

# Low-Drift Odometry, Mapping and Ground Segmentation Using a Backpack LiDAR System

Pengxin Chen<sup>ID</sup>, Wenzhong Shi, Sheng Bao<sup>ID</sup>, Muyang Wang<sup>ID</sup>, Wenzheng Fan, and Haodong Xiang<sup>ID</sup>

**Abstract**—We propose a framework for odometry, mapping and ground segmentation using a backpack LiDAR system that achieves both real-time and low-drift performance. First, we present a spatio-temporal calibration method to carefully merge scans from the two laser scanners on a backpack. Second, we propose a feature extraction method which generalizes a point’s geometrical characteristics as two groups (*disjoint, continuous*) and three types (*edge, corner, plane*). The extracted features are used in point cloud registration in the odometry and mapping tasks. Third, a fast ground segmentation method is customized for the backpack LiDAR system. Finally, we evaluate the proposed method in four datasets logged by the backpack across different scales and environments. Furthermore, NTU VIRAL dataset is used to benchmark our method quantitatively. Experiments show that our method consistently outperforms the state-of-the-art methods before using loop closure optimization and sensor fusion techniques.

**Index Terms**—SLAM, mapping, backpack LiDAR.

## I. INTRODUCTION

LIGHT detection and ranging (LiDAR) is a technique to measure accurate distances between laser scanners and object surfaces. It is widely used in many applications such as navigation, surveying and mapping. There are different types of LiDAR systems, including Terrestrial Laser Scanning (TLS), Airborne Laser Scanning (ALS) and Mobile Laser Scanning (MLS) [1]. In the MLS group, various platforms of laser scanner are designed according to different passing conditions. Vehicle LiDARs can only operate on outdoor roads. Trolley LiDARs can be used indoors, but it is still difficult for them to pass through stairs. Handheld LiDARs can handle most indoor and outdoor scenarios, but the limited sensor and computing resources are their obvious bottleneck. While these LiDAR systems serve a large number of applications, none can be optimized for fast and flexible scanning in challenging locations [2]. Then Backpack LiDAR System (BLS) emerged to fill such a gap. Typically, the BLS is intended to gather 3D information in areas impassable by

Manuscript received February 23, 2021; accepted June 28, 2021. Date of publication July 14, 2021; date of current version July 29, 2021. This work was supported by The Hong Kong Polytechnic University under Grants 1-ZVN6 and 4-BCF7, in part by The State Bureau of Surveying and Mapping, P.R. China under Grant 1-ZVE8, and in part by Hong Kong Research Grants Council under Grant T22-505/19-N. This letter was recommended for publication by Associate Editor J. Civera and M. Walter upon evaluation of the reviewers’ comments. M. Walter (*Corresponding author: Wenzhong Shi*)

The authors are with the Department of Land Surveying and Geo-informatics, and the Smart Cities Research Institute, The Hong Kong Polytechnic University, Hong Kong, China (e-mail: pengxin.chen@connect.polyu.hk; john.wz.shi@polyu.edu.hk; sheng.bao@connect.polyu.hk; muyang.wang@connect.polyu.hk; vincent.wz.fan@connect.polyu.hk; haodong.hd.xiang@connect.polyu.hk).

Digital Object Identifier 10.1109/LRA.2021.3097060

vehicles, especially those GNSS-denied indoor and underground environments.

Although BLS brings flexibility, convenience and agility without compromising sensor and computing resources, it is challenging to execute Simultaneous Localization And Mapping (SLAM) on it. Unlike vehicle LiDAR systems which move smoothly, BLS will undergo fast and irregular mechanical jitter when the user’s feet hit the ground. This would cause aggressive pose changes in a very short time. Meanwhile, the ground plane shakes in the laser scanner’s coordinate system, thus increasing the difficulty of ground segmentation task.

Here we address three tasks of point cloud processing: odometry, mapping and ground segmentation. The odometry task is the use of LiDAR data to estimate the rigid motion of a BLS. A trajectory comprising 6-DOF poses will be obtained from the odometry. The mapping task registers coming laser scans into a global map in a fixed coordinate system, whereby reconstructing the 3D map of surrounding environments. Finally, the goal of ground segmentation is to separate a point cloud into two subsets representing ground plane and others respectively. In this paper, we propose a framework for LiDAR odometry, mapping and ground segmentation using a BLS. The proposed method achieves both real-time and low-drift performance without using any sensor fusion techniques. By this method we make the following contributions:

- 1) We present a spatio-temporal calibration method that carefully merges scans from multiple laser scanners. First, the method adopts a plane-based scan matching approach to obtain the extrinsic parameters. Second, a “split-and-merge” strategy is used to handle the time difference between scans from different scanners.
- 2) We present an effective feature extraction method which generalizes a point’s geometrical characteristics as two groups (*disjoint, continuous*) and three types (*edge, corner, plane*). Then three types of key points are selected by analyzing the scatter matrix of a point’s neighborhoods. These key points are highly discriminative and can be directly used in ground segmentation and point cloud registration.
- 3) We present a ground segmentation method for a BLS. The extracted *plane* features near the user’s feet position are used as seed points for ground growing. The *corner* features, on the contrary, are an end sign of the ground growing. By this way the efficiency of the ground segmentation is improved. The runtime of the feature extraction and the ground segmentation altogether is less than 10

*ms* per merged scan ( $2 \times$  VLP-16) on a portable computer (Intel NUC10i7FNH), with the assistance of *OpenMP* library [3].

## II. RELATED WORK

BLS has a wide application prospect in mapping places that LiDAR-equipped vehicles or robots cannot access to. Corso *et al.* [4] developed a mobile mapping system equipped with five 2D range finders to recover 3D indoor information, which is regarded as one of the earliest backpack LiDAR prototypes. Polewski *et al.* [5] used a single-LiDAR BLS to collect point clouds in forested areas, and co-registered such point clouds with those from an unmanned ALS. Hence, the point clouds acquired from ground and air can be stitched and georeferenced. More recently, Su *et al.* [6] developed a dual-LiDAR BLS to measure tree heights and diameters at breast in forest inventory applications. Two laser scanners in [6] are installed orthogonally to gather points with complementary field of view (FOV): one scanner collect points horizontally while the other one emits laser beams vertically to the ground like a push broom. Furthermore, Zheng *et al.* [2] tilted one of the laser scanners in their dual-LiDAR BLS, so that the user's feet would not interfere with laser beams during walking.

When application scenarios are confined to small-scale and short-time mapping tasks, the operating LiDAR system could be further compressed as handheld form. Recent related work includes, for example, a lightweight handheld SLAM platform [7] equipped with a spinning 2D LiDAR, and a handheld LiDAR dataset [8] which incorporates a 64-beam laser scanner together with a camera, an onboard PC and batteries. It is worth mentioning that inertial, visual and georeferencing sensors are usually added to a BLS, which however is not the focus of this paper.

In terms of the LiDAR-based SLAM approaches, a classical LiDAR-based pose estimation algorithm is Iterative Closest Point (ICP) algorithm [9]. It formulates the point cloud registration problem as minimizing the point-to-point distance. But it remains challenging to apply ICP in real-world scan matching because there are no exact one-to-one correspondences between two scans [10]. Considering the sparsity of laser scans, scholars replaces ICP's object function with point-to-line [11], point-to-plane [12], point-to-projection [13], and plane-to-plane [14] metrics. Such substitutions take into account the point spacing and line spacing of scans, which directly improves the quality of correspondences and the accuracy of scan matching.

The Normal Distribution Transform (NDT) algorithm [15] and its variants are another popular type of point cloud registration methods. They subdivide 2D or 3D space into cell grids and assign a normal distribution to each cell. Since for each cell NDT only needs to compute the centroid and covariance of points as features, the biggest advantage of NDT-based methods is their speed. The limitation of NDT, however, lies in the cell size tradeoff. Specifically, a large cell will blur the cell features, thus reducing the registration accuracy, whereas a small cell will narrow the convergence basin as a cell only contributes to the score function for scan points within its bounds.

Over the past recent years, the LiDAR Odometry And Mapping (LOAM) [16] has been considered as the state-of-the-art method. It extracts edge and surface features from laser scans, and estimates ego-motion by jointly optimizing point-to-edge and point-to-surface metrics using the Levenberg-Marquardt method. It has two modules, namely odometry and mapping, running in parallel. The result of odometry module functions as the initial motion guess for the mapping module. T. Shan *et al.* proposed lightweight and ground-optimized LOAM in 2018 (LeGO-LOAM) [17], and further fused it with IMU preintegration and factor graph in 2020 (LIOSAM) [18]. LeGO-LOAM applies a ground plane extraction method [19] and an image-based cloud segmentation method [20] in its feature extraction step so as to leverage the presence of a ground plane to filter out noise. LIO-Mapping [21] and MILIOM [22] are tightly coupled inertial LiDAR odometry methods. M-LOAM [23] combines multiple LiDARs with online extrinsic calibration.

With the widespread application of deep learning, learning-based methods have achieved promising results on the issues of point cloud registration [24] and vehicle LiDAR odometry [25]. Unfortunately, it still remains challenging to solve backpack LiDAR odometry using deep learning methods. This is mainly due to the lack of publicly available BLS datasets with accurate 6-DOF ground truths of scan poses in large scale.

## III. METHODOLOGY

### A. System Overview

The method proposed in this paper is validated on a BLS equipped with two laser scanners. One scanner is mounted horizontally on the top and the other one is tilted  $60^\circ$ .

The design of our BLS is motivated by observations of the weaknesses of existing BLSs. Firstly, the single-LiDAR backpack with only 16 beams in [5] can hardly gather points from ground in narrow environments, such as corridors. Secondly, the dual-LiDAR backpack in [6] can provide complementary FOV, but due to the orthogonal fashion of scanners' placement, the rear scanner can frequently receive laser bounces from user's hind leg in walks. Thirdly, the tilted rear scanner in [2] can prevent point clouds being contaminated by user's feet, but both scanners on their backpack are placed too close to user's head. Thus, another issue is raised: two scanners will interfere with each other and there is a risk of receiving laser bounces from user's head. To conclude, achieving an optimal arrangement of two scanners is a tradeoff between more FOV and less interference. This prompts us to design our BLS prototype as shown in Fig. 2. The green and blue areas are the FOVs of two scanners, respectively. Such design can ensure largest stitched FOV with basically no laser interference among user's head, feet and two scanners themselves (i.e. these objects will never appear in the green or blue areas). Lastly, the prototype is made of carbon fiber, and a cant beam is placed under scanners to ensure structure's rigidity.

The BLS prototype and overall framework of the BLS is shown in Fig. 1. The system receives point clouds from two laser scanners as input. The first step is the spatio-temporal calibration which aims to merge two point clouds into one. In the feature

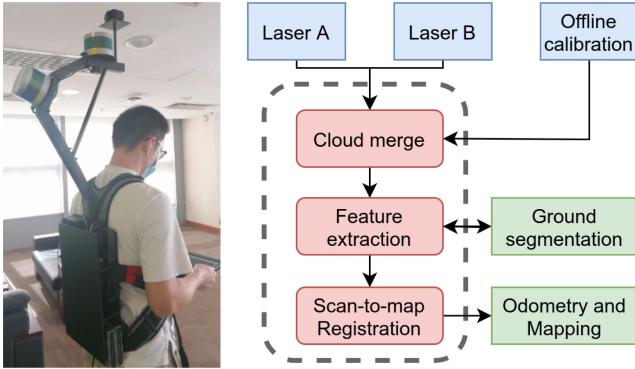


Fig. 1. Backpack LiDAR system overview.

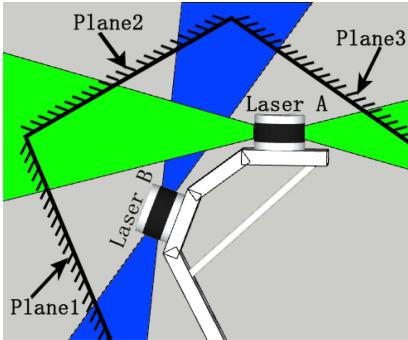


Fig. 2. Extrinsic parameter calibration.

extraction module, three types of features, namely the “*edge*, *corner* and *plane*” key points, are extracted from the merged cloud. These key points are then fed into the ground segmentation and the scan-to-map registration modules. Finally, the system outputs a LiDAR map containing ground information. The details of these modules are introduced below.

### B. Spatio-Temporal Calibration

To obtain the extrinsic parameters between the two laser scanners, we set up a plane-based calibration method as shown in Fig. 2. Three planes are placed around these two scanners, ensuring that each plane can be partly observed by both scanners. Let  $\mathcal{P}_A$  and  $\mathcal{P}_B$  be the point clouds acquired from Laser A and B. We thus can get three pairs of coplanar equations from  $\mathcal{P}_A$  and  $\mathcal{P}_B$ . Thereafter, we change the positions of planes to get more pair of equations. Finally, solving a plane-based registration problem [26] will provide us the transformation matrix  $T_{ex}$  between Laser A and B.

The transformation matrix  $T_{ex}$  has provided the spatial constraints between the two scanners, but we still lack the temporal information to merge  $\mathcal{P}_A$  and  $\mathcal{P}_B$ . Different starting timestamps of  $\mathcal{P}_A$  and  $\mathcal{P}_B$ , denoted by  $t_{an}$  and  $t_{bn}$ , necessitate the temporal calibration. As shown in Fig. 3, we adopt a “split-and-merge” strategy for the spatio-temporal calibration. That is, each  $\mathcal{P}_B$  is first split according to the starting timestamps of the closest  $\mathcal{P}_A$  in the point sequence. Thereafter, the split points of  $\mathcal{P}_B$  are transformed by  $T_{ex}$ , and merged into  $\mathcal{P}_A$  that shares the same

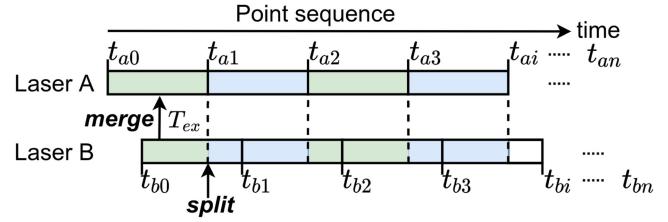


Fig. 3. Spatio-temporal calibration: split-and-merge; dotted lines are extended from  $t_{ai}$ , referring to the timestamp for splitting  $\mathcal{P}_B$ . points between two dotted lines will be merged; green and blue areas represent adjacent merged scan.

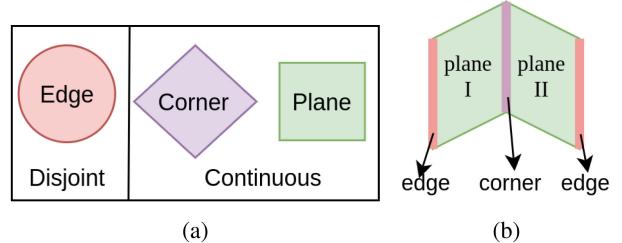


Fig. 4. Point labeling and feature types. (a) relationship chart of labels; (b) distinction between *edge* and *corner*.

time domain (i.e. in the same color). The individual timestamp of each point can be obtained from Velodyne ROS driver.<sup>1</sup> Note that the extrinsic parameter  $T_{ex}$  is obtained offline, but the “split-and-merge” is processed online.

### C. Feature Extraction

The feature extraction module starts with labeling each point in the merged cloud. Five labels are used to describe a point’s geometrical characteristics (see Fig. 4a). Let  $\mathcal{P}_t = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n\}$  be the merged cloud acquired at time  $t$ , where  $\mathbf{p}_i$  is a point in  $\mathcal{P}_t$ . Points are first determined to be either *disjoint* or *continuous* according to whether a leap of point spacing exists. Formally, the ratio of distances between adjacent points in the same scan line, denoted by  $r_{ps}^i$ , is defined as (1). A point  $\mathbf{p}_i$  will be labeled as *disjoint* if  $r_{ps}^i > \tau_{ps}$  and *continuous* otherwise, where  $\tau_{ps}$  is a given parameter. Note that  $r_{ps}^i$  should be set as its reciprocal if  $r_{ps}^i < 1$ .

$$r_{ps}^i = \frac{\|\mathbf{p}_{i+1} - \mathbf{p}_i\|_2}{\|\mathbf{p}_i - \mathbf{p}_{i-1}\|_2} \not\propto 1 \quad (1)$$

In the *disjoint* group, some points are close to each other. We regard them as unstable points arising from occlusion or outliers. After excluding them, the rest in this group are further classified into the *edge* points. In the *continuous* group on the other side, the eigenvalue ratio, denoted by  $r_{eg}^i$ , is defined to extract *corner* and *plane* features. the derivation of  $r_{eg}^i$  is as follows:

<sup>1</sup><https://github.com/ros-drivers/velodyne>

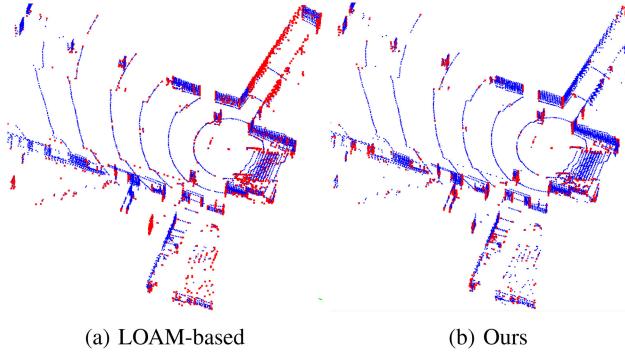


Fig. 5. Feature points; blue for *plane* features, red for *edge|corner* features. Note that feature points are not downsampled for better visualization, but they will be downsampled by a 0.2m voxel filter in registration.

- 1) For a point  $\mathbf{p}_i$ , select its neighboring points  $\mathcal{C} = \{\mathbf{p}_i, \mathbf{p}_{i\pm 1}, \mathbf{p}_{i\pm 2}, \dots, \mathbf{p}_{i\pm n}\}$  in the same scan line and normalize the point spacing:

$$\tilde{\mathbf{p}}_{i\pm j} = \mathbf{p}_{i\pm(j-1)} + \frac{\mathbf{p}_{i\pm j} - \mathbf{p}_{i\pm(j-1)}}{\|\mathbf{p}_{i\pm j} - \mathbf{p}_{i\pm(j-1)}\|_2}, j \in \mathbb{Z}^+ \quad (2)$$

- 2) Compute the scatter matrix  $\Sigma$  for  $\mathbf{p}_i$ :

$$\Sigma = \frac{1}{2n+1} \sum_{\tilde{\mathbf{p}} \in \mathcal{C}} (\tilde{\mathbf{p}} - \mu)(\tilde{\mathbf{p}} - \mu)^T, \mu = \frac{1}{2n+1} \sum_{\tilde{\mathbf{p}} \in \mathcal{C}} \tilde{\mathbf{p}} \quad (3)$$

- 3) Compute the eigenvalue ratio of  $\Sigma$ :

$$r_{eg}^i = \lambda_2 / \lambda_1 \quad (4)$$

where  $\lambda_1$  and  $\lambda_2$  are the first two largest eigenvalues of  $\Sigma$ . The range of  $r_{eg}^i$  is  $[0, 1]$ . Then, a certain number of points with large  $r_{eg}^i$  values are selected as *corner* features, and meanwhile, points with small values as *plane* features.

- 4) Non-maximum Suppression (NMS) with respect to  $r_{eg}^i$  for the *corner* feature points. This aims to retain only one point per scan line at each corner place.

Fig. 4(b) draws a distinction between *edge* and *corner* intuitively by displaying two intersecting planes: the line of intersection is labeled as *corner* instead of *edge*. Fig. 5 gives a comparison of feature extraction between LOAM-based methods [16][17] [18] and ours. It is worth noting that the blue areas are often corrupted by the red in Fig. 5(a), but in contrast Fig. 5(b) shows more discriminative features.

#### D. Ground Segmentation

The purposes of ground segmentation are twofold: i) reject non-planar feature points on the ground to facilitate scan matching; ii) be the pre-processing step for downstream applications such as dynamic objects detection and semantic labeling.

Fig. 6 shows the schematic diagram of ground segmentation for the proposed backpack system. We take advantage of the backpack structure and the extracted features to segment ground LiDAR points efficiently. Two knacks contribute to the efficiency. First, as the LiDAR coordinate origin is at the center of Laser A, namely the red point  $\mathbf{p}_a$  in Fig. 6, then the feet

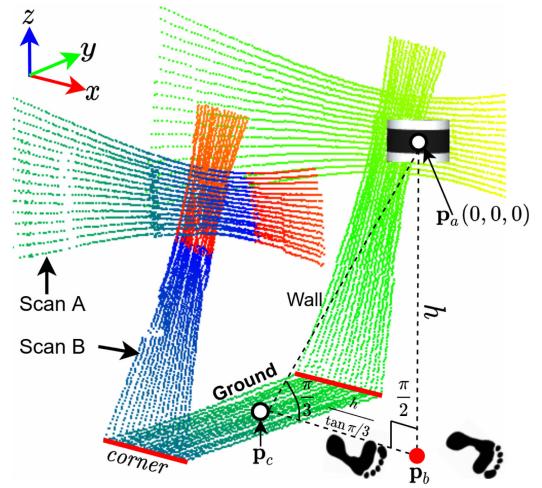


Fig. 6. Ground segmentation for a backpack LiDAR system; the scans displayed are acquired in a narrow corridor; here we only keep ground and side walls for better visualization.

---

#### Algorithm 1: Ground Segmentation.

---

```

Input: scans:  $\mathcal{P}_A$  and  $\mathcal{P}_B$  after calibration
Output: ground points set:  $\mathcal{G}$ 
Parameters: seed point  $\mathbf{p}_c$ , scalar  $\tau_g$ 
Start:  $\mathcal{G} \leftarrow \emptyset$ 
for each scan line  $\mathcal{L} = \{\mathbf{p}^{0-n}\}$  in  $\mathcal{P}_B$  do
    Extract corner set  $\mathcal{F}_C$  and plane set  $\mathcal{F}_P$ .
    Find the nearest point  $\{\mathbf{p}^i\}$  of  $\mathbf{p}_c$  in  $\mathcal{F}_P$ .
    for t=1:n do
        if  $\mathbf{p}^{i\pm t} \in \mathcal{F}_C$  or  $|\mathbf{p}^{i\pm t}[z] - \mathbf{p}_c[z]| > \tau_g$  then break.
        Add  $\mathbf{p}^{i\pm t}$  to  $\mathcal{G}$ .
        Fit ground plane equation using  $\mathcal{G}$ :  $\mathbf{n} \cdot (\mathbf{x} - \boldsymbol{\mu}) = 0$ ;  $\mathbf{n}$ : unit normal vector;  $\boldsymbol{\mu}$ : a point on the ground.
        for  $\mathbf{p} \in \{\mathcal{P}_A, \mathcal{P}_B\}$  do
            if  $|(\mathbf{p} - \boldsymbol{\mu})^T \mathbf{n}| \leq \tau_g$  then
                Add  $\mathbf{p}$  to  $\mathcal{G}$ .

```

---

center ( $\mathbf{p}_b$ ) and the ground points center ( $\mathbf{p}_c$ ) will be roughly fixed accordingly, provided the height of human is known. For instance, if the person carrying the backpack is 1.8 meters tall, then the coordinate of point  $\mathbf{p}_c$  will be  $(-1, 0, -1.9)$ , assuming Laser A is 0.1 meters higher than head.  $\mathbf{p}_c$  is utilized as a “seed point” for fast searching ground points. Second, the searching process directly treats *plane* features as ground candidates and terminates when encountering *corner* features. Details are shown in Algorithm 1, where  $[\cdot]$  denotes the  $z$ -axis coordinate value of a point.

#### E. Scan-to-Map Registration

Given a feature scan  $\mathcal{S}$  and a feature map  $\mathcal{M}$ , the registration problem can be formulated as solving a non-linear least square problem:

$$\min_{\xi \in \mathbb{R}^6} f = \frac{1}{2} \sum_{i=1}^m \|\rho_i \cdot D(T(\mathbf{p}_i, \xi))\|^2 : \mathbb{R}^6 \longrightarrow \mathbb{R} \quad (5)$$

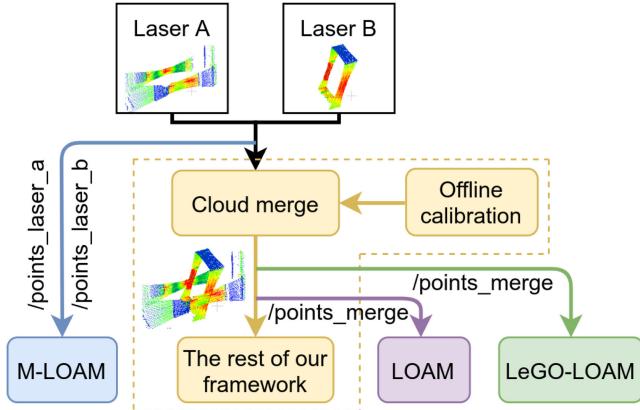


Fig. 7. Configuration of compared methods; the yellow modules belong to our framework; “/points\_laser\_a” and “/points\_laser\_b” are ROS topics from two laser scanners, respectively. “/pointsmerge” is output from our “Cloud merge” module.

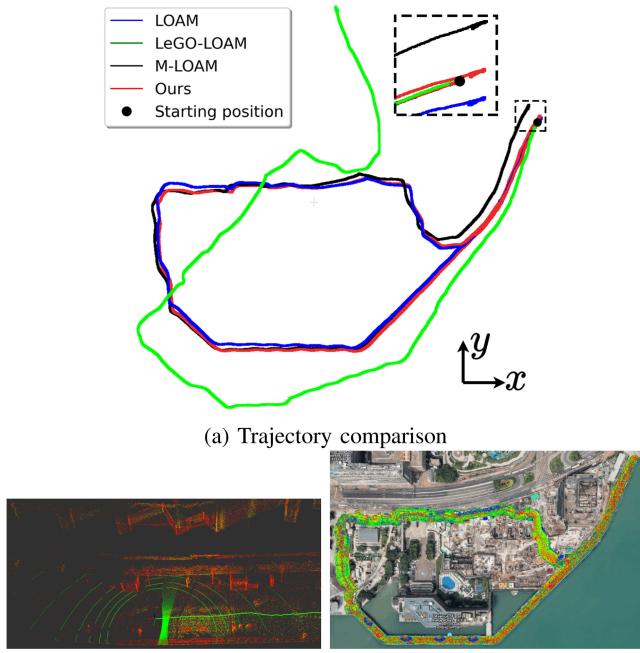


Fig. 8. Results of compared methods using the *Victoria Harbor* dataset.

where  $\xi = [t_x, t_y, t_z, \alpha, \beta, \gamma]^T$  denotes the 6-DOF rigid motion from  $\mathcal{S}$  to  $\mathcal{M}$ ;  $t_x, t_y$  and  $t_z$  are translations along the  $x$ -,  $y$ - and  $z$ -axes in a Cartesian coordinate system, respectively; and  $\alpha, \beta, \gamma$  are Euler angles;  $\mathbf{p}_i$  is a feature point in  $\mathcal{S}$ ; the functor  $T(\cdot)$  transforms  $\mathbf{p}_i$  to a new point  $\hat{\mathbf{p}}_i$  according to  $\xi$ .

For each  $\hat{\mathbf{p}}_i$ , its  $k$  nearest neighboring points of the same feature type in  $\mathcal{M}$  are extracted and denoted as  $\mathcal{N}_f$ . If  $\mathbf{p}_i$  is a *edge* or *corner* point,  $\mathcal{N}_f$  will be fitted as a line, and the cost function  $D(\cdot)$  will be defined as computing a point-to-line distance. In contrast, a plane equation will be fitted if  $\mathbf{p}_i$  is a *plane* point, and  $D(\cdot)$  will be a point-to-plane distance function accordingly. Unlike LOAM [16] which directly uses the distance function

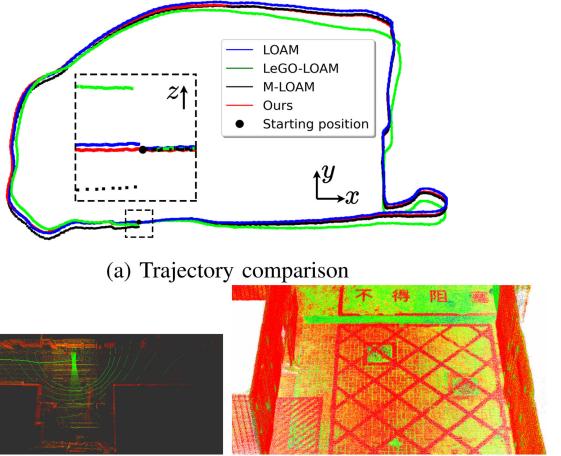


Fig. 9. Results of compared methods using the *Campus* dataset. Gaps among ground bricks are visible in (c).

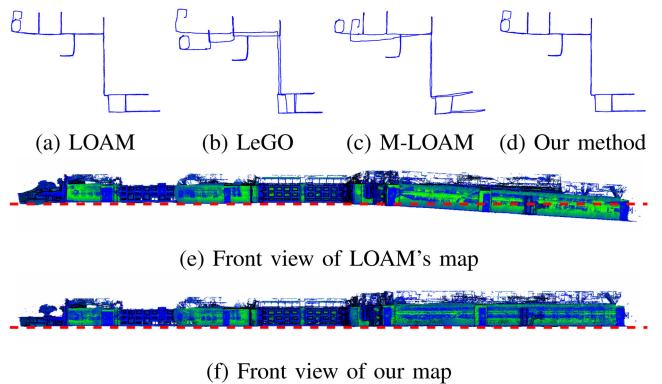


Fig. 10. Results of compared methods using the *Building floor* dataset. In (d) and (e), the red dash line represents the horizontal plane.

as residuals, we take into account the line/plane fitting quality to weight the residual block. The fitting quality  $\rho_i$  is evaluated by (6) where  $\lambda_1, \lambda_2$  and  $\lambda_3$  are the eigenvalues in descending order of  $\mathcal{N}_f$ 's scatter matrix.  $\lambda_f$  is  $\lambda_2$  in the line fitting case, and becomes  $\lambda_3$  in the plane fitting case. Intuitively,  $\rho_i$  measures the compression near a fitted line or plane, whereby a larger  $\rho_i$  means higher fitting quality and yields a heavier weighting residual unit.

$$\rho_i = \sqrt{(\lambda_1^2 - \lambda_f^2)/(\lambda_1^2)} \quad (6)$$

Finally, Gaussian-Newton method is used to solve (5). Thus, we can compute the motion increment  $\Delta\xi$  by the recurrence formula:

$$\Delta\xi = -(\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T \mathbf{r} \quad (7)$$

where  $\mathbf{J}$  and  $\mathbf{r}$  are the Jacobian matrix and the residual vector of  $\rho_i \cdot (D \circ T)(\xi)$ , respectively.

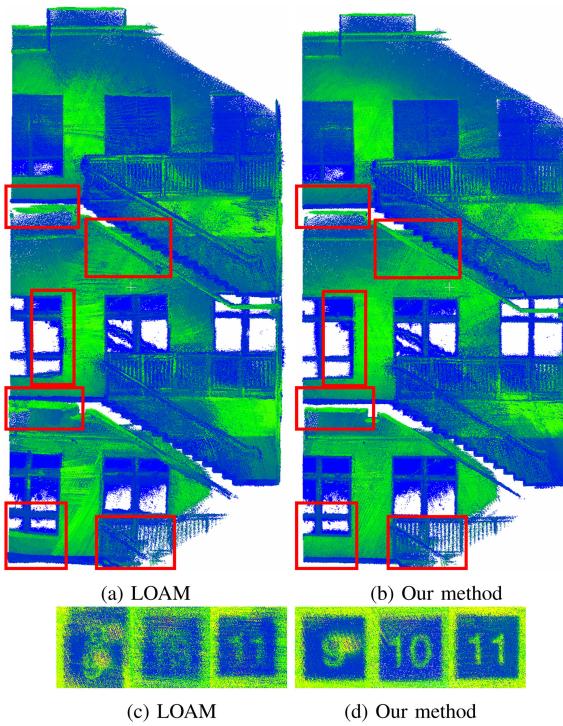


Fig. 11. Staircase test between LOAM and our method; (a) and (b) are an overview of mapping results; (c) and (d) are floor numbers on the wall extracted from both maps. colors are rendered by the intensity of points.

TABLE I  
DATASET DETAILS

Dataset	Scans number	Duration (s)	Elevation change (m)	Trajectory length (m)
Victoria harbor	47201	1190	3.7	1250
Campus	25082	632	0.8	588
Building floor	39334	994	0	513
Staircase	10119	255	8.9	54

## IV. EXPERIMENTS

### A. Qualitative Experiment

We use a Velodyne Puck for the horizontal laser scanner on the backpack, and a Puck Hi-Res for the tilted one. A portable computer (Intel NUC) is used for both data logging and processing. Four typical indoor and outdoor datasets are logged to evaluate the proposed framework. Details of the datasets are listed in Table I. In the experiments, we compare our method with LOAM [16], LeGO-LOAM [17] and M-LOAM [23]. No loop closure optimization will be used.

**Feature extraction configurations:** we set  $\tau_{ps} = 4$  to distinguish the *disjoint*|*continuous* groups, and accordingly the *edge* feature. Each scan line is subdivided into 12 parts uniformly. The number of neighborhoods  $n$  is set as 5 to compute scatter matrix  $\Sigma$ . For each part the point with the largest  $r_{eg}^i$  value is selected as a *corner* feature if meanwhile  $r_{eg}^i > 0.1$  holds, and the smallest  $r_{eg}^i$  value for a *plane* feature if  $r_{eg}^i < 0.01$  holds.

TABLE II  
END-TO-END TRANSLATION ERROR (METERS)

Dataset	LOAM	LeGO-LOAM	M-LOAM	Ours
Victoria harbor	8.5	251	18.8	3.0
Campus	0.35	3.29	2.06	0.04

**Submap configurations:** A submap is maintained during scan-to-map registration. For a coming scan with known timestamp, we first search for historical poses both within (indoor: 20, outdoor: 50) meters range spatially and within 10 seconds temporally. Each retrieved pose corresponds to a set of feature points which are then transformed to the *World* coordinate system and stacked together. These feature points are finally downsampled by a 0.2m voxel filter to compose the submap. In terms of motion priori, the pose of the last registered scan is set as the initial guess for a coming scan.

Since LOAM and LeGO-LOAM are designed for single-LiDAR odometry, we have to make some adjustments so that they can consume our dual-LiDAR data. Firstly, as shown in Fig. 7, we merge two 16-beam scans via our ‘‘Cloud merge’’ module to generate a 32-beam scan. By this way, our dual-LiDAR point clouds are disguised as a 32-beam single-LiDAR data which are then fed to and trick LOAM and LeGO-LOAM. Moreover, the original LOAM implementation recovers laser *ring* and point sequence information by calculating pitch and yaw angles, respectively. Instead, we modify LOAM’s code as directly retrieving such information from Velodyne ROS driver. Finally, as M-LOAM inherently starts with multiple extrinsic-uncalibrated LiDARs, we let it directly subscribe ROS<sup>2</sup> topics from both laser scanners.

1) *Victoria Harbor*: This dataset was logged at the Victoria Harbor in Hong Kong and intended to test the performance of our method in widely open areas. The user walked along the harbor at an average speed of 1.05 m/s and returned to the starting position again. This dataset is challenging because in many scans we can only obtain points on one side of the user (the other side is ocean), which occurs over half of the time during data logging. Fig. 8(a) compares the trajectories of four methods. Among them, LeGO-LOAM’s trajectory diverges quickly. Other three trajectories accumulate drifts as well when given a closer watch at the starting position. Since the ground-truth trajectories are not available, we instead compare the relative translational error when the backpack returns to the start (see Table II), which indicates that our method performs best in this test. Fig. 8(b) is an example of our ground segmentation results, and Fig. 8(c) shows that the map produced by our method is well-aligned with Google Earth.

2) *Campus*: This dataset covers continuous scenarios including walkway, sheltered passage, narrow alley and gentle slope. Again, we compare trajectories of four methods in Fig. 9 and Table II. Comparison shows that LOAM does not handle well the drift along  $z$ -axis, which might be due to the unbalanced contributions of the edge and surface features in LOAM’s registration

<sup>2</sup><https://www.ros.org>

TABLE III  
ATE REPORT OVER THE NTU VIRAL DATASETS. THE BEST RESULTS IN 1-LIDAR AND 2-LiDAR GROUPS ARE HIGHLIGHTED IN BOLD. NOTE THAT LIO-M, LIO-SAM AND MILIOM METHODS EMPLOY IMU MEASUREMENTS ADDITIONALLY WHILE M-LOAM AND OUR METHOD ONLY USE LiDAR DATA. THE RESULTS FOR LIO-M, LIO-SAM, M-LOAM, MILIOM ARE TAKEN FROM [22]

Dataset	1-LiDAR						2-LiDAR			
	LIO-M	LIO-SAM	MILIOM	Ablation study				M-LOAM	MILIOM	$\mathcal{PECW}$
				$\mathcal{P}$	$\mathcal{PE}$	$\mathcal{PEC}$	$\mathcal{PECW}$			
eee-01	1.0542	0.0915	0.1042	-	0.0853	0.0758	<b>0.0691</b>	0.1945	0.0666	<b>0.0615</b>
eee-02	0.7234	0.0815	0.0650	-	0.0688	0.0628	<b>0.0624</b>	0.2996	0.0656	<b>0.0608</b>
eee-03	1.0314	0.1176	<b>0.0628</b>	-	0.0967	0.0933	0.0927	0.1555	<b>0.0518</b>	0.0870
nya-01	2.2436	0.0899	0.0832	0.0560	0.0628	0.0555	<b>0.0536</b>	0.2334	0.0565	<b>0.0423</b>
nya-02	1.9664	0.1068	0.0721	0.1115	0.0775	0.0808	<b>0.0716</b>	0.2859	0.0668	<b>0.0663</b>
nya-03	2.9934	0.3655	<b>0.0577</b>	0.1045	0.0720	0.0677	0.0683	0.1925	<b>0.0423</b>	0.0601
sbs-01	1.6737	0.0966	0.0764	-	0.0766	0.0714	<b>0.0707</b>	0.1925	<b>0.0658</b>	0.0698
sbs-02	1.8056	0.0961	0.0806	-	0.0674	0.0675	<b>0.0661</b>	0.1778	0.0816	<b>0.0653</b>
sbs-03	2.0006	0.0960	0.0884	1.6818	0.0695	0.0686	<b>0.0678</b>	0.1863	0.0933	<b>0.0654</b>
Average	1.7214	0.1268	0.0767	-	0.0752	0.0715	<b>0.0691</b>	0.2131	0.0656	<b>0.0643</b>

$\mathcal{P}$ : Our method using only *plane* features. Residuals are not weighted.

$\mathcal{PE}$ : Our method using *plane, edge* features. Residuals are not weighted.

$\mathcal{PEC}$ : Our method using *plane, edge, corner* features. Residuals are not weighted.

$\mathcal{PECW}$ : Our method using *plane, edge, corner* features. Residuals are weighted by the  $\rho_i$  proposed in Section III-E.

process. By contrast, our method maintains good consistency of the ground plane. As Fig. 9(c) shows, the paint characters and ground bricks are clearly visible.

3) *Building Floor*: This dataset is representative of an office floor with corridor, lobby and terrace. In this test, corridors are only 1.3 meters wide, and thus the horizontal laser scanner on the backpack can hardly capture ground points. In such a case, the tilted laser scanner is essential to obtain the ground information, and accordingly prevents the odometry from drifting. Fig. 10(a)(b)(c)(d) shows the trajectories of four methods among which LeGO-LOAM and M-LOAM does not generate a correct trajectory in such a perspective. We further compare the maps of LOAM and our method in the front view, as shown in Fig. 10(e)(f). It apparently shows that there is a downward drift in LOAM’s map. We measure the maximum distance from trajectory points to the floor plane of four methods: 2.12(LOAM), 1.84(LeGO-LOAM), 35.8(M-LOAM) and 0.14(Ours) meters. Our method achieves the lowest offset ( $0.14m$ ) which is within the normal range of sensor’s up-down shaking during user walking.

4) *Staircase*: Finally, we evaluate the mapping quality of our method in a staircase. This dataset was logged from the 9<sup>th</sup> to the 11<sup>th</sup> floor at Building Z in Hong Kong PolyU. As LeGO-LOAM fails in this test and M-LOAM does not produce a meaningful result, we only compare the results of LOAM and our method. From Fig. 11(a) and 11(b), we can see that the map produced by our method has a cleaner skeleton, especially in the regions labeled by red boxes. A more obvious contrast between the two methods is the visibility of floor numbers (see 11(c) and 11(d)). It is difficult to recognize three numbers from LOAM’s map but is much easier from ours.

## B. Quantitative Experiment

We use NTU VIRAL dataset [27] to benchmark our algorithm. This publicly available<sup>3</sup> dataset provides trajectory groundtruth by laser-tracking a prism placed on a UAV. It also provides an

evaluation kit to measure the average translational error (ATE). Other than M-LOAM, we compare our method with three latest LiDAR-IMU fusion methods: LIO-M [21], LIO-SAM [18] and MILIOM [22]. In this experiment, our ground segmentation function is turned off. The comparison results on the ATE metric is report in Table III. We see that our method consistently outperforms these four methods, even though we only use LiDAR data.

### C. Ablation Studies

In order to investigate the importance of different feature types proposed in Section III-C and the residual weighting in Section III-E, we perform an ablation study on the NTU VIRAL benchmark by using different combinations of features and by switching on/off the residual weighting. Such an ablation study is carried out on the single-LiDAR setting since this is a more general case. As shown in Table III, our method achieves the best average performance when all feature types are used and residuals are weighted.

### D. Runtime Analysis

We run our algorithm on two Velodyne VLP-16 which can provide us with approximately 600 k points per second. A detailed runtime analysis is reported in Table IV. Typically, there are about 600 feature points in a scan and 20 k (indoor) or 50 k (outdoor) feature points in the submap. The Gaussian-Newton solver is fixed at 15 iterations. Scan lines are independent of each other w.r.t the feature extraction, as are feature points w.r.t the point-to-line/plane residuals computation. In light of this, *OpenMP* is used to make them run in parallel. Altogether, our algorithm operates faster than 20 Hz on a portable computer (Intel NUC, i7-10710U@1.10 GHz×12).

<sup>3</sup>[https://ntu-aris.github.io/ntu\\_viral\\_dataset/](https://ntu-aris.github.io/ntu_viral_dataset/)

TABLE IV  
RUNTIME ANALYSIS PER FRAME (UNIT: MS)

Module	Pipeline	Unit time	Module time	All
Feature extraction	merge clouds	0.2	9.2	38 //
	extract features	7		
	segment ground	2		
Scan-to-map registration	update submap	3	38	
	registration	35		

//: Two modules are running in parallel in ROS.

## V. CONCLUSION

This paper presented a framework for odometry, mapping and ground segmentation using a backpack LiDAR system that comprises four modules: spatio-temporal calibration, feature extraction, ground segmentation and scan-to-map registration. While the calibration method is presented with two laser scanners, it can be extended to calibrate more scanners. The extracted features are used to register point cloud and accelerate the ground segmentation task. Both qualitative and quantitative experiments are conducted to prove that the proposed framework achieves low-drift performance. With the assistance of *OpenMP* library for parallel computing, the runtime is faster than 20 Hz per frame, which well guarantees the real-time performance of odometry and mapping tasks. The future work will include the fusion with IMU and other sensors. With an IMU assisting, a motion prior can de-skew point clouds and accelerate the optimization process in scan matching.

## REFERENCES

- [1] C. Toth and G. Jkw, "Remote sensing platforms and sensors: A survey," *ISPRS J. Photogrammetry Remote Sens.*, vol. 115, pp. 22–36, 2016.
- [2] Z. Gong, J. Li, Z. Luo, C. Wen, C. Wang, and J. Zelek, "Mapping and semantic modeling of underground parking lots using a backpack lidar system," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 2, pp. 734–746, Feb. 2021.
- [3] L. Dagum and R. Menon, "Openmp: An industry standard api for shared-memory programming," *IEEE Comput. Sci. Eng.*, vol. 5, no. 1, pp. 46–55, Jan.–Mar. 1998.
- [4] N. Corso and A. Zakhori, "Indoor localization algorithms for an ambulatory human operated 3 d mobile mapping system," *Remote Sens.*, vol. 5, no. 12, pp. 6611–6646, 2013.
- [5] P. Polewski, W. Yao, L. Cao, and S. Gao, "Marker-free coregistration of uav and backpack lidar point clouds in forested areas," *ISPRS J. Photogrammetry Remote Sens.*, vol. 147, pp. 307–318, 2019.
- [6] Y. Su, Q. Guo, S. Jin, H. Guan, X. Sun, Q. Ma, T. Hu, R. Wang, Y. Li, "The development and evaluation of a backpack lidar system for accurate and efficient forest inventory," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 7, pp. 1–5, 2020.
- [7] T. Lowe, S. Kim, and M. Cox, "Complementary perception for handheld slam," *IEEE Robot. Automat. Lett.*, vol. 3, no. 2, pp. 1104–1111, Apr. 2018.
- [8] M. Ramezani, Y. Wang, M. Camurri, D. Wisth, M. Mattamala, and M. Fallon, "The newer college dataset: Handheld lidar," *Inertial Vis. Ground Truth*, 2020, *arXiv:2003.05691*.
- [9] P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, Feb. 1992.
- [10] Z. Dong *et al.*, "Registration of large-scale terrestrial laser scanner point clouds: A review and benchmark," *ISPRS J. Photogrammetry Remote Sens.*, vol. 163, pp. 327–342, 2020.
- [11] A. Censi, "An icp variant using a point-to-line metric," in Proc. *IEEE Int. Conf. Robot. Automat.*, 2008, pp. 19–25.
- [12] Y. Chen and G. Medioni, "Object modeling by registration of multiple range images," in Proc. *IEEE Int. Conf. Robot. Automat.*, 1991, pp. 2724–2729.
- [13] R. J. Campbell and P. J. Flynn, "A survey of free-form object representation and recognition techniques," *Comput. Vis. Image Understanding*, vol. 81, no. 2, pp. 166–210, 2001.
- [14] A. Segal, D. Haehnel, and S. Thrun, "Generalized-icp," *Robotics: Sci. Syst.*, Seattle, WA, vol. 2, no. 4, pp. 435–442, 2009.
- [15] M. Magnusson, "The three-dimensional normal-distributions transform: an efficient representation for registration, surface analysis, and loop detection," Ph.D. dissertation, Dept. Tech, Orebro Univ., Orebro, Sweden, 2009.
- [16] J. Zhang and S. Singh, "Low-drift and real-time lidar odometry and mapping," *Auton. Robots*, vol. 41, no. 2, pp. 401–416, 2017.
- [17] T. Shan and B. Englot, "Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," in Proc. *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 4758–4765.
- [18] T. Shan *et al.*, "Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping," in Proc. *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 5135–5142.
- [19] M. Himmelsbach, F. V. Hundelhausen, and H-J. Wuensche, "Fast segmentation of 3D point clouds for ground vehicles," in Proc. *IEEE Intell. Veh. Symp.*, 2010, pp. 560–565.
- [20] I. Bogoslavskyi and C. Stachniss, "Fast range image-based segmentation of sparse 3D laser scans for online operation," in Proc. *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 163–169.
- [21] H. Ye, Y. Chen, and M. Liu, "Tightly coupled 3 d lidar inertial odometry and mapping," in Proc. *IEEE Int. Conf. Robot. Automat.*, 2019, pp. 3144–3150.
- [22] T.-M. Nguyen, S. Yuan, M. Cao, Y. Lyu, T. H. Nguyen, and L. Xie, "MIL-IOM: Tightly coupled multi-input lidar-inertia odometry and mapping," 2021, *arXiv:2104.11888*.
- [23] J. Jiao, H. Ye, Y. Zhu, and M. Liu, "Robust odometry and mapping for multi-lidar systems with online extrinsic calibration," 2020, *arXiv:2010.14294*.
- [24] W. Lu, G. Wan, Y. Zhou, X. Fu, P. Yuan, and S. Song, "Deepvcp: An end-to-end deep neural network for point cloud registration," in Proc. *IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 12–21.
- [25] Q. Li *et al.*, "Lo-net: Deep real-time lidar odometry," in Proc. *IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8473–8482.
- [26] W. Fan, W. Shi, H. Xiang, and K. Ding, "A novel method for plane extraction from low-resolution inhomogeneous point clouds and its application to a customized low-cost mobile mappi," *Remote Sens.*, vol. 11, no. 23, pp. 2789–2810, 2019.
- [27] T.-M. Nguyen, S. Yuan, M. Cao, Y. Lyu, T. H. Nguyen, and L. Xie, "Ntu viral: A visual-inertial-ranging-lidar dataset, from an aerial vehicle viewpoint," Submitted to IJRR, 2021. [Online]. Available: <https://doi.org/10.21979/N9/X39LEK>.