FMCW-LIO: A Doppler LiDAR-Inertial Odometry

Mingle Zhao, Jiahao Wang, Tianxiao Gao, Chengzhong Xu, *Fellow, IEEE*, and Hui Kong

***Abstract*—Conventional LiDAR-inertial odometry (LIO) or SLAM methods heavily rely on geometric features of environments, as LiDARs primarily provide range measurements instead of motion measurements. From now on, however, the situation changes thanks to the novel Frequency Modulated Continuous Wave (FMCW) LiDARs. FMCW LiDARs not only offer the point range with high resolution but also capture the instant point Doppler velocity through the Doppler effect. In the letter, we propose FMCW-LIO, a novel and robust LIO, leveraging intrinsic Doppler measurements from FMCW LiDARs. To correctly exploit Doppler velocities, a motion compensation method is designed, and a Doppler-aided observation model is applied for on-manifold state estimation. Then, dynamic points can be effectively removed by the Doppler criteria, deriving more consistent geometric observations. FMCW-LIO eventually achieves accurate state estimation and static mapping, even in structure-degenerated environments. Extensive experiments in diverse scenes are performed and FMCW-LIO outperforms other algorithms on both accuracy and robustness.**

***Index Terms*—Sensor Fusion, Localization, SLAM, State Estimation, FMCW LiDAR.**

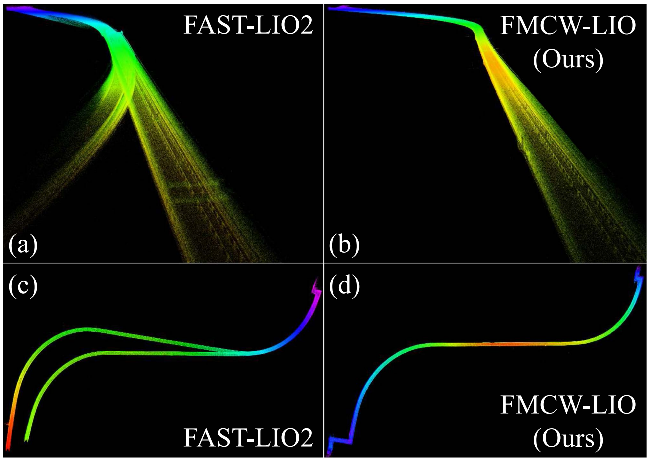

Fig. 1. Online mapping comparison of the whole S-shaped tunnel. (a) and (c) show the drifted and inconsistent maps of FAST-LIO2. (b) and (d) show the neat and consistent maps of FMCW-LIO.

I. Introduction

LASER Light Detection And Ranging (LiDAR) sensors are indispensable and play a crucial role in the advancements of robotics and automation. In recent years, LiDAR sensors have undergone dramatic innovations with more advanced features. Among them, a remarkable milestone is the technology of Frequency Modulated Continuous Wave (FMCW) LiDAR that can capture both the range and radial Doppler velocity of points based on the modulation and demodulation of laser waves in frequency domain [1], [2]. Additionally, the FMCW LiDAR shows promise of miniaturization, thus holding significant potential in robotic applications.

The new dimension of Doppler velocity in the FMCW LiDAR implicitly conveys the motion information of the LiDAR, which opens up new pathways for robotic perception and state estimation. Consequently, numerous unexplored topics arise in the integration of conventional methods, previously dominated by geometric feature-based methods, with the intrinsic Doppler velocity. Besides, FMCW LiDARs can output precise range sensing of massive points with significantly high temporal-spatial resolution. On the other hand, FMCW LiDARs possess the inherent capability to measure the Doppler velocity. More importantly, Doppler measurements can provide a direct observation of the LiDAR motion state (LiDAR velocity), which has a correspondence-free nature. The correspondence-free observation is irrelevant to the locally geometric features (e.g., line or plane features) of the surrounding environment. As a result, FMCW LiDARs, representing the next generation of laser sensing technology, hold the potential to address various challenging issues in the conventional LiDAR field, including the degeneracy problem [2].

In this work, harnessing the advantages of FMCW LiDAR and Inertial Measurement Unit (IMU) sensors, we propose a tightly-coupled FMCW LiDAR-inertial odometry (LIO) and mapping framework, with Doppler and geometric observations. Specifically, the contributions can be summarized as:

1) A robust Doppler-aided LIO, FMCW-LIO, is proposed for FMCW LiDAR systems. To our knowledge, FMCW-LIO is the first real-time and complete framework for FMCW LiDAR-inertial state estimation and mapping.
2) To precisely utilize Doppler velocities in an accumulated LiDAR scan, a novel motion compensation method for Doppler measurements is designed by fusing with IMU.
3) A Doppler-aided observation, integrated in the on-manifold Iterated Error-State Extended Kalman Filter (IESEKF), is applied for state estimation. An efficient dynamic point removal method is also exploited to obtain consistent geometric observations and static maps.

Manuscript received 16 January 2024; accepted 26 April 2024. Date of publication 3 May 2024; date of current version 9 May 2024. This letter was recommended for publication by Associate Editor L. Heng and Editor S. Behnke upon evaluation of the reviewers' comments. This work was supported by the Science and Technology Development Fund of Macau SAR under Grant AGJ-2021-0046, Grant A2-2022-0081, Grant AFJ-2022-0123, and Grant AFJ-2023-0067. *(Corresponding author: Hui Kong.)*

The authors are with the State Key Laboratory of Internet of Things for Smart City (SKL-IOTSC), Faculty of Science and Technology, University of Macau, Taipa 999078, China (e-mail: zhao.mingle@connect.um.edu.mo; kahowang0228@foxmail.com; ga0.tianxiao@connect.um.edu.mo; czxu@um.edu.mo; huikong@um.edu.mo).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2024.3396636, provided by the authors.

Digital Object Identifier 10.1109/LRA.2024.3396636

4) Comprehensive experiments on multiple mobile platforms equipped with an FMCW LiDAR are conducted to validate and demonstrate the effectiveness, accuracy, and robustness of the proposed framework. The video is available at https://youtu.be/2yuZYw91AP8.

II. Related Work

In this section, we review the most related works focusing on LiDAR-inertial odometry, FMCW LiDAR-based algorithms and datasets, and recent automotive radar-inertial odometry.

A. LiDAR-Inertial Odometry

To handle dynamic motions of robots, the tightly-coupled LiDAR-inertial fusion has attracted significant research attention. LIO-SAM [3] employs a factor-graph framework, and it can also fuse Global Navigation Satellite System (GNSS) and loop closure factors. LIO-mapping [4] uses the IMU pre-integration in a constrained refinement on estimation and mapping. LINS [5] introduces a framework based on the IESEKF to recursively update the state, but it heavily relies on the ground assumption. In [6], authors apply the IESEKF and a novel Kalman gain formulation for efficient computing. FAST-LIO2 [7] further enhances the performance by using a direct scan-to-map observation model and the incremental k-dimensional tree (*ikd-Tree*) for the online map. Faster-LIO [8] uses an incremental voxel map (iVox) to achieve higher search speed. Point-LIO [9] uses a point-wise framework and a novel IMU modeling method that allows a high-frequency odometry output. A hierarchical geometrical observer and a higher-order motion correction are applied in [10] which can efficiently improve the estimation accuracy and map quality.

B. FMCW LiDAR-Based Algorithms and Datasets

Recently, DICP [2] introduces a variant of conventional geometric Iterative Closest Point (ICP) methods by including a Doppler objective function in the registration problem, revealing significant improvements in accuracy even in feature-denied scenes. Guo et al. [11] propose a velocity estimation and clustering framework with Doppler velocities in the CARLA simulation [12]. Gu et al. [13] present a moving-object tracking method using an FMCW LiDAR. In [14] and [15], authors present a continuous-time LiDAR odometry (LO) and a velocity estimator supporting Doppler measurements. In [16], authors use FMCW LiDARs to simultaneously estimate the kinematic state and the shape of moving objects. More recently, Jung et al. [17] release a heterogeneous LiDAR dataset (HeLiPR) for place recognition tasks, including an FMCW LiDAR. However, HeLiPR is not appropriate for online state estimation because the velocity of points is the post-processed absolute velocity using external sensors, instead of the raw Doppler velocity measurement.

C. Automotive Radar-Inertial Odometry

3D or 4D imaging radars can measure the position and Doppler velocity of points similar to FMCW LiDARs, but the point density is far less than that of FMCW LiDARs. In [18] [19], authors propose EKF-based radar-inertial odometry (RIO) systems, with ego-velocity estimates. In [20], authors use the radar intensity in the RIO. Ng et al. [21] propose a radar-inertial odometry which uses a continuous-time framework. Lately, 4D iRIOM [22] shows the impressive localization and mapping accuracy of a RIO, integrated with the scan-to-submap matching and loop closure modules.

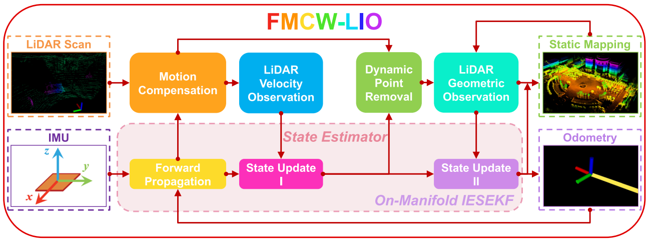

Fig. 2. System overview of FMCW-LIO.

III. System Overview

The system overview is illustrated in Fig. 2. FMCW-LIO takes 6-axis IMU signals as system inputs. LiDAR scans from an FMCW LiDAR are fed into the motion compensation to correct the distortion of position and Doppler measurements. The undistorted scans are first used in the LiDAR velocity observation model. Adopting the updated velocity, potential dynamic points in the scan can be removed, according to raw Doppler measurements. Then, the remaining static points are sent to the LiDAR geometric observation, resulting in more accurate and consistent associations for the second state update and static mapping. Overall, the Doppler-aided FMCW-LIO can achieve accurate and robust state estimation and output a consistent, neat, and static map.

For notations, the inertial frame is set as the world frame w and the body frame b is the IMU frame. ${}^{w}(\cdot)$, ${}^{b}(\cdot)$, and ${}^{l}(\cdot)$ respectively represent a 3D vector projected in the world, body, and LiDAR frame. The gravity ${}^{w}\mathbf{g}$ is constant in the world frame under the steady and flat earth assumption. A 3D vector $\mathbf{p}$ from point A to B projected in the frame C is ${}^{C}\mathbf{p}_{AB}$. Similarly, a 3D linear velocity of the frame B with respect to the frame A expressed in the frame C is ${}^{C}\mathbf{v}_{AB}$. We use $\mathbf{R} \in SO(3)$ to represent rotations and a rotation matrix rotating a vector in the frame B to A is $\mathbf{R}_{AB}$. $\mathbf{T}_{AB} \in SE(3)$ physically transforms a vector in the frame B to the frame A, considering the lever arm. $\mathbf{0}$ is the 3×1 zero vector, $\mathbf{0}_{n \times m}$ is the $n \times m$ zero matrix, $\mathbf{I}$ is the 3×3 identity matrix, and $\mathbf{I}_n$ is the $n \times n$ identity matrix. $\widetilde{(\cdot)}$, $\widehat{(\cdot)}$, and $\overline{(\cdot)}$ are respectively the measurement, the propagated state, and the updated state.

IV. Methodology

A. System State Description

The state $\mathbf{x} \in \mathcal{M}$ evolves on the 24-dimensional manifold $\mathcal{M}$, including the rotation $\mathbf{R}_{wb}$ and position ${}^{w}\mathbf{p}_{wb}$ of body frame with respect to the world frame (the first body frame), velocity ${}^{w}\mathbf{v}_{wb}$, gyroscope and accelerometer biases ${}^{b}\mathbf{b}_{g}$ and ${}^{b}\mathbf{b}_{a}$, and LiDAR-IMU extrinsic parameters $\mathbf{R}_{bl}$ and ${}^{b}\mathbf{p}_{bl}$:

$$\mathcal{M} \triangleq SO(3) \times \mathbb{R}^{15} \times SO(3) \times \mathbb{R}^{3}; \quad \dim(\mathcal{M}) = 24$$

$$\mathbf{x} \triangleq \left[\mathbf{R}_{wb}^{\top} \quad {}^{w}\mathbf{p}_{wb}^{\top} \quad {}^{w}\mathbf{v}_{wb}^{\top} \quad {}^{b}\mathbf{b}_{g}^{\top} \quad {}^{b}\mathbf{b}_{a}^{\top} \quad {}^{w}\mathbf{g}^{\top} \quad \mathbf{R}_{bl}^{\top} \quad {}^{b}\mathbf{p}_{bl}^{\top} \right]^{\top} \tag{1}$$

To parameterize the error state on the tangent space of the locally homeomorphic manifold, two operators (“boxplus”: $\boxplus$

and "boxminus": $\boxminus$) are utilized [7], [23]. Hence the error-state $\delta\mathbf{x} \triangleq \mathbf{x} \boxminus \mathbf{x}_{esti} \in \mathbb{R}^{24}$ between the true state $\mathbf{x}$ and the nominal estimate $\mathbf{x}_{esti}$ ($\mathbf{x}_{esti} = \widehat{\mathbf{x}}$ or $\mathbf{x}_{esti} = \overline{\mathbf{x}}$) is:

$$\delta\mathbf{x} = \begin{bmatrix} \delta\boldsymbol{\theta}_{wb}^\top & {}^w\delta\mathbf{p}_{wb}^\top & {}^w\delta\mathbf{v}_{wb}^\top & {}^b\delta\mathbf{b}_g^\top & {}^b\delta\mathbf{b}_a^\top & {}^w\delta\mathbf{g}^\top & \delta\boldsymbol{\theta}_{bl}^\top & {}^b\delta\mathbf{p}_{bl}^\top \end{bmatrix}^\top$$

B. State Propagation

1) System Input: The system input is the IMU measurement (angular velocity ${}^b\widetilde{\boldsymbol{\omega}}$ and specific force ${}^b\widetilde{\mathbf{a}}$). The input is denoted as $\mathbf{u}$, interrupted by zero-mean Gaussian noises $\mathbf{n}_g$ and $\mathbf{n}_a$. $\mathbf{n}_{bg}$ and $\mathbf{n}_{ba}$ are Gaussian noises of IMU biases, modeled as random walks. The input $\mathbf{u}$ and the noise $\mathbf{w}$ are:

$$\mathbf{u} \triangleq \begin{bmatrix} {}^b\widetilde{\boldsymbol{\omega}}^\top & {}^b\widetilde{\mathbf{a}}^\top \end{bmatrix}^\top \in \mathbb{R}^6, \ \mathbf{w} \triangleq \begin{bmatrix} \mathbf{n}_g^\top & \mathbf{n}_a^\top & \mathbf{n}_{bg}^\top & \mathbf{n}_{ba}^\top \end{bmatrix}^\top \in \mathbb{R}^{12}$$

2) Continuous-Time Model: Considering the input signal and the rigid connection between LiDAR-IMU frames, the state dynamics model in continuous time can be derived as:

$$\begin{aligned} &\dot{\mathbf{R}}_{wb} = \mathbf{R}_{wb}\left[{}^b\widetilde{\boldsymbol{\omega}} - {}^b\mathbf{b}_g - \mathbf{n}_g\right]_\times, \ {}^w\dot{\mathbf{p}}_{wb} = {}^w\mathbf{v}_{wb} \\ &{}^w\dot{\mathbf{v}}_{wb} = \mathbf{R}_{wb}\left({}^b\widetilde{\mathbf{a}} - {}^b\mathbf{b}_a - \mathbf{n}_a\right) + {}^w\mathbf{g}, {}^b\dot{\mathbf{b}}_g = \mathbf{n}_{bg}, {}^b\dot{\mathbf{b}}_a = \mathbf{n}_{ba} \\ &{}^w\dot{\mathbf{g}} = \mathbf{0}, \quad \dot{\mathbf{R}}_{bl} = \mathbf{0}_{3\times3}, \quad {}^b\dot{\mathbf{p}}_{bl} = \mathbf{0} \end{aligned} \tag{2}$$

where $[\mathbf{t}]_\times$ is the skew-symmetric matrix of a 3D vector $\mathbf{t}$.

3) Discrete-Time Model: Using discrete IMU measurements, we can propagate the nominal state from (2) when every IMU measurement (denoting i the index) arrives [7]:

$$\mathbf{x}_{i+1} = \mathbf{x}_i \boxplus \left(\mathbf{f}\left(\mathbf{x}_i, \mathbf{u}_i, \mathbf{w}_i\right)\Delta t\right) \tag{3}$$

where Δt is the sampling period between two adjacent IMU measurements. $\mathbf{f}$ is the derivative of the discrete dynamics model [7]. The estimated nominal state $\widehat{\mathbf{x}}$ is propagated using the above function where the noise is set to zero:

$$\widehat{\mathbf{x}}_{i+1} = \widehat{\mathbf{x}}_i \boxplus \left(\mathbf{f}\left(\widehat{\mathbf{x}}_i, \mathbf{u}_i, \mathbf{0}_{12\times1}\right)\Delta t\right) \tag{4}$$

Retaining first-order terms from (3) and (4), the error state $\delta\mathbf{x}$ and the covariance $\widehat{\mathbf{P}}$ follow the linear dynamics model:

$$\begin{aligned} \delta\mathbf{x}_{i+1} &= \mathbf{F}_{\delta\mathbf{x}_i}\delta\mathbf{x}_i + \mathbf{F}_{\mathbf{w}_i}\mathbf{w}_i \\ \widehat{\mathbf{P}}_{i+1} &= \mathbf{F}_{\delta\mathbf{x}_i}\widehat{\mathbf{P}}_i\mathbf{F}_{\delta\mathbf{x}_i}^\top + \mathbf{F}_{\mathbf{w}_i}\mathbf{Q}_i\mathbf{F}_{\mathbf{w}_i}^\top \end{aligned} \tag{5}$$

where $\mathbf{F}_{\delta\mathbf{x}_i}$ and $\mathbf{F}_{\mathbf{w}_i}$ are respectively the transition matrix and noise Jacobian matrix linearized at $\widehat{\mathbf{x}}_i$ [6]. $\mathbf{Q}_i$ is the covariance of the noise $\mathbf{w}_i$, which can be obtained from the offline IMU calibration [24]. The initial state $\widehat{\mathbf{x}}_0$ and covariance $\widehat{\mathbf{P}}_0$ of the current propagation are the updated state $\overline{\mathbf{x}}_{k-1}$ and covariance $\overline{\mathbf{P}}_{k-1}$ at the last scan-end time t_{k-1}.

C. Motion Compensation and State Update

1) Motion Compensation for Doppler Measurements: A generic LiDAR usually returns accumulated scans at a fixed frequency (e.g., 10 Hz), instead of returning each individual LiDAR point separately. The system is propagated from the last update at the last scan-end time t_{k-1} until the current scan arrives at time t_k. However, although a LiDAR scan arrives at a certain time, each LiDAR point in this scan is sampled at its individual discrete time. Hence, for a moving LiDAR, each point is sampled with respect to the LiDAR position at its sampling time, instead of the LiDAR position at the scan-end time. To exactly use LiDAR points in state update, we need to correct point positions in a scan by motion compensation.

For a LiDAR point p_j sampled at t_j between two adjacent IMU sampling moments t_{i-1} and t_i, we can perform backward propagation [6] with IMU to correct its position measurements, obtaining the corrected position ${}^{l_k}\widetilde{\mathbf{p}}_{l_k p_j}$ of p_j with respect to the LiDAR position at the scan-end time t_k. We denote the raw position measurement of p_j as ${}^{l_j}\widetilde{\mathbf{p}}_{l_j p_j}$, which is with respect to the LiDAR position at its sampling time t_j. $(\check{\cdot})$ represents the backward propagated state at the point sampling time. The motion compensation for the point position measurement is:

$$\begin{aligned} {}^{l_k}\widetilde{\mathbf{p}}_{l_k p_j} = \widehat{\mathbf{R}}_{bl}^\top \Big(\widehat{\mathbf{R}}_{wb_k}^\top \big(\check{\mathbf{R}}_{wb_j} \big(\check{\mathbf{R}}_{bl}\, {}^{l_j}\widetilde{\mathbf{p}}_{l_j p_j} + {}^b\check{\mathbf{p}}_{bl}\big) \\ + {}^w\check{\mathbf{p}}_{wb_j} - {}^w\widehat{\mathbf{p}}_{wb_k}\big) - {}^b\widehat{\mathbf{p}}_{bl}\Big) \end{aligned} \tag{6}$$

where $(\widehat{\mathbf{R}}_{wb_k}, {}^w\widehat{\mathbf{p}}_{wb_k})$ and $(\check{\mathbf{R}}_{wb_j}, {}^w\check{\mathbf{p}}_{wb_j})$ are respectively system poses at the scan-end time t_k and the sampling time t_j of point p_j. This step can achieve the motion compensation for positions except for Doppler measurements. Next, we design an accurate motion compensation method for Doppler measurements, enabling the state update at t_k.

Under the implicit assumption in the above position compensation that LiDAR points are static in the world frame (i.e., ${}^w\mathbf{v}_{wp_j} \equiv \mathbf{0}$), we can derive a motion compensation method for Doppler measurements. For a static point p_j, its Doppler velocity $\widetilde{v}_j^{d_j}$ is measured with respect to the LiDAR position and LiDAR velocity at its sampling time t_j [2]:

$$\widetilde{v}_j^{d_j} = -\frac{{}^{l_j}\widetilde{\mathbf{p}}_{l_j p_j}^\top}{\left\|{}^{l_j}\widetilde{\mathbf{p}}_{l_j p_j}\right\|}\, {}^{l_j}\check{\mathbf{v}}_{wl_j} \tag{7}$$

where ${}^{l_j}\check{\mathbf{v}}_{wl_j}$ is the LiDAR velocity with respect to the world frame and projected in the LiDAR frame at t_j. For p_j, its Doppler velocity $\widetilde{v}_j^{d_k}$, with respect to the LiDAR position and LiDAR velocity at the scan-end time t_k, is unequal to the raw measurement $\widetilde{v}_j^{d_j}$ at t_j. Similar to (7), $\widetilde{v}_j^{d_k}$ can be derived as:

$$\widetilde{v}_j^{d_k} = -\frac{{}^{l_k}\widetilde{\mathbf{p}}_{l_k p_j}^\top}{\left\|{}^{l_k}\widetilde{\mathbf{p}}_{l_k p_j}\right\|}\, {}^{l_k}\widehat{\mathbf{v}}_{wl_k} \tag{8}$$

where ${}^{l_k}\widehat{\mathbf{v}}_{wl_k}$ is the LiDAR velocity with respect to the world frame and projected in the LiDAR frame at t_k. To compensate the Doppler measurement from $\widetilde{v}_j^{d_j}$ to $\widetilde{v}_j^{d_k}$, we can use the forward and backward propagated states, LiDAR-IMU extrinsic parameters, and IMU measurements with the transformation between ${}^{l_j}\check{\mathbf{v}}_{wl_j}$ and ${}^{l_k}\widehat{\mathbf{v}}_{wl_k}$. Then substituting (6), (7), and ${}^l\mathbf{v}_{wl} = \mathbf{R}_{bl}^\top(\mathbf{R}_{wb}^\top\, {}^w\mathbf{v}_{wb} + {}^b\boldsymbol{\omega}_{wb} \times {}^b\mathbf{p}_{bl})$ into (8):

$$\begin{aligned} \widetilde{v}_j^{d_k} = \frac{\left\|{}^{l_j}\widetilde{\mathbf{p}}_{l_j p_j}\right\|}{\left\|{}^{l_k}\widetilde{\mathbf{p}}_{l_k p_j}\right\|}\widetilde{v}_j^{d_j} - \frac{1}{\left\|{}^{l_k}\widetilde{\mathbf{p}}_{l_k p_j}\right\|}\Big\{\Big[{}^w\check{\mathbf{p}}_{l_j p_j}^\top\left({}^w\widehat{\mathbf{v}}_{wl_k} - {}^w\check{\mathbf{v}}_{wl_j}\right) \\ + \left({}^w\check{\mathbf{p}}_{wl_j} - {}^w\widehat{\mathbf{p}}_{wl_k}\right)^\top {}^w\widehat{\mathbf{v}}_{wl_k}\Big]\Big\} \end{aligned} \tag{9}$$

where

$$\begin{aligned} &{}^w\check{\mathbf{p}}_{l_j p_j} = \check{\mathbf{R}}_{wb_j}\check{\mathbf{R}}_{bl}\, {}^{l_j}\widetilde{\mathbf{p}}_{l_j p_j} \\ &{}^w\check{\mathbf{p}}_{wl_j} = {}^w\check{\mathbf{p}}_{wb_j} + \check{\mathbf{R}}_{wb_j}\, {}^b\check{\mathbf{p}}_{bl}, \quad {}^w\widehat{\mathbf{p}}_{wl_k} = {}^w\widehat{\mathbf{p}}_{wb_k} + \widehat{\mathbf{R}}_{wb_k}\, {}^b\widehat{\mathbf{p}}_{bl} \end{aligned}$$

$$\begin{aligned} {}^{w}\check{\mathbf{v}}_{wl_j} &= {}^{w}\check{\mathbf{v}}_{wb_j} + \check{\mathbf{R}}_{wb_j}\left({}^{b_j}\check{\boldsymbol{\omega}}_{wb_j} \times {}^{b}\check{\mathbf{p}}_{bl}\right) \\ {}^{w}\widehat{\mathbf{v}}_{wl_k} &= {}^{w}\widehat{\mathbf{v}}_{wb_k} + \widehat{\mathbf{R}}_{wb_k}\left({}^{b_k}\widehat{\boldsymbol{\omega}}_{wb_k} \times {}^{b}\widehat{\mathbf{p}}_{bl}\right) \end{aligned} \tag{10}$$

In (10), $(\widehat{\mathbf{R}}_{wb_k}, {}^{w}\widehat{\mathbf{p}}_{wb_k}, {}^{w}\widehat{\mathbf{v}}_{wb_k})$ are forward propagated states at t_k, $(\check{\mathbf{R}}_{wb_j}, {}^{w}\check{\mathbf{p}}_{wb_j}, {}^{w}\check{\mathbf{v}}_{wb_j})$ are backward propagated states at t_j. ${}^{b_j}\check{\boldsymbol{\omega}}_{wb_j}$ is the unbiased gyroscope measurement (i.e., ${}^{b_j}\check{\boldsymbol{\omega}}_{wb_j} = {}^{b_j}\widetilde{\boldsymbol{\omega}}_j - {}^{b_j}\check{\mathbf{b}}_g$) at t_j from the IMU measurement interpolation, and ${}^{b_k}\widehat{\boldsymbol{\omega}}_{wb_k}$ is the unbiased gyroscope measurement at t_k. Notably, beyond motion compensation, (9) essentially reveals the interrelation between the Doppler velocities of a static point w.r.t. different LiDAR frames. Finally, we can align all points in a scan to the scan-end time t_k under a unified temporal-spatial frame, by the motion compensation for both position (6) and Doppler (9) measurements.

2) LiDAR Velocity Observation: In a motion-compensated scan at t_k, each point can provide the radial information of the LiDAR velocity in the LiDAR frame. As in (8), the Doppler velocity of a static point can be predicted using the point direction and the LiDAR velocity. The normalized direction vector ${}^{l_k}\widetilde{\mathbf{d}}_{l_kp_j}$ of point p_j after motion compensation can be computed through: ${}^{l_k}\widetilde{\mathbf{d}}_{l_kp_j} = \frac{{}^{l_k}\widetilde{\mathbf{p}}_{l_kp_j}}{\|{}^{l_k}\widetilde{\mathbf{p}}_{l_kp_j}\|}$. For n points, a matrix formulation can be yielded by stacking all points:

$$\widetilde{\mathbf{D}}_k {}^{l_k}\mathbf{v}_{wl_k} = \widetilde{\mathbf{V}}_k \tag{11}$$

where $\widetilde{\mathbf{D}}_k = [\ldots \quad -{}^{l_k}\widetilde{\mathbf{d}}_{l_kp_j} \quad \ldots]^{\top}$ is a $n \times 3$ matrix and $\widetilde{\mathbf{V}}_k = [\ldots \quad \widetilde{v}_j^{d_k} \quad \ldots]^{\top}$ is a $n \times 1$ vector. Accordingly, we can estimate a LiDAR velocity measurement at the scan-end time t_k by solving a least squares problem [18]:

$${}^{l_k}\widetilde{\mathbf{v}}_{wl_k} = \underset{{}^{l_k}\mathbf{v}_{wl_k}}{\arg\min}\left\|\widetilde{\mathbf{D}}_k {}^{l_k}\mathbf{v}_{wl_k} - \widetilde{\mathbf{V}}_k\right\|^2 = \left(\widetilde{\mathbf{D}}_k^{\top}\widetilde{\mathbf{D}}_k\right)^{-1}\widetilde{\mathbf{D}}_k^{\top}\widetilde{\mathbf{V}}_k$$

Meanwhile, the covariance of the estimated LiDAR velocity, $\mathbf{R}_{\mathbf{v}}$, can be computed from the estimation residual [18], serving as the observation covariance in the state update.

For an FMCW LiDAR scan containing tens of thousands of points, directly solving the above problem is computationally burdensome and error-prone, due to potential dynamic points. Thus, a 3-point Random Sample and Consensus (RANSAC) method [18] is applied to efficiently obtain a robust solution. For example, setting the outlier probability to 30% and the desired success probability to 99%, the iteration number is 11, which leads to a high efficiency. Besides, to prevent the solution from falling into the null-space of (11) (i.e., $\widetilde{\mathbf{D}}_k {}^{l_k}\mathbf{v}_{wl_k} = \mathbf{0}_{n\times 1}$) and improve the ratio of valid samples, the sampling probability is weighted by the absolute cosine of the angle $\phi_{\widehat{\mathbf{v}}_l\widetilde{\mathbf{d}}_j}$ between the point direction and the forward propagated LiDAR velocity in the LiDAR frame:

$$\left\|\cos\left(\phi_{\widehat{\mathbf{v}}_l\widetilde{\mathbf{d}}_j}\right)\right\| = \frac{\left\|\langle {}^{l_k}\widetilde{\mathbf{d}}_{l_kp_j}, {}^{l_k}\widehat{\mathbf{v}}_{wl_k}\rangle\right\|}{\left\|{}^{l_k}\widetilde{\mathbf{d}}_{l_kp_j}\right\|\left\|{}^{l_k}\widehat{\mathbf{v}}_{wl_k}\right\|} \tag{12}$$

where ${}^{l_k}\widehat{\mathbf{v}}_{wl_k} = (\widehat{\mathbf{R}}_{wb_k}\widehat{\mathbf{R}}_{bl})^{\top} {}^{w}\widehat{\mathbf{v}}_{wl_k}$ is the forward propagated LiDAR velocity in the LiDAR frame at the scan-end time t_k, and ${}^{w}\widehat{\mathbf{v}}_{wl_k}$ can be computed from (10).

The above estimated measurement of LiDAR velocity provides a correspondence-free observation for the system state. The LiDAR velocity observation model $\mathbf{h}_{\mathbf{v}}(\mathbf{x}, {}^{l}\mathbf{n}_{\mathbf{v}})$ is:

$$\begin{aligned} {}^{l}\widetilde{\mathbf{v}}_{wl} &= \mathbf{h}_{\mathbf{v}}(\mathbf{x}, {}^{l}\mathbf{n}_{\mathbf{v}}) \\ &\triangleq \mathbf{R}_{bl}^{\top}\left(\mathbf{R}_{wb}^{\top} {}^{w}\mathbf{v}_{wb} + \left({}^{b}\widetilde{\boldsymbol{\omega}} - {}^{b}\mathbf{b}_g - \mathbf{n}_g\right) \times {}^{b}\mathbf{p}_{bl}\right) + {}^{l}\mathbf{n}_{\mathbf{v}} \end{aligned} \tag{13}$$

where the covariance of ${}^{l}\mathbf{n}_{\mathbf{v}}$ is the aforementioned $\mathbf{R}_{\mathbf{v}}$. To reach the optimal estimate at t_k, the LiDAR velocity residual $\mathbf{r}_{\mathbf{v}_k} = {}^{l_k}\widetilde{\mathbf{v}}_{wl_k} - {}^{l_k}\widehat{\mathbf{v}}_{wl_k}$ can be linearized as:

$$\mathbf{r}_{\mathbf{v}_k} = \mathbf{h}_{\mathbf{v}}(\mathbf{x}_k, \mathbf{n}_{\mathbf{v}}) - \mathbf{h}_{\mathbf{v}}(\widehat{\mathbf{x}}_k, \mathbf{0}) \approx \mathbf{H}_{\mathbf{v}_k}\delta\mathbf{x}_k + {}^{l}\mathbf{n}_{\mathbf{v}} \tag{14}$$

The observation Jacobian matrix $\mathbf{H}_{\mathbf{v}_k} \in \mathbb{R}^{3\times 24}$ is:

$$\mathbf{H}_{\mathbf{v}_k} = \left[\mathbf{H}_{\delta\boldsymbol{\theta}_k} \quad \mathbf{0}_{3\times 3} \quad \mathbf{H}_{\delta\mathbf{v}_k} \quad \mathbf{H}_{\delta\mathbf{b}_{gk}} \quad \mathbf{0}_{3\times 6} \quad \mathbf{H}_{\delta\boldsymbol{\theta}_{bl}} \quad \mathbf{H}_{\delta\mathbf{p}_{bl}}\right] \tag{15}$$

The different block components of $\mathbf{H}_{\mathbf{v}_k}$ are:

$$\begin{aligned} &\mathbf{H}_{\delta\boldsymbol{\theta}_k} = \widehat{\mathbf{R}}_{bl}^{\top}[\widehat{\mathbf{R}}_{wb_k}^{\top} {}^{w}\widehat{\mathbf{v}}_{wb_k}]_{\times},\ \mathbf{H}_{\delta\mathbf{v}_k} = (\widehat{\mathbf{R}}_{wb_k}\widehat{\mathbf{R}}_{bl})^{\top} \\ &\mathbf{H}_{\delta\mathbf{b}_{gk}} = \widehat{\mathbf{R}}_{bl}^{\top}[{}^{b}\widehat{\mathbf{p}}_{bl}]_{\times},\ \mathbf{H}_{\delta\mathbf{p}_{bl}} = \widehat{\mathbf{R}}_{bl}^{\top}[{}^{b_k}\widetilde{\boldsymbol{\omega}}_k - {}^{b_k}\widehat{\mathbf{b}}_g]_{\times} \\ &\mathbf{H}_{\delta\boldsymbol{\theta}_{bl}} = [\widehat{\mathbf{R}}_{bl}^{\top}(\widehat{\mathbf{R}}_{wb_k}^{\top} {}^{w}\widehat{\mathbf{v}}_{wb_k} + ({}^{b_k}\widetilde{\boldsymbol{\omega}}_k - {}^{b_k}\widehat{\mathbf{b}}_g) \times {}^{b}\widehat{\mathbf{p}}_{bl})]_{\times} \end{aligned} \tag{16}$$

On account of the low dimension of 3D LiDAR velocity observation ($\dim(\mathbf{r}_{\mathbf{v}_k}) \ll \dim(\mathcal{M})$), the conventional formulation of Kalman gain instead of (23) is applied:

$$\mathbf{K}_{\mathbf{v}_k} = \widehat{\mathbf{P}}_k\mathbf{H}_{\mathbf{v}_k}^{\top}\left(\mathbf{H}_{\mathbf{v}_k}\widehat{\mathbf{P}}_k\mathbf{H}_{\mathbf{v}_k}^{\top} + \mathbf{R}_{\mathbf{v}}\right)^{-1} \tag{17}$$

The updated state $\overline{\mathbf{x}}_{\mathbf{v}_k}$ and covariance $\overline{\mathbf{P}}_{\mathbf{v}_k}$ using the LiDAR velocity observation are respectively:

$$\begin{aligned} \overline{\mathbf{x}}_{\mathbf{v}_k} &= \widehat{\mathbf{x}}_k \boxplus \overline{\delta\mathbf{x}}_{\mathbf{v}_k} = \widehat{\mathbf{x}}_k \boxplus (\mathbf{K}_{\mathbf{v}_k}\mathbf{r}_{\mathbf{v}_k}) \\ \overline{\mathbf{P}}_{\mathbf{v}_k} &= \mathbf{L}_{\mathbf{v}_k}(\mathbf{I}_{24\times 24} - \mathbf{K}_{\mathbf{v}_k}\mathbf{H}_{\mathbf{v}_k})\widehat{\mathbf{P}}_k\mathbf{L}_{\mathbf{v}_k}^{\top} \end{aligned} \tag{18}$$

where $\mathbf{L}_{\mathbf{v}_k}$ is the projection matrix between tangent spaces arising from the on-manifold increment $\overline{\delta\mathbf{x}}_{\mathbf{v}_k}$ [23].

3) LiDAR Geometric Observation: Apart from the instant correspondence-free Doppler-aided observation, the geometric association from LiDAR points can also provide observations. Here we use the point-plane distance model [7], [25]:

$$\begin{aligned} 0 &= h_{g_j}(\mathbf{x}, {}^{l}\mathbf{n}_{p_j}) \\ &\triangleq {}^{w}\mathbf{n}_{plane_j}^{\top}\left(\mathbf{T}_{wb}\mathbf{T}_{bl}\left({}^{l}\widetilde{\mathbf{p}}_{lp_j} - {}^{l}\mathbf{n}_{p_j}\right) - {}^{w}\widetilde{\mathbf{q}}_{wq_j}\right) \end{aligned} \tag{19}$$

where $\mathbf{T}_{wb}, \mathbf{T}_{bl} \in SE(3)$ are respectively the system pose and extrinsic parameters, ${}^{l}\mathbf{n}_{p_j}$ is the point position noise, ${}^{w}\mathbf{n}_{plane_j}$ is the unit normal vector of the nearest neighbor plane of point p_j in the world frame, and ${}^{w}\widetilde{\mathbf{q}}_{wq_j}$ is a point in that plane projected in the world frame.

In IESEKF, the state updated by the LiDAR velocity observation give a prior Gaussian distribution in the tangent space at $\overline{\mathbf{x}}_{\mathbf{v}_k}$, but the optimal estimate is derived in the tangent space at $\widehat{\mathbf{x}}_k^{\kappa}$, i.e., the state of κ-th update iteration (equals the updated state after $(\kappa - 1)$-th update iteration):

$$\widehat{\mathbf{x}}_k^{\kappa} = \overline{\mathbf{x}}_k^{\kappa-1} = \widehat{\mathbf{x}}_k^{\kappa-1} \boxplus \overline{\delta\mathbf{x}}_k^{\kappa-1},\ \widehat{\mathbf{x}}_k^{0} = \overline{\mathbf{x}}_{\mathbf{v}_k} \tag{20}$$

With $\delta\mathbf{x}_k = \mathbf{x}_k \boxminus \overline{\mathbf{x}}_{\mathbf{v}_k} \sim \mathcal{N}(\mathbf{0}_{24\times 1}, \overline{\mathbf{P}}_{\mathbf{v}_k})$, the prior distribution projected in the tangent space at $\widehat{\mathbf{x}}_k^{\kappa}$ imposing a prior error-state distribution in the κ-th update iteration:

$$\delta\mathbf{x}_k^{\kappa} \sim \mathcal{N}\left(-\mathbf{J}_k^{\kappa}(\widehat{\mathbf{x}}_k^{\kappa} \boxminus \overline{\mathbf{x}}_{\mathbf{v}_k}), \mathbf{J}_k^{\kappa}\overline{\mathbf{P}}_{\mathbf{v}_k}\mathbf{J}_k^{\kappa\top}\right) \tag{21}$$

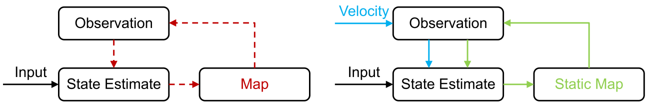

Fig. 3. Illustration of the feedback loop between the state estimate and the online map.

where $\mathbf{J}_k^\kappa$ is the tangent projection matrix because of the on-manifold update step $(\widehat{\mathbf{x}}_k^\kappa \boxminus \overline{\mathbf{x}}_{\mathbf{v}_k})$, similar to (18). The LiDAR geometric observation residual of point p_j at t_k is:

$$\mathbf{r}_{g_{kj}}^\kappa = 0 - h_{g_j}(\widehat{\mathbf{x}}_k^\kappa, \mathbf{0}) = -h_{g_j}(\widehat{\mathbf{x}}_k^\kappa, \mathbf{0}) \tag{22}$$

Then, fusing the prior distribution and the likelihood observation distribution from all observations, an equivalent maximum a-posteriori estimate (MAP) can be obtained [23].

Therefore, the iterated state update follows:

$$\begin{aligned}\mathbf{K}_{g_k}^\kappa &= \left(\mathbf{H}_{g_k}^{\kappa\top}\mathbf{R}_{g_k}^{-1}\mathbf{H}_{g_k}^\kappa + \left(\mathbf{J}_k^\kappa\overline{\mathbf{P}}_{\mathbf{v}_k}\mathbf{J}_k^{\kappa\top}\right)^{-1}\right)^{-1}\mathbf{H}_{g_k}^{\kappa\top}\mathbf{R}_{g_k}^{-1}\\ \overline{\delta\mathbf{x}}_k^\kappa &= \mathbf{K}_{g_k}^\kappa\mathbf{r}_{g_k}^\kappa + \left(\mathbf{K}_{g_k}^\kappa\mathbf{H}_{g_k}^\kappa - \mathbf{I}_{24\times24}\right)\mathbf{J}_k^\kappa\left(\widehat{\mathbf{x}}_k^\kappa \boxminus \overline{\mathbf{x}}_{\mathbf{v}_k}\right)\\ \widehat{\mathbf{x}}_k^{\kappa+1} &= \widehat{\mathbf{x}}_k^\kappa \boxplus \overline{\delta\mathbf{x}}_k^\kappa\end{aligned} \tag{23}$$

where $\mathbf{r}_{g_k}^\kappa$, $\mathbf{H}_{g_k}^\kappa$, and $\mathbf{R}_{g_k}$ are formulated by stacking matrices of all valid points. Finally, after the convergence of IESEKF, the optimally updated state and covariance are:

$$\begin{aligned}\overline{\mathbf{x}}_k &= \overline{\mathbf{x}}_{g_k} = \widehat{\mathbf{x}}_k^{\kappa+1}\\ \overline{\mathbf{P}}_k &= \overline{\mathbf{P}}_{g_k} = \mathbf{L}_{g_k}\left(\mathbf{I}_{24\times24} - \mathbf{K}_{g_k}^\kappa\mathbf{H}_{g_k}^\kappa\right)\left(\mathbf{J}_k^\kappa\overline{\mathbf{P}}_{\mathbf{v}_k}\mathbf{J}_k^{\kappa\top}\right)\mathbf{L}_{g_k}^\top\end{aligned} \tag{24}$$

In (24), $\overline{\mathbf{x}}_{g_k}$ and $\overline{\mathbf{P}}_{g_k}$ are the updated state and covariance using the LiDAR geometric observation, $\overline{\mathbf{x}}_k$ and $\overline{\mathbf{P}}_k$ are the optimal state estimate and covariance after both LiDAR velocity and geometric update at the scan-end time t_k.

D. Dynamic Point Removal and Static Mapping

The LiDAR geometric observation in (19) depends on the online established map. Dynamic points in the LiDAR scan can break the static world assumption and introduce significant errors. Importantly, dynamic points can contaminate the online map, which is prone to provide erroneous geometric correspondences. This dependence creates a loop between the state estimate and the map that can destabilize the system, as shown in the left of Fig. 3. Specifically, if the state estimate is disturbed by dynamic points, the online map will be inaccurate and chaotic, and the erroneous correspondences from dynamic points will amplify the estimation error. This positive feedback loop of the estimation error may lead to divergence even failure of the state estimator.

Fortunately, the above problem in traditional LiDAR systems can be solved directly and efficiently thanks to the inherent Doppler measurements from FMCW LiDARs. After the state update in (18), we can predict a Doppler velocity $\widetilde{v}_{j,pred}^{d_k}$ for point p_j. If the discrepancy between the prediction and the measurement exceeds a given threshold, the point is removed to prevent it from being added to the LiDAR geometric observation and the online map. The method is similar to the outlier rejection mentioned in [2] but the threshold δv_{thr}^d is adjustable according to the angle $\phi_{\overline{\mathbf{v}}_l\widetilde{\mathbf{d}}_j}$ between the normalized direction of a point and the LiDAR velocity from the updated state in (18). The condition that the point p_j is regarded as a dynamic point is:

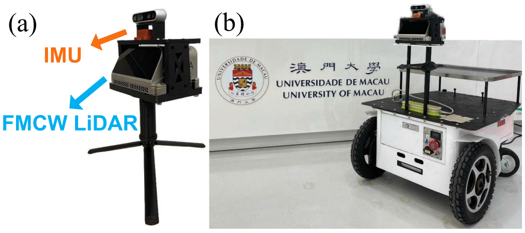

Fig. 4. The sensor suite on the (a) handheld and the (b) wheeled platforms.

$$\left\|\widetilde{v}_{j,pred}^{d_k} - \widetilde{v}_j^{d_k}\right\| > \delta v_{thr}^d\left\|\cos\left(\phi_{\overline{\mathbf{v}}_l\widetilde{\mathbf{d}}_j}\right)\right\| \tag{25}$$

where the cosine is in (12) but uses the updated state ${}^{l_k}\overline{\mathbf{v}}_{wl_k}$.

Significantly, as shown in (12) and (25), this cosine reveals a crucial physical insight: the effectiveness and the signal-to-noise ratio of a LiDAR point in estimating the 3D linear LiDAR velocity depend on the angle (relative direction) between the point direction and the LiDAR velocity projected in the LiDAR frame (i.e., ${}^l\mathbf{v}_{wl}$), rather than directly on the LiDAR velocity projected in the world frame (i.e., ${}^w\mathbf{v}_{wl}$) or the Field-of-View (FoV) of an FMCW LiDAR. Meanwhile, this cosine represents the radial nature of the Doppler effect.

Consequently, only almost static points are fed into the LiDAR geometric observation and the map, which can further improve the accuracy and robustness of the online system. FMCW-LIO can eventually maintain a neat and static map.

Meanwhile, the Doppler-aided correspondence-free observations can partially alleviate the drawback of the aforementioned positive error feedback loop. This observation model reduces the reliance on historical associations from the online map, as shown in the right of Fig. 3.

V. Experimental Results

A. FMCW LiDAR-Inertial Data Collection

It is necessary to collect the FMCW LiDAR-inertial data in the real world to validate the proposed method, due to the absence of public FMCW LiDAR-inertial datasets including raw Doppler measurements, as mentioned in Section II-B. We use an Aeva Aeries II [26] FMCW LiDAR and an Xsens MTi-G-710 IMU to assemble our sensor suite. This sensor suite can be mounted on handheld or wheeled platforms, as in Fig. 4. The LiDAR FoV is 120°×28.8°. The LiDAR returns 10 Hz scans including position and raw Doppler measurements. The frequency of IMU measurements is 200 Hz. The computing device is the Intel NUC with an i7-1165G7 CPU.

The data description is shown in Table I. For the handheld platform, the sensor suite is held by a walking person with persistent shaking and swing. The speed of the handheld platform is slow (about 1.5 m/s). For the wheeled platform, the sensor suite is mounted on a vehicle and the moving speed is relatively fast (up to 7.0 m/s). To comprehensively evaluate different algorithms, diverse conditions (structured vs. structure-degenerated scenes, wide vs. narrow scenes, slow vs. fast speeds) are considered, as shown in Fig. 5. In *campus_01* and *campus_02*, we traverse the campus of the University of Macau and the scene is wide enough to extract and match geometric features for general LIO systems. On the contrary, *street_01* and *street_02* are collected

TABLE I
DESCRIPTION OF THE COLLECTED DATASET

Sequence	Platform	Speed	Scene Type	Length (m)	Duration (s)
campus_01	Handheld	Slow	Wide	425	403
campus_02	Wheeled	Fast	Wide	454	172
street_01	Handheld	Slow	Narrow	182	204
street_02	Handheld	Slow	Narrow	186	223
tunnel_01	Handheld	Slow	Degenerated	1129	740
tunnel_02	Wheeled	Fast	Degenerated	232	90
tunnel_03	Handheld	Slow	Degenerated	283	179
tunnel_04	Wheeled	Fast	Degenerated	1712	368

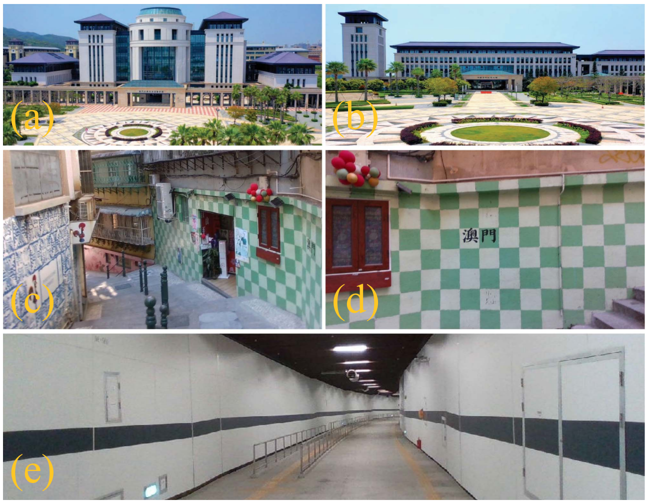

Fig. 5. The scenes of the collected dataset. (a) and (b) are campus scenes. (c) and (d) are narrow street scenes. (e) is the tunnel scene.

in narrow streets in downtown Macao, thus the scene is considerably narrow leading to significant challenges for LIO systems. Furthermore, to validate the Doppler-aided observation, four structure-degenerated sequences are collected in a subaqueous pedestrian tunnel. The tunnel is S-shaped and the one-side length is about 850 m. Since the dataset is challenging and the GNSS accuracy is highly unreliable in these narrow, among-building, or indoor scenes, we choose the trajectory optimized by offline post-processing as the ground-truth reference. We utilize a state-of-the-art loop closure [27] and the pose graph optimization [28] to generate the ground truth. Notably, all post-processed ground-truth trajectories are meticulously examined to ensure alignment with the real world. Besides, we use two metrics, absolute translational error (RMSE) and end-to-end error (E2E) [7], to evaluate algorithms. Therefore, the starting and ending positions of all sequences are confirmed to be exactly coincident to enable the end-to-end evaluation.

For the thorough evaluation and ablation study of different algorithms and modules, we compare FMCW-LIO with conventional mainstream LIO systems under diverse framework designs, i.e., LINS [5], LIO-SAM [3], DLIO [10], and FAST-LIO2 [7]. We also evaluate the recently released Doppler-aided LO, STEAM-DICP [14], which can serve as an ablation study to validate the role of IMU on handheld or small and medium-scale robot platforms with narrow sensing FoV, narrow workspaces, and high-frequency dynamic motions. We tune the parameters of all algorithms to achieve their optimal accuracy and use these tuned parameters in all experiments. In FMCW-LIO, we use *ikd-Tree* [7] as the map structure, and the local map size is set to 1000 m. We also test two variants of FMCW-LIO to perform ablation studies and validate the effectiveness of two modules, including 1) the variant without the Doppler compensation (w/o DC) and 2) the variant without the dynamic point removal (w/o DR).

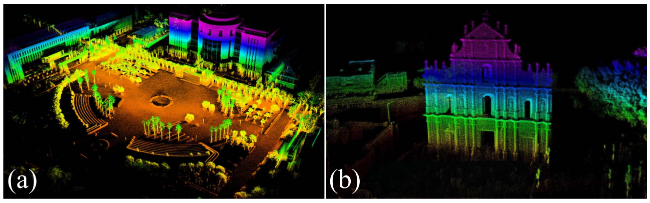

Fig. 6. Mapping in structured scenes. (a) Campus scene. (b) Street scene.

B. Experiments in Structured Environments

The results in four structured scenes are in the left four columns of Table II and the loop closure of LIO-SAM is disabled for fair comparisons. FMCW-LIO achieves accuracy comparable to FAST-LIO2 in *campus* scenes. However, in *street_01*, the narrow scene and intense on-stairs motions (as shown in Fig. 5(c)) result in failures or large drifts of other algorithms, but FMCW-LIO can achieve remarkably low errors. The reason is that, in *street_01*, the sensor suite faces a feature-less wall within an extremely narrow scene which leads to erroneous or invalid geometric observations (as in Fig. 5(d)). Nevertheless, FMCW-LIO can use the Doppler-aided LiDAR velocity observation that is independent of the geometric features of environments. Meanwhile, the LiDAR velocity observation can perform the Zero-Velocity Update (ZUPT) when the platform is steady when facing the wall or offer z-axis velocity when a large motion in the z direction occurs. As a result, the estimated trajectory of FMCW-LIO can exactly return to the starting position.

In addition, due to the presence of numerous moving pedestrians and cars in *street* sequences, the variant (w/o DR) exhibits larger localization errors. Conversely, the complete FMCW-LIO and the variant (w/o DC) achieve accurate estimation performance. In structured sequences, the Doppler compensation effect is not significant due to the slow-moving speeds of both handheld and wheeled platforms operating within the campus or street blocks. Hence, the variant (w/o DC) achieves comparable accuracy with the complete FMCW-LIO. The mapping results of FMCW-LIO in structured scenes are shown in Fig. 6 and the average running time per scan of algorithms is listed in Table III. FMCW-LIO can achieve real-time and efficient performance that is comparable to that of FAST-LIO2. Online demonstrations can be found in the video.

Furthermore, we observe that STEAM-DICP demonstrates estimation accuracy comparable to other LIO methods in *campus_01*. However, in other sequences, it fails when encountering dynamic motions, fast rotations, or narrow scenes due to the lack of keypoints. In *campus_02*, STEAM-DICP can successfully run up to the halfway point, until rapid rotations occur. Thus, we only calculate the RMSE of the segments where it can successfully run without system failures. Therefore, it can be observed that IMU is essential for handheld or robotic platforms that frequently encounter narrow workspaces and highly dynamic motions.

TABLE II
ABSOLUTE TRANSLATIONAL ERRORS (RMSE, METERS) AND END-TO-END ERRORS (E2E, METERS) ON THE COLLECTED DATASET

Method/Sequence	*campus_01*		*campus_02*		*street_01*		*street_02*		*tunnel_01*		*tunnel_02*		*tunnel_03*		*tunnel_04*	
	RMSE	E2E	RMSE	E2E	RMSE	E2E	RMSE	E2E	RMSE	E2E	RMSE	E2E	RMSE	E2E	RMSE	E2E
LINS	3.18	34.36	4.24	38.98	–	–	–	–	–	–	–	–	–	–	–	–
LIO-SAM	1.24	4.29	4.89	3.16	×	×	0.23	0.18	×	×	2.11	0.10	×	×	×	×
DLIO	1.11	1.51	3.44	2.53	×	×	2.00	7.96	×	×	5.61	11.87	×	×	×	×
FAST-LIO2	0.84	1.08	0.54	**0.07**	11.74	4.33	0.19	0.05	×	×	14.99	17.50	46.11	99.83	60.09	23.84
STEAM-DICP	1.23	1.49	1.76	×	×	×	×	×	35.16	×	4.39	×	0.81	×	49.32	×
FMCW-LIO (w/o DC)	0.11	0.14	0.87	0.31	1.14	0.23	0.20	0.08	0.96	0.52	1.94	1.65	0.59	0.32	2.04	0.86
FMCW-LIO (w/o DR)	0.13	0.13	0.16	0.09	2.73	0.72	1.20	0.35	0.48	0.20	0.12	0.06	0.28	**0.05**	**0.11**	0.06
FMCW-LIO	**0.08**	**0.12**	**0.13**	**0.07**	**0.19**	**0.11**	**0.18**	**0.04**	**0.47**	**0.12**	**0.07**	**0.03**	**0.27**	**0.05**	0.13	**0.02**

– As LINS is designed for spinning LiDARs and ground vehicles, its applicability on our LiDAR and platforms is highly challenging.
× The algorithm failed or severely drifted in the corresponding sequence.

TABLE III
AVERAGE RUNNING TIME (MILLISECONDS) PER LIDAR SCAN

	LIO-SAM	DLIO	FAST-LIO2	STEAM-DICP	FMCW-LIO
Structured	40.08	45.93	33.53	178.25	39.71
Degenerated	28.76	36.00	20.54	85.46	21.33

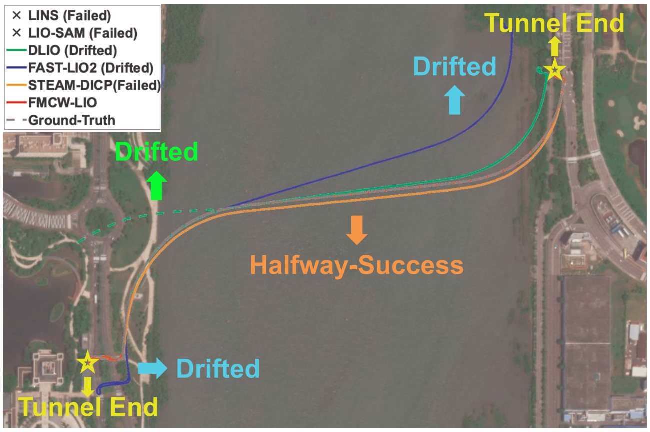

Fig. 7. Estimated trajectories overlaid with the satellite map.

C. Experiments in Structure-Degenerated Environments

To further validate the Doppler-aided observation model, we evaluate algorithms on four *tunnel* sequences targeting geometric structure-degenerated scenes. These sequences have diverse distributions in length and moving speed. The results are shown in the right four columns of Table II. As expected, all conventional LIO systems that depend on geometric observations fail in most *tunnel* sequences. FMCW-LIO, however, accomplishes incredibly low errors, accurately returning to starting points. In the shortest sequence *tunnel_02*, LIO-SAM can achieve accurate localization with a factor graph framework which can maintain relatively long-term constraints in this short length. Although DLIO and FAST-LIO2 do not fail, they fail to achieve accurate localization, generating chaotic, inconsistent and drifted tunnel maps, as in Fig. 1(a)–(c). In the longest sequence *tunnel_04*, FMCW-LIO can perfectly achieve precise localization (2 cm end-to-end error over 1.7 km traversed round-trip distance) through the whole tunnel. The Doppler-aided observation provides a direct observation of the LiDAR velocity state. However, in long tunnel sequences, the factor graph framework in LIO-SAM that only relies on geometric features also proves ineffective in handling incorrect correspondences. Finally, FMCW-LIO can generate a neat, complete, and consistent map of the whole tunnel (as shown in Fig. 1(d)), and the estimated tunnel length is highly close to the real data. Trajectories of algorithms overlaid with a satellite map are shown in Fig. 7. It can be found that FMCW-LIO outputs a precious trajectory accurately aligned with the tunnel ends while other algorithms suffer from severe drifts in such structure-degenerated scenes.

Similarly, STEAM-DICP can only achieve successful running up to the halfway point of each *tunnel* sequence. When the platform executes a U-turn to return, STEAM-DICP fails due to the narrow scene, fast rotations, and a scarcity of keypoints. Despite STEAM-DICP achieving a halfway success in *tunnel* sequences compared to conventional LIO methods, it still exhibits significant drifted errors, especially in the z-axis. We only compute the RMSE of the successful segments for STEAM-DICP. An insight from the results is that while the LiDAR-only continuous-time model of STEAM-DICP can partially cope with dynamic motions, it may encounter challenges or fail to handle the narrow workspaces and intense motion on handheld or robot platforms, without the assistance of exteroception-irrelevant IMU sensors.

In *tunnel_02* and *tunnel_04*, due to high speeds (about 7.0 m/s) of the wheeled platform in the tunnel, the variant (w/o DC) exhibits a significant accuracy decrease which shows the effectiveness of the Doppler compensation in a low-frequency scan pattern. Additionally, in *tunnel_01* and *tunnel_03*, the presence of significant vibrations during walking on the handheld platform may lead to a further deterioration of IMU estimation performance when the geometric degeneration occurs, resulting in failures of other algorithms.

D. Dynamic Point Removal and Static Mapping Results

As described in Section IV-D, FMCW-LIO is capable of removing dynamic points and generating static maps according to the Doppler residual and the updated LiDAR velocity. The method operates at the point level rather than the semantic or object level, resulting in direct and efficient performance. The results and comparisons of dynamic point detection and static mapping are shown in Fig. 8. It is obvious that, for conventional methods without removal, ghost points and residual blur of moving persons and buses may persist in the online map. Contrarily, FMCW-LIO can detect dynamic points (red points in Fig. 8(c)–(g)) in the current scan and finally retain points that

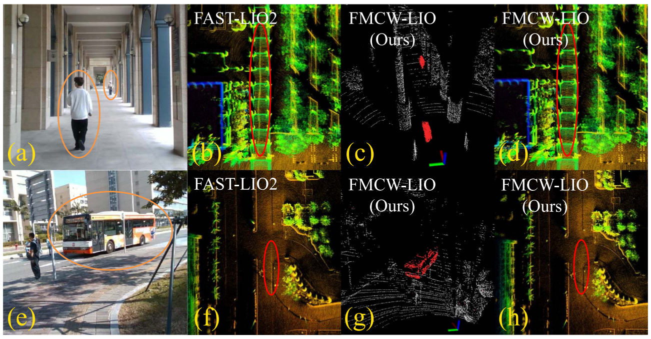

Fig. 8. Dynamic point removal and static mapping. (a), (e) Moving persons and bus (orange ellipsoids). (b), (f) Ghost points (red ellipsoids) in the maps of FAST-LIO2. (c), (g) Online dynamic point detection (red points) of FMCW-LIO. (d), (h) Dynamic point removal (no ghost point in red ellipsoids) of FMCW-LIO.

are nearly static in the map. As a result, ghost points and blur artifacts do not manifest in the map generated by FMCW-LIO. More online results can be found in the video.

VI. Conclusion

In this letter, the Doppler-aided FMCW-LIO is presented for robotic state estimation and mapping. To achieve accurate and robust state estimation, an on-manifold IESEKF is formulated, fusing IMU and FMCW LiDAR measurements. The motion compensation for Doppler measurements is addressed by the IMU-driven backward propagation. FMCW-LIO utilizes the inherent Doppler-aided LiDAR velocity observation, enabling robust state estimation and static mapping, even in severely narrow or structure-degenerated scenes. Diverse experiments on multiple platforms validate the effectiveness and show the superior performance of FMCW-LIO.

References

[1] J. Anderson, R. Massaro, J. Curry, R. Reibel, J. Nelson, and J. Edwards, “Ladar: Frequency-modulated, continuous wave laser detection and ranging,” *Photogrammetric Eng. Remote Sens.*, vol. 83, no. 11, pp. 721–727, 2017.

[2] B. Hexsel, H. Vhavle, and Y. Chen, “DICP: Doppler iterative closest point algorithm,” in *Proc. Robot.: Sci. Syst.*, 2022. [Online]. Available: http://www.roboticsproceedings.org/rss18/p015.html

[3] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, “LIO-SAM: Tightly-coupled lidar inertial odometry via smoothing and mapping,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 5135–5142.

[4] H. Ye, Y. Chen, and M. Liu, “Tightly coupled 3D LiDAR inertial odometry and mapping,” in *Proc. Int. Conf. Robot. Automat.*, 2019, pp. 3144–3150.

[5] C. Qin, H. Ye, C. E. Pranata, J. Han, S. Zhang, and M. Liu, “LINS: A lidar-inertial state estimator for robust and efficient navigation,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 8899–8906.

[6] W. Xu and F. Zhang, “FAST-LIO: A fast, robust LiDAR-inertial odometry package by tightly-coupled iterated kalman filter,” *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 3317–3324, Apr. 2021.

[7] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, “FAST-LIO2: Fast direct LiDAR-inertial odometry,” *IEEE Trans. Robot.*, vol. 38, no. 4, pp. 2053–2073, Aug. 2022.

[8] C. Bai, T. Xiao, Y. Chen, H. Wang, F. Zhang, and X. Gao, “Faster-LIO: Lightweight tightly coupled LiDAR-inertial odometry using parallel sparse incremental voxels,” *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 4861–4868, Apr. 2022.

[9] D. He, W. Xu, N. Chen, F. Kong, C. Yuan, and F. Zhang, “Point-LIO: Robust high-bandwidth light detection and ranging inertial odometry,” *Adv. Intell. Syst.*, vol. 5, 2023, Art. no. 2200459.

[10] K. Chen, R. Nemiroff, and B. T. Lopez, “Direct LiDAR-inertial odometry: Lightweight LIO with continuous-time motion correction,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2023, pp. 3983–3989.

[11] M. Guo, K. Zhong, and X. Wang, “Doppler velocity-based algorithm for clustering and velocity estimation of moving objects,” in *Proc. 7th Int. Conf. Automat., Control Robot. Eng.*, 2022, pp. 216–222.

[12] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, “CARLA: An open urban driving simulator,” in *Proc. 1st Annu. Conf. Robot Learn.*, 2017, pp. 1–16.

[13] Y. Gu, H. Cheng, K. Wang, D. Dou, C. Xu, and H. Kong, “Learning moving-object tracking with FMCW LiDAR,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, pp. 3747–3753.

[14] Y. Wu et al., “Picking up speed: Continuous-time LiDAR-only odometry using Doppler velocity measurements,” *IEEE Robot. Automat. Lett.*, vol. 8, no. 1, pp. 264–271, Jan. 2023.

[15] D. J. Yoon et al., “Need for speed: Fast correspondence-free LiDAR-inertial odometry using doppler velocity,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2023, pp. 5304–5310.

[16] M. Yoneda, K.-M. Dahlén, and T. Ogawa, “Extended object tracking with Doppler velocity-based point registration,” in *Proc. IEEE Symp. Sensor Data Fusion Int. Conf. Multisensor Fusion Integration*, 2023, pp. 1–8.

[17] M. Jung, W. Yang, D. Lee, H. Gil, G. Kim, and A. Kim, “HeLiPR: Heterogeneous LiDAR dataset for inter-LiDAR place recognition under spatial and temporal variations,” 2023, *arXiv:2309.14590*.

[18] C. Doer and G. F. Trommer, “An EKF based approach to radar inertial odometry,” in *Proc. IEEE Int. Conf. Multisensor Fusion Integration Intell. Syst.*, 2020, pp. 152–159.

[19] J. Michalczyk, R. Jung, and S. Weiss, “Tightly-coupled EKF-based radar-inertial odometry,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, pp. 12336–12343.

[20] 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 J.*, vol. 21, no. 3, pp. 3314–3323, Feb. 2021.

[21] Y. Z. Ng, B. Choi, R. Tan, and L. Heng, “Continuous-time radar-inertial odometry for automotive radars,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 323–330.

[22] Y. Zhuang, B. Wang, J. Huai, and M. Li, “4D iRIOM: 4D imaging radar inertial odometry and mapping,” *IEEE Robot. Automat. Lett.*, vol. 8, no. 6, pp. 3246–3253, Jun. 2023.

[23] D. He, W. Xu, and F. Zhang, “Kalman filters on differentiable manifolds,” 2021, *arXiv:2102.03804*.

[24] N. El-Sheimy, H. Hou, and X. Niu, “Analysis and modeling of inertial sensors using Allan variance,” *IEEE Trans. Instrum. Meas.*, vol. 57, no. 1, pp. 140–149, Jan. 2008.

[25] J. Zhang and S. Singh, “LOAM: Lidar odometry and mapping in real-time,” in *Proc. Robot.: Sci. Syst.*, 2014, vol. 2, pp. 1–9.

[26] “Aeva Inc. Aeries II.” Accessed: Aug. 8, 2023. [Online]. Available: https://www.aeva.com/aeries-ii/

[27] Y. Wang, Z. Sun, C.-Z. Xu, S. E. Sarma, J. Yang, and H. Kong, “LiDAR iris for loop-closure detection,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 5769–5775.

[28] F. Dellaert, “Factor graphs and GTSAM: A hands-on introduction,” Georgia Institute of Technology, Tech. Rep, vol. 2, p. 4, 2012.