A Novel 3D LiDAR SLAM Based on Directed Geometry Point and Sparse Frame

Shuang Liang, Zhiqiang Cao, *Senior Member, IEEE*, Chengpeng Wang, and Junzhi Yu, *Senior Member, IEEE*

***Abstract*—Simultaneous localization and mapping is an indispensable yet challenging direction for mobile robots. Attracted by 3D LiDAR with accurate depth information and robustness to illumination variations, many 3D LiDAR SLAM methods based on scan-to-map matching have been developed. However, there is a critical issue of existing approaches, where a large and dense map is generally required to achieve satisfactory localization accuracy, leading to low efficiency of scan-to-map matching. To solve this problem, in this letter, we propose a novel 3D LiDAR SLAM based on directed geometry point (DGP) and sparse frame. The former is used to provide a sparse distribution of points in the spatial dimension and the latter gives rise to a sparse distribution of frames in the temporal sequence. The sparsity of points and frames impove the efficiency of 3D LiDAR SLAM, and the strict data association based on directed geometric points also brings in good accuracy of pose estimation. To compensate the accuracy loss of the localization and mapping caused by frame sparsity, point propagation is proposed to improve the quality of directed geometric points in the map and the accuracy of scan-to-map matching. Also, loop detection and pose graph optimization are conducted for global consistency. The experimental results demonstrate the effectiveness of the proposed method in terms of accuracy and efficiency.**

***Index Terms*—3D LiDAR SLAM, directed geometric point, sparse frame, point propagation.**

I. Introduction

LOCALIZATION is an essential prerequisite for autonomous mobile robots to execute tasks in unknown environments [1]–[3], where simultaneous localization and mapping (SLAM) is one of the mainstream solutions. SLAM has been explored using a variety of sensors in recent decades, which include monocular camera, stereo camera, RGBD camera, 2D LiDAR, and 3D LiDAR, etc. From the viewpoint of better navigation, RGBD camera and 3D LiDAR are preferable because they can output rich point cloud. The former can provide relatively dense point cloud with color information, while the latter offers the point cloud with more accurate depth information and broader view. Compared with the former, the latter is robust to illumination variations and textures. In this paper, we focus on 3D LiDAR SLAM.

Scan matching is a general way to solve LiDAR SLAM, which calculates relative pose transformation between source data and target data via various registration methods based on classical ICP [4], [5] and NDT [6], etc. Typically, the source and target data are composed of points, where geometric point [7], [8] and directed point [9], [10] are two common types in 3D LiDAR SLAM.

Specifically, the geometric point refers to a point with significant geometric feature, which eliminates other points and increases the efficiency of scan matching. However, the direction of each geometric point typically needs to be recalculated in scan matching, such as point-to-plane ICP. This leads to a waste of time. Moreover, a dense geometric point set has to be maintained to achieve accurate direction estimation, and the processing of dense point set is also time consuming. For the directed point, it is a point with direction represented by normal vector of the surface where the point is located. The time of recalculating direction in scan matching is saved, however, the directed point-based methods are time-consuming due to the fact that all points are involved in the scan matching. Geometric point and directed point have their benefits and weaknesses. If they can be combined, a satisfactory performance is expected. In this paper, directed geometric point (DGP) is introduced which is defined as a geometric point with direction. On one hand, the direction recalculation of geometric point is no longer required in scan matching and thus the corresponding dense point set can be replaced by a sparse point set. On the other hand, only directed points with significant geometric features are kept, which reduces the computational cost and is suitable for CPU-only machines.

The target data of scan matching is commonly composed of recent or all historical point clouds, which is called map. The corresponding scan matching is called scan-to-map matching. However, a disadvantage of typical scan-to-map matching is that the point number in the map is too much so that the efficiency is low. Actually, it is not necessary to keep all recent frames because point clouds in neighboring frames are similar. An idea is to only utilize those frames sparsely distributed in space, which are termed as fusion frames. This solution leads to a small-size map and the efficiency of scan-to-map matching can be improved. However, a problem arises where the frame sparsity affects the localization and mapping precision. To compensate the loss of precision, the concept of point propagation is introduced to improve the quality of points.

Manuscript received July 28, 2020; accepted November 17, 2020. Date of publication December 8, 2020; date of current version December 22, 2020. This letter was recommended for publication by Editor Sven Behnke upon evaluation of the Associate Editor and reviewers' comments. This work was supported in part by the National Natural Science Foundation of China under Grants 62073322, 61633020, 61633017, 61836015. *(Corresponding author: Zhiqiang Cao.)*

Shuang Liang, Zhiqiang Cao, and Chengpeng Wang are with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, CAS, Beijing 100190, China, and also with School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: liangshuang2017@ia.ac.cn; zhiqiang.cao@ia.ac.cn; wangchengpeng2019@ia.ac.cn).

Junzhi Yu is with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, CAS, Beijing 100190, China, and also with State Key Laboratory for Turbulence and Complex System, Department of Mechanics and Engineering Science, BIC-ESAT, College of Engineering, Peking University, Beijing 100871, China (e-mail: junzhi.yu@ia.ac.cn).

Digital Object Identifier 10.1109/LRA.2020.3043200

The main contributions of this paper are as follows. Firstly, a new point feature termed directed geometric point is proposed for LiDAR SLAM. It significantly decreases the point number in the map, which thus increases the efficiency of scan-to-map matching. Furthermore, the strict data association based on directed geometric point guarantees the accuracy of pose estimation. Secondly, the sparse distribution of fusion frames in the map further improves the efficiency of scan-to-map matching and point propagation is also designed to improve the quality of points and compensate the loss of accuracy caused by frame sparsity. The paper is organized as follows. Section II presents related work. Section III gives the problem statement. In Section IV, the proposed LiDAR SLAM is described in detail. The experimental evaluations are shown in Section V and Section VI concludes the paper.

II. Related Work

Scan matching is a general solution to LiDAR SLAM by aligning source and target data via various registration methods. A typical registration method is iterative closest point (ICP) [4], [5], which calculates pose transformation by minimizing the alignment error between source and target data. Pomerleau *et al*. [11] provided a comparison between ICP variants and an open-source ICP library. Ding *et al*. [12] employed this ICP library to achieve multi-session map construction in large scale outdoor environment using 3D LiDAR. Ramezani *et al*. [13] presented a LiDAR SLAM method for legged robots with Autotuned ICP. In ICP, there are two common alignment error metrics: point-to-point distance [4] and point-to-plane distance [5]. Compared to the former, the latter is more suitable for sparse point cloud because there may not exist a strict point-point correspondence between two sparse point clouds. Behley *et al*. [9] proposed a LiDAR SLAM method called SuMa with projection data association and point-to-plane error metric. Chen *et al*. [14] improved the robustness to dynamic environment of SuMa by introducing semantic information.

The efficiency of ICP is positively correlated with the size of source and target data. To achieve real-time performance, feature extraction strategies are proposed. The most common feature is point feature [7]–[10]. Zhang and Singh [7] extracted two kinds of geometric point features: edge point and planar point. Afterwards, Shan and Englot [8] improved the geometric point extraction method via ground plane segmentation and clustering. Behley and Stachniss [9] employed directed point feature augmented by radius, timestamps, and stability in the point-to-plane ICP. Serafin and Grisetti [10] presented NICP with directed point feature, where the difference of normal vectors is considered for robust data association. Recently, line and plane features are also employed in the LiDAR SLAM [15], [16]. Schaefer *et al*. [15] used pole landmark as a type of line feature for 3D LiDAR localization and mapping. Grant *et al*. [16] adopted point and plane features simultaneously in LiDAR SLAM to improve the accuracy and efficiency of scan matching. However, the aforementioned large-scale line and plane features are relatively rare in some unstructured environments, which limits the applications of these methods. For general purpose, a directed geometric point feature is presented in this paper by taking the advantages of both geometric and directed point features.

In terms of composition of target data, the scan matching can be also categorized into scan-to-scan and scan-to-map matching. For the former, the target data comes from a single frame [6], which is fast but inaccurate. The latter combines points from multiple frames to construct the target data [7], [9] and achieves higher accuracy, which is preferred by the mainstream LiDAR SLAM methods. The scan-to-map matching is further divided into scan-to-global map matching [17], [18] and scan-to-local map [9], [19], [20] matching. Ji *et al*. [17] represented the global map with the combination of grids and point features and optimized the pose transformation by expectation-maximization (EM) algorithm. Weingarten and Siegwart [18] constructed global map with planar segments and tracked the pose of robot by extended Kalman filter. Behley and Stachniss [9] built the local map with directed point features in recent frames for scan-to-local map matching. Similarly, Deschaud [19] obtained the local map with sampling geometric features for scan matching. Konolige *et al*. [20] generated the local map combining topological graph and metric map for efficient navigation. Besides, the combination of scan-to-scan and scan-to-map matching is also considered to balance accuracy and efficiency [7], [8], [21]. Zhang and Singh [7] presented LOAM, which first combined the scan-to-scan and scan-to-global map matching for real-time performance. Afterwards, Qin and Cao [21] simplified the feature extraction process and improved the efficiency of scan-to-map matching of LOAM. Similarly, Shan and Englot [8] combined the scan-to-scan and scan-to-local map matching with loop closure to improve global consistency of pose estimation. However, the low efficiency of scan-to-map matching hampers the performance of LiDAR SLAM. To solve this problem, we propose a novel efficient scan-to-map matching approach with directed geometric point, sparse fusion frame and point propagation techniques, which can be executed in real time without the help of scan-to-scan matching.

III. Symbol Definition and Problem Statement

In this section, we present the formal problem statement with definitions of concepts and symbols. We label the LiDAR frame at timestamp $t(t = 0, 1, 2, \cdots)$ and the world frame as C_t and W, respectively, where W coincides with C_0. Considering the redundancy of point clouds between nearby frames, we only focus on the sparse fusion frames. Further, keyframe is defined as the special fusion frame where the adjacent keyframes are separated by ω fusion frames. Mathematically, the keyframe is denoted as $K_x(x = 0, 1, 2, \cdots)$, and the fusion frames between K_x and K_{x+1} are described by $K_x^y(y = 0, 1, \ldots, \omega - 1)$. Specially, $K_x = K_x^0$ and K_0 is in coincidence with C_0. Fig. 1 illustrates different frames, where K_α^0 and K_α^β are latest keyframe and fusion frame before the new frame C_t, respectively.

With different types of frames, the definitions of directed geometric point (DGP) sets are given. Due to the distinct geometric features of edge and planar points, we focus on directed edge point and directed planar point in this paper. Formally, directed edge points and directed planar points extracted from the frame C_t constitute the directed edge point set L_t and directed planar point set P_t, respectively, where $L_t = \{l_i^t\}_{i=1}^{|L_t|}$ and $P_t = \{p_i^t\}_{i=1}^{|P_t|}$. Specifically, $l_i^t \in \mathbb{R}^6$ is the i^{th} directed edge point and the first three entries $l_i^t(1:3)$ refers to the position of i^{th} directed edge point in L_t, and the last three entries $l_i^t(4:6)$ is the corresponding unit direction vector of the line where the i^{th} directed edge point is located. Similarly, $p_i^t(1:3)$ and $p_i^t(4:6)$ describe the position of i^{th} directed planar point in P_t and unit

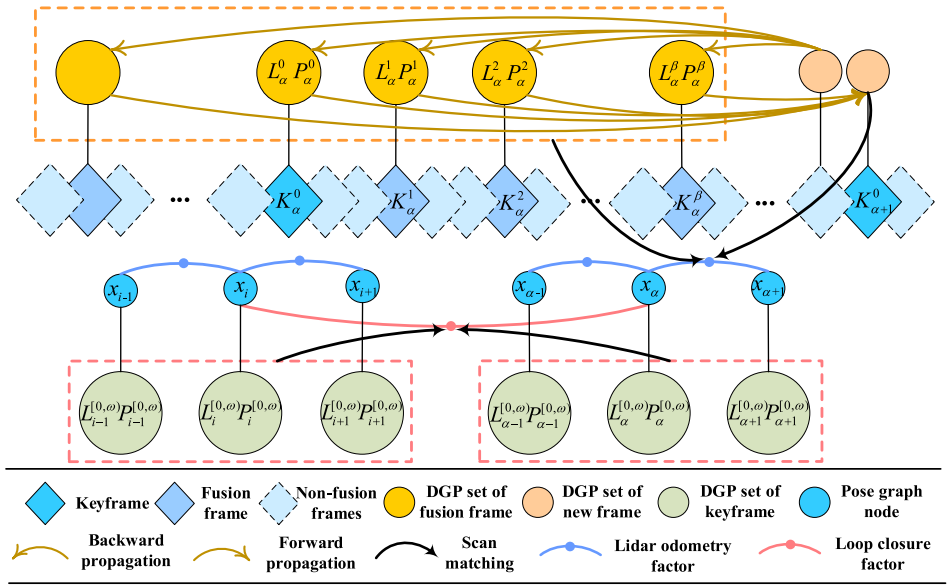

Fig. 1. Illustration of our DGP-SLAM pipeline. $K_\alpha^y (y = 0, 1, \ldots, \omega - 1)$ are fusion frames, and K_α^0, K_α^β refer to the latest keyframe and fusion frame before the new frame C_t. The local map is comprised of directed geometric point (DGP) sets in the dotted orange boxes, and the scan-to-map matching is conducted between the directed geometric point sets of C_t and local map. If the new frame C_t is not a fusion frame, the directed geometric points in C_t are propagated to the fusion frames in the local map; otherwise, points are propagated from the local map to the new frame C_t. A pose graph node x_α corresponds to the keyframe K_α^0. When C_t is recognized as new keyframe $K_{\alpha+1}^0$, the pose estimation would be added into the pose graph as the lidar odometry factor between x_α and $x_{\alpha+1}$. Besides, the loop detection occurs between the loop frame K_i^0 and K_α^0, and the loop closure factor between x_i and x_α is obtained by scan matching between two maps in the dotted red boxes.

normal vector of the plane i^{th} directed planar point located. The directed edge point set and directed planar point set in the fusion frame K_x^y are denoted as L_x^y and P_x^y, respectively.

$$L_x^y = \{l_i^{x,y}\}_{i=1}^{|L_x^y|} \tag{1}$$

$$P_x^y = \{p_i^{x,y}\}_{i=1}^{|P_x^y|} \tag{2}$$

Also, the directed edge point set $L_x^{[0,\omega)}$ and directed planar point set $P_x^{[0,\omega)}$ are attached in the keyframe K_x, where $L_x^{[0,\omega)}$ is comprised of L_x^0, L_x^1, …, $L_x^{\omega-1}$, and $P_x^{[0,\omega)}$ consists of P_x^0, P_x^1, …, $P_x^{\omega-1}$.

After that, the pose transformation between frames is characterized. The estimated pose transformation by scan-to-map matching from the latest keyframe K_α to the frame $C_t (t = 0, 1, 2, \cdots)$ is labelled as $T_{K_\alpha C_t} \in \mathbb{SE}_3$. Similarly, the relative pose estimation of scan-to-map matching from K_x to K_{x+1} is denoted as $T_{K_x K_{x+1}} \in \mathbb{SE}_3$. On the basis, the pose estimation of scan-to-map matching from W to C_t can be obtained as follows.

$$T_{WC_t} = T_{WK_1}\, T_{K_1K_2} \cdots T_{K_{\alpha-1}K_\alpha} T_{K_\alpha C_t} \tag{3}$$

Notice that the pose transformation between K_x and K_{x+1} changes after global pose graph optimization, we use $\hat{T}_{K_x K_{x+1}} \in \mathbb{SE}_3$ to refer to the optimized relative pose from K_x to K_{x+1}. Also, the optimized global pose from the world frame W to the frame C_t is referred to as $\hat{T}_{WC_t} \in \mathbb{SE}_3$, which is given by

$$\hat{T}_{WC_t} = \hat{T}_{WK_1}\, \hat{T}_{K_1K_2} \cdots \hat{T}_{K_{\alpha-1}K_\alpha} T_{K_\alpha C_t} \tag{4}$$

Overall, the problem statement is as follows. Given 3D LiDAR point clouds in frames $C_t (t = 0, 1, 2, \ldots)$, extract directed edge point and directed planar point sets L_t and P_t, estimate the pose transformation $T_{K_\alpha C_t}$ and maintain the direct geometric point sets L_x^y and P_x^y of fusion frames by scan-to-map matching, on this basis, detect loops with direct geometric point sets $L_x^{[0,\omega)}$ and $P_x^{[0,\omega)}$ of keyframes, and execute global pose graph optimization between keyframes $K_x (x = 0, 1, 2, \cdots)$ so that ego-motion estimation $\hat{T}_{WC_t}$ of LiDAR and a global consistent map are obtained with real-time performance and competitive accuracy.

IV. The Proposed 3D LiDAR SLAM

The framework of the proposed DGP-SLAM is given in Fig. 2, which includes three modules: DGP extraction, scan-to-map matching and loop closure. The DGP extraction module first extracts directed geometric points from new frame C_t. Then scan-to-matching module takes the directed geometric points as inputs to perform scan matching between C_t and the local map via iterative data association and pose optimization. Afterwards, frame selection and point propagation are conducted successively to effectively update local map. Further, the loop closure module combines the directed geometric point sets $L_x^{[0,\omega)}$, $P_x^{[0,\omega)}$ and pose transformation $T_{K_x K_{x+1}}$ to detect loop and conduct global pose graph optimization. Finally, a global pose $\hat{T}_{WC_t}$ and a global map of traversed area are outputted.

A. DGP Extraction

The DGP extraction module extracts directed geometric point sets L_t and P_t from raw 3D LiDAR data. In this paper, we

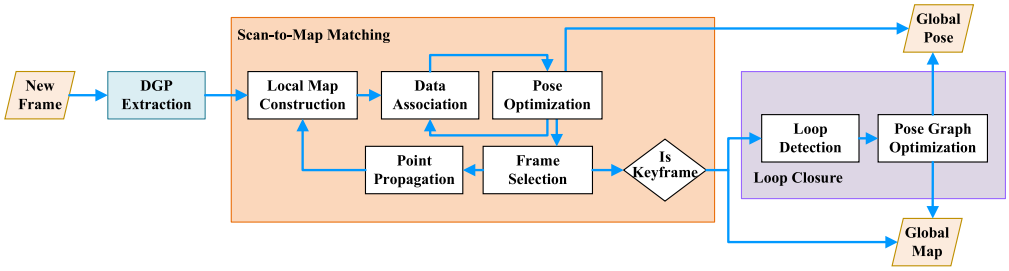

Fig. 2. The framework of the proposed DGP-SLAM.

concern two geometric points: edge and planar points. Because there are no edge points in the ground plane, we firstly separate ground points [22] as planar points. Then other edge and planar points are extracted from non-ground points using the strategy in [7]. On this basis, for each un-traversed edge or planar point, the neighboring points $u_1, u_2, \cdots u_m$ are obtained by the radius searching of KD tree [23]. Then, the center c_u and direction $\bar{d}_u$ of neighboring points are calculated, where $c_u = \frac{1}{m}\sum_{i=1}^{m} u_i$ and $\bar{d}_u$ is the unit eigenvector corresponding to minimum or maximum eigenvalue of matrix $\Sigma_U = \frac{1}{m}\sum_{i=1}^{m}(u_i - c_u)(u_i - c_u)^\top$ depending on whether they are planar or edge points. And then $(c_u, \bar{d}_u)$ is added into L_t or P_t. After that, $u_1, u_2, \cdots u_m$ are labelled as traversed. Repeat this process until each edge or planar point is traversed, and DGP extraction is accomplished. Note that the number of directed geometric points are much less than the number of geometric points as a directed geometric point is generated based on multiple geometric points.

B. Scan-to-Map Matching

The scan-to-map matching module is the core of DGP-SLAM. In the following, we address its three submodules: frame selection, pose estimation and point propagation in detail.

1) Frame Selection: As mentioned above, there are mainly three types of frames: fusion frame, keyframe, and non-fusion frame. The goal of the frame selection is to determine the type of a frame. The selection process is described below. Due to that the nearby frames have similar measurements, we keep distance between adjacent fusion frames. Specially, we label the latest fusion frame before new frame C_t as K_α^β , and then C_t is recognized as a new fusion frame when the distance between it and K_α^β denoted as $\|t_{WK_\alpha^\beta} - t_{WC_t}\|$ is larger than d_F, where $t_{WK_\alpha^\beta}$ and t_{WC_t} are the translation vectors of $T_{WK_\alpha^\beta}$ and T_{WC_t}. d_F is a threshold. On the basis, C_t is labelled as $K_\alpha^{\beta+1}$ if $\beta < \omega - 1$; otherwise, C_t is labelled as a new keyframe $K_{\alpha+1}$.

2) Pose Estimation: The pose estimation module endeavors to obtain the relative pose from the latest keyframe K_α to C_t denoted as $T_{K_\alpha C_t}$, which is conducted by iterative data association and pose optimization. For each round, data association between directed geometric point sets of C_t and local map is employed and then $T_{K_\alpha C_t}$ is updated by solving the optimization problem built by above data associations. The details are as follows.

First, the local map is constructed by projecting directed geometric points in the latest N fusion frames into a reference frame. In this paper, the latest keyframe K_α before the new frame C_t is selected as the reference frame of local map.

Next, data association and pose optimization are performed in an iterative way. Initially, the iteration number $r = 0$ and the pose to be optimized $T_{K_\alpha C_t} \in \mathbb{SE}_3$ is initialized by:

$$T_{K_\alpha C_t} = T_{K_\alpha C_{t-1}} \left(T_{K_\alpha C_{t-2}}\right)^{-1} T_{K_\alpha C_{t-1}},\ t \geq 2 \tag{5}$$

and $T_{K_\alpha C_0} = T_{K_\alpha C_1} = I$. Then at the r^{th} iteration $(1 \leq r \leq N_I - 1)$, $T_{K_\alpha C_t} \in \mathbb{SE}_3$ is initialized by the value of the optimized $T_{K_\alpha C_t}$ in the $(r-1)^{th}$ iteration, where $N_I = 3$ is the maximum iteration number.

After initialization, data association between directed geometric point sets of C_t and local map is performed. For i^{th} directed edge point l_i^t and i^{th} directed planar point p_i^t in C_t, they are projected to the reference frame of local map K_α by $T_{K_\alpha C_t}$, and the candidate associated points in the local map are then determined by the following three strict criteria:

1) point-point distance. The distance between the projected position of i^{th} directed geometric point and that of the candidate point is lower than r_1;
2) point-line or point-plane distance. The distance from i^{th} directed geometric point projection to the line or plane where the candidate point is located is lower than d_1;
3) direction difference. The angle between the directions corresponding to i^{th} directed geometric point projection and the candidate point is lower than θ_1.

The candidate with the minimal point-line or point-plane distance is regarded as the best association. Formally, we assume that the best associations for l_i^t and p_i^t are j^{th} directed edge point $l_j^{a,b}$ in fusion frame K_a^b and k^{th} directed planar point $p_k^{c,d}$ in fusion frame K_c^d, respectively. The projections of the best associations in the reference frame K_α are denoted as $\hat{l}_j^{a,b}$ and $\hat{p}_k^{c,d}$, which can be calculated by:

$$\left(\hat{l}_j^{a,b}, 1\right)^\top = \Omega \circ \left(l_j^{a,b}, 1\right)^\top \tag{6}$$

$$\left(\hat{p}_k^{c,d}, 1\right)^\top = \Omega \circ \left(p_k^{c,d}, 1\right)^\top \tag{7}$$

$$\Omega = \begin{bmatrix} R_{WK_\alpha}^\top R_{WK_a^b} & 0 & R_{WK_\alpha}^\top \left(t_{WK_a^b} - t_{WK_\alpha}\right) \\ 0 & R_{WK_\alpha}^\top R_{WK_a^b} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

where $R_{WK_a^b}$ and $t_{WK_a^b}$ are the rotation matrix and translation vector of $T_{WK_a^b}$, and R_{WK_α} and t_{WK_α} are the rotation matrix and translation vector of T_{WK_α}. $\top$ is transpose operator and the matrix multiplication is explicitly expressed by $\circ$ to avoid ambiguity.

With all valid data associations, the optimization problem is built by minimizing the objective function $f(T_{K_\alpha C_t})$ as follows:

$$\underset{T_{K_\alpha C_t}\in\mathbb{SE}_3}{min}\ f\ (T_{K_\alpha C_t}) = \sum_{i=1}^{|L_t|}\rho\left(\left\|e_i^l\right\|_2^2\right) + \sum_{i=1}^{|P_t|}\rho\left(\left\|e_i^p\right\|_2^2\right) \tag{8}$$

where

$$e_i^l = \left(T_{K_\alpha C_t} * l_i^t\,(1:3) - \hat{l}_j^{a,b}\,(1:3)\right) \times \hat{l}_j^{a,b}\,(4:6) \tag{9}$$

$$e_i^p = \left(T_{K_\alpha C_t} * p_i^t\,(1:3) - \hat{p}_k^{c,d}\,(1:3)\right) \cdot \hat{p}_k^{c,d}\,(4:6) \tag{10}$$

are the corresponding point-to-line and point-to-plane distance residuals. $\rho(\cdot)$ is the Huber loss function [24], $*:\mathbb{SE}_3\times\mathbb{R}^3\rightarrow\mathbb{R}^3$ is rigid transformation operator. $\times$ and $\cdot$ mean the cross product and dot product operators, respectively. It is noted that (8) is an optimization problem in manifold $\mathbb{SE}_3$, which can be converted to the following issue in Euclidean space by re-parameterization [25]:

$$\min_{\delta\phi\in\mathbb{R}^3,\delta t\in\mathbb{R}^3} f\left(\mathcal{R}_{T_{K_\alpha C_t}}\left(\delta\phi,\delta t\right)\right) \tag{11}$$

where the retraction $\mathcal{R}_{T_{K_\alpha C_t}}:\mathbb{R}^6\rightarrow\mathbb{SE}_3$ is a bijective map between the increment $(\delta\phi,\delta t)\in\mathbb{R}^6$ of the tangent space at $T_{K_\alpha C_t}\in\mathbb{SE}_3$ and neighborhood of $T_{K_\alpha C_t}$. The form of retraction is not unique. Following [25], the retraction is defined as follows:

$$\mathcal{R}_{T_{K_\alpha C_t}}\ (\delta\phi,\delta t) = \left(R_{K_\alpha C_t}\mathrm{Exp}\left(\delta\phi\right), t_{K_\alpha C_t} + R_{K_\alpha C_t}\delta t\right) \tag{12}$$

$R_{K_\alpha C_t}$ and $t_{K_\alpha C_t}$ are the rotation matrix and translation vector of $T_{K_\alpha C_t}$, respectively. (11) can be solve by the LM algorithm [26], where the derivatives of e_i^l and e_i^p with respect to $(\delta\phi,\delta t)$ are as follows.

$$\frac{\partial e_i^l}{\partial\left(\delta\phi,\delta t\right)} = -\left[\hat{l}_j^{a,b}\,(4:6)\right]_\times \left[-R_{K_\alpha C_t}[l_i^t\,(1:3)]_\times\ R_{K_\alpha C_t}\right] \in \mathbb{R}^{3\times3} \tag{13}$$

$$\frac{\partial e_i^p}{\partial\left(\delta\phi,\delta t\right)} = \hat{p}_k^{c,d}(4:6)^\top\ \left[-R_{K_\alpha C_t}[p_i^t\,(1:3)]_\times\ R_{K_\alpha C_t}\right] \in \mathbb{R}^{1\times3} \tag{14}$$

where $[\cdot]_\times$ and $\top$ are skew symmetric matrix and transpose operators. We denote the optimization results of (11) as $\delta\phi^*$ and δt^*, and then the solution of (8) is as follows.

$$T_{K_\alpha C_t} \leftarrow \mathcal{R}_{T_{K_\alpha C_t}}\left(\delta\phi^*,\delta t^*\right) \tag{15}$$

The above iteration process of data association and pose optimization is repeated N_I times to obtain the final pose estimation result.

3) Point Propagation: With the pose estimation result $T_{K_\alpha C_t}$, directed geometric points are created or updated via point propagation. First, data association between C_t and local map is re-performed using optimized $T_{K_\alpha C_t}$ with more strict criteria where θ_1 and d_1 are replaced by smaller thresholds θ_2 and d_2. With slight abuse of notation, the best associations of l_i^t and p_i^t are also described as j^{th} directed edge point $l_j^{a,b}$ in fusion frame K_a^b and k^{th} directed planar point $p_k^{c,d}$ in fusion frame K_c^d. Based on the best association results, forward or backward propagation occurs according to whether C_t is a fusion frame or not. If C_t is recognized as a fusion frame, l_i^t and p_i^t are added into the directed geometric point set of C_t. On this basis, forward propagation is performed where the associated points $l_j^{a,b}$ and $p_k^{c,d}$ are used to update l_i^t and p_i^t, and then they are deleted from K_a^b and K_c^d. Otherwise, backward propagation is conducted by updating $l_j^{a,b}$ and $p_k^{c,d}$ with l_i^t and p_i^t. No matter the propagation is forward or backward, each update can be regarded as the problem of modifying a point-direction pair $(p_1,\bar{d}_1)$ in the frame A by another point-direction pair $(p_2,\bar{d}_2)$ in the frame B. We denote the transformation from A to B as T. An intuitional way is to project $(p_2,\bar{d}_2)$ from B to A by T and then update $(p_1,\bar{d}_1)$ by weighted average. However, this scheme depends on the weight with an inaccurate result. Recall that a directed geometric point is acquired by searching neighbor points and eigenvalue decomposition. If we merge neighboring point sets U and V of $(p_1,\bar{d}_1)$ and $(p_2,\bar{d}_2)$, and then recalculate the covariance matrix and eigenvalues, a better updating would be expected, where $U=\{u_1,u_2,\ldots,u_m\}$ and $V=\{v_1,v_2,\ldots,v_n\}$. However, the direct merge of U and V is time consuming. Inspired by the incremental normal and curvature estimation in [27], we design an incremental update scheme. Instead of storing the point sets U and V, we store sum matrixes $\Phi_U=\sum_{i=1}^m u_iu_i^T$, $\Phi_V=\sum_{i=1}^n v_iv_i^T$ and sum vectors $\tau_U=\sum_{i=1}^m u_i$ and $\tau_V=\sum_{i=1}^n v_i$. After projection by T, Φ_V and τ_V become Φ_V' and τ_V':

$$\Phi_V' = R\Phi_VR^\top + R\tau_Vt^\top + t\tau_V^\top R^\top + tt^\top \tag{16}$$

$$\tau_V' = R\tau_V + t \tag{17}$$

where R and t are rotation matrix and translation vector of T. After that, Φ_U, τ_U and m are updated as follows.

$$\Phi_U \leftarrow \Phi_U + \Phi_V'\ \tau_U \leftarrow \tau_U + \tau_V'\ m \leftarrow m+n \tag{18}$$

With the updated Φ_U, τ_U and m, the new covariance $\Sigma_{U\cup V}$ of merged point set can be easily obtained as follows.

$$\Sigma_{U\cup V} = \frac{1}{m}\ \Phi_U - \frac{1}{m^2}\tau_U\tau_U^\top \tag{19}$$

Finally, the position and direction of updated directed geometric point can be calculated similar to the process of DGP extraction. It is noted that the sum matrixes and vectors of old fusion frames have already been calculated and our update scheme can achieve high efficiency.

C. Loop Closure

The loop closure module performs loop detection and pose graph optimization in a separate thread among keyframes. When C_t is recognized as a keyframe $K_{\alpha+1}$, the directed geometric point sets $L_\alpha^{[0,\omega)}$ and $P_\alpha^{[0,\omega)}$ of K_α are obtained, and then loop detection begins. First, we look for loop candidate frame by searching the spatially closest keyframe to K_α within keyframes between the first keyframe K_0 and $K_{\alpha-S}$, where S is a given threshold in avoidance of selecting temporally close keyframes. Suppose that the loop candidate frame is K_i, two local maps around K_i and K_α are built, as illustrated in the dotted red boxes of Fig. 2. After that, scan matching between above maps are performed. Note that loop candidate keyframe may be far

TABLE I
THE COMPARISION OF DIFFERENT VARIANTS OF OUR 3D LIDAR SLAM AVERAGED OVER KITTI 00-10 SEQUENCES IN RELATIVE TRANSLATION ERROR OF SCAN-TO-MAP MATCHING, RUNNING TIME OF SCAN-TO-MAP MATCHING AND POINT NUMBER OF LOCAL MAP

Method	GP*	DGP*	Strict data association	Sparse frame	Point propagation	Incremental update	RTE (%)	Time (ms/frame)	Point number
GP-SLAM	✓	—	—	—	—	—	0.95	151	92331
DGP-SLAM-I	—	✓	—	—	—	—	0.98	52	6003
DGP-SLAM-II	—	✓	✓	—	—	—	0.89	64	22036
DGP-SLAM-III	—	✓	✓	✓	—	—	0.93	56	18946
DGP-SLAM-IV	—	✓	✓	✓	✓	—	0.89	89	16499
DGP-SLAM	—	✓	✓	✓	✓	✓	0.89	55	17156

* GP: geometric point. DGP: directed geometric point.

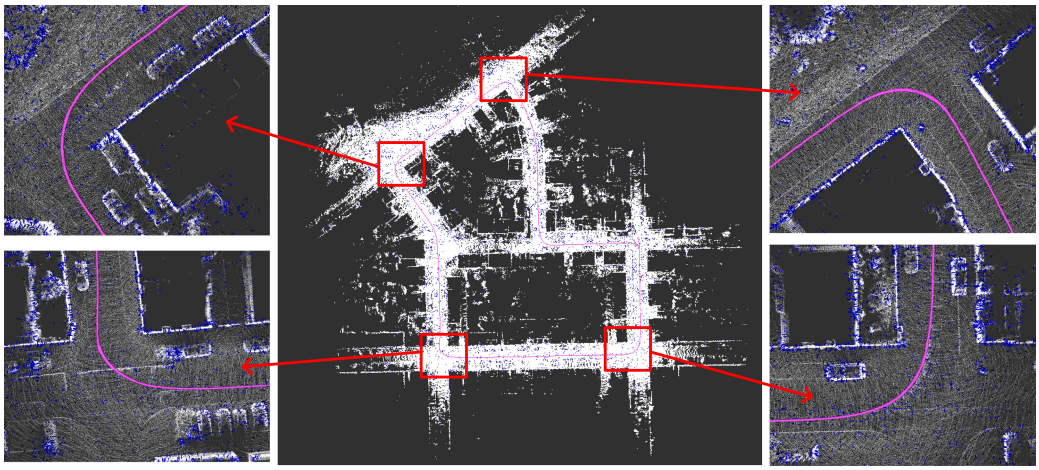

Fig. 3. The global directed geometric point map for sequence 07 of the KITTI dataset, which is visualized by blue lines. The endpoint and direction of each blue line correspond to the position and direction of a directed geometric point. Besides, the global geometric point map is also visualized by white points for comparison with the directed geometric point map.

away from new keyframe K_α, which is not suitable to use the strict data association. We instead use the ICP implementation in the PCL library [23] for scan matching in the loop detection. If the score of scan matching is larger than a given threshold, we consider that a valid loop is detected. In this case, the relative transformation between K_i and K_α is inserted into the pose graph as a loop closure factor. Also, another relative pose $T_{K_\alpha K_{\alpha+1}}$ from K_α to $K_{\alpha+1}$ is also added into the pose graph as a lidar odometry factor. On this basis, pose graph optimization is conducted [28] and then new poses $\hat{T}_{K_x K_{x+1}} (0 \leq x \leq \alpha)$ are produced. With the combination of directed geometric point sets $L_x^{[0,\omega)}$ and $P_x^{[0,\omega)}$ of K_x, a global consistent directed geometric point map is produced.

V. EXPERIMENTS

We evaluate the proposed DGP-SLAM method on the public KITTI dataset [29] and Newer College dataset [30]. The KITTI dataset includes sequences 00-10 with ground truth and sequences 11–21 without ground truth. The point cloud is collected by a Velodyne HDL64 mounted on a car, and ground truth is calculated by an accurate GPS/INS. This dataset covers a variety of scenes, such as urban, highway, and county, which provides a benchmark platform for different approaches. The Newer College dataset is a dataset collected using a handheld device at typical walking speeds through New College, Oxford. The point cloud is obtained by Ouster OS-1 64. Five sequences are provided in this dataset, and the ground truth poses of two sequences are currently available. In the following experiments, the parameters are set as follows: $\omega = 8$, $d_F = 2m$, $r_1 = 2m$, $d_1 = 1m$, $\theta_1 = 30^\circ$, $d_2 = 0.1m$, $\theta_2 = 20^\circ$, $N = 32$.

A. Ablation Experiment

In order to demonstrate the performance of our proposed DGP-SLAM, its five variants GP-SLAM, DGP-SLAM-I, DGP-SLAM-II, DGP-SLAM-III and DGP-SLAM-IV are considered according to whether directed geometric point, strict data association, sparse frame, point propagation, and incremental update are utilized. Table I presents the comparison results averaged over KITTI 00-10 sequences in terms of accuracy, efficiency, and point number of the local map. Take sequence 07 of the KITTI dataset as an example, the global directed geometric point map is illustrated in Fig. 3. The directed geometric point map is visualized by blue lines where the endpoint and direction of each blue line correspond to the position and direction of a directed geometric point. In addition, the global geometric point map constructed by white points is demonstrated for comparison. Clearly, the density of our DGP map is much less than that of geometric point map, which leads to obvious efficiency improvement of DGP-SLAM, as shown in Table I. Compared with the results of GP-SLAM, DGP-SLAM-I and DGP-SLAM-II, without the help of strict data association, the accuracy of

TABLE II
THE RELATIVE TRANSLATION (%)/ROTATION (DEG/100M) ERROR COMPARISON OF THE PROPOSED DGP-SLAM WITH EXISTING METHODS ON SEQUENCES 00-10 OF THE KITTI DATASET

	00	01	02	03	04	05	06	07	08	09	10
LeGO-LOAM without loop closure [8]	1.35/0.61	27.43/2.45	2.09/0.85	1.24/0.82	1.25/0.79	0.89/0.47	0.87/0.45	0.77/0.55	1.34/0.54	1.28/0.60	1.68/0.69
LeGO-LOAM [8]	1.38/0.57	28.03/2.69	2.14/0.84	1.21/0.79	1.27/0.84	0.91/0.48	0.80/0.43	0.74/0.54	1.40/0.54	1.25/0.60	1.70/0.69
A-LOAM [21]	0.79/0.32	1.96/0.50	4.57/1.46	0.95/0.49	0.77/0.42	0.50/0.25	0.62/0.28	0.45/0.26	1.11/0.35	0.74/0.33	1.01/0.39
LOAM [7]	0.78/-	1.43/-	0.92/-	0.86/-	0.71/-	0.57/-	0.65/-	0.63/-	1.12/-	0.77/-	0.80/-
DGP-SLAM without loop closure	0.89/0.41	1.94/0.46	0.99/0.38	0.79/0.41	0.71/0.41	0.43/0.30	0.49/0.23	0.58/0.63	0.97/0.34	0.66/0.30	1.09/0.41
DGP-SLAM	0.85/0.34	1.94/0.46	0.96/0.38	0.79/0.41	0.71/0.41	0.42/0.26	0.44/0.20	0.44/0.44	1.00/0.35	0.73/0.35	1.09/0.41

TABLE III
THE ACCURACY COMPARISON OF THE PROPOSED DGP-SLAM WITH EXISTING METHODS ON SHORT AND LONG EXPERIMENTS OF THE NEWER COLLEGE DATASET

	Short experiment				Long experiment			
	$RE_{pos}(\%)$	$RE_{rot}(deg/100m)$	$ATE_{pos}(m)$	$ATE_{rot}(deg)$	$RE_{pos}(\%)$	$RE_{rot}(deg/100m)$	$ATE_{pos}(m)$	$ATE_{rot}(deg)$
LeGO-LOAM without loop closure [8]	1.26	1.40	8.10	3.54	1.55	1.90	20.70	13.62
LeGO-LOAM [8]	1.03	1.18	5.26	2.50	1.11	1.44	6.07	5.18
A-LOAM [21]	1.79	5.09	13.59	5.26	4.21	3.84	61.83	20.57
DGP-SLAM without loop closure	0.88	0.97	6.18	2.47	0.80	1.08	11.99	10.83
DGP-SLAM	0.79	0.95	3.86	2.20	0.73	1.02	4.30	7.89

DGP-SLAM-I is even worse than that of GP-SLAM due to false data associations. By combining strict data association with directed geometric points, a high performance is achieved. Besides, the sparsity of frame can improve efficiency with a certain loss of accuracy, which can be compensated by point propagation. Furthermore, compared to DGP-SLAM-IV without incremental update, DGP-SLAM with incremental update can further improve efficiency without the loss of accuracy.

B. Accuracy Evaluation

We evaluate the accuracy of our method quantitatively on the KITTI dataset. Table II gives the relative translation and rotation errors of our method and existing methods including LOAM [7], A-LOAM [21], and LeGO-LOAM [8] with/without loop closure. It is noted that LOAM [7] runs at 10% of the real-time speed to achieve the best results due to the heavy computational burden of scan-to-map matching, whereas other methods run at a real-time speed on a desktop with 3.6 GHz Intel Core i7-7700 CPU. It can be seen that our approach achieves better accuracy in 7 sequences in terms of relative translation error, and in 6 sequences in terms of relative rotation error. Also, the average relative translation error of the proposed method on sequences 11–21 of the KITTI dataset is 1.00%, which is lower than 1.26% of A-LOAM [21].

Besides, we also evaluate our method on the Newer College dataset. It contains two sequences with ground truth: short experiment and long experiment. Table III presents the results of our method and existing methods including A-LOAM [21], and LeGO-LOAM [8] with/without loop closure, in terms of relative translation error RE_{pos}, relative rotation error RE_{rot}, absolute translation error ATE_{pos} and absolute rotation error ATE_{rot}. The relative translation and rotation errors RE_{pos}, RE_{rot} are calculated in the same way with KITTI dataset [29]. Besides, the absolute translation and rotation errors ATE_{pos}, ATE_{rot} are described by Root Mean Square Error (RMSE) of absolute errors between the whole estimated and ground truth trajectories [31]. One can see from the Table III that our method achieves the best results in terms of relative translation and rotation errors, which demonstrates the high accuracy of our scan-to-map matching. Besides, the absolute translation and rotation errors of our method with loop closure are smaller than that without loop closure, which verifies that the global consistency of trajectory is improved using loop detection and pose graph optimization.

C. Efficiency Evaluation

In this section, the efficiency of proposed SLAM is addressed where the methods LOAM, LeGO-LOAM, A-LOAM are involved in comparison. LOAM is implemented using the widely used public version LOAM_Velodyne [32]. Considering that LOAM and A-LOAM belong to the methods without explicit loop closure, the comparison of the running time is mainly conducted in terms of feature extraction, scan-to-scan matching and scan-to-map matching, where feature extraction in our approach refers to DGP extraction. Different from these three methods, our solution does not need the scan-to-scan matching due to the fast scan-to-map matching. Besides, to obtain a fair evaluation, sequences 00-10 of the KITTI dataset are divided into loop sequences (00, 02, 05, 06, 07, 08, 09) and acyclic sequences (01, 03, 04, 10), respectively, where acyclic sequences correspond to those without any loop. Table IV gives average running times of different methods for sequences 00-10 of the KITTI dataset. One can see from the results of acyclic sequences that the efficiency of our scan-to-map matching is obviously higher than others, which demonstrates the superiority of our scan-to-map matching over

TABLE IV
AVERAGED RUNNING TIMES (MS) OF DIFFERENT METHODS WITH RESPECT TO FEATURE EXTRACTION, SCAN-TO-SCAN MATCHING AND SCAN-TO-MAP MATCHING FOR SEQUENCES 00-10 OF THE KITTI DATASET

	LOAM_ Velodyne	LeGO-LOAM	A-LOAM	DGP-SLAM
Feature extraction	53	24	39	54
Scan-to-scan matching	36	18	71	—
Scan-to-map matching (loop sequences)	268	135	143	54
Scan-to-map matching (acyclic sequences)	231	170	125	57

TABLE V
AVERAGE NUMBER OF POINTS (PN) IN THE MAP FOR SCAN-TO-MAP MATCHING OF THE PROPOSED METHOD AND EXISTING METHODS

	LOAM_ Velodyne	LeGO-LOAM	A-LOAM	DGP-SLAM
PN	337142	54543	159070	17156

other methods in terms of efficiency. Moreover, the average running times of scan-to-map matching for three compared methods are all larger than 100 ms, which means that each frame cannot be processed in time, and this is why the scan-to-scan matching has to be introduced to ensure the real-time performance. On the contrary, our method is efficient where average running times of feature extraction and scan-to-map matching are both 50+ ms, and the real-time performance of our approach can be guaranteed by performing these two modules in two separate cores even without scan-to-scan matching.

Table V presents the average number of points in the map for scan-to-map matching of our method and existing methods in sequences 00-10 of the KITTI dataset. It can be seen that the map size of our approach is obviously less than others due to the sparseness of fusion frames with the utilization of DGP features, which paves the foundation of our efficient scan-to-map matching.

VI. CONCLUSION

This letter presents a novel 3D LiDAR SLAM method denoted as DGP-SLAM based on directed geometry point and sparse frame. The sparsity of directed geometry points and frames impove the efficiency of 3D LiDAR SLAM, and the strict data association based on directed geometric points also achieves good accuracy of pose estimation. Moerever, point propagation is proposed to compensate the accuracy loss of the localization and mapping caused by frame sparsity, and loop detection and pose graph optimization are conducted for global consistency. The experimental results reveal the effectiveness of the proposed DGP-SLAM.

REFERENCES

[1] L. Lecrosnier, R. Boutteau, P. Vasseur, X. Savatier, and F. Fraundorfer, "Camera pose estimation based on PnL with a known vertical direction," *IEEE Robot. Automat. Lett.*, vol. 4, no. 4, pp. 3852–3859, Oct. 2019.

[2] B. Bescos, J. M. Facil, J. Civera, and J. Neira, "DynaSLAM: Tracking, mapping and inpainting in dynamic scenes," *IEEE Robot. Automat. Lett.*, vol. 3, no. 4, pp. 4076–4083, Oct. 2018.

[3] A. L. Ballardini, L. Ferretti, S. Fontana, A. Furlan, and D. G. Sorrenti, "An indoor localization system for telehomecare applications," *IEEE Trans. Syst. Man, Cybern.: Syst.*, vol. 46, no. 10, pp. 1445–1455, Oct. 2016.

[4] P. J. Besl and N. D. McKay, "A method for registration of 3D shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, Feb. 1992.

[5] Y. Chen and G. Medioni, "Object modelling by registration of multiple range images," *Image Vis. Comput.*, vol. 10, no. 3, pp. 145–155, 1992.

[6] M. Magnusson, A. Lilenthal, and T. Duckett, "Scan registration for autonomous mining vehicles using 3D-NDT," *J. Field Robot.*, vol. 24, no. 10, pp. 803–827, 2007.

[7] J. Zhang and S. Singh, "LOAM: Lidar odometry and mapping in Real-time," in *Proc. Robotics: Sci. Syst. Conf.*, 2014.

[8] 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. Robot. Syst.*, 2018, pp. 4758–4765.

[9] J. Behley and C. Stachniss, "Efficient surfel-based SLAM using 3D laser range data in urban environments," in *Proc. Robot.: Sci. Syst. Conf.*, 2018.

[10] J. Serafin and G. Grisetti, "NICP: Dense normal based point cloud registration," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot Syst.*, 2015, pp. 742–749.

[11] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, "Comparing ICP variants on real-world data sets Open-source library and experimental protocol," *Auton. Robots*, vol. 34, no. 3, pp. 133–148, 2013.

[12] X. Ding, Y. Wang, H. Yin, L. Tang, and R. Xiong, "Multi-session map construction in outdoor dynamic environment," in *Proc. IEEE Int. Conf. Real-time Comput. Robot.*, 2018, pp. 384–389.

[13] M. Ramezani, G. Tinchev, E. Iuganov, and M. Fallon, "Online LiDAR-SLAM for legged robots with robust registration and deep-learned loop closure," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020 pp. 4158–4164.

[14] X. Chen, A. Milioto, E. Palazzolo, P. Giguere, J. Behley, and C. Stachniss, "SuMa++: Efficient lidar-based semantic SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 4530–4537.

[15] A. Schaefer *et al.*, "Long-Term urban vehicle localization using pole landmarks extracted from 3-D lidar scans," in *Proc. Europ. Conf. Mobile Robots*, 2019.

[16] W. S. Grant, R. C. Voorhies, and L. Itti, "Efficient velodyne SLAM with point and plane features," *Auton. Robots*, vol. 43, no. 5, pp. 1207–1224, 2019.

[17] K. Ji *et al.*, "CPFG-SLAM: A robust simultaneous localization and mapping based on LIDAR in off-road environment," in *Proc. IEEE Intell. Vehicles Symp.*, 2018, pp. 650–655.

[18] J. Weingarten and R. Siegwart, "3D SLAM using planar segments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2006, pp. 3062–3067.

[19] J. Deschaud, "IMLS-SLAM: Scan-to-model matching based on 3D data," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 2480–2485.

[20] K. Konolige, E. Marder-Eppstein, and B. Marthi, "Navigation in hybrid metric-topological maps," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2011, pp. 3041–3047.

[21] T. Qin and S. Cao, A-LOAM. 2018. [Online]. Available: https://github.com/HKUST-Aerial-Robotics/A-LOAM

[22] D. Zermas, I. Izzat, and N. Papnikolopoulos, "Fast segmentation of 3D point clouds: A paradigm on LiDAR data for autonomous vehicle applications," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2017, pp. 5067–5073.

[23] R. Bogdan and S. Cousins, "3D is here: Point cloud library (PCL)," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2011.

[24] S. Agarwal *et al.*, Ceres Solver. 2012. [Online]. Available: http://ceres-solver.org

[25] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-Manifold preintegration for real-time visual-inertial odometry," *IEEE Trans. Robot.*, vol. 33, no. 1, pp. 1–21, Feb. 2017.

[26] T. D. Barfoot. *State Estimation for Robotics*. Cambridge, England: Cambridge Univ. Press, 2017.

[27] R. Dube *et al.*, "Incremental-Segment-Based localization in 3-D point clouds," *IEEE Robot. Automat. Lett.*, vol. 3, no. 3, pp. 1832–1839, Jul. 2018.

[28] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, "ISAM2: Incremental smoothing and mapping using the bayes tree," *Int. J. Robot. Res.*, vol. 31, no. 2, pp. 216–235, 2012.

[29] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3354–3361.

[30] M. Ramezani, Y. Wang, M. Camurri, D. Wisth, M. Mattamala, and M. Fallon, *The Newer College Dataset: Handheld Lidar, Inertial and Vision with Ground Truth*, The Oxford Robotics Institute, University of Oxford, UK., 2020.

[31] Z. Zhang and D. Scaramuzza, "A tutorial on quantitative trajectory evaluation for visual(-inertial) odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 7244–7251.

[32] L. Laboshin, et al., 2015. [Online]. Available: https://github.com/laboshinl/loam_velodyne