Fast and Versatile Feature-Based LiDAR Odometry via Efficient Local Quadratic Surface Approximation

Seungwon Choi, Hee-Won Chae, Yunsuk Jeung, Seokjoon Kim, Kyusung Cho, and Tae-wan Kim

***Abstract*—We present a fast and versatile feature-based LiDAR odometry method using local quadratic surface approximation and point-to-surface alignment. Unlike most feature-based methods, our approach approximates the local geometry of the LiDAR scan as a quadratic surface to mitigate performance degradation caused by the inconsistency between the feature class and the map's local geometry. For computational efficiency, we leverage a symmetric objective function to align features on the local surface of the map without requiring time-consuming surface parameter evaluation. Evaluation on the KITTI and Newer College dataset demonstrates that the proposed method performs better than other feature-based methods. In particular, our approach exhibits robust performance in environments where the ambiguity of feature classifications is considerable. In addition, to demonstrate the robustness of the proposed method when LiDAR scans are relatively sparse, we evaluated the proposed method on datasets collected using LiDAR with a relatively small number of scan channels.**

***Index Terms*—LiDAR odometry, localization, mapping, SLAM.**

I. Introduction

SIX degrees of freedom (6-DoF) sensor pose estimation is a fundamental research topic in robotics, as it can provide important information regarding navigation and three-dimensional (3D) reconstruction of the environment. In particular, LiDAR odometry estimation, which provides robust localization performance and creates a dense 3D map, has been explored in various ways. The performance of LiDAR odometry depends on the quality of scan-to-scan matching using the iterative closest point (ICP) [1], [2] method. In this method, the nearest points within different scans are paired using the distance between two scan points, and an incremental transform that minimizes the sum of the distance between the point pairs is repeatedly found.

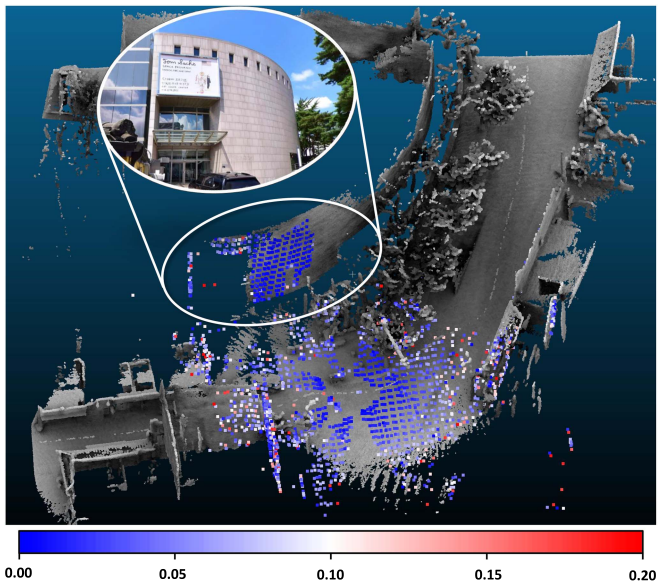

Fig. 1. Example of an integrated point cloud map (grey) and feature points extracted from a raw scan. The color of a feature point depends on the local smoothness, and the color corresponding to the smoothness value is shown in the bar below.

To achieve real-time performance, a feature-based approach proposed by LiDAR odometry and mapping (LOAM) [3] is generally used for LiDAR odometry estimation. In the feature-based approach, structural features, such as edges and planes are extracted from the raw scan using local smoothness. Then, the LiDAR pose is estimated by aligning the features with existing correspondences existing in the map, using the ICP method. Feature-based approaches provide computationally efficient and robust performance relative to direct methods [4], [5], [6], [7], [8]. However, feature classification using evaluated local curvature involves considerable ambiguity. For example, in Fig. 1, it can be seen that the curvature of the point cloud observed on the curved surface is lower than that of the one observed on the flat plane, and this surface point is classified as a planar feature, based on its curvature. The inconsistency between the feature class and the actual local geometry reduces the convergence and accuracy of registration between LiDAR scans and map points [9], which impairs the performance of LiDAR odometry.

In this letter, we focus on circumventing the inconsistency by defining the feature as the point observed on the smooth

Manuscript received 16 August 2022; accepted 29 November 2022. Date of publication 8 December 2022; date of current version 23 December 2022. This letter was recommended for publication by Associate Editor P. Checchin and Editor S. Behnke upon evaluation of the reviewers' comments. This work was supported by Industrial Strategic Technology Development Program- Electronic System Industrial Technology Development Project under Grant 1415178235, Development of Industrial AR Support Platform Technology for Integrated Work Support in Manufacturing Site funded by the Ministry of Trade, Industry & Energy (MOTIE), Korea. *(Corresponding author: Tae-wan Kim.)*

Seungwon Choi is with the Department of Naval Architecture and Ocean Engineering, Seoul National University (SNU) Seoul 08826, Republic of Korea, and also with the SLAM Team, MAXST Seoul 08826, Republic of Korea (e-mail: csw3575@snu.ac.kr).

Hee-Won Chae, Yunsuk Jeung, Seokjoon Kim, and Kyusung Cho are with the SLAM Team, MAXST Seoul 06267, Republic of Korea (e-mail: hwchae@maxst.com; ysjeung@maxst.com; seokjoon@maxst.com; kscho@maxst.com).

Tae-wan Kim is with the Department of Naval Architecture and Ocean Engineering, Seoul National University (SNU) Seoul 08826, Republic of Korea (e-mail: taewan@snu.ac.kr).

Digital Object Identifier 10.1109/LRA.2022.3227875

surface and approximating the local geometry of the map as a quadratic surface. For this, we use the symmetric objective function proposed by Rusinkieicz [10] in the field of computer graphics. The zero-set of this function is the condition that the two points lie on a quadratic surface, and there is no need to evaluate parameters of the surface because the function can be constructed from point pair positions and normal vectors.

The main contribution of this letter is to propose a method for approximating the local geometry of map points to a quadratic surface for robust LiDAR odometry, even in environments where the ambiguity of feature classifications is considerable. In addition, we propose a theoretical background for treating this problem as an on-manifold non-linear optimization problem and a derivation of the optimization process for this problem.

We evaluated the performance of the proposed method using the KITTI [11] and Newer College datasets [12], [13]. These evaluation results are compared with those of other feature-based methods, and we demonstrate that the proposed method not only improves the performance of LiDAR odometer odometry but also ensures robust performance in environments where there is considerable ambiguity in feature classifications, where existing methods often fail. Furthermore, we aslo collected datasets via Velodyne VLP-32 LiDAR, which has a relatively small number of channels, to demonstrate that the proposed method can improve the performance of LiDAR odometry even when the approximation of the local geometry is challenging due to the sparsity of the LiDAR scan.

The remainder of this letter is organized as follows: Section II presents a review of the existing feature-based LiDAR odometry methods and variant ICP algorithms for scan-to-scan matching. Section III provides a detailed description of the method proposed in this letter. Section IV details a comparison of the performance of the proposed algorithm with that of methods, using public and our datasets. Finally, Section V gives the conclusions of this letter.

II. Related Work

LiDAR odometry is typically performed using the iterative closest point (ICP) algorithm [1], [2]. In this method, a relative transformation is estimated that aligns two point clouds thorough iteratively performance of correspondence matching and non-linear optimization. The convergence and stability of ICP in any instance depend on the distance function to be minimized [14]. For this reason, many approaches related to the modification of distance metrics have been proposed to improve ICP performance [9], [14], [15], [16], [17]. Point-to-plane ICP [2] aligns a point to the local plane. This method provides better convergence and stable alignment than point-to-point ICP [1], which minimizes the distances between paired points. In Generalized-ICP (G-ICP) [4], ICP is related to a probabilistic model to enable robust point-to-plane minimization. As this method provides robust scan alignment, it is also applied to various estimation methods for LiDAR odometry.

To achieve real-time performance, the feature-based approach proposed in LOAM [3] is generally used to LiDAR odometry. Following this method, when a newly measured LiDAR scan is obtained, feature points that are considered to be measured on a plane or edge are extracted using local smoothness, and the current pose of LiDAR is estimated by matching these newly extracted features with map points via ICP. This method has become the standard, as it has produced improved computational efficiency and robustness compared to the direct methods performed without feature extraction.

To accelerate the feature-based method, Shan et al. proposed lightweight and ground-optimized LiDAR odometry and mapping (LeGO-LOAM) [18]. In this method, efficient point segmentation and two-setp scan matching were proposed, and improved computational efficiency over that possible using LOAM was achieved. Furthermore, in Fast LiDAR odometry and mapping (Fast LOAM) [19], the initial pose estimation, performed via matching features between two consecutive scans was simplified under the assumption that the linear and angular velocities of sensor motion were held constant for a short time interval to further improve computational efficiency. In versatile LiDAR SLAM via multi-metric linear least square (MULLS) [20], a method is proposed of extending the feature class to ground, facade, roof, pillar, beam, and vertex for robust performance in various environments.

The method of evaluating local smoothness involving adjacent points in horizontal scans that is used in most feature-based LiDAR odometry methods [3], [18], [19], [20] is computationally efficient, but the ambiguity of the feature classifications is considerable, as shown in Fig. 1 This ambiguity leads inconsistency between the feature class and the actual local geometry of the map points, which is a major factor that impairs the convergence and robustness of ICP in LiDAR odometry [9]. In the field of computer graphics, several studies have been proposed to mitigate such inconsistency. In particular, Mitra et al. [21] approximated local geometry to a quadratic surface and minimized the point-to-surface distance. This method demonstrated an improvement in the convergence of ICP in various experiments, but its computational efficiency is reduced due to the process used for evaluation of the surface parameters. For point-to-surface alignment without this evaluation, Rusinkiewicz [10] proposed a symmetric objective function that is minimized when two points lie on the same quadratic surface. This symmetric objective function can be constructed using coordinates and normal vectors of point pairs, and it can be compared to the point-to-plane error; the convergence basin is widened from the extended zero-set, thus providing improved convergence for ICP.

In computer graphics, the main use of the point registration via ICP is to align two correlated dense point clouds that belong to the objects having smooth surfaces. Such assumption is rather difficult to be applied in LiDAR odometry. The challenge of LiDAR odometry is to align scan points only with the given sparse sets of the visible surfaces of the obstacles. In other words, the observable surfaces are limited due to the occlusion and the blind spots. Therefore, unlike the work of Rusinkieicz, we adopt the feature extraction process to deal with sparse information of the given surfaces, described in Section III-B. In this process, the points observed from the smooth surfaces are extracted based on the local smoothness of the horizontal scan. Also, unlike

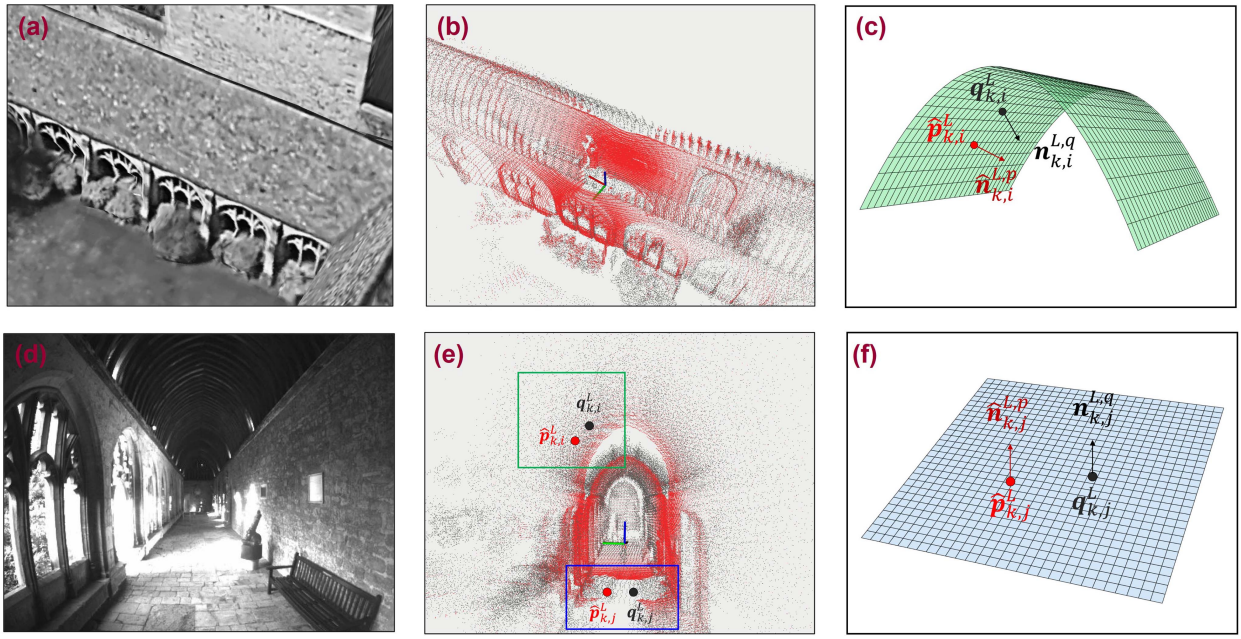

Fig. 2. Example of feature-based point-to-surface alignment in cloister sequence of the Newer College dataset. Map points (in black) and surface features (in red) in (b), (e) are aligned on an evaluated quadratic surface (c) including the flat plane (f). (c) and (f) represent quadratic surface approximations using the point coordinates and their normal.

the work of Rusinkieicz, our approach handled this non-linear problem on the manifold. We have added a detailed explanation of this content in Section III-D and derived the equations for this problem.

III. Methodology

Our approach focuses on efficient point-to-surface alignment for LiDAR odometry to provide robust performance in environments where the ambiguity of feature classifications is considerable. To efficiently approximate the local geometry of LiDAR scans as surfaces, we leverage the symmetric objective function proposed by Rusinkiewicz in the field of computer graphics. Before describing this method in detail, we first present the work of Rusinkiewicz, which is the foundation of our approach (Section III-A). Then, we describe the steps we take to extend the work of Rusinkiewicz to LiDAR odometry. First, surface feature extraction for point-to-surface alignment will be described (Section III-B). We then formulate the problem as a non-linear optimization problem (Section III-C). Finally, we derive efficient on-manifold non-linear optimization approach for the problem (Section III-D).

A. Symmetric Objective Function

A symmetric objective function was proposed to approximate the local geometry of the point cloud in terms of a quadratic surface, without evaluating the surface parameters. In this section, we present the physical meaning and derive the objective function.

A quadratic surface is generally specified in the following form:

$$Ax^2+By^2+Cz^2+Dyz+Ezx+Fxy+Gx+Hy+Iz+J=0, \tag{1}$$

where A through J are surface parameters, and $\mathbf{x}=[x,y,z]\in\mathbb{R}^3$ is the position of any point on the surface. This equation can be written in a quadratic form as:

$$\tilde{\mathbf{x}}^T\mathbf{Q}\tilde{\mathbf{x}}=0, \tag{2}$$

where $\tilde{\mathbf{x}}=[x\ y\ z\ 1]$ and $\mathbf{Q}$ is 4 by 4 symmetric matrix. The physical meaning of symmetric objective function is the condition that a point pair $\{\mathbf{x}_i,\ \mathbf{x}_j\}\in\mathbb{R}^3$ is on a quadratic surface defined by $\mathbf{Q}$. Using (2), this condition can be written as:

$$\tilde{\mathbf{x}}_i^T\mathbf{Q}\tilde{\mathbf{x}}_i-\tilde{\mathbf{x}}_j^T\mathbf{Q}\tilde{\mathbf{x}}_j=(\tilde{\mathbf{x}}_i-\tilde{\mathbf{x}}_j)^T\mathbf{Q}(\tilde{\mathbf{x}}_i+\tilde{\mathbf{x}}_j)=0, \tag{3}$$

and considering $\mathbf{Q}\tilde{\mathbf{x}}_i=[\mathbf{n}_i^T\ \mathbf{q}_4^T\tilde{\mathbf{x}}_i]^T$, where $\mathbf{n}_i$ and $\mathbf{q}_4$ are the normal vector of tangent plane on quadratic surface at $\mathbf{x}_i$ and the fourth column vector of $\mathbf{Q}$, respectively, (3) can be rewritten as:

$$(\mathbf{x}_i-\mathbf{x}_j)^T(\mathbf{n}_i+\mathbf{n}_j)=0 \tag{4}$$

The left side of (4) gives the definition of the symmetric objective function. Using this function, the ICP between the point clouds $\mathcal{P}$ and $\mathcal{Q}$ can be modeled as the point-to-surface alignment. Let transformation $\mathbf{T}$ be composed of a rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$, as:

$$\mathbf{T}=\begin{bmatrix}\mathbf{R} & \mathbf{t}\\ \mathbf{0}^{\mathrm{T}} & 1\end{bmatrix}\in\mathbb{R}^{4\times4},\ \mathbf{R}\in\mathbb{R}^{3\times3},\ \mathbf{t}\in\mathbb{R}^3 \tag{5}$$

Then, ICP via point-to-surface alignment can be written as:

$$\mathbf{T}^* = \underset{\mathbf{T}}{\operatorname{argmin}} \sum_{\{\mathbf{p},\mathbf{q}\}\in\mathcal{S}} \left[(\mathbf{R}\mathbf{p} + \mathbf{t} - \mathbf{R}^{\mathrm{T}}\mathbf{q}) \cdot (\mathbf{n}_{\mathrm{p}} + \mathbf{n}_{\mathrm{q}})\right]^2, \quad (6)$$

where $\mathcal{S}$ is a set of point pairs and $\mathbf{n}_{\mathrm{p}} \in \mathbb{R}^3$ and $\mathbf{n}_{\mathrm{q}} \in \mathbb{R}^3$ are the normal vectors of $\mathbf{p} \in \mathcal{P}$ and $\mathbf{q} \in \mathcal{Q}$ respectively. Note that a split in the rotation is introduced in (5) to reduce the linearization error. Therefore, the rotation matrix and translation vector of final transform that align $\mathcal{P}$ to $\mathcal{Q}$ is **RR** and **Rt** respectively.

B. Surface Feature Extraction

When a new LiDAR scan is observed, feature extraction is performed for efficient scan matching. Unlike other feature-based methods, our method assume a quadratic surface for the local geometry of feature points. In extraction of surface feature, local smoothness is estimated using adjacent points in the horizontal scan, as suggested in LOAM. The local smoothness of the k-th point of the m-th horizontal $c_{m,k}$ is estimated as follows:

$$c_{m,k} = \frac{1}{|S_{m,k}|} \sum_{\mathbf{p}^{\mathcal{L}}_{m,i}\in S_{m,k}} \|\mathbf{p}^{\mathcal{L}}_{m,i} - \mathbf{p}^{\mathcal{L}}_{m,k}\| \quad (7)$$

where $\mathbf{p}^{\mathcal{L}}_{m,k} \in \mathbb{R}^3$ and $\mathbf{p}^{\mathcal{L}}_{m,i} \in \mathbb{R}^3$ are the k-th point of the m-th horizontal scan and i-th point of $S_{m,k}$, which is the set of adjacent points of $\mathbf{p}^{\mathcal{L}}_{m,k}$, respectively, in LiDAR frame $\mathcal{L}$. If this value falls below the threshold, the point is considered to have been observed on a smooth surface. To distribute the features evenly over the scan, each horizontal scan is divided into several sub-regions and surface features less than a certain number are extracted from each region. As suggested by LOAM, if one or more of the points adjacent to a point have already been selected as a feature, or if this point is extracted from a plane parallel to the horizontal scan or the boundary of an occluded region, this point is not itself selected as a feature, even if the curvature of this point is below the threshold.

C. LiDAR Odometry With Point-to-Surface Alignment

Our point-to-surface alignement for LiDAR odometry is divided into four steps. First, the normals of the extracted features are estimated through sphere fitting, with the assumption that the local geometry forms a quadratic surface. Then, the motion compensation of the observed scan is performed, using the prediction of the current LiDAR pose. Then, the pose of the current LiDAR is estimated through point-to-surface alignment between map points and features. Finally, undistorted features are added as new map points.

We assume that LiDAR motion during a sweep has a constant linear and angular velocities. Therefore, the motion compensations are performed with the following:

$$\hat{\mathbf{p}}^{\mathcal{L}}_{k,i} = \mathbf{R}^{\mathcal{L}}_{k-1,i}\mathbf{p}^{\mathcal{L}}_{k,i} + \mathbf{t}^{\mathcal{L}}_{k-1,i} \quad (8)$$

$$\hat{\mathbf{n}}^{\mathcal{L}}_{p,k,i} = \mathbf{R}^{\mathcal{L}}_{k-1,i}\mathbf{n}^{\mathcal{L}}_{p,k,i} \quad (9)$$

where $\mathbf{p}^{\mathcal{L}}_{k,i} \in \mathbb{R}^3$ and $\mathbf{n}^{\mathcal{L}}_{p,k,i} \in \mathbb{R}^3$ are the position and normal of the i-th feature extracted from the k-th LiDAR frame, respectively. The subscript p of $\mathbf{n}^{\mathcal{L}}_{p,k,i}$ means that this is normal vector of p. This is used to distinguish it from the normal vector of the corresponding map point q in (13). $\mathbf{R}^{\mathcal{L}}_{k-1,i} \in \mathbb{R}^{3\times3}$ and $\mathbf{t}^{\mathcal{L}}_{k-1,i} \in \mathbb{R}^3$ are the lineary interpolated rotation matrix and translation vector of the LiDAR relative transform $\mathbf{t}^{\mathcal{L}}_{k-1} \in \mathbb{R}^{4\times4}$ between times $[t_{k-2},\ t_{k-1}]$, respectively. We use the standard spherical linear interpolation (slerp) for rotation. We denote the de-skewed feature point and its normal as $\hat{\mathbf{p}}^{\mathcal{L}}_{k,i} \in \mathbb{R}^3$ and $\hat{\mathbf{n}}^{\mathcal{L}}_{p,k,i} \in \mathbb{R}^3$, respectively.

The LiDAR pose is corrected by estimating the relative transform that aligns the de-skewed features to the nearest quadratic surface on the map through point-to-surface distance minimization. For simplicity of implementation, all operations are described in the local frame of the LiDAR. By this means, the map points and their normal vectors are also transformed into the local LiDAR coordinate system using the current LiDAR pose as:

$$\hat{\mathbf{T}}^{\mathcal{W}}_{k} = \mathbf{T}^{\mathcal{L}}_{k-1}\mathbf{T}^{\mathcal{W}}_{k-1} \quad (10)$$

$$\mathbf{q}^{\mathcal{L}}_{k,i} = \left(\hat{\mathbf{R}}^{\mathcal{W}}_{k}\right)^{\mathrm{T}} \mathbf{q}^{\mathcal{W}}_{k,i} - \left(\hat{\mathbf{R}}^{\mathcal{W}}_{k}\right)^{\mathrm{T}} \hat{\mathbf{t}}^{\mathcal{W}}_{k} \quad (11)$$

$$\mathbf{n}^{\mathcal{L}}_{q,k,i} = \left(\hat{\mathbf{R}}^{\mathcal{W}}_{k}\right)^{\mathrm{T}} \mathbf{n}^{\mathcal{W}}_{q,k,i} \quad (12)$$

where $\hat{\mathbf{T}}^{\mathcal{W}}_{k} \in \mathbb{R}^{4\times4}$ is the prediction of k-th LiDAR pose in the world coordinate system, $\mathbf{T}^{\mathcal{W}}_{k-1} \in \mathbb{R}^{4\times4}$ is the optimized LiDAR pose at time t_{k-1}, and $\mathbf{q}^{\mathcal{L}}_{k,i} \in \mathbb{R}^3$ and $\mathbf{n}^{\mathcal{L}}_{q,k,i} \in \mathbb{R}^3$ are the position and normal vectors of the corresponding map point in local LiDAR coordinate system, respectively.

Let **R** and **t** are the rotation matrix and translation vector that align features to map points. Then, the objective function of i-th point pair is formed as:

$$f_i(\mathbf{T}) = \left(\mathbf{R}\hat{\mathbf{p}}^{\mathcal{L}}_{k,i} + \mathbf{t} - \mathbf{R}^{\mathrm{T}}\mathbf{q}^{\mathcal{L}}_{k,i}\right) \cdot \left(\hat{\mathbf{n}}^{\mathcal{L}}_{p,k,i} + \mathbf{n}^{\mathcal{L}}_{q,k,i}\right) \quad (13)$$

As shown in (5), **T**, which minimizes the sum of the objective functions of all point pairs can be estimated using a non-linear optimization method and the relative LiDAR transform during $[t_{k-1},\ t_k]$ is calculated using the result as:

$$\mathbf{T}^{\mathcal{L}}_{k} = \mathbf{T}^{\mathcal{L}}_{k-1} \begin{bmatrix} \mathbf{RR} & \mathbf{Rt} \\ \mathbf{0} & 1 \end{bmatrix} \quad (14)$$

Finally, the optimized LiDAR pose at t_k in the world coordinate system is calculated as:

$$\mathbf{T}^{\mathcal{W}}_{k} = \mathbf{T}^{\mathcal{L}}_{k}\mathbf{T}^{\mathcal{W}}_{k-1}, \quad (15)$$

and the extracted feature points are undistorted using the current LiDAR pose and added as new map points.

D. On-Manifold Non-Linear Optimization

The rotation matrix **R** and transformation **T** in (5) belong to the *Special Orthogonal Group* SO(3) and *Special Euclidean Group* SE(3) respectively defined as:

$$\mathrm{SO}(3) = \{\mathbf{R} \in \mathbb{R}^{3\times3} \mid \mathbf{R}^{\mathrm{T}}\mathbf{R} = \mathbf{I},\ \det(\mathbf{R}) = 1\}, \quad (16)$$

TABLE I
ABSOLUTE TRANSLATION ERROR (IN METERS) OF EACH METHOD WITH KITTI TRAINING SEQUENCES. ALL ERRORS ARE REPRESENTED AS [M]. THE AVERAGE TIME CONSUMPTION FOR A SCAN ΔT ACCORDING TO EACH METHOD IS REPRESENTED AS [MS]. BOLD CELLS IN A SINGLE ROW PRESENT THE BEST PERFORMER FOR THAT METRIC

Sequence (Num. of frames)	00 (4541)	01 (1011)	02 (4661)	03 (801)	04 (271)	05 (2761)	06 (1101)	07 (1101)	08 (4071)	09 (1591)	10 (1201)	Avg. per frame [m]	ΔT [ms]
Proposed	**3.60**	18.83	8.57	0.91	0.39	**1.18**	0.91	0.55	4.62	2.15	2.13	**4.63**	**40**
F-LOAM	4.99	18.85	**8.00**	**0.70**	0.54	2.74	0.69	0.45	4.21	**1.46**	1.42	4.80	**40**
MULLS	6.04	**2.38**	10.61	0.71	0.91	2.42	**0.66**	0.64	4.37	2.51	**1.14**	4.80	75
LeGo-LOAM	6.31	119.42	14.67	0.92	0.85	2.82	0.81	0.72	**3.51**	2.14	1.82	11.16	60
A-LOAM	4.03	20.17	61.03	0.72	**0.30**	2.83	0.72	**0.40**	4.66	1.59	3.14	15.51	130

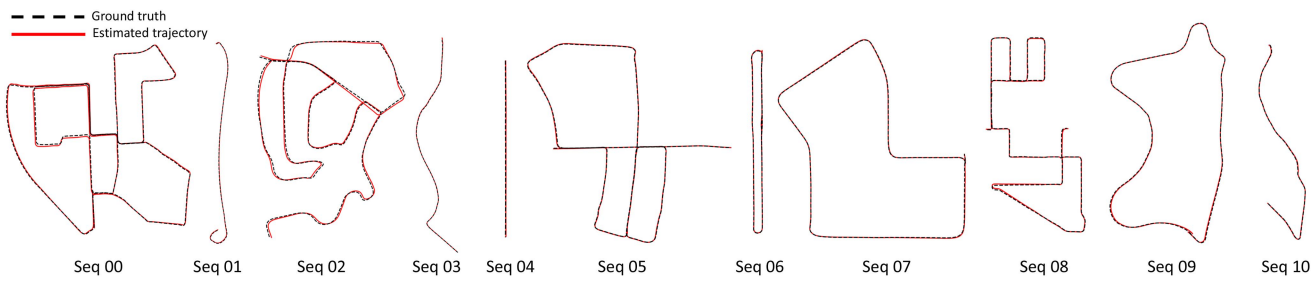

Fig. 3. Estimated and ground truth trajectory of KITTI training sequences (00–10).

$$\mathrm{SE}(3) = \left\{ \mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix} \in \mathbb{R}^{4\times 4} \mid \mathbf{R} \in \mathrm{SO}(3),\ \mathbf{t} \in \mathbb{R}^3 \right\}, \tag{17}$$

Because these groups belong to the *Lie Group*, a set of differentiable manifolds, not to linear vector space, LiDAR odometry is typically treated as the on-manifold non-linear optimization problem. A standard approach for this problem consists of three steps: mapping from the manifold space to the linear vector space, solving the problem via first-order approximation and a normal equation, and mapping the solution back to the manifold space. In this section, we introduce this method for SO(3) and SE(3) and derive the optimization process for our problem. Our presentation is based on [22]

Every Lie Group can be associated with *Lie algebra*, which is linear vector space. Associated Lie algebra of SO(3) and SE(3) are $\mathfrak{so}(3)$ and $\mathfrak{se}(3)$ respectively defined as follow:

$$\mathfrak{so}(3) = \left\{ \boldsymbol{\phi}^{\wedge} \in \mathbb{R}^{3\times 3} \mid \boldsymbol{\phi} \in \mathbb{R}^3 \right\}, \tag{18}$$

$$\mathfrak{se}(3) = \left\{ \boldsymbol{\xi}^{\wedge} \in \mathbb{R}^{4\times 4} \mid \boldsymbol{\xi} = \begin{bmatrix} \boldsymbol{\rho} \\ \boldsymbol{\phi} \end{bmatrix} \in \mathbb{R}^6 \right\}, \tag{19}$$

where $(\cdot)^{\wedge}$ is the skew-symmetric operator and $\boldsymbol{\rho} \in \mathbb{R}^3$ and $\boldsymbol{\phi} \in \mathbb{R}^3$ represent the translation and rotation part of $\boldsymbol{\xi}$, respectively.

In terms of rotation, linear vector space can be related with manifold by means of *Exponential map* defined as:

$$\mathbf{R} = \exp(\boldsymbol{\phi}^{\wedge}) = \mathbf{I} + (\sin\theta)\mathbf{a}^{\wedge} + (1-\cos\theta)\mathbf{a}\mathbf{a}^{\mathrm{T}}, \tag{20}$$

where $\theta \in \mathbb{R}$ and $\mathbf{a} \in \mathbb{R}^3$ are norm and direction vectors for $\boldsymbol{\phi}$, respectively, and the opposite operation is possible through *Logarithm map*. Similarly, for SE(3), the exponential map is defined as:

$$\mathbf{T} = \exp(\boldsymbol{\xi}^{\wedge}) = \begin{bmatrix} \exp(\boldsymbol{\phi}^{\wedge}) & \mathbf{J}_l\boldsymbol{\rho} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix}. \tag{21}$$

In (21), $\mathbf{J}_l$ is linear jacobian matrix of the form:

$$\mathbf{J}_l = \frac{\sin\theta}{\theta}\mathbf{I} + \left(1 - \frac{\sin\theta}{\theta}\right)\mathbf{a}\mathbf{a}^{\mathrm{T}} + \frac{1-\cos\theta}{\theta}\mathbf{a}^{\wedge}, \tag{22}$$

We use the left perturbation model [22] to derive linearization and the normal equation. Let small perturbation in $\mathfrak{se}(3)$ be $\delta\boldsymbol{\xi}$. Then, we can linearize (13) through first-order approximation as:

$$f_i(\exp(\delta\boldsymbol{\xi}^{\wedge})\mathbf{T}) \approx f_i(\mathbf{T}) + \mathbf{J}_i^T\delta\boldsymbol{\xi}, \tag{23}$$

where $\mathbf{J}_i$ is calculated as:

$$\mathbf{J}_i = \begin{bmatrix} \mathbf{I}_{3\times 3} & -(\mathbf{R}\mathbf{p}+\mathbf{t})^{\wedge} - (\mathbf{R}^{\mathrm{T}}\mathbf{q})^{\wedge} \end{bmatrix}^{\mathrm{T}} \cdot (\mathbf{n}_{\mathrm{p}} + \mathbf{n}_{\mathrm{q}}) \tag{24}$$

For simplicity, all subscripts in (13) are omitted.

Then, we can calculate $\delta\boldsymbol{\xi}$ by solving the normal equation and update T by remapping the optimized $\delta\boldsymbol{\xi}$ by the exponential map and multiplying it to T as:

$$\mathbf{T} \leftarrow \exp(\delta\boldsymbol{\xi})\mathbf{T} \tag{25}$$

IV. EXPERIMENTAL RESULT

To demonstrate the efficiency and versatility of the proposed method, we evaluated it on the KITTI odometry and Newer College dataset. In particular, we evaluated the absolute translation error [23] between the ground truth pose provided in the dataset and the trajectory estimated in the proposed method; we compared this with the results of other feature-based methods, such as A-LOAM, LeGo-LOAM [18], MULLS [20], and F-LOAM [19]. We also collected LiDAR scan data using Velodyne

TABLE II
ABSOLUTE TRANSLATION ERROR (IN METER) OF EACH METHOD WITH NEWER COLLEGE DATASET. ALL ERRORS ARE REPRESENTED AS [M]. AVERAGE TIME CONSUMPTION FOR THE SCAN ΔT OF EACH METHOD IS REPRESENTED AS [MS]. BOLD CELLS IN A SINGLE ROW PRESENT THE BEST PERFORMER FOR THAT METRIC

Sequence (Num. of frames)	quad_easy (1988)	math_easy (2160)	cloister (2787)	stairs (1190)	short_experiment (15301)	Avg. per frame [m]	ΔT [ms]
Proposed	**0.08**	**0.08**	**0.31**	**0.12**	**0.38**	**0.31**	30
A-LOAM	**0.08**	0.11	3.72	0.37	0.43	0.88	110
F-LOAM	0.22	0.13	3.43	0.13	3.22	1.59	**20**

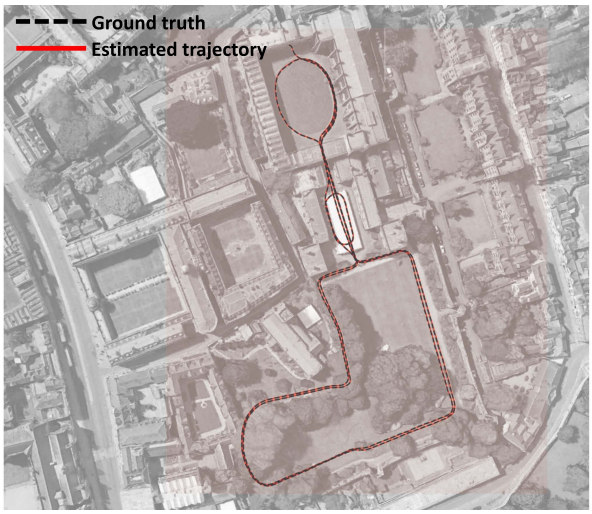

Fig. 4. Result of proposed method on *short_experiment* for the Newer College dataset and the corresponding image from Google Maps: estimated trajectory (red line) and ground truth trajectory (dashed black line).

VLP-32 and evaluated the proposed method to demonstrate that it can provide robust performance, even when the LiDAR scan is sparse.

A. KITTI Odometry Dataset

KITTI provided 21 sequences of LiDAR scan data that were obtained from a Velodyne HDL-64E mounted on a driving car and corresponding to ground truth trajectory for training sequences (0–10) acquired from GPS/IMU sensor. These sequences are collected from various environments, such as urban road, highway, and country road. We used training sequences (0–10) of the KITTI dataset for evaluation.

For comparison, we set map point resolutions for A-LOAM, Lego-LOAM, and F-LOAM in all KITTI sequences to 0.8 m and 0.4 m for planar and edge cases, respectively. In the case of MULLS, the parameters proposed in the original work are used. In the proposed method, we set the map resolution of the surface features to 0.8, the same value as the planar case for other methods.

Because the proposed method is a LiDAR odometry without loop closure, the trajectory of other methods is also estimated without pose graph optimization using loop closure. Table I shows absolute translation error [23] in meters and average time consumption to process a scan using each method. The proposed method has lowest average translation error per frame and best computational efficiency.

To present the performance of the proposed method, we show the ground truth and estimated trajectories for all training sequences of KITTI in Fig. 3. The results in this figure, indicate low drift and global consistency without pose graph optimization using loop closure.

B. Newer College Dataset

The Newer College dataset contains LiDAR scan data collected from LiDAR mounted on a hand-held stick. There are two models of LiDAR used for data collection: Ouster OS-1 64 and OS-0 128. Using this dataset is more challenging due to the instability of the sensor platform. Most of the data are collected when the platform is moving sharply and dynamically making the evaluation of the performance of the LiDAR odometry difficult. Therefore, we only used sequences with relatively few sharp motions, although the platform was still unstable during the experiment. In addition, we compared our results on this data with the results of comparable counterparts, except for MULLS and LeGo-LOAM, which are specialized for driving dataset and 6-DoF ground vehicles, respectively.

For comparison, we set the map point resolutions of A-LOAM, and F-LOAM at all sequences, except for the stairs sequence, to 0.8 m and 0.4 m for planar and edge cases, respectively. In the case of the stairs sequence, as the space is smaller than that of other sequences, the resolution was increased fourfold. The resolution of the surface features of the proposed method is set to be the same as that for the planar feature resolution of other methods.

Table II presents the results of evaluation on the Newer College dataset. *quad_easy* and *math_easy* are sequences collected in a square surrounded by buildings and sequence *stairs* is collected in narrow stairway. These environments provide sufficient plane and edge features for other LiDAR odometry methods. Therefore, all of the methods mentioned in the Table II show acceptable performance for these sequences. However, as can be seen in Fig. 2, in the *cloister* dataset obtained from a narrow corridor, the surface features are obtained from the ceiling with a quadratic surface, resulting in the failure of existing methods. By contrast, the proposed method shows robust performance even in this environment, as it aligns the features to a local quadratic surface that includes a flat plane. Furthermore, in the *short_experiment* sequence, which contains many unstructured objects, such as trees, bushes, and rocks, the proposed method performs better than other LiDAR odometry method. Fig. 4. shows the estimated and ground truth trajectory for the last sequence drawn on the corresponding Google map image. Although this sequence is quite long, it is

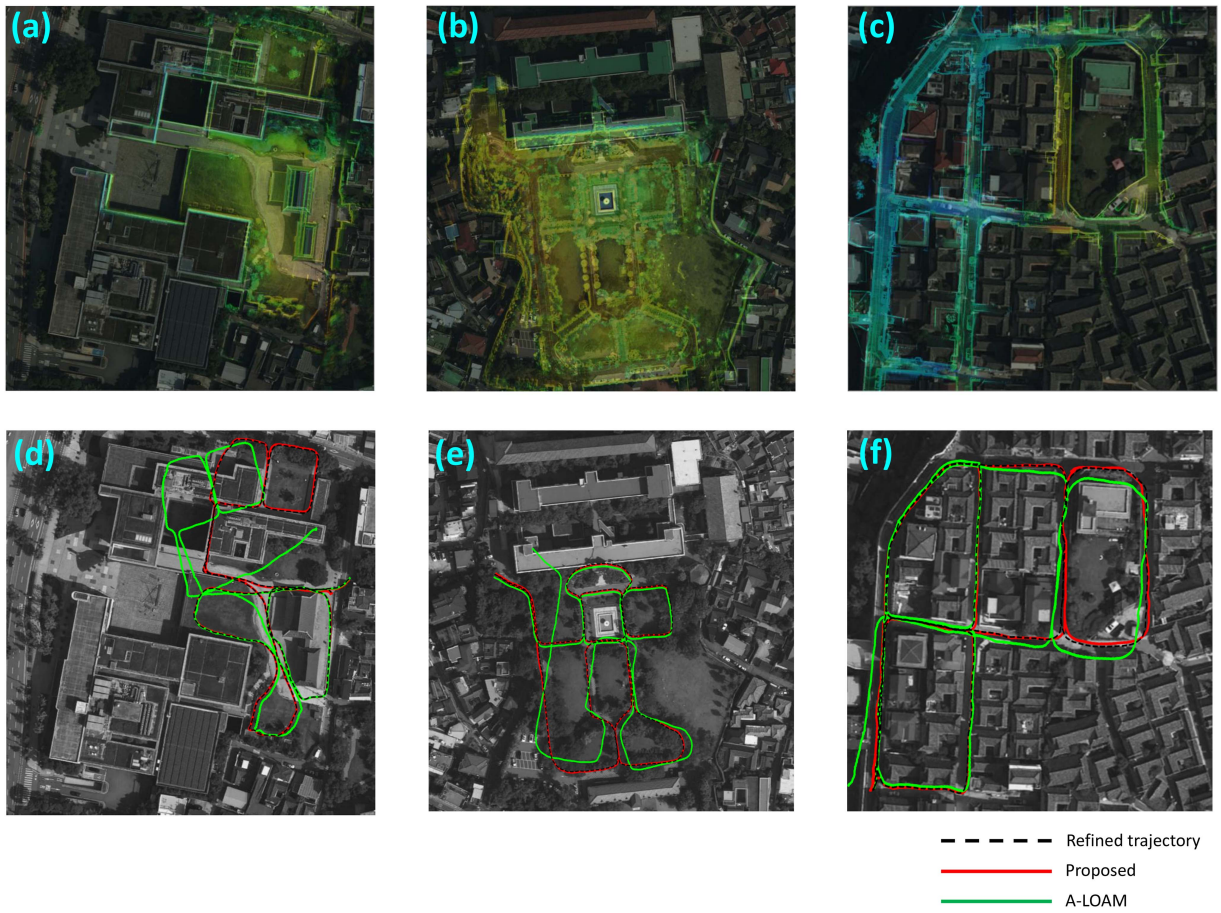

Fig. 5. Evaluation result of the proposed method with our datasets. In (a), (b), and (c), map images from the map aligned with the aggregated point cloud maps of $museum$, $park$, and $village$ sequence, respectively. (d), (e), and (f) show the refined trajectory, estimated trajectory via the proposed method and A-LOAM, and the corresponding map image.

TABLE III
ABSOLUTE TRANSLATION ERROR (IN METERS) FOR EACH METHOD WITH OUR DATASET. ALL ERRORS ARE REPRESENTED AS [M]. AVERAGE TIME CONSUMPTION FOR A SCAN ΔT OF EACH METHODS ARE REPRESENTED AS [MS]. BOLD TEXT DENOTES THE FIRST PLACE

Sequence (Num. of frames)	museum (6202)	park (7253)	village (7196)	Avg [m]	ΔT [ms]
Proposed	**1.51**	**0.71**	**1.07**	**1.08**	**50**
A-LOAM	2.94	1.75	1.23	1.92	135

clear that the proposed method shows very little drift without loop closure. All datasets except for the last sequence were collected using OS-0 128 LiDAR.

C. Our Dataset

To demonstrate the robustness of the proposed method for sparse LiDAR scans, we collected data using a KAARTA Stencil Pro, a commercial sensor platform involving Velodyne VLP-32 LiDAR, GNSS sensor, IMU, and cameras. It provides a refined trajectory obtained by a fusion of all sensors and a global optimization using loop closure detection, and we have assumed this trajectory as the ground truth. We collected three data sequences in various environments, as shown in Fig. 5 . These datasets are referred to as $museum$, $park$, and $village$. The $museum$ sequence was collected around the National Modern Art Museum of Korea. The second sequence was collected in a park containing unstructured objectives. The last sequence, was collected in the residential area of North Village of Seoul. The parameters related to map resolution are set as they are set in the KITTI dataset.

Table III and Fig. 5 present the evaluation result on our dataset. Because F-LOAM failed for all sequences, we only compared the results of our method with those of A-LOAM. We not only demonstrated that the proposed method outperforms compartments for all sequences but also that our method can provide robust LiDAR odometry performance, even where LiDAR scans are relatively sparse, as can be seen in Fig. 5.

V. CONCLUSION

In this letter, we proposed a fast and versatile LiDAR odometry method that mitigate the inconsistency between feature class and local geometry of on a map. We used local quadratic surface approximation and point-to-surface minimization to align the extracted surface features to the nearest local quadratic surface on the map. We evaluated the proposed method on public

and our datasets and demonstrated that the proposed method provides robust and efficient performance relative to the other feature-based methods. In particular, we demonstrated that this method can provide superior performance to other methods when the ambiguity of feature classifications is considerable or when the LiDAR scan is relatively sparse. In the future, we will study robust loop closure to improve global consistency and the continuous method for motion compensation to improve the scan alignment performance.

REFERENCES

[1] 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.

[2] Y. Chen and G. Medioni, "Object modeling by registration of multiple range images," in *Proc. IEEE Int. Conf. Robot. Automat.*, 1991, vol. 3, pp. 2724–2729.

[3] J. Zhang and S. Singh, "LOAM: LiDAR odometry and mapping in real-time," in *Proc. Robot.: Sci. Syst.*, 2014, pp. 1–9.

[4] A. Segal, D. Hähnel, and S. Thrun, "Generalized-ICP," in *Robotics: Science and Systems V*, J. Trinkle, Y. Matsuoka, and J. A. Castellanos, Eds. Cambridge, MA, USA: MIT Press, 2009, pp. 161–168. [Online]. Available: http://dblp.uni-trier.de/db/conf/rss/rss2009.html#SegalHT09

[5] K. Koide, M. Yokozuka, S. Oishi, and A. Banno, "Voxelized GICP for fast and accurate 3D point cloud registration," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 11054–11059.

[6] P. Biber and W. Strasser, "The normal distributions transform: A new approach to laser scan matching," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2003, vol. 3, pp. 2743–2748.

[7] K. Koide, J. Miura, and E. Menegatti, "A portable three-dimensional LiDAR-based system for long-term and wide-area people behavior measurement," *Int. J. Adv. Robot. Syst.*, vol. 16, 2019, doi: 10.1177/1729881419841532.

[8] F. Huang, W. Wen, J. Zhang, and L.-T. Hsu, "Point wise or feature wise? A benchmark comparison of publicly available LiDAR odometry algorithms in urban canyons," *IEEE Intell. Transp. Syst. Mag.*, vol. 14, no. 6, pp. 155–173, Nov./Dec. 2022.

[9] N. Gelfand, L. Ikemoto, S. Rusinkiewicz, and M. Levoy, "Geometrically stable sampling for the ICP algorithm," in *Proc. IEEE 4th Int. Conf. 3-D Digit. Imag. Model.*, 2003, pp. 260–267.

[10] S. Rusinkiewicz, "A symmetric objective function for ICP," *ACM Trans. Graph.*, vol. 38, no. 4, pp. 1–7, Jul. 2019, doi: 10.1145/3306346.3323037.

[11] 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.

[12] 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," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 4353–4360.

[13] L. Zhang, M. Camurri, and M. Fallon, "Multi-camera LiDAR inertial extension to the newer college dataset," 2021, *arXiv:2112.08854*.

[14] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in *Proc. IEEE 3rd Int. Conf. 3-D Digit. Imag. Model.*, 2001, pp. 145–152.

[15] A. W. Fitzgibbon, "Robust registration of 2D and 3D point sets," *Image Vis. Comput.*, vol. 21, no. 13/14, pp. 1145–1153, 2003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0262885603001835

[16] T. Jost and H. Hugli, "A multi-resolution ICP with heuristic closest point search for fast and robust 3D registration of range images," in *Proc. IEEE 4th Int. Conf. 3-D Digit. Imag. Model.*, 2003, pp. 427–433.

[17] G. Champleboux, S. Lavallee, R. Szeliski, and L. Brunie, "From accurate range imaging sensor calibration to accurate model-based 3D object localization," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 1992, pp. 83–89.

[18] 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.

[19] H. Wang, C. Wang, C.-L. Chen, and L. Xie, "F-LOAM: Fast LiDR odometry and mapping," *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 4390–4396.

[20] Y. Pan, P. Xiao, Y. He, Z. Shao, and Z. Li, "MULLS: Versatile LiDAR SLAM via multi-metric linear least square," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 11633–11640.

[21] N. J. Mitra, N. Gelfand, H. Pottmann, and L. Guibas, "Registration of point cloud data from a geometric optimization perspective," in *Proc. Eurograph./ACM SIGGRAPH Symp. Geometry Process.*, New York, NY, USA, Association for Computing Machinery, 2004, pp. 22–31, doi: 10.1145/1057432.1057435.

[22] T. D. Barfoot, *State Estimation for Robotics*, 1st ed. New York, NY, USA: Cambridge Univ. Press, 2017.

[23] 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.