Robust Multi-Modal Multi-LiDAR-Inertial Odometry and Mapping for Indoor Environments

Li Qingqing*, Yu Xianjia*, Jorge Peña Queralta*, Tomi Westerlund*

*Turku Intelligent Embedded and Robotic Systems (TIERS) Lab, University of Turku, Finland.
Emails: [1]{qingqli, xianjia.yu, jopequ, tovewe}@utu.fi

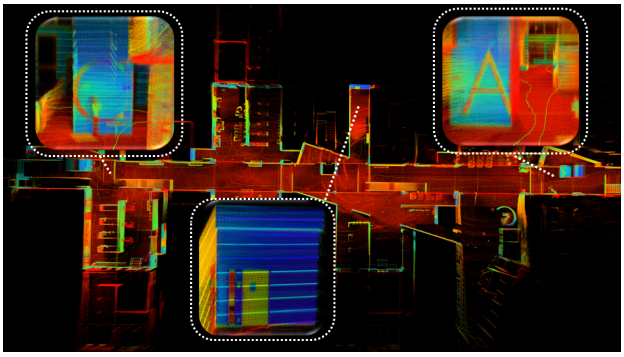
***Abstract*—Integrating multiple LiDAR sensors can significantly enhance a robot's perception of the environment, enabling it to capture adequate measurements for simultaneous localization and mapping (SLAM). Indeed, solid-state LiDARs can bring in high resolution at a low cost to traditional spinning LiDARs in robotic applications. However, their reduced field of view (FoV) limits performance, particularly indoors. In this paper, we propose a tightly-coupled multi-modal multi-LiDAR-inertial SLAM system for surveying and mapping tasks. By taking advantage of both solid-state and spinnings LiDARs, and built-in inertial measurement units (IMU), we achieve both robust and low-drift ego-estimation as well as high-resolution maps in diverse challenging indoor environments (e.g., small, featureless rooms). First, we use spatial-temporal calibration modules to align the timestamp and calibrate extrinsic parameters between sensors. Then, we extract two groups of feature points including edge and plane points, from LiDAR data. Next, with pre-integrated IMU data, an undistortion module is applied to the LiDAR point cloud data. Finally, the undistorted point clouds are merged into one point cloud and processed with a sliding window based optimization module. From extensive experiment results, our method shows competitive performance with state-of-the-art spinning-LiDAR-only or solid-state-LiDAR-only SLAM systems in diverse environments. More results, code, and dataset can be found at https://github.com/TIERS/multi-modal-loam.**

***Index Terms*—LiDAR-inertial odometry, multi-LiDAR systems, sensor fusion, solid-state LiDAR, SLAM, mapping**

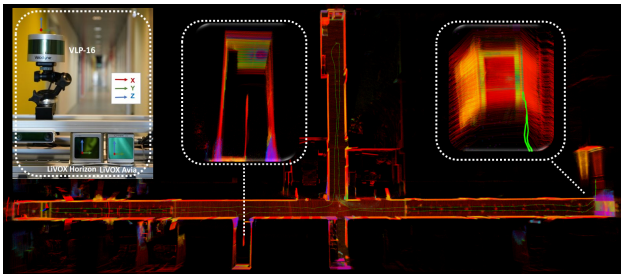
I. Introduction

Autonomous mobile robots rely heavily on spinning 3D LiDAR sensors, which offer high-quality geometric data at long ranges, with a full horizontal FoV, and robust performance across environmental conditions. As a result, this technology has found widespread application in a variety of fields, including self-driving vehicles [1], unmanned aerial vehicles [2], and forest surveying systems [3], [4], among other areas.

In odometry and mapping tasks, achieving denser 3D geometry measurements of the surrounding environment is crucial for enhancing 3D environment understanding. Unfortunately, spinning LiDAR with higher resolution and multiple beams can be costly due to its more complicated architecture. Low-resolution spinning LiDAR, while more affordable, produces sparse point clouds with limited features, making the problem difficult to tackle [5] and leading to inevitable inherent alignment errors [6].



(a) Mapping result with the proposed system at a hall environment. Thanks to the high resolution of solid-state LiDAR with a non-repetitive scan pattern, the mapping result is able to show clear detail of object's surface.



(b) Hardware and Mapping result in long corridor environment. Our proposed methods show robust performance in long corridor environments and survive in narrow spaces where 180°U-turn occurs.

Fig. 1: Our proposed methods show high-resolution mapping results and robust performance in challenging environment

LiDAR technology has advanced rapidly in recent years, with new sensors able of generating image-like data in addition to point clouds [7], [8], and new solid-state LiDAR sensors that offer dense 3D point clouds with non-repetitive scan patterns, while at the same time lowering the cost [9], [10]. Despite the clear advantages of solid-state LiDARs, the naturally narrow horizontal FoV leads, in a similar way to monocular pinhole camera systems, to the sensing volume being blocked by objects, or even entirely occupied by a near wall, resulting in an insufficient number of feature points to estimate a 6-degree-of-freedom (6-DoF) pose. This limitation has made current solid-state-based SLAM methods challenging to apply in outdoor or large indoor scenarios [10], [11].

Multiple studies in the literature have focused on improving LiDAR maps by integrating point clouds from multiple LiDAR sensors [12], [13]. However, the low frame publishing frequency of typical LiDARs (e.g., 10 Hz) can hinder an accurate 6-DOF pose estimation in multi-LiDAR systems. In contrast, IMUs have been widely used in state-of-the-art SLAM systems [14], [10], due to their ability to measure acceleration and angular velocity at a high frequency (e.g., 200 Hz) in three-dimensional space. Nonetheless, there remains a lack of methods that can effectively exploit multi-LiDAR inertial systems for odometry estimations.

To improve the robustness of the SLAM system, we propose a novel tightly-coupled multi-modal multi-LiDAR-inertial odometry and mapping system, which takes advantage of both the large horizontal FoV from a spinning LiDAR and the dense measurements from a solid-state LiDAR as Table I shows. The proposed system first performs spatial-temporal calibration to align the timestamp and calibrate the extrinsic parameters between sensors. Then, we extract two group feature points, edge and planar points, from LiDAR data. Next, with pre-integrated IMU data, an un-distortion module is employed on Lidar point cloud data. Finally, the un-distorted point cloud is merged into one point cloud and sent to sliding window based optimization module. This work is, to the best of our knowledge, the first multi-LiDAR-inertial SLAM system able to effectively integrate LiDAR sensors with heterogeneous scan modalities within a single estimation and optimization framework. This work is inspired by the limitations found in state-of-the-art algorithms for different LiDAR sensors in our previous works [15], [11], where we show that low-cost solid-state LiDARs outperform high-resolution spinning LiDAR in an outdoor environment, while at the same time perform poorly in indoor environments. The unique characteristics and main contributions of our work can be summarized as follows:

1) Present a complete solution for multi-modal LiDAR spatio-temporal calibration and feature extraction. The method adopts an ICP-based scan-matching approach to obtain the extrinsic parameters, split-and-merge based timestamp alignment, and unified channel based feature extraction for both spinning and solid-state-lidar.
2) Design and implementation of a novel tightly-coupled multi-modal multi-LiDAR-inertial mapping framework that is able to combine LiDARs with different scanning modalities and IMU for odometry estimations.
3) The demonstration of a SLAM method for taking advantage of low-cost spinning LiDARs and solid-state LiDARs that outperform the state-of-the-art in high-resolution mapping with high levels of detail.

We provide a unique open-source implementation for tightly-coupled multi-modal LiDAR and IMU fusion available to the community. Through extensive experiments, our proposed methods show state-of-the-art capabilities in various environments, with comparable odometry estimations and higher map quality. The structure of the paper is as follows. Section II surveys the existing research in multi-LiDAR systems and mapping and SLAM with solid-state LiDAR. Section III introduces our proposed mythology. Section IV delves into the details of the implementation and experimental results. Finally, Section V concludes the study and suggests future work.

TABLE I: Characterization of off-the-shelf LiDAR sensors based on horizontal resolution (H. Res.), vertical resolution (V. Res.), and cost.

Lidar Types	**High H. Res.**	**High V. Res.**	**Low-cost**
Spinning, 64+ channels	✔	✔	✗
Spinning, 16-32 channels	✔	✗	✔
Solid-State	✗	✔	✔
Ours (solid-state + spinning-16)	✔	✔	✔

II. Related Works

A. Feature-based LiDAR odometry and mapping

LiDAR odometry generally employs scan-matching techniques including ICP, KISS-ICP [16], GICP [17], and others to determine the relative transformation between two successive frames. Feature-based matching approaches have gained popularity as a computationally efficient alternative to full point cloud matching. For example, in [18], Zhang et al. propose the registration of edge and plane features for real-time LiDAR odometry. This type of operation assumes that the LiDAR moves within a structured environment, with edge and plane points clearly identifiable from the point clouds. The matching of consecutive scans is then performed by solving a least-squares optimization problem. SLAM with features of planes for indoor environments has attracted many researchers' interests as planes ubiquitously exist in indoor environments [19].

B. SLAM with solid-state LiDARs

With the development of LiDAR technology, low-cost and high-performance solid-state LiDAR has attracted significant researcher interest. The different sensing characteristics means that new challenges in point cloud registration and mapping arise. In [9], Lin et al. address several fundamental challenges with a robust, real-time LiDAR odometry and mapping algorithm for solid-state LiDAR (LOAM Livox) that accounts for the reduced FoV and the non-repetitive sampling patterns have been presented by taking effort on both front-end and back-end in. Other recent results have also presented tightly-coupled LiDAR-inertial odometry and mapping schemes for both solid-state and mechanical LiDARs [10], [20]. Regarding the inherently limited FoV of a single solid-state LiDAR, a decentralized approach for simultaneous calibration, localization, and mapping utilizing multiple solid-state LiDARs was introduced in [21] to enhance system resilience.

C. Multi-modal LiDAR-based SLAM

Relying solely on spinning LiDARs for pose estimation is suboptimal, as registering skewed point clouds or features can eventually result in substantial drift. Contemporary LiDAR SLAM systems commonly integrate data from multiple sensors to enhance accuracy. LIO-SAM was proposed as a method

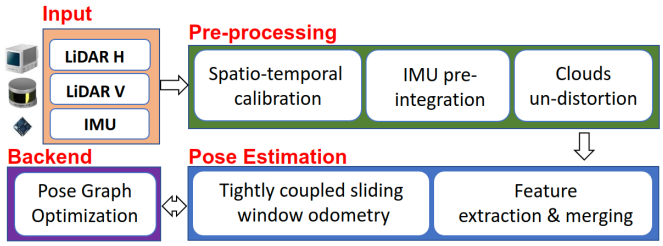


Fig. 2: The pipeline of proposed multi-modal LiDAR-inertial odometry and mapping framework. The system starts with preprocessing module which takes the input from sensors and performs IMU pre-integration, calibrations, and un-distortions. The scan registration module extracts features and sent the features to a tightly coupled sliding window odometry. Finally, a pose graph is built to maintain global consistency.

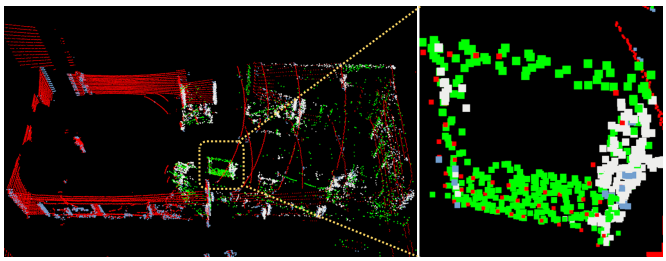


Fig. 3: Extracted features points in office room environment(left) and zoom-in view of one gate object (right). Plane and edge feature points from Velodyne are in red and blue, and from Horizon are in green and white separately.

to eliminate the accumulated drift of LiDAR-inertial odometry over extended periods or in feature-sparse environments, by tightly coupling LiDAR and inertial measurement unit (IMU), and optionally GNSS sensors, via smoothing and mapping [22]. SLAM approaches utilizing the fusion of LiDAR and IMU data can be found in other studies, for example, LIO-Mapping [23] and Fast-LIO with iterated Kalman filter [20]. Moreover, integrating LiDAR-inertia odometry with visual-inertia odometry (VIO) can further improve accuracy while keeping the robustness either in texture-less or feature-less environments [24].

D. Multi-LiDAR odometry and mapping

More recently, the research focus has also shifted towards the fusion of data from multiple LiDARs. For example, [12] addresses multi-LiDAR online extrinsic calibration, odometry, and mapping, where extracted edge and planar features are utilized and data uncertainty is modeled with Gaussian distribution. A tightly coupled LiDAR-inertial odometry and mapping approach with low drift is proposed in [25], which utilizes features extracted from multiple time-synchronized LiDARs with complementary FoV. Rather than utilizing spinning LiDARs, a decentralized extended Kalman filter (EKF) approach for simultaneous calibration, localization, and mapping for multiple solid-state LiDARs was introduced to improve upon the limited FoV of a single solid-state LiDAR [21]. However, we find a lack of work in the literature in terms of fusing multiple LiDARs with different scanning modalities, which has potential due to the different advantages they have in terms of map quality and odometry accuracy.

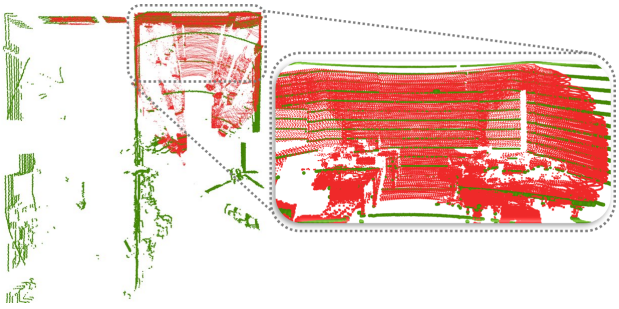


Fig. 4: Multi-LiDAR extrinsic parameter calibration result in an office room environment. Red points come from Horizon and green points from VLP-16. Top view (left), and detail of the matching (right).

III. System Overview

To simplify the system design, we have made follow assumptions:1) The LiDARs are synchronized at the software level. The time offset between sensors is not considered in our system 2)The extrinsic parameters between IMU and at least one LiDAR are known. In our case, we use LiDAR's built-in IMU and use the extrinsic parameter from factory settings.

A. Overview

The design of our system is motivated by our previous work [15] where solid-state LiDAR shows significant performance outdoors but failed all tests indoors. To combine the high situation awareness ability and robust performance, here we proposed multi-modal LiDAR-inertial odometry and mapping scheme. In this paper, we consider a perception system consisting of multiple modal LiDARs and IMU, and LiDAR sensors are not triggered by hardware based external clock. The pipeline of the proposed method is illustrated in Fig. 2 and the hardware system is shown in Fig. 5. Our hardware is composed of a spinning LiDAR Velodyne VLP-16, a low-cost solid-state LiDAR Livox Horizon, and its built-in IMU.

1) System pipeline: The system starts with a data pre-processing module, in which IMU data are pre-integrated, extrinsic parameters between sensors are calibrated, timestamps of clouds with different starting times are aligned, the clouds from multiple LiDARs are un-distorted with IMU pre-integrated results. After pre-processing, feature point clouds representing plane and edge points are extracted and merged into one cloud. The merged feature cloud will be sent to the sliding window odometry module where the feature cloud will be matched against the local map. Together with pre-integrated IMU, fixed size of feature clouds, and local map, six-DoF egomotions and IMU parameters are estimated and optimized by keyframe-based sliding window optimization. At the backend, the system maintains a global pose graph with selected keyframes. Loop closure is detected in a keyframe basis graph using ICP, and a global graph optimization is invoked to guarantee the reconstructed map is globally consistent.

2) Notation and Problem Formulation: We treat IMU coordinate as the base local coordinate indicated as $()^b/()^I$. The merged cloud will be transformed to $()^b/()^I$. We use the first keyframe received by the system as the origin of the world coordinates denoted as $()^w$. The coordinate of spinning

LiDAR is denoted as $()^v$, and the coordinates of solid-state LiDAR are denoted as $()^h$. We use $\mathcal{P}^v_{t_1} = \{\mathbf{p}^v_1, \mathbf{p}^v_2, ...\mathbf{p}^v_n\}$ be the point cloud acquired at time t_1 with spinning LiDAR, and $\mathcal{P}^h_{t_2} = \{\mathbf{p}^h_1, \mathbf{p}^h_2, ...\mathbf{p}^h_n\}$ be the cloud acquired at time t_2 with solid state LiDAR, where $\mathbf{p}^v_i$ and $\mathbf{p}^h_j$ are a point in $\mathcal{P}^v_{t_1}$ and $\mathcal{P}^h_{t_2}$.

We denote $\mathbb{F}_{Ek}$ and $\mathbb{F}_{Pk}$ as the edge and plane feature point cloud extracted from the LiDARs' data at time k. The transformation matrix is denoted as $\mathbf{T}^b_a \in SE(3)$, which transforms a point from frame $()^a$ into the frame $()^b$. $\mathbf{R}^b_a \in SO(3)$ and $\mathbf{t} \in R^3$ are the rotation matrix and the translation vector of $\mathbf{T}^b_a$ respectively.The quaternion $\mathbf{q}^b_a$ under Hamilton notation is used, which corresponds to $\mathbf{R}^b_a$. $\otimes$ is used for the multiplication of two quaternions. $\mathbf{q}^b_a$ and $\mathbf{R}^b_a$ can be convert by Rodrigues formula. With a given point cloud from multi-modal LiDAR sensor and IMU info, the state needs to be optimized for keyframe k is defined as (1) where $\mathbf{t}_k$ is the translation vector, $\mathbf{q}_k$ represents orientation in quaternion, $\mathbf{v}_k$ is the velocity, $\mathbf{b}_{a_k}$ and $\mathbf{b}_{g_k}$ are the bias vector of the accelerator and gyroscope.

$$\mathbf{X}_k = [\mathbf{p}_k, \mathbf{q}_k, \mathbf{v}_k, \mathbf{b}_{a_k}, \mathbf{b}_{g_k}] \in \mathbb{R}^3 \times \mathbb{S}^3 \times \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^3 \quad (1)$$

B. Pre-processing

1) Spatial-temporal Calibration and Initialization: As the extrinsic parameter between IMU and one LiDAR is known, so here we focus on calibrating the extrinsic parameters between two LiDARs. We assume the sensor platform is stationary during the extrinsic calibration process. As solid-state LiDAR with the non-repetitive pattern is able to obtain more details from the environment within the FoV, therefore, we integrated n consecutive frames(e.g., ten) to new point cloud $\mathcal{P}^h_{t_1 \sim t_n}$ for the calibration process. Let $\mathcal{P}^v_{t_m}$ be one cloud data obtained at $t_m \in (t_1, t_n)$ from spinning LiDAR. Generalized Iterated Closest Point (GICP)[17] method is employed to caculate the relative transformation matrix $\mathbf{T}^h_v$ between $\mathcal{P}^h_{t_1 \sim t_n}$ and $\mathcal{P}^v_{t_m}$ from the overlapped region. The extrinsic calibration results with data collected from a classroom environment are shown in Fig. 4. As the transformation matrix $\mathbf{T}^i_h$ between IMU and one LiDAR given by the factory, we can get $\mathbf{T}^i_v$ by $\mathbf{T}^i_v = \mathbf{T}^h_v * \mathbf{T}^i_h$.

We consider a system that LiDARs are not triggered with the external clock(e.g. GNSS) like [12], each cloud $\mathcal{P}^h$ and $\mathcal{P}^v$ are collected at different starting timestamps. To merge clouds into one combined cloud $\mathcal{P}^m$, we need to align the starting timestamp and ending timestamp. We adopt a split-and-merge method similar to [13].The individual timestamp of $p^h_i \in \mathcal{P}^h$ and $p^v_i \in \mathcal{P}^v$ can be obtained from the sensors driver. If the timestamp for a point $p^v_i \in \mathcal{P}^v$ is not available, it also can be calculated by orientation difference [18]. When a new cloud $\mathcal{P}^h_{t_k}$ received at time t_k, we put all points $p^v_i \in \mathcal{P}^h_{t_k}$ to a queue $\mathbb{Q}^h$ which ordered by timestamp. When a new cloud $\mathcal{P}^v_{t_m}$ received at time t_m, we first get its start time t_{m_s} and end time t_{m_e}, then all points in $\mathbb{Q}^h$ which timestamp $t_i < t_{m_s}$will be dropped, and $t_i \in (t_{m_s}, t_{m_e})$ will be popped to a new frame $\mathcal{P}^h_{t_m}$ which share the same time domain with $\mathcal{P}^v_{t_m}$.

2) IMU Initialization and Preintegration: The IMU sensor output angular velocity and acceleration measurements are defined as $\tilde{\omega}_t$ and $\tilde{\mathbf{a}}_t$ using equations. 2 and 3:

$$\boldsymbol{\omega}_t = \boldsymbol{\omega}_t + \mathbf{b}^{\boldsymbol{\omega}}_t + \mathbf{n}^{\boldsymbol{\omega}}_t \quad (2)$$

$$\tilde{\mathbf{a}_t} = \mathbf{R}^{WL}_t(\mathbf{a}_t - \mathbf{g}) + \mathbf{b_t}^{\mathbf{a}} + \mathbf{n_t}^{\mathbf{a}} \quad (3)$$

Where $\mathbf{b}_t$ is the measurement bias and $\mathbf{n}_t$ is white noise. $\mathbf{R}^{LW}_t$ is the rotation matrix from World coordinate $()^L$ to local coordinate $()^W$, $\mathbf{g}$ is the gravity vector in world coordinate. Based on the raw measurement $\boldsymbol{\omega}_t$ and $\tilde{\mathbf{a}}_t$, we can infer the motion of the robot as follows:

$$\begin{aligned}\mathbf{p}_{t+\Delta t} = \mathbf{p}_t + \mathbf{v}_t\Delta t + \frac{1}{2}\mathbf{g}\Delta t^2 + \\ \frac{1}{2}\mathbf{R}_t(\hat{\mathbf{a}}_t - \mathbf{b}^{\mathbf{a}}_t - \mathbf{n}^{\mathbf{a}}_t)\Delta t^2\end{aligned} \quad (4)$$

$$\mathbf{v}_{t+\Delta t} = \mathbf{v}_t + \mathbf{g}\Delta t + \mathbf{R}_t(\hat{\mathbf{a}}_t - \mathbf{b}^{\mathbf{a}}_t - \mathbf{n}^{\mathbf{a}}_t)\Delta t \quad (5)$$

$$\begin{aligned}\mathbf{q}_{t+\Delta t} &= \mathbf{q}_t \otimes \mathbf{q}_{\Delta t} \\ &= \mathbf{q}_t \otimes \begin{bmatrix} exp(\frac{1}{2}\Delta t(\tilde{\boldsymbol{\omega}}_t - \mathbf{b}^{\boldsymbol{\omega}}_t - \mathbf{n}^{\boldsymbol{\omega}}_t)) \\ 1 \end{bmatrix}\end{aligned} \quad (6)$$

Where the $\mathbf{p}_t, \mathbf{v}_t$ and $\mathbf{q}_t$ are the estimated position, velocity and orientation in quaternion at time t, $\mathbf{p}_{t+\Delta t}, \mathbf{v}_{t+\Delta t}$ and $\mathbf{q}_{t+\Delta t}$ are the estimated state at time $t+\Delta t$. We apply the IMU preintegration method proposed in [14]. The relative motion between two timestamp $\Delta\mathbf{V}, \Delta\mathbf{P}, \Delta\mathbf{Q}$ can be caulated based on equations 4∼6 and will be used for initial guess in III-C3.

C. Multi-modal LiDAR Pose Estimation

1) Union Feature Extraction: For computing efficiency, feature extraction is essential for the SLAM system. We focus on extracting the general features that exist in different modal LiDARs and can be shared in the optimization process. Here we extract feature points based on [10] that selects a set of feature points from measurements according to their continuous and surface normal vector. We extend the method and adapt it to both spinning and solid-state LiDARs. The set of extracted features consists of two subsets: plane points $\mathbb{F}_P$ and edge points $\mathbb{F}_E$. By checking the continuity, the edge feature included two types of points $\mathbb{F}_{E_l}$ and $\mathbb{F}_{E_b}$ where $\mathbb{F}_{E_l}$ represents the line feature where two surface meets, and $\mathbb{F}_{E_b}$ represents breaking points where plane end.

Let $\mathcal{P}^v_t$ be the point cloud acquired at time t from spinning LiDAR, $\mathcal{P}^h_t$ be the point cloud acquired from solid-state LiDAR at the same time domain after temporal alignment. If the channel number of each point $p_v \in \mathcal{P}^v_t$ is unavailable, then We first project points in $\mathcal{P}^v_t$ onto a range image based on the horizontal and vertical angle w.r.t. the origin. Each row represents data from one channel of the spinning lidar. Then the points are divided into N subsets $\{\mathbf{L}^v_i\}_{i \in N}$ where N is the total channel numbers of spinning lidar. For point cloud $\mathcal{P}^h_t$, we divide the point based on line number which can be obtained from Livox ROS driver [1]. Similarly, we first divided the points into M subsets $\{\mathbf{L}^h_i\}_{i \in M}$ where M is the total line numbers of solid-state lidar. The points in each $\mathbf{L}^v_{i\,i \in N}$ and $\mathbf{L}^h_{i\,i \in M}$ are ordered by timestamp. For each subset $\mathbf{L}^v_i$ in $\{\mathbf{L}^v_i\}_{i \in N}$ and $\mathbf{L}^h_j$ $\{\mathbf{L}^h_j\}_{j \in M}$, we first extract the continues points $\mathcal{P}^v_{i\,C}$ and $\mathcal{P}^h_{j\,C}$ by checking the depth difference with its neighbour points. If the depth difference between the point in

[1]https://github.com/Livox-SDK/livox_ros_driver

$\mathbf{L}_i^v$ or $\mathbf{L}_j^h$ and nearest neighbor points within the same subset is smaller than the depth threshold d_{th}, then the point is added to continuous points subset $\mathcal{P}_{i\,C}^h$ or $\mathcal{P}_{j\,C}^v$. Then we follow the feature extraction methods in [13], where a scatter matrix $\mathbf{\Sigma}$ is calculated based on neighbor points. By analyzing the two largest eigenvalues λ_1 and λ_2 of $\mathbf{\Sigma}$, the plane points are detected and labeled as a plane, the point where two plane meet is labeled as corner features, the points where one plane ends and neighboring with discontinuous points are labeled as break points. We merge the corner points and break points together and use them as edge feature points. After this process, plane feature points $\mathbb{F}_{Pt}^h$ and $\mathbb{F}_{Pt}^v$, edge feature points $\mathbb{F}_{Et}^h$ and $\mathbb{F}_{Et}^v$ are extracted. We show extracted feature points in the office room environment in Fig. 3

2) Feature Clouds Merging: Instead of keeping two different system state $\mathbf{X}_t^w$ for clouds $\mathcal{P}_t^h$ and $\mathcal{P}_t^v$, we merge the same type of feature point cloud into one frame and use a single system state. As the cloud $\mathcal{P}_t^h$ from solid-state lidar can be more easily blocked by near objects, with extreme cases leading to all points reflecting on a single plane (e.g., a close wall) in an indoor environment, this means that there is a certain probability of most of the points in $\mathbb{F}_{Pt}^h$ belonging to a single plane. This means that insufficient $\mathbb{F}_{Et}^h$ points can be extracted. In this case, we treat the $\mathcal{P}_t^h$ as *bad frame*, with the corresponding feature cloud not being considered to the union feature cloud $\mathbb{F}_{Et}^i$ and $\mathbb{F}_{Pt}^i$. This ensures more consistent behavior and robust estimation across environments and over time. To detect such *bad frames*, We first remove the points in $\mathbb{F}_{Pt}^h$ and $\mathbb{F}_{Et}^h$ that is close to the origin of the sensor (e.g., 2 m threshold), and then check the amount n_e of edge point in the feature cloud $\mathbb{F}_{Et}^h$. If n_e is smaller than the edge feature threshold τ_e (e.g., 100 in our experiments), then the cloud $\mathcal{P}_t^h$ is treated as a *bad frame*. If no such *bad frame* is detected, we transform the complete feature clouds $\mathbb{F}_{Pk}^h$, $\mathbb{F}_{Pt}^v$, $\mathbb{F}_{Et}^h$ and $\mathbb{F}_{Et}^v$ to the $()^i$ coordinate frame and merge them to fused, unified feature clouds $\mathbb{F}_{Ek}^i$ and $\mathbb{F}_{Pk}^i$ using the extrinsic transformation matrices $\mathbf{T}_v^i$ and $\mathbf{T}_h^i$, calculated as described in Section. III-B1 with Eq. (7).

$$\mathbb{F}_E^i = \mathbf{T}_v^i * \mathbb{F}_E^v + \mathbf{T}_h^i * \mathbb{F}_E^h, \mathbb{F}_P^i = \mathbf{T}_v^i * \mathbb{F}_P^v + \mathbf{T}_h^i * \mathbb{F}_P^h. \quad (7)$$

If $\mathbb{P}_t^h$ is "bad frame", then $\mathbb{F}_E^i = \mathbf{T}_v^i * \mathbb{F}_E^v$ and $\mathbb{F}_P^i = \mathbf{T}_v^i * \mathbb{F}_P^v$. The union feature clouds $\mathbb{F}_E^i$ and $\mathbb{F}_P^i$ will be down-sampled before sending them into sliding window optimization module.

3) Keyframe Selection & Undistortion: Given feature point cloud and preintegrated IMU within the same time domain $\mathbb{F}_{Ek}^i, \mathbb{F}_{Pk}^i, \mathbb{I}_{preg_k}^i$ and a point cloud feature map $\mathbb{M}_k^w$ in world coordinate, the registration problem can be formulated as solving a non-linear least square problem. The initial guess of the state $\mathbf{X}^w$ is estimated with Eq. (8):

$$\begin{aligned} \tilde{\mathbf{p}}_k &= \bar{\mathbf{p}}_{k-1} + \bar{\mathbf{q}}_{k-1} * \Delta\mathbf{P}_{k-1}^k \\ \tilde{\mathbf{q}}_k &= \bar{\mathbf{q}}_{k-1} * \Delta\mathbf{Q}_{k-1}^k, \tilde{\mathbf{b}_{\mathbf{g}_k}} = \bar{\mathbf{b}_{\mathbf{g}_{k-1}}} \\ \tilde{\mathbf{v}}_k &= \bar{\mathbf{v}}_{k-1} + \bar{\mathbf{q}}_{k-1} * \Delta\mathbf{V}_{k-1}^k, \tilde{\mathbf{b}_{\mathbf{a}_k}} = \bar{\mathbf{b}_{\mathbf{a}_{k-1}}} \end{aligned} \quad (8)$$

As sliding window optimization is a relatively heavy process, therefore, maintain the sparsity of the frames in the window can significantly affect the real-time performance. Here we check the IMU drift during the time interval of two consecutive keyframes, and select the frame as keyframe if the orientation difference is higher than a certain degree (e.g., 30°) or the time difference between a current frame and the last keyframe larger then certain time (e.g., 2 seconds). Then each point in the selected keyframe will be un-distorted with ΔQ and ΔP provided by IMU pre-integration. Each keyframe contains deskewed feature clouds $\mathbb{F}_{Ek}^i$ and $\mathbb{F}_{Pk}^i$, pre-integrated IMU $\mathbb{I}_{preg_k}^i$, and initial guess of state $\tilde{\mathbf{X}}_k^w \sim [\tilde{\mathbf{p}}_k, \tilde{\mathbf{q}}_k, \tilde{\mathbf{v}}_k, \tilde{\mathbf{b}_a}_k, \tilde{\mathbf{b}_g}_k]$ which will be optimized by sliding window optimization.

4) Sliding Window Optimization: In this paper, we follow keyframe based tightly coupled lidar-inertial sliding window optimization strategy in [10]. The merged feature points $\mathbb{F}_{Ek}^i$ and $\mathbb{F}_{Pk}^i$ of keyframe k are treated as feature clouds that are extracted from single lidar sensor as in [10]. We build a window with τ consecutive keyframes where the states that need to be optimized for each frame are $\tilde{\mathbf{X}^w} = [\tilde{\mathbf{X}_1^w}, \tilde{\mathbf{X}_2^w}, ..., \tilde{\mathbf{X}_\tau^w}]$. The optimal state can be obtained by minimizing the function:

$$\min_{\tilde{\mathbf{X}}}\{||\mathbb{D}_{prior}(\tilde{\mathbf{X}^w})||^2 + \sum_{k=1}^{\tau}\mathbb{D}_L(\tilde{\mathbf{X}_k^w}) + \sum_{k=1}^{\tau}\mathbb{D}_I(\tilde{\mathbf{X}_k^w})\} \quad (9)$$

Where $||\mathbb{D}_{prior}(\tilde{\mathbf{X}^w})||^2$ represents the prior residual term which is generated by marginalizing oldest frames before the current window via Schur-complement [14],$\mathbb{D}_I(\tilde{X^w})$ represents the pre-integrated IMU terms as defined in [10].$\mathbb{D}_L(\tilde{\mathbf{X}_k^w})$ is lidar term defined as (10).

$$\sum_{a=1}^{m}(\mathbb{D}_e(\mathbf{X}_k^w, \mathbf{p}_{k,a}^i, \mathbb{M}_k^w))^2 + \sum_{b=1}^{n}(\mathbb{D}_s(\mathbf{X}_k^w, \mathbf{p}_{k,b}^i, \mathbb{M}_k^w))^2 \quad (10)$$

$\mathbb{D}_e$ is the point-to-edge residual term defined as (11) and $\mathbb{D}_s$ point-to-plane residual term defined as (12).

$$\mathbb{D}_e(\mathbf{X}^w, \mathbf{p}^i, \mathbb{M}^w)) = \frac{||(\mathbf{p}^w - \grave{\boldsymbol{e}}^{\boldsymbol{w}}) \times (\mathbf{p}^w - \grave{\boldsymbol{e}}^{\boldsymbol{w}})||}{||\acute{\boldsymbol{e}}^{\boldsymbol{w}} - \grave{\boldsymbol{e}}^{\boldsymbol{w}}||} \quad (11)$$

$$\mathbb{D}_s(\mathbf{X}^w, \mathbf{p}^i, \mathbb{M}^w)) = |\mathbf{n}_s^T\mathbf{p}^w + 1/||\mathbf{n}_s||| \quad (12)$$

where $\mathbf{p}^i$ represents a feature point belonging to $\mathbb{F}_{Ek}, \mathbb{F}_{Pk}$. Then, $\mathbf{p}^w = \mathbf{R(q)}\mathbf{p}^i + \mathbf{t}$ represents the scan point $\mathbf{p}^i$ at local frame $()^i$, which is transformed to world frame $()^w$ given the state estimation $[\mathbf{q}, \mathbf{t}]$ in $\mathbf{X}^w$. We denote by $\grave{\boldsymbol{e}}^{\boldsymbol{w}}$ and $\acute{\boldsymbol{e}}^{\boldsymbol{w}}$ the two closest corresponding edge feature points on the feature map $\mathbb{M}^w$, while $\mathbf{n}_s^w$ is the plane normal vector that is calculated by neighbor plane feature points in the $\mathbb{M}^w$ cloud. We solve the non-linear Eq. (9) using the Ceres Solver toolbox [26]. To ensure global consistency, we also maintain a pose-graph structure with optimized states $\mathbf{X}^w$ and pre-integrated IMU measurements as optimization constraints.

IV. Evaluation

A. Sensor Configuration and Implementation

We implement the proposed multi-modal multi-LiDAR-inertial odometry and mapping system in C++ with ROS melodic environment that can be shared in the robotic community. The system shown in Fig. 2 is structured in four nodes: preprocessing, feature extraction, scan registration, and graph

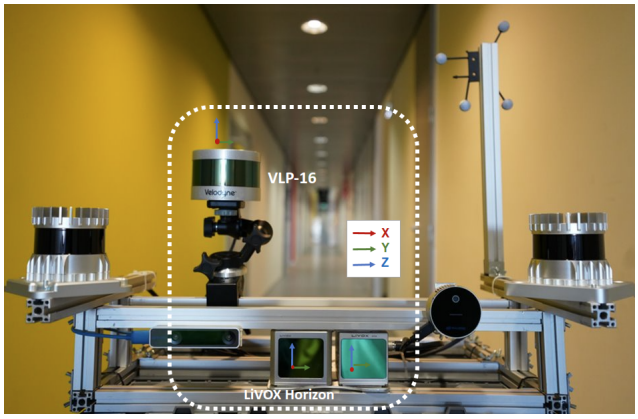


Fig. 5: Hardware platform used for data acquisition. The sensors used in this work are the Livox Horizon LiDAR, with its built-in IMU, and the Velodyne VLP-16 LiDAR. The platform is mounted on a moving ground vehicle.

TABLE II: End-to-end position error in meters (N/A when odometry estimations diverge; V: Velodyne VLP-16, H: Livox Horizon, I: IMU). Numbers in bold indicate the best performance, while underscored numbers indicate the second best in each environment.

Dataset	Hall	Corridor	Office
LeGo (V)	0.567	0.336	0.127
FLIO (HI / VI)	0.109 / 0.069	N/A / **0.062**	N/A / 0.188
LIOM (HI / VI)	N/A / 0.736	N/A / 1.951	NA / **0.102**
Ours (HI / VI)	N/A / 0.107	N/A / 0.132	NA / 0.165
Ours (HVI)	**0.051**	0.085	0.124

optimization. The factor graph optimization is maintained by GTSAM 4.0 [27], and non-linear optimization is performed by Ceres Solver 2.0 [26]. The framework proposed in this paper is validated using datasets gathered by Velodyne VLP-16 (V), Livox Horizon (H) 3D LiDAR, and its built-in IMU (I). The VLP-16 measurement range is up to 100 m with an accuracy of ± 3cm. It has a vertical FoV of 30°(± 15°) and a horizontal FoV of 360°. The 16-channel sensor provides a vertical angular resolution of 2°and The horizontal angular resolution varies from 0.1°to 0.4°. For solid-state LiDAR, we selected Livox Horizon, which is designed with an FoV of 81.7° × 25.1°. Horizon was scanning at 10 HZ and reaches a similar but more uniform FoV coverage compared with typical 64-line mechanical LiDARs. The extrinsic parameter between Horizon and its built-in IMU is provided by factory instruction. The sensors are connected to a laptop directly with Ethernet and synchronized with software-based precise timestamp protocol (PTP) [28]. We run ROS drivers for Velodyne and Horizon and recorded the data in rosbag format.

B. Qualitative Experiment

From our previous research [15], [11], a tightly coupled solid-state LiDAR-inertial system shows competitive performance outdoors but poorly in indoor environments. Therefore, here we aim to compare our proposed system with a typical and challenging indoor environment: an office room, a long corridor, and a large hall. The data are gathered with the platform as Fig. 5 shows at ICT-City in Turku, Finland.

We compare our proposed method with several state-of-the-art SLAM algorithms: LeGO-LOAM [29] [2], Fast-LIO [20] [3] and LILI-OM [10] [4]. LeGO-LOAM is LiDAR only odometry, Fast-LIO and LILI-OM are tightly coupled LiDAR inertial odometry which can both work with solid state LiDAR and spinning LiDARs. Fast-LIO features with a tightly-coupled iterated extended Kalman filter framework and iKD-tree data structure which show efficient and robust performance [20]. Similar to our proposed method, LILI-OM employs keyframe-based sliding window optimization but only fuse single LiDAR and pre-integrated IMU measurements. As Fast-LIO, LI During the experiments, we use the default configurations from the official Github repository, and loop closure detection is off for each method. To compare the odometry accuracy, all three datasets were starting and ended at the same place. The mean square distance (MSE) between the starting and ending positions is treated as the error. The results generated by selected methods in all dataset shows in Table II.

1) Hall: This data was recorded at a hall environment around 127m × 35 m to compare odometry and mapping performance in a relatively large indoor environment. The recording started at a narrow space and was followed by a 180°U-turn where most of the FoV of the solid-state-LiDAR are covered by near walls, therefore, solid-state LiDAR only based methods cannot receive enough features that might cause huge drift. We show the trajectory in Fig. 7 and mapping results in Fig. 1a. From the position error shown in Table II, our proposed methods show the best performance reaching 5.1 cm, and Fast-LIO (VI) shows the second best performance.

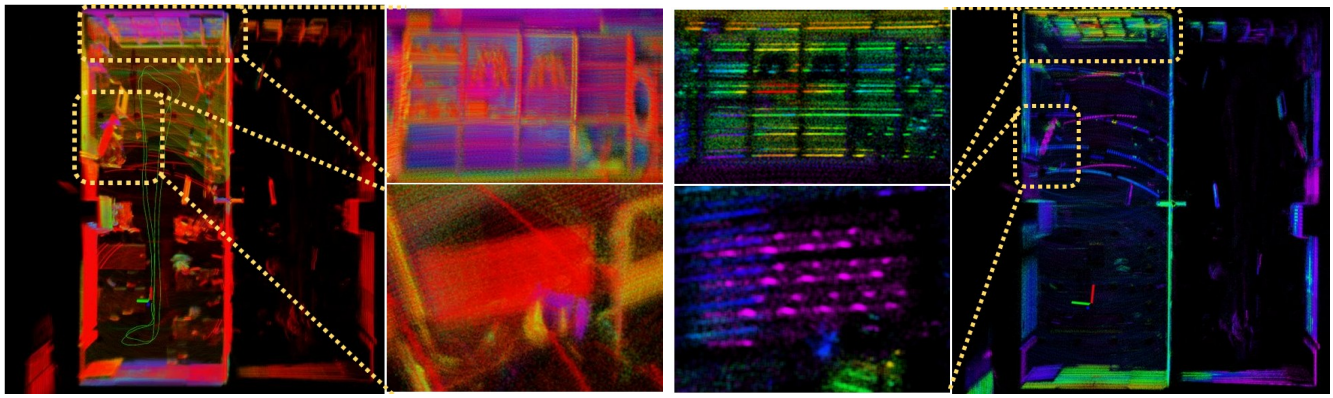
2) Corridor: The corridor environment is another challenging environment as low-resolution LiDARs might not get enough feature points from the environment to perform robust localization. The data sequence was recorded at a 60 m long corridor. The mapping results are shown in Fig. 1b. Fast-LIO (VI) performs the best, while our proposed method follows closely in performance.

3) Office: Another data sequence is recorded in a small office room with a size of $12\,m \times 3.7\,m$. To make the mapping task more challenging in this environment, we perform several fast 180°U-turns during the data recording. From the result, we can see LIOM (VI) shows the best performance while our proposed method with HVI ranks second.

C. Outdoor Mapping

For the sake of completeness, we also test the proposed method with an outdoor data sequence on a city road and a forest environment. The resulting map is shown in Fig. 8. A qualitative analysis of the map point cloud shows a high level of detail. However, the extrinsic calibration method has been designed for indoor environments with enough edge and planar features within the overlapped FoV between the two LiDAR sensors; therefore, the extrinsic parameters are not well calibrated in the forest environment, which leads to a decrease in sharpness in the final map. In any case,

[2]https://github.com/RobustFieldAutonomyLab/LeGO-LOAM
[3]https://github.com/hku-mars/FAST_LIO
[4]https://github.com/KIT-ISAS/lili-om



(a) Map results generated by our methods in HVI mode.

(b) Map results generated Fast_LIO in VI mode.

Fig. 6: Qualitative comparison of map details in the office room dataset sequence. The color of the points represents the reflectivity provided by raw sensor data. The point size is $1\,cm^3$, and transparency is set to 0.05. The middle two columns show the zoom-in view of the wall (top) and TV (bottom).

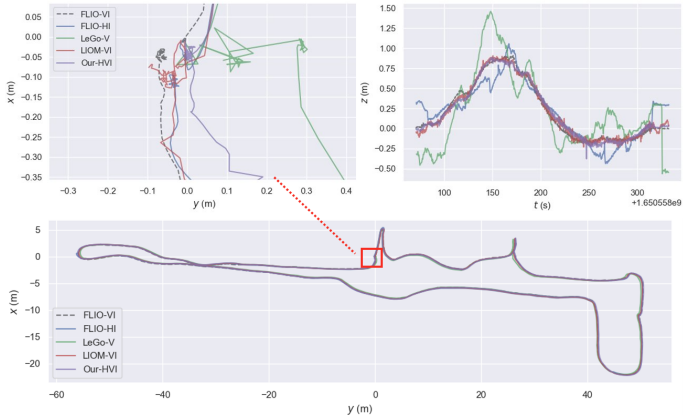


Fig. 7: The trajectory result on dataset Hall. Our proposed methods show the smallest error when returning to the start point. The trajectory from different methods (bottom), the zoom-in view of starting and ending point(top left), the changes along Z-axis(top right)

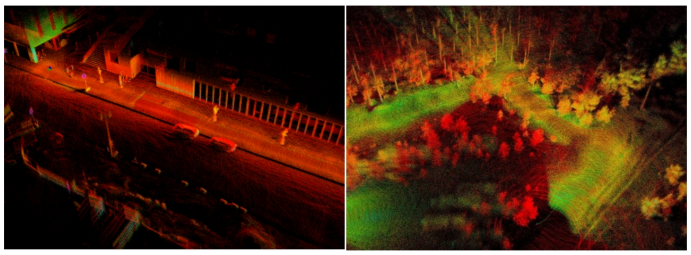


Fig. 8: Mapping results with the proposed method outdoors: urban street (left) and forest environment (right).

this demonstrates the potential for generalization to more unstructured environments and enables high-density mapping even with sub-optimal extrinsic parameters calibration

Our proposed multi-modal multi-LiDAR-inertial method demonstrates competitive and consistent performance compared to other selected methods. However, we observed that our method did not always outperform other approaches, despite utilizing more measurements from the environment. One possible reason for this could be inaccurate time synchronization between the sensors, our time synchronization between LiDARs is on the software level with sub-microsecond accuracy. The inaccurate timestamp for each point will bring the error to the system during the cloud undistortion and cloud merging steps, which is hard to eliminate.

From the results, the VI-based method is able to track the position of the sensor in all dataset sequences. However, the HI based methods fail in most of the sequences except for Fast_LIO(VI) in the large hall environment. To understand the difference between each LiDAR, we also test our methods with Velodyne-Inertial odometry. The results show the accuracy of the proposed method VI is less accurate than the HVI which indicates the horizon LiDAR can improve the system's performance. It is also worth noting that our focus is on integrating various multi-modal LiDAR sensors at the point cloud registration and feature extraction stages while aiming at a more consistent framework across environments.

D. Mapping Quality Comparison

One of the key benefits of the multi-LiDAR system is its high perception awareness ability. Here we compare the mapping quality in terms of resolution. Part of the mapping results by our proposed methods has shown in Figure. 1a and 1b where the color represents intensity value. From the result, we can see many objects (e.g., door, wall letters). We compare the mapping result between our proposed method in Fig. 6a and Fast_LIO (VI) in Fig. 6b. Our method shows the most stable performance in our experiments in an office room environment. By zooming in the same area, we can see a more uniform point cloud from the wall and a TV in the map generated with our method.

E. Runtime Analysis

Our evaluations were conducted on a laptop with an Intel Core i7-10875H CPU and 64 GB RAM on Unbuntu 18.04.6 LTS system. We show the average runtime per frame in Table III and feature numbers in Table IV. Preprocessing and feature extraction are lightweight. Runtime is dominated by the sliding-window-based pose estimation and optimization.

TABLE III: Analysis of processing time (ms) for the different algorithm stages on an Intel i7-10875H CPU.

	Hall	**Corridor**	**Office**
Pre-processing stage	4.14	4.44	4.37
Multi-LiDAR feature extraction	80.32	86.74	94.21
Pose estimation, optimization	139.71	123.51	132.73

TABLE IV: Average number of feature points per frame extracted from the different LiDARs in the three tested environments. Only points in a range between 2 m, and 50 m are considered.

	V-raw	**V-edge**	**V-plane**	**H-raw**	**H-edge**	**H-plane**
Hall	17370	201	15680	21109	405	1934
Corridor	5965	101	5493	14623	302	4007
Office	20235	329	18430	17462	409	1029

V. Conclusion

We have presented in this paper a tightly coupled multi-modal multi-LiDAR-inertial odometry and mapping framework with sliding window optimization for pose estimation. This is, to the best of our knowledge, the first SLAM algorithm to leverage the advantages of both spinning LiDARs and solid-state LiDARs within a single framework. Specifically, we have focused on demonstrating that high-robustness odometry and high-quality mapping are possible with an adequate combination of low-cost sensors. The proposed system effectively fuses the high situational awareness through dense point clouds in a solid-state LiDAR with the larger FoV from a spinning LiDAR. Despite the odometry accuracy being compared to other methods in certain environments, key properties of our method are consistency across environments and, specially, higher-quality maps where environment details can be appreciated.

In the next steps, we plan to explore further multi-modal LiDAR sensors providing image-like data and multi-IMU fusion within a single framework. We expect to focus further on online calibration in such more heterogeneous systems.

Acknowledgment

This research work is supported by the Academy of Finland's AeroPolis project (Grant 348480) and the Finnish Foundation for Technology Promotion (Grants 7817 and 8089).

References

[1] Q. Li, J. P. Queralta, T. N. Gia, Z. Zou, and T. Westerlund, "Multi-sensor fusion for navigation and mapping in autonomous vehicles: Accurate localization in urban environments," *Unmanned Systems*, vol. 8, no. 03, 2020.

[2] N. Varney, V. K. Asari, and Q. Graehling, "Dales: a large-scale aerial lidar data set for semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020.

[3] J. Yang, Z. Kang, S. Cheng, Z. Yang, and P. H. Akwensi, "An individual tree segmentation method based on watershed algorithm and three-dimensional spatial distribution analysis from airborne lidar point clouds," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, 2020.

[4] Q. Li, P. Nevalainen, J. Peña Queralta, J. Heikkonen, and T. Westerlund, "Localization in unstructured environments: Towards autonomous robots in forests with delaunay triangulation," *Remote Sensing*, vol. 12, 2020.

[5] H. Ye, Y. Chen, and M. Liu, "Tightly coupled 3d lidar inertial odometry and mapping," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019.

[6] Y. Xu, J. Lin, J. Shi, G. Zhang, X. Wang, and H. Li, "Robust self-supervised lidar odometry via representative structure discovery and 3d inherent error modeling," *IEEE Robotics and Automation Letters*, 2022.

[7] A. Tampuu, R. Aidla, J. A. van Gent, and T. Matiisen, "Lidar-as-camera for end-to-end driving," *arXiv preprint arXiv:2206.15170*, 2022.

[8] Y. Xianjia, S. Salimpour, J. P. Queralta, and T. Westerlund, "Analyzing general-purpose deep-learning detection and segmentation models with images from a lidar as a camera sensor," *arXiv preprint arXiv:2203.04064*, 2022.

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

[10] K. Li, M. Li, and U. D. Hanebeck, "Towards high-performance solid-state-lidar-inertial odometry and mapping," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, 2021.

[11] H. Sier, L. Qingqing, Y. Xianjia, J. P. Queralta, Z. Zou, and T. Westerlund, "A benchmark for multi-modal lidar slam with ground truth in gnss-denied environments," *arXiv preprint arXiv:2210.00812*, 2022.

[12] J. Jiao, H. Ye, Y. Zhu, and M. Liu, "Robust odometry and mapping for multi-lidar systems with online extrinsic calibration," *IEEE Transactions on Robotics*, 2021.

[13] P. Chen, W. Shi, S. Bao, M. Wang, W. Fan, and H. Xiang, "Low-drift odometry, mapping and ground segmentation using a backpack lidar system," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, 2021.

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

[15] L. Qingqing, Y. Xianjia, J. P. Queralta, and T. Westerlund, "Multi-modal lidar dataset for benchmarking general-purpose localization and mapping algorithms," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022.

[16] I. Vizzo, T. Guadagnino, B. Mersch, L. Wiesmann, J. Behley, and C. Stachniss, "Kiss-icp: In defense of point-to-point icp simple, accurate, and robust registration if done the right way," *IEEE Robotics and Automation Letters*, 2023.

[17] A. Segal, D. Haehnel, and S. Thrun, "Generalized-icp." in *Robotics: science and systems*, 2009.

[18] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time." in *Robotics: Science and Systems*, vol. 2, no. 9, 2014.

[19] L. Zhou, D. Koppel, and M. Kaess, "Lidar slam with plane adjustment for indoor environment," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, 2021.

[20] W. Xu and F. Zhang, "Fast-lio: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, 2021.

[21] J. Lin, X. Liu, and F. Zhang, "A decentralized framework for simultaneous calibration, localization and mapping with multiple lidars," in *IEEE/RSJ IROS*. IEEE, 2020.

[22] 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 *2020 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2020.

[23] H. Ye, Y. Chen, and M. Liu, "Tightly coupled 3d lidar inertial odometry and mapping," in *2019 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2019.

[24] T. Shan, B. Englot, C. Ratti, and D. Rus, "Lvi-sam: Tightly-coupled lidar-visual-inertial odometry via smoothing and mapping," in *2021 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2021.

[25] T.-M. Nguyen, S. Yuan, M. Cao, L. Yang, T. H. Nguyen, and L. Xie, "Miliom: Tightly coupled multi-input lidar-inertia odometry and mapping," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, 2021.

[26] S. Agarwal and K. Mierle. Ceres solver. [Online]. Available: http://ceres-solver.org

[27] F. Dellaert, "Factor graphs and gtsam: A hands-on introduction," Georgia Institute of Technology, Tech. Rep., 2012.

[28] M. Lixia, A. Benigni, A. Flammini, C. Muscas, F. Ponci, and A. Monti, "A software-only ptp synchronization for power system state estimation with pmus," *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 5, 2012.

[29] T. Shan and B. Englot, "Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018.