will be inferior when an automobile stops at an intersection or turns sharply.

On the other hand, enhancing mmWave radar perception by fusing with other sensors has been attractive. Among them, some take radars to improve the perception of LiDARs or cameras to perform better in challenging environments [22]–[30]. The radar in these works cannot work alone. On the contrary, some other works try to enhance the spatial resolution of radars by the supervision of LiDAR's dense point clouds [6], [8], [11]–[13]. Although the training requires data from multiple sensors, the radar alone performs robustly in various robot applications. However, costly LiDAR prevents massive deployment and thus makes the system collect training data inefficient.

Dynamic visual feature tracking. Dynamic features have always been a challenge in SLAM. Currently, most SLAM techniques [14], [31]–[34] reject dynamic features to ensure pose estimation accuracy. Qiu *et al.* [35] simultaneously tracked the camera poses and dynamic objects by motion correlation analysis. They have proved that this problem is partially observable. Our dynamic 3D reconstruction is observable since we assume that dynamic features are outliers, so the camera pose can be first estimated. Moreover, we only need to recover the 3D positions of dynamic features rather than an object's 6-DoF pose. Ren *et al.* [36] and Xu *et al.* [37] focused on object-level tracking by learning-based semantic segmentation. In contrast, we focus on feature-level tracking for our point cloud enhancement.

[1]The open-source code and dataset of mmEMP is available at `https://github.com/bella-jy/mmEMP`.



(a) Statistical RPCDL. (b) Chamfer and Hausdorff distances.

Fig. 2. A preliminary study shows that dynamic visual features with wrong 3D positions significantly degenerate the performance of point cloud generation.

III. System Design of mmEMP

A. Dynamic Visual-Inertial 3D Reconstruction

mmEMP aims to enhance the spatial resolution of mmWave radar and generate dense point clouds using a low-cost VI sensor suite. The opportunity for this idea comes from the 3D reconstruction of VI SLAM [14], [15]. The reconstructed 3D positions of dense visual features may provide LiDAR-like guidance for radar point cloud generation.

Preliminary study. To verify this idea, we perform the VI 3D reconstruction by VINS [14] and input the 3D positions of visual features into the SOTA radar-based point cloud generation method [13]. Extensive experiments show that the moving visual features in the scene, *i.e.*, dynamic features, cause performance degeneration. We conduct the experiments by controlling the number of dynamic features from 0 to 30 with a step of 5. We conduct ten trials for each setting and plot the statistical result of the radar point cloud density level (RPCDL) [13] in Fig. 2(a). In addition, we calculate the mean Chamfer [38] and Hausdorff [39] distances to measure the point cloud similarity, as shown in Fig. 2(b).

When we select the scene without moving objects, the RPCDL (the larger, the better) is similar to [13]. However, the performance exhibits a remarkable decrease if we increase the number of dynamic features. In contrast, the mean Chamfer and Hausdorff distances increase along with more dynamic features, meaning that the generated point cloud differs from the ground truth point cloud obtained by the depth map from Intel Realsense D435. The reason is that VI SLAM techniques cannot restore dynamic features' 3D positions because they break the epipolar constraint for bundle adjustment. The VI 3D reconstruction cannot correctly capture all dynamic objects in the scene. Next, we elaborate on the problem of dynamic 3D reconstruction.

Problem formulation. Fig. 3 illustrates the geometry of a dynamic feature between two consecutive frames. Consider a moving point i observed on the previous frame at pixel[2] $\hat{\mathbf{p}}_i = [\hat{u}_{pi}, \hat{v}_{pi}, 1]^\top$ and on the current frame at pixel $\hat{\mathbf{q}}_i = [\hat{u}_{qi}, \hat{v}_{qi}, 1]^\top$, features $\hat{\mathbf{p}}_i$ and $\hat{\mathbf{p}}_i$ are paired by optical flow.

[2]In our formulation, we mark $\hat{(\cdot)}$ as known variables or constants. $\|\cdot\|$ denotes the L^2 norm unless otherwise specified. We also express the 2D pixel as its homogeneous coordinate.

The camera pose $\hat{\mathbf{T}} = [\hat{\mathbf{R}}, \hat{\mathbf{t}}]$, where $\hat{\mathbf{R}} \in \mathrm{SO}(3)$ and $\hat{\mathbf{t}} \in \mathbb{R}^3$, is known by VI SLAM [14], [15] as long as dynamic features are substantially fewer than background features, which is usually true in practice. The outlier rejection technique (RANSAC [40]) of VI SLAM makes the robot pose estimation robust to these dynamic features.

Our goal is to calculate the 3D position $\mathbf{P}_i \in \mathbb{R}^3$ *w.r.t.* the previous frame and the translation $\Delta\mathbf{d} \in \mathbb{R}^3$ when it is observed on the current frame. The epipolar constraint requires a fixed point in the scene, while the dynamic point has a translation when observed in the second frame. Existing techniques calculate the intersection $\mathbf{P}'_i$ of the moving point *w.r.t.* the camera center.

The translation makes the point re-projection no longer exists. Instead, both frames share the projection of the translation. Consider $\mathbf{O}_1 = (0, 0, 0)$, from the cosine law, we have the following equations (refer to Fig. 3):

$$\|\mathbf{P}_i\|^2 + \|\mathbf{Q}_i\|^2 - 2\|\mathbf{P}_i\| \cdot \|\mathbf{Q}_i\| \cos\theta_1 - \|\Delta\mathbf{d}\|^2 = 0, \quad (1)$$

$$\|\mathbf{P}_i - \hat{\mathbf{t}}\|^2 + \|\mathbf{Q}_i - \hat{\mathbf{t}}\|^2 - 2\|\mathbf{P}_i - \hat{\mathbf{t}}\| \cdot \|\mathbf{Q}_i - \hat{\mathbf{t}}\| \cos\theta_2 - \|\Delta\mathbf{d}\|^2 = 0, \quad (2)$$

where $\mathbf{Q}_i = \mathbf{P}_i + \Delta\mathbf{d}$.

To explicitly express $\cos\theta_1$ and $\cos\theta_2$ *w.r.t.* $\mathbf{P}_i$ and $\Delta\mathbf{d}$, we define the pseudo-projections of the moving point i. $\mathbf{m}_i \in \mathbb{R}^3$ denotes the pseudo-pixel homogeneous coordinate of point $\mathbf{Q}_i$ in the previous frame. $\mathbf{n}_i \in \mathbb{R}^3$ is the pseudo-projection homogeneous coordinate of point $\mathbf{P}_i$ in the current frame. Then, we have

$$\cos\theta_1 = \frac{\hat{\mathbf{p}}_i \cdot \mathbf{m}_i}{\|\hat{\mathbf{p}}_i\| \cdot \|\mathbf{m}_i\|}, \quad \cos\theta_2 = \frac{\hat{\mathbf{q}}_i \cdot \mathbf{n}_i}{\|\hat{\mathbf{q}}_i\| \cdot \|\mathbf{n}_i\|}. \quad (3)$$

Next, we express the pseudo-projections $\mathbf{m}_i$ and $\mathbf{n}_i$ *w.r.t.* the target unknowns $\mathbf{P}_i$ and $\Delta\mathbf{d}$.

Given the camera intrinsic matrix $\hat{\mathbf{K}} = \begin{bmatrix} \hat{f}_x & 0 & \hat{c}_x \\ 0 & \hat{f}_y & \hat{c}_y \\ 0 & 0 & 1 \end{bmatrix}$ and $\mathbf{P}_i = [p_{ix},\ p_{iy},\ p_{iz}]^\top$, $\Delta\mathbf{d} = [\Delta d_x,\ \Delta d_y,\ \Delta d_z]^\top$, the pinhole imaging model gives:

$$\mathbf{m}_i = \begin{bmatrix} \hat{f}_x \frac{p_{ix}+\Delta d_x}{p_{iz}+\Delta d_z} + \hat{c}_x & \hat{f}_y \frac{p_{iy}+\Delta d_y}{p_{iz}+\Delta d_z} + \hat{c}_y & 1 \end{bmatrix}^\top. \quad (4)$$

In addition, the imaging model also gives a measurement model,

$$\hat{\mathbf{p}}_i = \begin{bmatrix} \hat{f}_x \frac{p_{ix}}{p_{iz}} + \hat{c}_x & \hat{f}_y \frac{p_{iy}}{p_{iz}} + \hat{c}_y & 1 \end{bmatrix}^\top. \quad (5)$$

We now define the homogeneous coordinate of point i in the previous frame as $\tilde{\mathbf{P}}_i = [\mathbf{P}_i; 1]$. In addition, we re-parameterize the known pose $\hat{\mathbf{T}} = \begin{bmatrix} \hat{\mathbf{R}}, \hat{\mathbf{t}} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{t}}_1 & \hat{\mathbf{t}}_2 & \hat{\mathbf{t}}_3 \end{bmatrix}^\top$, where $\hat{\mathbf{t}}_j \in \mathbb{R}^{1\times 4}$, $j = \{1,\ 2,\ 3\}$, denotes j^{th}row of $\hat{\mathbf{T}}$. Then the image projection theory gives:

$$s \cdot \mathbf{n}_i = \begin{bmatrix} \hat{\mathbf{t}}_1 & \hat{\mathbf{t}}_2 & \hat{\mathbf{t}}_3 \end{bmatrix}^\top \tilde{\mathbf{P}}_i, \quad (6)$$

where s denotes the scale variable. Eliminating s by the last row of the above equation writes

$$\mathbf{n}_i = \begin{bmatrix} \frac{\hat{\mathbf{t}}_1 \tilde{\mathbf{P}}_i}{\hat{\mathbf{t}}_3 \tilde{\mathbf{P}}_i} & \frac{\hat{\mathbf{t}}_2 \tilde{\mathbf{P}}_i}{\hat{\mathbf{t}}_3 \tilde{\mathbf{P}}_i} & 1 \end{bmatrix}^\top. \quad (7)$$

Similarly, we have another measurement model with pixel $\hat{\mathbf{q}}_i$

$$\hat{\mathbf{q}}_i = \begin{bmatrix} \frac{\hat{\mathbf{t}}_1 (\tilde{\mathbf{P}}_i + \Delta\tilde{\mathbf{d}})}{\hat{\mathbf{t}}_3 (\tilde{\mathbf{P}}_i + \Delta\tilde{\mathbf{d}})} & \frac{\hat{\mathbf{t}}_2 (\tilde{\mathbf{P}}_i + \Delta\tilde{\mathbf{d}})}{\hat{\mathbf{t}}_3 (\tilde{\mathbf{P}}_i + \Delta\tilde{\mathbf{d}})} & 1 \end{bmatrix}^\top. \quad (8)$$

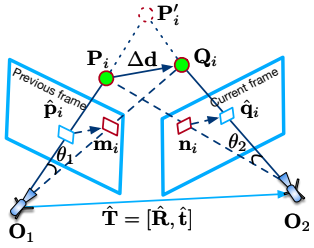


Fig. 3. The geometry of a dynamic feature between two camera frames.

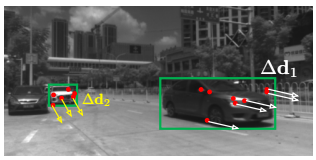


Fig. 4. Dynamic features on a rigid object share the same translation.

At this stage, all terms in Eqn. (1) and (2) are expressed *w.r.t.* the target unknowns $\mathbf{P}_i$ and $\Delta\mathbf{d}$.

Combing Eqn. (1), (2), (5), and (8) writes the overall measurement model

$$\hat{\mathbf{z}}_i = \begin{bmatrix} 0 & 0 & \hat{u}_{pi} & \hat{v}_{pi} & \hat{u}_{qi} & \hat{v}_{qi} \end{bmatrix}^\top = F(\mathbf{P}_i,\ \Delta\mathbf{d}), \quad (9)$$

where $F(\mathbf{P}_i,\ \Delta\mathbf{d})$ denotes the stacked vector of 6 expressions *w.r.t.* unknown $\mathbf{P}_i$ and $\Delta\mathbf{d}$. Apparently, we cannot find the unique solution from Eqn. (9) due to its non-linearity.

Our key observation is that dynamic features on a rigid object share the same translation $\Delta\mathbf{d}$ between camera frames, as shown in Fig. 4. One more dynamic feature incurs 3 more variables, *i.e.*, its 3D position, but adds 6 more measurement equations. Moreover, the SOTA image segmentation tool [41] can separate the features of different rigid objects.

Thus, we formulate the dynamic 3D reconstruction as a non-linear least square problem. We assume N dynamic features on a rigid object. The unknown vector is $\mathbf{X} = (\mathbf{P}_1, \mathbf{P}_2, \cdots, \mathbf{P}_N, \Delta\mathbf{d})$. Then, we have the following optimization problem

$$\mathbf{X}^* = \arg\min_{\mathbf{X}} \frac{1}{2} \sum_{i=1}^{N} \|\mathbf{e}_i(\mathbf{X})\|^2, \quad (10)$$

where $\mathbf{e}_i(\mathbf{X}) = \hat{\mathbf{z}}_i - F(\mathbf{P}_i,\ \Delta\mathbf{d})$. As long as $N \geq 2$ for a rigid object, the features' 3D positions and the translation can be recovered. We use Ceres Solver [42], an open-source C++ library, to solve this non-linear optimization problem.

B. Point Cloud Generation and Refinement

The point cloud generation using the supervision of 3D visual point clouds obtained in § III-A is similar to [13]. Our technical contribution lies in the point cloud refinement. We refine the point cloud by eliminating spurious points via inferring the vehicle's drift-free rigid transformations and checking the point spatial stability. The data processing pipeline is shown in Fig. 5.

Point cloud generation. We input raw RDMs and output dense point clouds with the supervision of a ground-truth label matrix *w.r.t.* real objects from our dynamic VI 3D reconstruction. In particular, we apply a generative adversarial network [43] that consists of a generator and a discriminator. Since the number of real target cells in the RDM is a minority, we use the focal loss [44] to deal with the sample imbalance problem. For more network details, please refer to [13].

Point cloud refinement. It is well known that RF multipath reflections cause spurious points [12], [45]. Our idea to

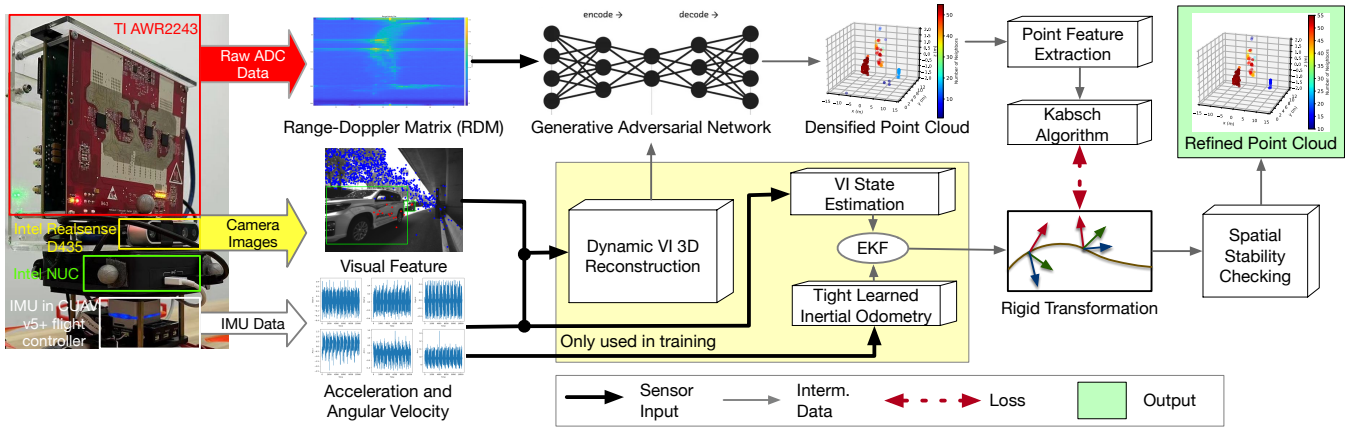


Fig. 5. An overview of our data processing pipeline. The measurements from the visual-inertial sensor suit are only used in training (modules in the yellow box) so that mmEMP works fine in adverse weather conditions.

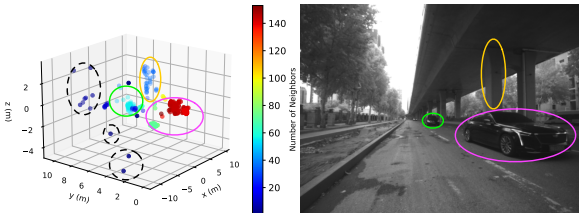


Fig. 6. The naive spatial stability checking with known rigid transformations over a few adjacent frames. Solid circles are colored for different objects. Dotted dark circles highlight the spurious points.

remove such spurious points is to leverage spatial stability. Intuitively, points on real objects show stable appearances in adjacent frames. On the contrary, spurious points are the mirror points of real objects *w.r.t.* the receiving path of multipath reflections. Their appearances show a random spatial pattern over time. Thus, if we superimpose multiple frames of point clouds *w.r.t.* the initial frame, points on real objects should see many more neighbors than spurious points.

To verify our idea, we conduct experiments in the outdoor scene with ground-truth rigid transformations provided by VINS [14]. Fig. 6 shows the preliminary result. We color points with different numbers of neighbors. For ease of verification, we stack 5 adjacent frames of point clouds and set a distance threshold (0.5 m) to define a point's neighborhood. The result highlights the spurious points with dark blue colors in the scene. Some are positioned below the ground with a height of −4.2 meters. The result indicates that this approach is promising to identify spurious points.

In practice, mmEMP aims to work in adverse weather where VI SLAM fails. Moreover, fixing a distance threshold to define the neighborhood will result in false positives of spurious points. Thus, we have the following designs.

1) VI-supervised rigid transformation learning. We obtain reliable rigid transformations with radar point clouds through a supervised learning framework. The supervision of rigid transformation is from VI state estimation [14], [15]. However, VI state estimations require loop closure to optimize its temporal drift. To mitigate the drift, we leverage the SOTA inertial-only state estimator, tight learned inertial odometry (TLIO) [46], to be an independent source of rigid transformation. The inertial-only state estimation is enabled by extracting the motion pattern hidden in a segment of IMU measurements through a deep neural network. We fuse the rigid transformation of the VI state estimator [14] with that of TLIO [46] by applying the extended Kalman filter (EKF) to obtain pseudo-ground-truth rigid transformations $\mathbf{T}_{\text{truth}}$ between radar frames (refer to Fig. 5).

We then use $\mathbf{T}_{\text{truth}}$ to supervise the inference of rigid transformations from radar point clouds. We first input the densified point cloud into the SOTA point feature extraction network [8]. The network gives a scene flow prediction $\hat{\mathbf{F}} = \{\hat{\mathbf{f}}_i \in \mathbb{R}^3\}_{i=1}^{M}$, where M denotes the number of points in a radar frame, and a moving probability map $\hat{\mathbf{M}} = \{\hat{m}_i \in [0,1]\}_{i=1}^{M}$, where $\hat{m}_i \geq 0.5$ indicates point i is moving. Based on $\hat{\mathbf{F}}$ and $\hat{\mathbf{M}}$, we then infer the radar rigid transformation $\hat{\mathbf{T}} \in \text{SE}(3)$ by the Kabsch algorithm [47].

We aim to adjust the inferred radar transformation $\hat{\mathbf{T}}$ to approach the pseudo-ground-truth transformation $\mathbf{T}$ by a loss function. Consider two consecutive radar frames, namely, the previous frame $\mathcal{F}_p$ and the current frame $\mathcal{F}_c$, and the rigid transformation $\mathbf{T}_p^c = \begin{bmatrix} \mathbf{R}_p^c & \mathbf{t}_p^c \\ \mathbf{0} & 1 \end{bmatrix}$ from $\mathcal{F}_p$ to $\mathcal{F}_c$. Then we can transform a point $\mathbf{f}_{pi}$ in $\mathcal{F}_p$ into $\mathcal{F}_c$ as $\mathbf{f}_{ci} = \mathbf{R}_p^c \mathbf{f}_{pi} + \mathbf{t}_p^c$. Thus, we can write the loss function of our supervised learning as

$$L_{\text{trans.}} = \frac{1}{M} \sum_{i=1}^{M} \left\| \left(\mathbf{R}^\top \hat{\mathbf{R}} - \mathbf{I}_3 \right) \mathbf{f}_{pi} + \mathbf{t} - \hat{\mathbf{t}} \right\|, \tag{11}$$

where $(\mathbf{R}, \mathbf{t})$ and $\left(\hat{\mathbf{R}}, \hat{\mathbf{t}}\right)$ are the rotation and translation components in $\mathbf{T}$ and $\hat{\mathbf{T}}$, respectively.

2) Velocity-adaptive spatial stability checking. Through extensive experiments on the naive algorithm for idea verification, we realize there is no fixed threshold to determine a point's neighborhood for spatial stability checking. Intuitively, if a vehicle travels faster, the spatial distribution of radar points across multiple frames will be larger. Thus, we design a spatial stability checking algorithm shown in Algorithm 1.

For each point cloud frame, we first compute the relative velocities of background and dynamic points *w.r.t.* the current

(0^{th}) frame. The velocity of background points is the vehicle's (Lines 4 – 5). However, dynamic points' relative velocities depend on the moving speeds of rigid objects. We compute this using the estimated translation by our dynamic 3D reconstruction in § III-A (Lines 8 – 11). After traversing all points of all frames, we compute the mean background velocity $\mathbf{v}_b$ and the mean velocity $\mathbf{v}_d^j$ for dynamic points on rigid object j (Lines 17 – 18). Then, background and dynamic points' neighborhoods are determined (Line 19). Note that d_0 is a constant to reserve a minimum neighborhood range. We set $d_0 = 0.5$ m and $F = 5$ in experiments. In the *for* loop, we superimpose the consecutive frames to the current frame (Line 7). At last, we count the neighbors of each point in the current frame and mark "spurious" if the number of a point's neighbors is lower than the 5^{th}percentile (Lines 20 – 23).

Algorithm 1 Velocity-adaptive Spatial Stability Checking

1: Input: Consecutive F frames of point clouds, corresponding relative timestamp Δt_i and rigid transformation $\mathbf{T}_i^0$ *w.r.t.* the current (0^{th}) frame, $i = 1, 2, \cdots, F-1$
2: Velocity set for dynamic points in rigid object j, $\mathcal{V}_d^j = \emptyset$, $j = 1, \cdots, G$; velocity set for background points $\mathcal{V}_b = \emptyset$
3: **for** $i = 1 : F-1$ **do**
4: Compute the radar's velocity $\mathbf{v}_i^0$ by $\mathbf{T}_i^0$ and timestamps
5: $\mathcal{V}_b \leftarrow \mathbf{v}_i^0$
6: **for** each point $\mathbf{p}^i$ in i^{th}frame **do**
7: Transform $\mathbf{p}^i$ to the current frame $\mathbf{p}_i^0$ by $\mathbf{T}_i^0$
8: **if** $\mathbf{p}^i$ is a dynamic point in rigid object j **then**
9: Compute this point's velocity $\mathbf{v}_p^j = \frac{\Delta \mathbf{d}}{\Delta t_i}$
10: $\mathbf{v}_d^j = \mathbf{v}_i^0 - \mathbf{v}_p^j$
11: **end if**
12: **end for**
13: **for** $j = 1 : G$ **do**
14: $\mathcal{V}_d^j \leftarrow \mathbf{v}_d^j$
15: **end for**
16: **end for**
17: Compute the time span of all frames $\Delta t = \sum_{i=1}^{F-1} \Delta t_i$
18: Compute mean background velocity $\mathbf{v}_b = \text{mean}\{\mathcal{V}_b\}$ and mean velocity $\mathbf{v}_d^j = \text{mean}\{\mathcal{V}_d^j\}$ for all dynamic objects
19: Background points' neighborhood range $d_b = \max\{d_0, \frac{1}{2}\|\mathbf{v}_b\|\Delta t\}$; dynamic points in object j have neighborhood range $d_j = \max\{d_0, \frac{1}{2}\|\mathbf{v}_d^j\|\Delta t\}$
20: **for** each point p in 0^{th}frame **do**
21: Count neighbors of point p as N_p
22: **end for**
23: Mark point p in 0^{th}frame "spurious" if N_p is lower than the 5^{th}percentile.

IV. System Implementation and Evaluation

A. Implementation and Experimental Setup

Currently, only the RPDNet dataset [48] provides raw radar RDMs with, however, LiDAR point clouds. No public dataset provides radar RDMs, monocular images, and IMU measurements. Therefore, we collect our dataset through a customized platform.

Hardware and dataset. As shown in the left picture in Fig. 5, the customized platform includes a 4D mmWave radar, a camera, and an IMU. The 4D radar is cascaded by four TI AWR2243, which contains 12 transmitting antennas and 16 receiving antennas. We use the Intel Realsense D435 to provide depth images and monocular images. We obtain the ground-truth 3D point cloud of visual features transformed from the depth images. We use the high-precision IMU integrated in CUAV v5+ flight controller to provide accelerations and angular velocities. All sensor data are timestamped and managed by the Intel NUC 11TNKi5 running Ubuntu 20.04 with robot operating system (ROS). At last, we transmit the dataset into a server for training.

Dataset and training. Our dataset includes 187200 IMU sequences and 62400 pairs of camera images and radar range-doppler matrices in indoor and outdoor scenes. We collect the outdoor data along a city road outside the campus, covering a duration over 2080 seconds in a total distance of 5720 meters. The indoor data are collected in our lab and offices. Across all our data, we use 2288 meters long trajectories with 24960 pairs of camera and radar data for training. During training, we take all sensor measurements. In the testing, we only input radar RDMs.

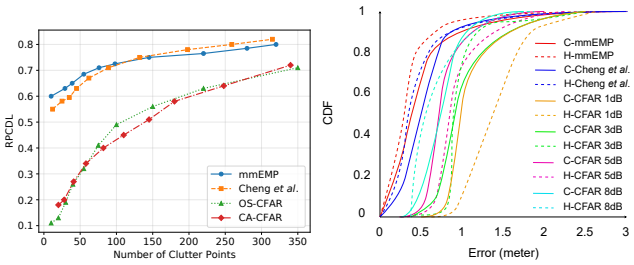
Ground-truth point cloud. We obtain the ground-truth point clouds from stereo images provided by Intel Realsense D435. Specifically, we set the same resolution for monocular images and stereo images. Then we first track scale-invariant feature transform features [49] and calculate their 3D points from the corresponding pixels in the stereo depth maps

Baselines. We compare with Cheng *et al.* [13], CA-CFAR, and OS-CFAR [50]. Cheng *et al.* [13] represents the SOTA 3D radar point cloud generation. CA-CFAR (Cell Averaging Constant False Alarm Rate) and OS-CFAR (Order Statistic CFAR) are two common techniques for target detection and estimation while maintaining a constant false alarm rate.

B. Performance Evaluation

1) Point cloud enhancement. We would like to see if the enhanced point cloud is similar to the ground-truth point cloud. We use three metrics to evaluate the similarity: RPCDL [13], Chamfer distance [38], and modified Hausdorff distance [39]. A higher RPCDL indicates that the radar point clouds are closer to the ground-truth point clouds. Chamfer and modified Hausdorff distances find the nearest neighbor for each point in one point cloud to another and takes the mean and median of all these distance respectively.

Fig. 7(a) shows the relationship between the number of clutter points and the RPCDL. A radar clutter point is defined as the point who fails to find any stereo-generated point within a distance threshold δ. In our experiments, $\delta = 1$ m. It can be seen that mmEMP performs similar to Cheng *et al.* [13], which enhances point clouds with a costly LiDAR. When the number of clutter points is about 50, mmEMP generates more than two times denser point clouds than the ones generated by OS-CFAR and CA-CFAR. Fig. 7(b) shows that mmEMP achieves the Chamfer and modified Hausdorff distances with the median errors of 0.38 m and 0.30 m, respectively, being



(a) RPCDL (b) Chamfer and M. Hausd.

Fig. 7. Point cloud similarity.

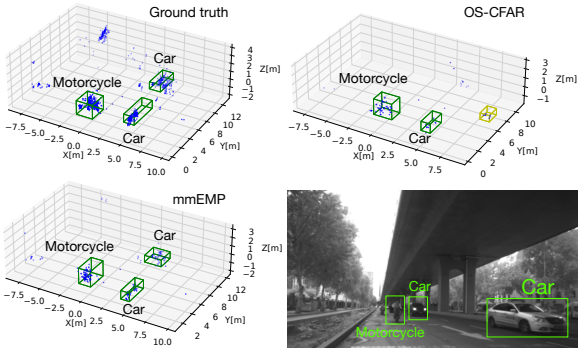


Fig. 8. Object detection from ground-truth point clouds and radar point clouds generated by OS-CFAR and mmEMP.

TABLE I

ABLATION STUDY FOR MMEMP MODULES

DVIR[1]	PR[2]	RPCDL↑	Chamfer (m)↓	mod. Hausdorff (m)↓
		59.07	0.342	0.387
✓		74.39	0.233	0.286
	✓	68.68	0.258	0.312
✓	✓	**78.64**	**0.192**	**0.248**

[1]DVIR denotes the dynamic VI 3D reconstruction module.
[2]PR denotes the point refinement module.

close to Cheng *et al.* [13] with the median errors of 0.50 m and 0.34 m. This figure evaluates the CA-CFAR with threshold from 1 dB to 8 dB. The increase of the threshold decreases the point cloud density. Compare with CFAR methods, mmEMP is much more similar to the ground-truth point cloud.

2) Ablation study. We evaluate the effectiveness of each module through ablation experiments, as shown in Table I. Without any proposed module, we input the conventional VI 3D reconstruction [14] into Cheng *et al.* [13]. The three metrics are the worst because dynamic visual features distort the 3D perception and spurious points bring fake objects. The results show that our dynamic 3D reconstruction leads to the biggest performance gain (refer to 2nd row). 3rd row shows the performance gain provided by the point cloud refinement.

3) Application on subtasks. We demonstrate the effectiveness of mmEMP enhanced point clouds by object detection, localization, and mapping. Fig. 8 shows that the spurious points in OS-CFAR lead to a false alarm detection, *i.e.*, the yellow box. mmEMP generates a much denser point cloud so as to produce accurate object bounding boxes. Fig. 9 shows the localization qualitative results of two trajectories

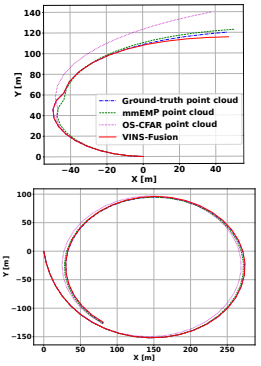


Fig. 9. Localization using ground-truth and radar point clouds from OS-CFAR and mmEMP.

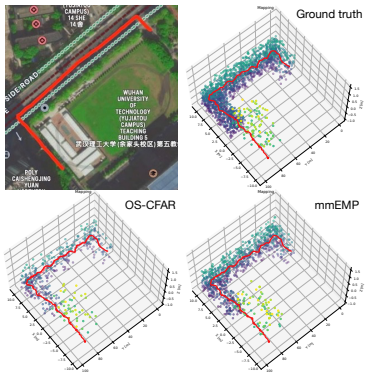


Fig. 10. Mapping using ground-truth point clouds and radar point clouds from OS-CFAR and mmEMP.

in the outdoor scene. The ground-truth odometry is obtained by VINS-Fusion [51] with GNSS readings to prevent the temporal drift of VINS-Mono [14] without loop closure. The localization results with different point clouds are from the Kabsch algorithm in our neural network pipeline (refer to Fig. 5). The mean localization errors *w.r.t.* the upper trajectory of using point clouds from the ground truth, mmEMP, and OS-CFAR are 1.38 m, 2.19 m, and 5.62 m, respectively. For the bottom trajectory, the errors are 1.43 m, 2.04 m, and 5.66 m, respectively. Fig. 10 shows the mapping qualitative results. The mapping errors in Chamfer distance of using point clouds from mmEMP and OS-CFAR are 2.51 m and 3.06 m, respectively.

V. CONCLUSION

mmEMP creates dense point clouds from a single-chip mmWave radar to support autonomous driving through a supervised learning pipeline. We use the supervision from a camera-IMU sensor suite, which is low-cost and commonly available in commercial vehicles, enabling crowdsourcing training data. mmEMP overcomes the challenges arising from dynamic visual features and spurious radar points with a dynamic 3D reconstruction algorithm and a neural network design. We collect a large dataset of image-radar pairs with IMU sequences and conduct experiments in indoor and outdoor scenes. The results show that mmEMP generates denser point clouds without spurious points and thus contributes more to the perceptional tasks, including object detection, localization, and mapping. In the future, we will explore the potential of the enhanced radar data by fusing it with other sensors, *e.g.*, event-based camera and WiFi beamforming reports, to build more robust robot perceptional systems at high speeds and in human-robot interactions.

ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China (NSFC) under grant No. 52031009, in part by the Natural Science Foundation of Hubei Province, China, under grant No. 2021CFA001, in part by the NSFC under grant No. 62272098, No. 62372265, and No. 62302254, in part by the National Key Research Plan under grant No. 2021YFB2900100.

References

[1] S. Wang, D. Cao, R. Liu, W. Jiang, T. Yao, and C. X. Lu, "Human parsing with joint learning for dynamic mmwave radar point cloud," *ACM Proc. IMWUT*, vol. 7, no. 1, pp. 1–22, 2023.

[2] H. Li, R. Liu, S. Wang, W. Jiang, and C. X. Lu, "Pedestrian liveness detection based on mmwave radar and camera fusion," in *IEEE Proc. SECON*, 2022, pp. 262–270.

[3] D. Cao, R. Liu, H. Li, S. Wang, W. Jiang, and C. X. Lu, "Cross vision-rf gait re-identification with low-cost rgb-d cameras and mmwave radars," *ACM Proc. IMWUT*, vol. 6, no. 3, pp. 1–25, 2022.

[4] K. Cai, B. Wang, and C. X. Lu, "Autoplace: Robust place recognition with single-chip automotive radar," in *IEEE Proc. ICRA*, 2022, pp. 2222–2228.

[5] P. Gao, S. Zhang, W. Wang, and C. X. Lu, "Dc-loc: Accurate automotive radar based metric localization with explicit doppler compensation," in *IEEE Proc. ICRA*, 2022, pp. 4128–4134.

[6] A. Prabhakara, T. Jin, A. Das, G. Bhatt, L. Kumari, E. Soltanaghai, J. Bilmes, S. Kumar, and A. Rowe, "High resolution point clouds from mmwave radar," in *IEEE Proc. ICRA*, 2023, pp. 4135–4142.

[7] M. Jiang, G. Xu, H. Pei, Z. Feng, S. Ma, H. Zhang, and W. Hong, "4d high-resolution imagery of point clouds for automotive mmwave radar," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–15, 2023.

[8] F. Ding, A. Palffy, D. M. Gavrila, and C. X. Lu, "Hidden gems: 4d radar scene flow learning using cross-modal supervision," in *IEEE Proc. CVPR*, 2023, pp. 9340–9349.

[9] F. Ding, Z. Pan, Y. Deng, J. Deng, and C. X. Lu, "Self-supervised scene flow estimation with 4-d automotive radar," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 8233–8240, 2022.

[10] H. Dong, Y. Xie, X. Zhang, W. Wang, X. Zhang, and J. He, "Gpsmirror: Expanding accurate gps positioning to shadowed and indoor regions with backscatter," in *Proc. ACM MobiCom*, 2023.

[11] C. X. Lu, M. R. U. Saputra, P. Zhao, Y. Almalioglu, P. P. De Gusmao, C. Chen, K. Sun, N. Trigoni, and A. Markham, "milliego: single-chip mmwave radar aided egomotion estimation via deep sensor fusion," in *ACM Proc. SenSys*, 2020, pp. 109–122.

[12] C. X. Lu, S. Rosa, P. Zhao, B. Wang, C. Chen, J. A. Stankovic, N. Trigoni, and A. Markham, "See through smoke: robust indoor mapping with low-cost mmwave radar," in *ACM Proc. MobiSys*, 2020, pp. 14–27.

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

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

[15] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[16] K. Qian, Z. He, and X. Zhang, "3d point cloud generation with millimeter-wave radar," *ACM Proc IMWUT*, vol. 4, no. 4, pp. 1–23, 2020.

[17] H. Yamada, T. Kobayashi, Y. Yamaguchi, and Y. Sugiyama, "High-resolution 2d sar imaging by the millimeter-wave automobile radar," in *IEEE Proc. CAMA*, 2017, pp. 149–150.

[18] M. T. Ghasr, M. J. Horst, M. R. Dvorsky, and R. Zoughi, "Wideband microwave camera for real-time 3-d imaging," *IEEE Transactions on Antennas and Propagation*, vol. 65, no. 1, pp. 258–268, 2016.

[19] C. M. Watts, P. Lancaster, A. Pedross-Engel, J. R. Smith, and M. S. Reynolds, "2d and 3d millimeter-wave synthetic aperture radar imaging on a pr2 platform," in *IEEE Proc. IROS*, 2016, pp. 4304–4310.

[20] J. Guan, S. Madani, S. Jog, S. Gupta, and H. Hassanieh, "Through fog high-resolution imaging using millimeter wave radar," in *IEEE Proc. CVPR*, 2020, pp. 11 464–11 473.

[21] Z. Luo, Q. Zhang, W. Wang, and T. Jiang, "Single-antenna device-to-device localization in smart environments with backscatter," *IEEE Internet of Things Journal*, vol. 9, no. 12, pp. 10 121–10 129, 2021.

[22] Y. Yang, J. Liu, T. Huang, Q.-L. Han, G. Ma, and B. Zhu, "Ralibev: Radar and lidar bev fusion learning for anchor box free object detection system," *arXiv preprint arXiv:2211.06108*, 2022.

[23] Y. Wang, J. Deng, Y. Li, J. Hu, C. Liu, Y. Zhang, J. Ji, W. Ouyang, and Y. Zhang, "Bi-lrfusion: Bi-directional lidar-radar fusion for 3d dynamic object detection," in *IEEE Proc. CVPR*, 2023, pp. 13 394–13 403.

[24] Y. Kim, J. W. Choi, and D. Kum, "Grif net: Gated region of interest fusion network for robust 3d object detection from radar point cloud and monocular image," in *IEEE Proc. IROS*, 2020, pp. 10 857–10 864.

[25] R. Nabati and H. Qi, "Centerfusion: Center-based radar and camera fusion for 3d object detection," in *IEEE Proc. WACV*, 2021, pp. 1527–1536.

[26] J.-J. Hwang, H. Kretzschmar, J. Manela, S. Rafferty, N. Armstrong-Crews, T. Chen, and D. Anguelov, "Cramnet: Camera-radar fusion with ray-constrained cross-attention for robust 3d object detection," in *Proc. ECCV*, 2022, pp. 388–405.

[27] Y. Wang, Z. Jiang, Y. Li, J.-N. Hwang, G. Xing, and H. Liu, "Rodnet: A real-time radar object detection network cross-supervised by camera-radar fused object 3d localization," *IEEE Journal of Selected Topics in Signal Processing*, vol. 15, no. 4, pp. 954–967, 2021.

[28] J.-T. Huang, C.-L. Lu, P.-K. Chang, C.-I. Huang, C.-C. Hsu, P.-J. Huang, H.-C. 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*, vol. 6, no. 2, pp. 3333–3340, 2021.

[29] R. Nabati and H. Qi, "Radar-camera sensor fusion for joint object detection and distance estimation in autonomous vehicles," *arXiv preprint arXiv:2009.08428*, 2020.

[30] K. Qian, S. Zhu, X. Zhang, and L. E. Li, "Robust multimodal vehicle detection in foggy weather using complementary lidar and radar signals," in *IEEE Proc. CVPR*, 2021, pp. 444–453.

[31] S. Song, H. Lim, A. J. Lee, and H. Myung, "Dynavins: A visual-inertial slam for dynamic environments," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11 523–11 530, 2022.

[32] Y. Fan, H. Han, Y. Tang, and T. Zhi, "Dynamic objects elimination in slam based on image fusion," *Pattern Recognition Letters*, vol. 127, pp. 191–201, 2019.

[33] B. Canovas, M. Rombaut, A. Nègre, D. Pellerin, and S. Olympieff, "Speed and memory efficient dense rgb-d slam in dynamic scenes," in *IEEE Proc. IROS*, 2020, pp. 4996–5001.

[34] W. Dai, Y. Zhang, P. Li, Z. Fang, and S. Scherer, "Rgb-d slam in dynamic environments using point correlations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 1, pp. 373–389, 2020.

[35] K. Qiu, T. Qin, W. Gao, and S. Shen, "Tracking 3-d motion of dynamic objects using monocular visual-inertial sensing," *IEEE Transactions on Robotics*, vol. 35, no. 4, pp. 799–816, 2019.

[36] Y. Ren, B. Xu, C. L. Choi, and S. Leutenegger, "Visual-inertial multi-instance dynamic slam with object-level relocalisation," in *IEEE Proc. IROS*, 2022, pp. 11 055–11 062.

[37] B. Xu, W. Li, D. Tzoumanikas, M. Bloesch, A. Davison, and S. Leutenegger, "Mid-fusion: Octree-based object-level multi-instance dynamic slam," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 5231–5237.

[38] A. Bell, B. Chambers, and H. Butler, "Point data abstraction library," https://pdal.io/en/latest/apps/chamfer.html, 2023.

[39] ——, "Point data abstraction library," https://pdal.io/en/latest/apps/hausdorff.html, 2023.

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

[41] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," *arXiv:2304.02643*, 2023.

[42] S. Agarwal, K. Mierle, and Others, "Ceres solver," http://ceres-solver.org.

[43] P. Luc, C. Couprie, S. Chintala, and J. Verbeek, "Semantic segmentation using adversarial networks," in *NIPS Workshop on Adversarial Training*, 2016.

[44] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *IEEE Proc. ICCV*, 2017, pp. 2980–2988.

[45] K. Bansal, K. Rungta, S. Zhu, and D. Bharadia, "Pointillism: Accurate 3d bounding box estimation with multi-radars," in *ACM Proc. Sensys*, 2020, pp. 340–353.

[46] W. Liu, D. Caruso, E. Ilg, J. Dong, A. I. Mourikis, K. Daniilidis, V. Kumar, and J. Engel, "Tlio: Tight learned inertial odometry," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5653–5660, 2020.

[47] W. Kabsch, "A solution for the best rotation to relate two sets of vectors," *Acta Crystallographica Section A: Crystal Physics, Diffraction,*

Theoretical and General Crystallography, vol. 32, no. 5, pp. 922–923, 1976.
[48] "Rpdnet," https://github.com/thucyw/RPDNet, Online; Accessed: 14 Sep., 2023.
[49] D. G. Lowe, "Object recognition from local scale-invariant features," in *IEEE Proc. ICCV*, vol. 2, 1999, pp. 1150–1157.
[50] M. A. Richards, *Fundamentals of radar signal processing*. McGraw-Hill Education, 2014.
[51] T. Qin, J. Pan, S. Cao, and S. Shen, "A general optimization-based framework for local odometry estimation with multiple sensors," 2019.