# ECTLO: Effective Continuous-time Odometry Using Range Image for LiDAR with Small FoV

Xin Zheng and Jianke Zhu

Abstract-Prism-based LiDARs are more compact and cheaper than the conventional mechanical multi-line spinning LiDARs, which have become increasingly popular in robotics, recently. However, there are several challenges for these new LiDAR sensors, including small field of view, severe motion distortions, and irregular patterns, which hinder them from being widely used in LiDAR odometry, practically. To tackle these problems, we present an effective continuous-time LiDAR odometry (ECTLO) method for the Risley-prism-based LiDARs with non-repetitive scanning patterns. A single range image covering historical points in LiDAR's small FoV is adopted for efficient map representation. To account for the noisy data from occlusions after map updating, a filter-based point-toplane Gaussian Mixture Model is used for robust registration. Moreover, a LiDAR-only continuous-time motion model is employed to relieve the inevitable distortions. Extensive experiments have been conducted on various testbeds using the prism-based LiDARs with different scanning patterns, whose promising results demonstrate the efficacy of our proposed approach.

## I. INTRODUCTION

Light detection and ranging (LiDAR) sensors can directly obtain the accurate range measurements in various scenarios by actively emitting the laser beams, which enables them to be the essential sensors for perception and navigation. Currently, the dominant LiDAR odometry approaches [1]–[5] make use of the multi-line mechanical spinning scanners due to their simplicity and success in many robotic applications [6], [7].

With the prevalence of autonomous driving, there is a demand for consumer-grade vehicle-borne LiDAR. However, the conventional multi-line spinning LiDAR cannot fulfill the massive deployment requirements on price, size and reliability. To this end, micro-electro-mechanical systems (MEMS) and rotating prism are two major alternative techniques used in the consumer market. Both of them try to reduce their sizes by using fewer pairs of semiconductor transceivers, and a mechanism to change the light direction at high frequency in order to cover the area in the limited field of view (FoV), which makes them appropriately integrate into the car. In this paper, we focus our attention on the Risleyprism-based LiDARs [8]. Despite that recent works [9]–[12] have adopted this LiDAR into odometry task, they are still variant methods of multi-line spinning LiDAR, and cannot take full consideration of the new characteristics of prismbased LiDARs.

E-mail: {xinzheng,jkzhu}@zju.edu.cn.



(d) Velodyne HDL-64E (360°×26.8°)

Fig. 1: Spherical projection results of different LiDARs within 0.1 second of integration duration. Compared to spinning LiDAR (d), the scanning patterns of SSL (a-c) are sparse with a small FoV.

A significant property of prism-based LiDAR is its limited horizontal FoV, as shown in Fig. 1, which was previously regarded as a drawback for registration. In this paper, we find this property can be utilized for a compact map structure. Current LiDAR odometry methods [9], [13] usually maintain a local map and adopt a scan-to-map strategy for more precise registration. Conventional data structures for map representation are either KD-tree [9], [11] or voxel hash table [13]–[15], which may not be the optimal choice for LiDARs with small FoV. The fundamental problem of LiDAR odometry is the registration between two consecutive point clouds, so only the overlap region affects the final optimization. Keeping all historical points around the current center within a fixed radius is unnecessary. The region of interest lies in an area even smaller than LiDAR's FoV. Thus, we extend a 2D range image map representation for prismbased LiDAR, which has been proven accurate and efficient for spinning LiDAR in diverse scenarios [5], [16], [17]. The image map is robocentric, and only covers the historical points in LiDAR's FoV. This map structure is memory efficient and friendly for the parallel implementation.

Unfortunately, directly bringing the idea of image map representation is ineffective for LiDARs with non-repetitive scanning pattern. Since occlusions are inevitable in this 2D map structure, the map will get quite noisy with continuous updating. Another common issue for different kinds of LiDAR is motion distortions, which is more serious in this new sensor affected by non-repetitive scanning mode. Moreover, the diverse sparse scanning patterns make the feature extraction complicated as in [1], [2].

To tackle the above critical issues, we present an effective Continuous Time LiDAR Odometry (ECTLO) method for LiDAR with small FoV in this paper. Without extra sensors,

Xin Zheng and Jianke Zhu are with the College of Computer Science, Zhejiang University, Hangzhou, China, 310027.

Jianke Zhu is the Corresponding Author.



Fig. 2: Reconstruction results of our proposed approach on the data collected by Livox Mid-40. Two pictures at the bottom show the details of structured facades and unstructured streets.

a continuous-time motion model [14] is applied during pose optimization. To account for the noisy data after 2D image map updating, we employ a point-to-plane Gaussian Mixture Model scheme [18] to effectively align the scans. In this framework, we directly register raw points from LiDAR to avoid the complicated feature extraction for the diverse scanning patterns. Fig. 2 shows the mapping result using a single Livox Mid-40 with a  $38.4^{\circ}$  sparse circular pattern.

In summary, the main contributions of this paper are: 1) an effective odometry method for LiDAR with small FoV by taking advantage of range image map representation; 2) a novel robust continuous-time filter registration scheme handling motion distortions and diverse sparse scanning patterns; 3) extensive experiments on a series of challenging datasets demonstrate that our proposed approach outperforms the state-of-the-art LiDAR-only odometry methods.

## II. RELATED WORKS

Our proposed approach aims to deal with the challenge of LiDAR with small FoV in odometry task. In this section, we briefly review the most relevant studies on prism-based LiDAR odometry, motion compensation, and point set registration.

## A. Prism-based LiDAR Odometry

Depending on the prism combinations, the prism LiDARs have different patterns that are irregular and non-repetitive, as shown in Fig. 1. Comparing to the conventional multi-line LiDARs with 360-degree coverage, these LiDARs have the sparse scanning patterns with small FoVs, which raises many issues on scan registration. Lin and Zhang [9] firstly employ a prism-based LiDAR Livox Mid-40 for outdoor scanning. Liu and Zhang [19] adopt a bundle adjustment framework with Livox Horizon for structural scene.

Due to the small FoV and low density of point cloud, the conventional methods cannot handle the fast motion and jittering. To overcome the limitation of single sensor, Li et al. [10] present a tightly-coupled LiDAR-inertial odometry framework for Livox Horizon. Moreover, Xu and Zhang [11] fuse multi-sensor information by an Iterated Extended Kalman Filter using Livox Avia. To achieve the accurate registration, both of them rely on the additional sensor and LiDARs having more dense scanning patterns with large FoV.

Those methods maintain a local map in KD-Tree that store points near the current LiDAR center within a fixed radius. However, overlap regions between the consecutive scans are restricted in the sensor's FoV. It is unnecessary to keep such abundant historical points for LiDAR with small FoV. Thus, we use a 2D robocentric image in this paper for efficient representation [5].

## B. Motion Compensation

Motion distortions in LiDARs are mainly due to the continuous movement of laser beam. To address this issue, a straightforward approach is to make use of the constant velocity motion model [1], [3], which undistorts the current point cloud with velocity from previous poses or extra IMU measurements. However, the constant velocity assumption may not be valid in real-world applications with fast orientation or position changes, especially for the handheld devices. Continuous-time trajectory [20]-[22] is more accurate to represent the nature continuous motions. To account for fast motion, it requires a lot of control poses within a scan. Meanwhile, the strict continuity of spline between control poses may be sensitive to noise during optimization. These drawbacks prohibit it from the real-time applications. Recently, Dellenbach et al. [14] compensate the distortions in the conventional spinning LiDAR by interpolating the beginning pose with the end one within a scan, where the discontinuity between consecutive scans is penalized by the additional position and velocity constraints.

Generally, motion distortions are more serious in nonrepetitive scanning mode, which brings the extra difficulty for point registration. Lin and Zhang [9] segment a single scan into several pieces to reduce the motion distortions. Xu and Zhang [11] de-skew the points by propagating the relative pose back from IMU measurements. In our pipeline, we employ the continuous-time model like [14] in optimization, which is capable of undistorting the raw points in a scan without extra sensors.

#### C. Point Set Registration

Generally, LiDAR odometry is treated as a point-set registration problem in literature. Given the input point clouds perceived by LiDAR at two consecutive timestamps, Iterative Closed Point (ICP) algorithm [23] is the typical solution to align them, which updates the relative transformation iteratively until convergence. LOAM [1] is the most successful LiDAR odometry approach, which selects the feature points by computing the roughness of scattered points on each scan line. Moreover, a variant of ICP algorithm is employed to make use of both point-to-line [24] and point-to-plane loss [25], which achieves low-drift odometry with real-time performance.

The accuracy of ICP registration depends on the reliable correspondences between point sets, which is suspect



Fig. 3: Overview of our proposed ECTLO approach. Once the scan is obtained, the optimal state  $\mathbf{s} = [\mathbf{T}_{WL_b}, \mathbf{T}_{WL_e}]$  is predicted by continuous-time filter registration. The mapping module combines the input scan  $\mathcal{P}$  with the previous range map by the odometry result  $\mathbf{s}$  to form an updated range image  $\mathcal{I}_{map}$ . The memory consumption of  $\mathcal{I}_{map}$  is fixed depending on the user-defined FoV and angular resolution of image.

to noise, outliers, and occlusions [26]. To this end, the probabilistic model is usually regarded as a more robust approach to align point sets. Generalized-ICP [27] formulates the point set registration problem into a probabilistic framework, which unifies point-to-point, point-to-plane and planeto-plane ICP by covariance estimation. Similarly, Normal Distribution Transformation (NDT) [28] replaces the single point by normal distribution within a pre-defined regular voxel. These correlation-based registration methods can be interpreted as minimizing the distance between distributions.

Assuming that each point has a Gaussian variance, point cloud can be regarded as a Gaussian Mixture Model (GMM) [26], [29]. However, GMM-based methods are computationally intensive, which hinder them from real-time applications. Gao and Tedrake [18] present a probabilistic model, which computes the filter-based correspondences in E step and updates pose by Gauss-Newton algorithm in M step. In this paper, we employ filter registration to tackle the problem from image map representation, in which the permutohedral filter is replaced by an efficient patch filter on range image [5].

## **III. METHODOLOGY**

In this section, we present the details of our proposed odometry approach. Firstly, we use a single 2D range image for map representation. Then, we suggest a continuoustime FilterReg method to achieve robust registration and compensate the severe motion distortions in cost function. Moreover, the analytic Jacobians are derived for efficient optimization. Fig. 3 shows the overview of our pipeline.

## A. Preliminary

Assume that the point measurement is relative to LiDAR coordinate system  $L_t$  at timestamp t. We denote the orientation and position of LiDAR as  $T_{WL_t}$  with respect to the world frame W at timestamp t, where the world frame is  $W = L_0$  at the starting location.

In this paper, the movement of LiDAR is in special Euclidean space. To obtain an unconstrained minimization problem, we apply the twist parameterization from Lie Algebra [30], where a vector  $\boldsymbol{\xi} \in \mathbb{R}^6$  represents rigid

transformation  $\mathbf{T}_{WL} \in SE(3)$ . All operators are right-version as below

right-
$$\oplus: \mathcal{Y} = \mathcal{X} \oplus \boldsymbol{\xi} = \mathcal{X} \operatorname{Exp}(\boldsymbol{\xi}) \in \operatorname{SE}(3)$$
  
right- $\ominus: \boldsymbol{\xi} = \mathcal{Y} \ominus \mathcal{X} = \operatorname{Log}(\mathcal{X}^{-1}\mathcal{Y}) \in \mathbb{R}^{6}.$  (1)

where  $\mathcal{X}, \mathcal{Y} \in SE(3)$ .  $Log(\cdot) : SE(3) \to \mathbb{R}^6$  is the logarithmic mapping, and  $Exp(\cdot) : \mathbb{R}^6 \to SE(3)$  is its inverse exponential mapping.

## B. Range Image Map Representation

As a registration problem, the pose estimation results are dependent on the overlap region of consecutive scans. Since the point cloud from multi-line spinning LiDAR has a 360-degree horizontal FoV, previous methods [9], [11], [15] keep a local map that stores the historical points around current center with a fixed radius to achieve better scan-to-map accuracy. However, the prism-based LiDAR has a limited horizontal FoV as the camera. The region of interest is smaller than spinning LiDAR. While we have the option to choose points within LiDAR's FoV and maintain KD-Tree [1], [2] or voxel [28] map representations, the range image [5] that retains points within FoV proves to be a more computationally efficient data structure with low memory consumption.

We extend this representation from the conventional multiline spinning LiDARs [5] to small FoV LiDAR.

1) Spherical Projection: A range image is an index table  $\mathcal{I} : \mathbb{R}^2 \to \mathbb{R}^3$  that reserves the spatial relationship of 3D point set within single 2D image. Given a point  $(x, y, z)^{\top}$  relative to range image origin, its correspondence can be indexed in previous image by spherical projection  $\Pi : \mathbb{R}^3 \to \mathbb{R}^2$ 

$$\begin{bmatrix} u \\ v \end{bmatrix} = \Pi(x, y, z) = \begin{bmatrix} \left(1/2 + \arctan\left(y/x\right) \cdot f_h^{-1}\right) w \\ \left(1/2 - \arcsin\left(z/r\right) \cdot f_v^{-1}\right) h \end{bmatrix}, \quad (2)$$

where  $r = \sqrt{x^2 + y^2 + z^2}$ . The image center correspondence to the current LiDAR center. w and h are the width and height of image. The pixel length of w and h is proportional to angular resolution  $\beta$  and FoV, where  $w = \beta f_h$  and  $h = \beta f_v$ . Note that  $f_h$  and  $f_v$  are the user-defined FoV rather than the original LiDAR's, which keeps the historical points out of the sensor's FoV to facilitate robust registration.

2) Map Generation and Updating: The generation of initial range image  $\mathcal{I}_{map}$  is just the process that project points onto an empty image. Fig. 1 shows the single scan projection results of different LiDARs. For Livox Mid-40, its single scan is too sparse to register, several consecutive scans at the static position are used for initialization. If the projection pixel is already occupied by another point, the collision is resolved by selecting the nearest points. In our proposed approach, we only maintain single range image  $\mathcal{I}_{map}$ . This means that the memory consumption of map is fixed when the FoV and angular resolution of the range image are determined.

Range Image  $\mathcal{I}_{map}$  is a robocentric map representation, where the projection origin is the last successful registered LiDAR position. Once current scan can register with previous  $\mathcal{I}_{map}$ , we transfer map points to the local scan coordinate. Then, all the previous map points and current scan points are projected onto a new image. We choose the nearest points when pixels conflict. This new image  $\mathcal{I}_{map}$  is the local map for future registration.

However, occlusion is inevitable when keeping historical map points in range image representation even with complicated conflict-resolving solutions. This phenomenon makes our  $\mathcal{I}_{map}$  noisier after continuously updating. Since the odometry output of current scan is affected by the local map, it may lead to the inferior pose optimization results. We handle this issue by applying an efficient GMM-based registration method.

## C. Continuous-time Filter Registration on Range Image

In general, robocentric range image usually reflects the surface environment in LiDAR's FoV so that the noisy data may be injected into this 2D map due to occlusions after updating. Therefore, the conventional registration methods like ICP and its variants [1], [4], [9] may not be effective for this map representation. Another common issue of LiDAR is motion distortion. Motivated by the probabilistic method FilterReg [18], we extend it into a continuous-time formulation, which is more robust to noise, outliers and occlusions. Moreover, a Gaussian filtering algorithm is employed to efficiently find the correspondences on 2D range image, where the analytic Jacobians are derived for Gauss-Newton optimization.

1) Odometry Formulation: The fundamental problem of LiDAR odometry is to find a series of discrete motion parameters properly describing the LiDAR movement. Once the raw scan  $\mathcal{P}$  is obtained, the transformation between the current scan  $\mathcal{P}$  and previous map  $\mathcal{Q}$  is estimated.  $\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_M$  and  $\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_N$  are points in  $\mathcal{P}$  and  $\mathcal{Q}$ , respectively.

We make use of a scan-to-map method, where map Q stores in  $\mathcal{I}_{map}$ . With the continuous updating of the range map, it becomes quite noisy. Conventional ICP methods aim to find the exact closest point in the data association procedure, which is susceptible to noise data. To this end, we introduce an efficient GMM-based method FilterReg [18] into our pipeline. To achieve better accuracy in optimization,

we adopt point-to-plane criteria. The original FilterReg estimates the current state s of LiDAR using the EM algorithm, as follows:

**E step**: for each point in scan  $\mathcal{P}$ , compute

$$\mathbf{m}_{\mathbf{p}_{i}}^{0} = \sum_{\mathbf{q}_{j}} \mathcal{N}(\mathbf{p}_{i}(\mathbf{s}^{\text{old}}); \mathbf{q}_{j}, \Sigma_{xyz})$$
$$\mathbf{m}_{\mathbf{p}_{i}}^{1} = \sum_{\mathbf{q}_{j}} \mathcal{N}(\mathbf{p}_{i}(\mathbf{s}^{\text{old}}); \mathbf{q}_{j}, \Sigma_{xyz})\mathbf{q}_{j}$$
$$\mathbf{n}_{\mathbf{p}_{i}} = (\sum_{\mathbf{q}_{j}} \mathcal{N}(\mathbf{p}_{i}(\mathbf{s}^{\text{old}}); \mathbf{q}_{j}, \Sigma_{xyz})\mathbf{n}_{\mathbf{q}_{j}})/\mathbf{m}_{\mathbf{p}_{i}}^{0}$$
(3)

**M** step: minimize the following objective function

$$E_{\text{reg}}(\mathbf{s}) = \frac{1}{M} \sum_{\mathbf{p}_i} \frac{\mathbf{m}_{\mathbf{p}_i}^0}{\mathbf{m}_{\mathbf{p}_i}^0 + c} \text{dot} \left( \mathbf{n}_{\mathbf{p}_i}, \mathbf{p}_i(\mathbf{s}) - \frac{\mathbf{m}_{\mathbf{p}_i}^1}{\mathbf{m}_{\mathbf{p}_i}^0} \right)^2,$$
(4)

where  $c = \frac{w_i}{1-w_i} \frac{J}{M}$ , and  $w_i$  is the parameter that accounts for the ratio of outliers.  $\mathcal{N}(\cdot)$  is Gaussian distribution. Gaussian kernel  $\Sigma_{xyz} = \text{diag}(\sigma^2, \sigma^2, \sigma^2)$  is the fixed parameter, and the variance  $\sigma$  decides the weight of each map points  $\mathbf{q}_j$ and its normal vector  $\mathbf{n}_{\mathbf{q}_j}$ .  $\mathbf{p}_i(\mathbf{s})$  transfers scan points from local coordinate into world frame, which may extend to continuous-time form. Practically, it is inefficient to employ all map points in E step. Thus, we select the neighborhood points of  $\mathbf{p}_i(\mathbf{s})$  in a window for approximation, where J is the size of valid neighborhoods. The implementation details are following.

2) Implementation on Range Image: Just like the closet point searching in ICP, the E step is to compute the correspondence of scan point  $\mathbf{p}_i$ . The form of  $\mathbf{m}_{\mathbf{p}_i}^0, \mathbf{m}_{\mathbf{p}_i}^1, \mathbf{n}_{\mathbf{p}_i}$ are Gaussian Transform. Obviously, the bottleneck of this EM algorithm is how to efficiently compute thousands of the above independent items in E step. FilterReg [18] employs a customized permutohedral lattice filter to enable efficient computation without compromising accuracy. Since the point registration problem occurs in 3D space rather than the general N-dimensional space, we utilize a more efficient 2D Gaussian filter on the range image.

The advantage of range image is to retain the 3D spatial relationship within a 2D index table, where its neighbor exists in adjacent pixels. Therefore, the Gaussian Transform can be efficiently computed by image filter methods like Gaussian Blur or Bilateral Filter. If the spherical projection result of  $\mathbf{p}_i$  is  $(u_i, v_i)$ , the related map points affecting the correspondence are in adjacent pixels. Hereby, Gaussian filter is employed to compute the filter-based correspondence  $\mathbf{m}_{\mathbf{p}_i}^0, \mathbf{m}_{\mathbf{p}_i}^1, \mathbf{n}_{\mathbf{p}_i}$  in a pre-defined window  $\mathcal{W}_i$  with center  $(u_i, v_i)$ .

During scan registration, map points Q are fixed so that their normal vectors do not change. By making use of pointto-plane criteria, we convert the previous map  $\mathcal{I}_{map}$  into two temporal index tables, vertex map  $\mathcal{I}_v$  storing 3D position information, and normal map  $\mathcal{I}_n$  saving their normal vector. We estimate normals through Eigendecomposition within a Wn window on the range image. To select valid planar points, we consider those surface curvatures [5] that fall below the threshold  $\delta\sigma$ . In cases where the angle between the normal vector and the point exceeds 90°, we flip the normal direction. It is important to note that we only need to calculate the normals once before the EM procedure.

3) Motion Compensation: A scan is the accumulated point cloud during a period of time  $t \in [t_b, t_e)$ .  $t_b$  is the starting timestamp of a scan, and  $t_e$  is the end timestamp. Motion distortion occurs due to assuming that all points are sampled simultaneously. To address this issue, we employ a continuous-time model to compensate the motion distortions in filter registration.

The continuous movement of LiDAR within a scan is parameterized by the beginning of scan  $\mathbf{T}_{WL_b}$  and end of scan  $\mathbf{T}_{WL_e}$ . Therefore, the estimation target is the state  $\mathbf{s} = [\mathbf{T}_{WL_b}, \mathbf{T}_{WL_e}]$ , where s is in  $\mathbb{R}^{12}$  through the minimal representation of Lie algebra. Given a raw point measurement  $\mathbf{p}_i \in \mathbb{R}^3$  at timestamp  $t_i \in [t_b, t_e)$ , its corrected position  $\mathbf{p}_i(\mathbf{s})$  in world coordinate is computed by

$$\tau = \mathbf{T}_{WL_e} \ominus \mathbf{T}_{WL_b}$$

$$\alpha = (t_i - t_b) / (t_e - t_b)$$

$$\mathbf{T}_{WL_i} = \mathbf{T}_{WL_b} Exp(\alpha \tau)$$

$$\mathbf{p}_i(\mathbf{s}) = \mathbf{T}_{WL_i} \mathbf{p}_i$$
(5)

where  $\tau \in \mathbb{R}^6$  is the tangent space of manifold SE(3), and  $\mathbf{T}_{WL_i}$  is the LiDAR origin at timestamp  $t_i$  in world coordinate. Each 3D point  $(x, y, z)^{\top}$  is represented in homogeneous coordinates as  $\mathbf{p}_i = (x, y, z, 1)^{\top}$ . As the estimated state s is updated iteratively until the convergence, the point  $\mathbf{p}_i(\mathbf{s})$  is recomputed with the newly updated pose at each iteration.

Ideally, the beginning pose of current scan should be consistent with the end pose of previous one. Thus, the new state is initialized by its previous pose as follows

$$\begin{aligned} \mathbf{T}_{\mathrm{WL}_{b}^{n}} &= \mathbf{T}_{\mathrm{WL}_{e}^{n-1}} \\ \mathbf{T}_{\mathrm{WL}_{e}^{n}} &= \mathbf{T}_{\mathrm{WL}_{b}^{n}} (\mathbf{T}_{\mathrm{WL}_{b}^{n-1}})^{-1} \mathbf{T}_{\mathrm{WL}_{e}^{n-1}}, \end{aligned}$$
(6)

where the first state  $\mathbf{s}_0$  is set to identity. However, the optimization may diverge on the noisy data with strict consistency constraints. Therefore, we impose two soft constraints, the location consistency  $E_{\text{loc}}(\mathbf{s}) = \mathbf{r}_{\text{loc}}^{\top}\mathbf{r}_{\text{loc}}$  and velocity consistency  $E_{\text{vel}}(\mathbf{s}) = \mathbf{r}_{\text{vel}}^{\top}\mathbf{r}_{\text{vel}}$ , where the position difference is defined as  $\mathbf{r}_{\text{loc}} = \mathbf{T}_{\text{WL}_{e}^{n}} \ominus \mathbf{T}_{\text{WL}_{e}^{n-1}}$  and velocity difference between the consecutive scans is computed by  $\mathbf{r}_{\text{vel}} = (\mathbf{T}_{\text{WL}_{e}^{n}} \ominus \mathbf{T}_{\text{WL}_{b}^{n-1}}) - (\mathbf{T}_{\text{WL}_{e}^{n-1}} \ominus \mathbf{T}_{\text{WL}_{b}^{n-1}})$ . Since handheld devices exhibit complex and irregular movement compared to driving scenarios, their scan data contain significant noise. Thus we apply the state constraint in the SE(3) space, rather than solely the translation component [14].

With the above proposed motion constraints, we reformulate the previous M step in Eqn. 4 into the continuous-time form,

$$E_{\text{ct-reg}}(\mathbf{s}) = E_{\text{reg}}(\mathbf{s}) + \lambda_l E_{\text{loc}}(\mathbf{s}) + \lambda_v E_{\text{vel}}(\mathbf{s}), \quad (7)$$

where  $\lambda_l$  and  $\lambda_v$  are two regularization coefficients to balance the above the energy terms.

4) Gauss-Newton in EM Procedure: The optimization in the M step involves a least square minimization problem, which can be reformulated into the generalized form  $\sum c_i \mathbf{r} i^{\top} \mathbf{r} i$ . Here,  $c_i$  represents the weight coefficient, and  $\mathbf{r}_i$  denotes the residual of the energy term. In our proposed approach, there are three kinds of residual, including  $\mathbf{r}_{\text{reg}}^i = \mathbf{n}_{\mathbf{p}_i}^{\top}(\mathbf{p}_i(\mathbf{s}) - \mathbf{m}_{\mathbf{p}_i}^1/\mathbf{m}_{\mathbf{p}_i}^0)$ ,  $\mathbf{r}_{\text{loc}}$  and  $\mathbf{r}_{\text{vel}}$ . By the first order Taylor expansion, the residual around  $\mathbf{s}$  is approximated as follows

$$\mathbf{r}_i(\mathbf{s} + \Delta \mathbf{s}) \simeq \mathbf{r}_i(\mathbf{s}) + \frac{\partial \mathbf{r}_i}{\partial \mathbf{s}} \Delta \mathbf{s},$$
 (8)

where  $\frac{\partial \mathbf{r}_i}{\partial \mathbf{s}}$  is the Jacobian, and  $\Delta \mathbf{s} = \begin{bmatrix} \Delta \boldsymbol{\xi}_b & \Delta \boldsymbol{\xi}_e \end{bmatrix}$  is the increment. In Gauss-Newton (GN) algorithm, the optimal increment minimizing  $E_{\text{ct-reg}}(\mathbf{s})$  is found by solving the linear equation  $\mathbf{H}\Delta \mathbf{s} = -\mathbf{b}$ , where the Hessian matrix  $\mathbf{H} = \sum c_i \left(\frac{\partial \mathbf{r}_i}{\partial \mathbf{s}}\right)^\top \frac{\partial \mathbf{r}_i}{\partial \mathbf{s}}$  and  $\mathbf{b} = \sum c_i \left(\frac{\partial \mathbf{r}_i}{\partial \mathbf{s}}\right)^\top \mathbf{r}_i$ . Since  $\mathbf{T}_{\text{WL}_b}$  and  $\mathbf{T}_{\text{WL}_e}$  are represented in SE(3), the new

Since  $\mathbf{T}_{WL_b}$  and  $\mathbf{T}_{WL_e}$  are represented in SE(3), the new state s for the next E step is updated through  $\mathbf{s} = \mathbf{s}^{\text{old}} \oplus \Delta \mathbf{s}$ . Specifically, the two poses are updated as follows

$$\mathbf{T}_{\mathrm{WL}_b} \leftarrow \mathbf{T}_{\mathrm{WL}_b} \oplus \Delta \boldsymbol{\xi}_b, \mathbf{T}_{\mathrm{WL}_e} \leftarrow \mathbf{T}_{\mathrm{WL}_e} \oplus \Delta \boldsymbol{\xi}_e.$$
(9)

The EM procedure will terminate when either the maximum increment falls below the threshold  $\delta_m$  or the number of iterations exceeds 15.

The computational cost in M step is mainly dominated in calculating the Jacobian. To this end, we derive the analytic Jacobians of each residual with respect to the beginning pose  $T_{WL_b}$  and the end pose  $T_{WL_e}$  for efficient optimization. The analytic Jacobian of registration term is computed by

$$\frac{\partial \mathbf{r}_{\text{reg}}^{i}}{\partial \mathbf{s}} = \mathbf{n}_{\mathbf{p}_{i}}^{\top} \frac{\partial (\mathbf{T}_{\text{WL}_{i}} \mathbf{p}_{i})}{\partial \mathbf{T}_{\text{WL}_{i}}} \begin{bmatrix} \partial \mathbf{T}_{\text{WL}_{i}} & \partial \mathbf{T}_{\text{WL}_{i}} \\ \partial \mathbf{T}_{\text{WL}_{i}} & \overline{\partial \mathbf{T}_{\text{WL}_{i}}} \end{bmatrix} \\
\frac{\partial (\mathbf{T}_{\text{WL}_{i}} \mathbf{p}_{i})}{\partial \mathbf{T}_{\text{WL}_{i}}} = \begin{bmatrix} \mathbf{R}_{\text{WL}_{i}} & -\mathbf{R}_{\text{WL}_{i}} [\mathbf{p}_{i}]_{\times} \end{bmatrix} \\
\frac{\partial \mathbf{T}_{\text{WL}_{i}}}{\partial \mathbf{T}_{\text{WL}_{b}}} = (1 - \alpha) \mathbf{J}_{r} ((\alpha - 1) \tau) \mathbf{J}_{l}^{-1} (\tau) \\
\frac{\partial \mathbf{T}_{\text{WL}_{i}}}{\partial \mathbf{T}_{\text{WL}_{i}}} = \alpha \mathbf{J}_{r} (\alpha \tau) \mathbf{J}_{r}^{-1} (\tau),$$
(10)

As in [30],  $\mathbf{J}_r(\cdot)$  and  $\mathbf{J}_l(\cdot)$  are the right- and left- Jacobians, respectively.  $[\cdot]_{\times}$  denotes the skew symmetric matrix.

Similarly, the analytic Jacobians of motion constraints can be derived as below

$$\frac{\partial \mathbf{r}_{\text{loc}}}{\partial \mathbf{s}} = \begin{bmatrix} \mathbf{J}_r^{-1}(\mathbf{r}_{\text{loc}}) & \mathbf{0} \end{bmatrix} 
\frac{\partial \mathbf{r}_{\text{vel}}}{\partial \mathbf{s}} = \begin{bmatrix} -\mathbf{J}_l^{-1}(\boldsymbol{\tau}) & \mathbf{J}_r^{-1}(\boldsymbol{\tau}) \end{bmatrix}.$$
(11)

#### IV. EXPERIMENT

In this section, we present the details of our experiments and discuss the results of LiDAR odometry. We evaluate our proposed approach on several datasets collected by various prism-based LiDAR with different scanning patterns. An ablation study proves the effectiveness of our continuoustime filter registration module. Additionally, we present the efficiency analysis on both commodity laptop and the embedded devices. Our method is implemented in C++ with

TABLE I: Parameter Settings of Our Method

Parameter	value	Parameter	value
$f_h, f_v$ (Livox Mid-40)	$50^\circ, 50^\circ$	$\lambda_l, \lambda_v$	0.1, 0.1
$f_h, f_v$ (Livox Horizon)	$90^\circ, 30^\circ$	$\sigma$	0.25
$f_h, f_v$ (Livox Avia)	$80^\circ, 80^\circ$	$w_i$	0.2
$f_h, f_v$ (Livox Mid-70)	$80^\circ, 80^\circ$	$\beta$	10 pixels/0
$\mathcal{W}_i$	$7 \times 7$	$\delta_{\sigma}$	0.055
$\mathcal{W}_n$	$5 \times 5$	$\delta_m$	5e-4

CUDA. The experiments are performed on a laptop computer with an Intel Core i7-9750H CPU@2.60 GHz having 16GB RAM and an NVIDIA GeForce RTX 2060 GPU. Besides, we evaluate the computational time of our proposed approach on NVIDIA Jetson AGX, which are the popular embedded devices in robotics. For simplicity, we name our method as 'ECTLO'. Table I gives the parameter settings.

### A. Performance Evaluation

We compare our presented ECTLO method against the state-of-the-art LiDAR approaches with publicly available implementations, including LOAM-Livox [9]<sup>1</sup>, BALM [19]<sup>2</sup>, and Livox-mapping<sup>3</sup>.

1) Qualitative Comparison: The testing data is adopted from the previous studies [9] with a Livox Mid-40. Since the dataset does not have ground truth for validation, we select two sequences 'loop\_hku\_main' (389m) and 'loop\_hku\_zym' (290m) whose beginning and end positions are the same.

Fig. 4 plots the trajectories of each valid method. It can be clearly seen that our proposed ECTLO approach achieves the lowest drift even without the extra loop closure module, which obtains the high reconstruction quality by taking advantage of the accurate continuous-time filter registration. Livox-mapping is the official odometry package from Livox, however, its mapping results have the obvious drifts. It is worthy of mentioning that BALM totally fails to obtain the correct results with this small FoV LiDAR.

To examine the gap between LiDAR-only odometry and sensor fusion methods, we compare our proposed LiDAR only approach against the state-of-the-art LiDAR-inertial odometry FAST-LIO2 [11] in a large-scale outdoor sequence 'hkust\_campus\_01' (1524m) from R3LIVE [31] collected by Livox Avia. The end-to-end drift of ECTLO is 4.72m in contrast to 0.31m drift of FAST-LIO2<sup>4</sup>. LOAM\_Livox cannot generalize to Livox Avia, and Livox\_mapping has large drift. Moreover, we plot the mapping result in Fig. 5.

2) *Quantitative Comparison:* The Hilti SLAM Challenge Dataset [32] is a multi-sensor collection containing challenging featureless areas and varying illumination conditions. We evaluate different odometry by computing absolute trajectory error (ATE) [33] using Livox Mid-70 data. Meanwhile, SVO2 [34] and hdl\_graph\_slam [35] are two baselines of VIO and multi-line spinning LiDAR odometry, respectively. The Mid-70 Data in Lab sequence is invalid at the beginning where the sensor is too close to the wall. Thus, we select other five sequences with ground truth for evaluation.

Table II shows the ATE on Hilti SLAM Challenge dataset, where LOAM\_Livox and BALM fail to track on all sequences. Sequence 'RPG Area' and 'Basement1' have the degenerated scenarios, where data is recorded in front of a textureless wall for a long time. It is challenging for both camera and LiDAR with small FoV. It can be observed that ECTLO outperforms SVO using the visual-inertial sensor and Livox\_mapping using Livox Mid-70 at a large margin. Our result is even close to the method using 360° scanning LiDAR Ouster OS0-64.

Although FAST-LIO2 using Livox Mid-70 and IMU has better performance at outdoor sequences, it drifts significantly in the indoor basement even with an extra sensor. This reflects that our robocentric image representation is a more appropriate structure for LiDAR with small FoV in narrow space. Overall, our ECTLO achieves the lowest average ATE on the Hilti Dataset with single small FoV LiDAR.

## B. Ablation Study

The main contributions of our proposed method are continuous-time model for motion compensation and GMMbased filter registration for LiDAR odometry. To investigate the effectiveness of each module, we conduct an ablation study with various settings. The experiments are performed on three sequences with different scanning patterns, including private 'ZJU-M0', 'ZJU-H0' and public 'hku\_park\_00'. The private datasets are collected by ourselves, which are publicly available<sup>5</sup>. Table III shows the end-to-end errors of different combinations. It can be observed that the continuous-time filter registration significantly improves the odometry accuracy. Although continuous time module performs well on the easy sequences like 'ZJU-M0' and 'hku\_park\_00', it fails on the challenging 'ZJU-H0' sequence. It can be concluded that both continuous-time and GMMbased registration are essential to achieving the robust and accurate odometry results in various scenarios.

To fairly evaluate each module, we compute the ATE on three sequences of Hilti Dataset. Sequence 'RPG Area' is in middle size indoor space. Sequence 'Basement4' has low-speed motion in the structural environment and 'Construction2' has fast motion in the complex outdoor largescale scenario. Continuous-time filter registration achieves the lowest ATE in all sequences.

#### C. Evaluation on Computational Efficiency

The computational load of our approach is mainly dominated by normal estimation and filter registration. By taking advantage of the effective range image representation, the spatial relationship among points can be used to speed up the millions of latent data associations. To demonstrate the efficiency of our presented method, we evaluate the computational costs at different stages, including data upload from CPU to GPU, normal estimation, filter registration, and

<sup>&</sup>lt;sup>1</sup>https://github.com/hku-mars/loam\_livox

<sup>&</sup>lt;sup>2</sup>https://github.com/hku-mars/BALM

<sup>&</sup>lt;sup>3</sup>https://github.com/Livox-SDK/livox\_mapping

<sup>&</sup>lt;sup>4</sup>https://github.com/hku-mars/FAST\_LIO

<sup>&</sup>lt;sup>5</sup>https://github.com/kevin2431/ECTLO



Fig. 4: The trajectory and reconstruction results of different methods on public Livox Mid-40 dataset. The green line in (a) is the trajectory with an extra loop closure module. The red line indicates the result without loop closure.

	Sensor	RPG Area	Basement1	Basement4	Construction2	Campus2   Average AT
SVO [34]	Camera + IMU	1.927	0.815	2.609	2.986	8.948 3.618
hdl_graph_slam [35]	Ouster-64	0.350	0.279	0.336	0.741	0.353 0.445
Livox_mapping	Mid-70	6.043	× <sup>1</sup>	1.988	15.479	10.924 -
Fast-LIO2 [11]	Mid-70 + IMU	0.242	2.334	1.407	0.206	<b>0.091</b> 0.856
ECTLO	Mid-70	0.394	0.334	0.124	0.253	0.356 0.282

TABLE II: ATE (m) on the Hilti SLAM Challenge Dataset

<sup>1</sup> Fail registration in this sequence.

TABLE I	II: Ablati	on Study	on Our	Proposed	Approach
		/			

CT Module	GMM Registration	En	End-to-end Error (m) ATE (m)				
		ZJU-M0	ZJU-H0	hku_park_00	RPG Area	Basement4	Construction2
$\mathbf{X}^{1}$	<b>X</b> 2	× <sup>3</sup>	×	37.02	17.32	7.67	×
×	$\checkmark$	6.74	17.33	24.95	6.16	0.25	×
$\checkmark$	×	0.53	44.33	0.38	0.46	0.54	×
✓	✓	0.77	2.52	0.32	0.39	0.12	0.25

<sup>1</sup> Without continuous-time motion model.

<sup>2</sup> Using point-to-plane ICP in [5] for registration.

<sup>3</sup> Fail registration in this sequence.

map updating. Once the input point cloud is transferred to GPU memory, all the remaining steps are computed on GPU using CUDA. Additionally, we deploy our approach into the embedded devices Jetson AGX, which is wildly used in autonomous driving.

The computational time is highly affected by the total number of scattered points within a scan. For fair comparison, we set the output frequency of each LiDAR to 10 Hz. Therefore, the total number of points in single scan is 10K for Livox Mid-40 ('ZJU-M0'), and 24K for Livox Horizon ('ZJU-H0') and Avia ('ZJU-A0'). We make use of all the raw points without downsampling. As depicted in Table IV, our proposed approach runs around 480 frame per second on the commodity laptop, and 110 frame per second on Jetson AGX. In practice, we do not have to update the map and compute normal for each scan. Instead, we just need update a

TABLE IV: Evaluation on Computational Time (ms)

	Dataset	Data Upload	Normal Estimation	Filter Registration	Map Updating	Total Time
AGX	ZJU-M0	0.36	2.36	4.46	1.87	9.05
	ZJU-H0	0.53	4.50	9.33	3.12	17.48
	ZJU-A0	0.57	9.23	7.72	4.75	22.27
Laptop	ZJU-M0	0.10	0.51	1.02	0.45	2.08
	ZJU-H0	0.19	1.04	2.73	0.77	4.73
	ZJU-A0	0.23	2.19	2.17	1.08	5.67

portion of points by downsampling the raw input data. Thus, our presented scheme can achieve real-time performance for various LiDAR even on a low-end embedded device.

# V. CONCLUSION

This paper proposed an effective continuous-time odometry approach to LiDAR with small FoV by taking advantage of filter registration. We made use of the continuoustime model to compensate the motion distortions. Moreover,



Fig. 5: Our mapping result on large-scale outdoor sequence 'hkust\_campus\_01'.

GMM-based filter registration was employed to robustly align sparse point clouds to the noisy map after map updating. Additionally, we employed the efficient range image representation, which not only consumes few memory in map updating but also can be easily implemented in parallel on GPUs. We have conducted the extensive evaluations, whose promising results demonstrated that our proposed approach is very effective.

Currently, our method only takes considerations of LiDAR data, which may lead to the sensor degeneration in some cases. For future work, we will incorporate various sensors into the optimization framework.

#### REFERENCES

- [1] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in realtime." in *Robotics: Science and Systems*, vol. 2, no. 9, 2014.
- [2] T. Shan and B. Englot, "Lego-loam: Lightweight and groundoptimized lidar odometry and mapping on variable terrain," in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018, pp. 4758–4765.
- [3] J.-E. Deschaud, "ImIs-slam: scan-to-model matching based on 3d data," in 2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2018, pp. 2480–2485.
- [4] J. Behley and C. Stachniss, "Efficient surfel-based slam using 3d laser range data in urban environments." in *Robotics: Science and Systems*, 2018.
- [5] X. Zheng and J. Zhu, "Efficient lidar odometry for autonomous driving," *IEEE Robotics and Automation Letters(RA- L)*, vol. 6, no. 4, pp. 8458–8465, 2021.
- [6] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, *et al.*, "Stanley: The robot that won the darpa grand challenge," *Journal of field Robotics*, vol. 23, no. 9, pp. 661–692, 2006.
- [7] A. Reinke, M. Palieri, B. Morrell, Y. Chang, K. Ebadi, L. Carlone, and A.-A. Agha-Mohammadi, "Locus 2.0: Robust and computationally efficient lidar odometry for real-time 3d mapping," *IEEE Robotics and Automation Letters*, pp. 1–8, 2022.
- [8] Z. Liu, F. Zhang, and X. Hong, "Low-cost retina-like robotic lidars based on incommensurable scanning," *IEEE/ASME Transactions on Mechatronics*, 2021.
- [9] J. Lin and F. Zhang, "Loam livox: A fast, robust, high-precision lidar odometry and mapping package for lidars of small fov," in 2020 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2020, pp. 3126–3131.
- [10] K. Li, M. Li, and U. D. Hanebeck, "Towards high-performance solid-state-lidar-inertial odometry and mapping," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5167–5174, 2021.
- [11] W. Xu and F. Zhang, "Fast-lio: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3317–3324, 2021.
- [12] J. Lin and F. Zhang, "R3live: A robust, real-time, rgb-colored, lidarinertial-visual tightly-coupled state estimation and mapping package," *arXiv preprint arXiv:2109.07982*, 2021.

- [13] I. Vizzo, T. Guadagnino, B. Mersch, L. Wiesmann, J. Behley, and C. Stachniss, "KISS-ICP: In Defense of Point-to-Point ICP – Simple, Accurate, and Robust Registration If Done the Right Way," *IEEE Robotics and Automation Letters (RA-L)*, vol. 8, no. 2, pp. 1–8, 2023.
- [14] P. Dellenbach, J.-E. Deschaud, B. Jacquet, and F. Goulette, "Cticp: Real-time elastic lidar odometry with loop closure," in 2022 International Conference on Robotics and Automation (ICRA). IEEE, 2022, pp. 5580–5586.
- [15] C. Bai, T. Xiao, Y. Chen, H. Wang, F. Zhang, and X. Gao, "Fasterlio: Lightweight tightly coupled lidar-inertial odometry using parallel sparse incremental voxels," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4861–4868, 2022.
- [16] A. Cowley, I. D. Miller, and C. J. Taylor, "Upslam: Union of panoramas slam," in 2021 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2021, pp. 1103–1109.
- [17] C. Qu, S. S. Shivakumar, W. Liu, and C. J. Taylor, "Llol: Low-latency odometry for spinning lidars," in 2022 International Conference on Robotics and Automation (ICRA). IEEE, 2022, pp. 4149–4155.
- [18] W. Gao and R. Tedrake, "Filterreg: Robust and efficient probabilistic point-set registration using gaussian filter and twist parameterization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 095–11 104.
- [19] Z. Liu and F. Zhang, "Balm: Bundle adjustment for lidar mapping," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3184–3191, 2021.
- [20] C. Park, P. Moghadam, S. Kim, A. Elfes, C. Fookes, and S. Sridharan, "Elastic lidar fusion: Dense map-centric continuous-time slam," in 2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2018, pp. 1206–1213.
- [21] D. Droeschel and S. Behnke, "Efficient continuous-time slam for 3d lidar-based online mapping," in 2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2018, pp. 5000–5007.
- [22] P. Furgale, T. D. Barfoot, and G. Sibley, "Continuous-time batch estimation using temporal basis functions," in 2012 IEEE International Conference on Robotics and Automation. IEEE, 2012, pp. 2088–2095.
- [23] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Sensor fusion IV: control paradigms and data structures*, vol. 1611. International Society for Optics and Photonics, 1992, pp. 586–606.
- [24] A. Censi, "An icp variant using a point-to-line metric," in 2008 IEEE International Conference on Robotics and Automation. IEEE, 2008, pp. 19–25.
- [25] K.-L. Low, "Linear least-squares optimization for point-to-plane icp surface registration," *Chapel Hill, University of North Carolina*, vol. 4, no. 10, pp. 1–3, 2004.
- [26] A. Myronenko and X. Song, "Point set registration: Coherent point drift," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 12, pp. 2262–2275, 2010.
- [27] A. Segal, D. Haehnel, and S. Thrun, "Generalized-icp." in *Robotics:* science and systems, vol. 2, no. 4. Seattle, WA, 2009, p. 435.
- [28] M. Magnusson, A. Lilienthal, and T. Duckett, "Scan registration for autonomous mining vehicles using 3d-ndt," *Journal of Field Robotics*, vol. 24, no. 10, pp. 803–827, 2007.
- [29] B. Jian and B. C. Vemuri, "Robust point set registration using gaussian mixture models," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 8, pp. 1633–1645, 2010.
- [30] J. Sola, J. Deray, and D. Atchuthan, "A micro lie theory for state estimation in robotics," arXiv preprint arXiv:1812.01537, 2018.
- [31] J. Lin and F. Zhang, "R 3 live: A robust, real-time, rgb-colored, lidarinertial-visual tightly-coupled state estimation and mapping package," in 2022 International Conference on Robotics and Automation (ICRA). IEEE, 2022, pp. 10672–10678.
- [32] M. Helmberger, K. Morin, B. Berner, N. Kumar, G. Cioffi, and D. Scaramuzza, "The hilti slam challenge dataset," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7518–7525, 2022.
- [33] M. Grupp, "evo: Python package for the evaluation of odometry and slam." https://github.com/MichaelGrupp/evo, 2017.
- [34] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, "Svo: Semidirect visual odometry for monocular and multicamera systems," *IEEE Transactions on Robotics*, vol. 33, no. 2, pp. 249– 265, 2016.
- [35] K. Koide, J. Miura, and E. Menegatti, "A portable three-dimensional lidar-based system for long-term and wide-area people behavior measurement," *International Journal of Advanced Robotic Systems*, vol. 16, no. 2, p. 1729881419841532, 2019.