IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

A&B-LO: Continuous-Time LiDAR Odometry with Adaptive Non-Uniform B-spline Trajectory Representation

Yuchu Lu, Chenpeng Yao, Jiayuan Du, Chengju Liu, *Member, IEEE*
and Qijun Chen, *Senior Member, IEEE*

***Abstract*—LiDAR odometry, fused by inertial measurement units (IMU), is an essential task in robotics navigation. Unlike the mainstream methods compensate the motion distortion of LiDAR data by high frequency inertial sensors, this paper deals with the distortion with continuous-time trajectory representation, and achieved competitive performance against state-of-the-art. We propose a compact framework of LiDAR odometry with adaptive non-uniform B-spline trajectory representation to formulate it as continuous-time estimation problem. We deploy point-to-plane registration and pseudo-velocity smoothing constraints to fully utilize geometric and kinematic information of odometry. For faster convergence of optimization, analytical Jacobian of constraints is derived to solve the non-linear least squares minimization. For more efficient B-spline representation, an adaptive knot spacing technique is proposed to adjust the time interval of control poses of spline. Extensive experiments on public and realistic datasets demonstrate validation and efficiency of our system compared with other LiDAR or LiDAR-inertial methods.**

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

I. INTRODUCTION

ODOMETRY estimation is an essential component in mobile robot navigation. Over decades, increasing research interest in LiDAR odometry (LO) has been raised due to its precise range sensing capabilities as well as adaptability to diverse scenarios. By fusion of inertial measurement units (IMU), LiDAR-inertial odometry (LIO) system provides high frequency odometry and robust estimation utilizing motion compensation from IMU preintegration. Although the deployment of IMU can achieve high-frequency estimation through transform integration, its consistency is relatively poor due to the approach of superimposing IMU-integrated poses onto LIO-optimized poses. Another extreme situation is Satu dataset provided by point-lio [1], where the peak angular velocity and acceleration are 75 rad/s, 80 m/s^2 exceeds the IMU measurement range (35 rad/s, 30 m/s^2). The mismatch between the actual state and measurements will introduce extra error into the estimation.

Classic LiDAR-only odometry methods [2] [3] [4] are mainly based on discrete-time estimation, which deskews in-frame distortion using constant velocity model. Under this assumption, sudden velocity changes within a LiDAR sampling period would introduce registration error into estimation. Therefore, trajectory representation based continuous-time methods, especially piece-wise linear interpolation [5] [6] [7] (PLI) and B-spline polynomials [8] [9], has gained popularity in recent years.

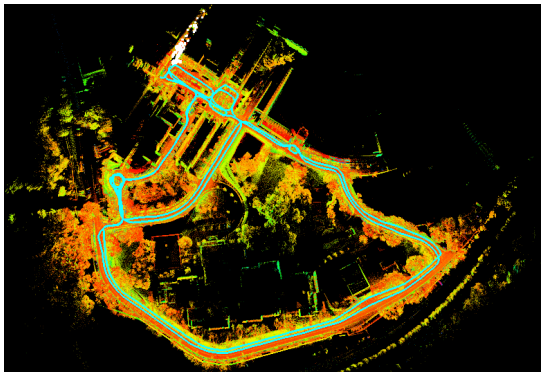

Fig. 1. Odometry and mapping results of our pipeline on MCD ntu_day_01. The point cloud is colored by intensity.

For continuous-time (CT) one, state estimation can be performed directly at the LiDAR points' sampling time. By directly performing estimation at points' sampling time, there is no need to consider in-frame deskewing anymore. The latest PLI-based LiDAR-only odometry [6] divided points into segments to be estimated with smaller time intervals than a single frame. Although impressive performances were achieved, it suffered from bad initialization when faced with aggressive motion and degeneracy caused by short of features. B-spline is widely adopted on offline multi-sensor calibration [10] due to its property of locality and continuity. However, its real-time performance on odometry [8] is unpromising. One reason is a point and its corresponding residual are computed from multiple control points of B-spline on Lie group, which increases the complexity of Jacobian computation. On the other hand, existing works on B-spline odometry employ nonlinear solver, e. g. Ceres, by which building of residuals and Jacobian are time-consuming.

In this paper, we present our LiDAR-only odometry (A&B-LO) with adaptive non-uniform B-spline representation. Illustrated as Fig. 1, the representation provides accurate continuous-time estimates to deal with motion distortion without an IMU. To fully utilize the motion prior, control point-based motion factors are included into the joint optimization, combining with adaptive point-to-plane registration and sliding window marginalization factors. The main contributions are:

- A compact LiDAR-only odometry pipeline with non-

Manuscript received: October, 29, 2025; Accepted December, 29, 2025.
This paper was recommended for publication by Editor Javier Civera upon evaluation of the Associate Editor and Reviewers' comments. This article was supported in part by the National Natural Science Foundation of China under Grant 62233013, Grant 62333017, Grant 62473295 and Grant 62403358.
The authors are with the Department of Control Science and Engineering, Tongji University, Shanghai, 201804, China, and with the State Key Laboratory of Autonomous Intelligent Unmanned Systems. luyc, 25167, dujiayuan, liuchengju, qjchen@tongji.edu.cn
Digital Object Identifier (DOI): see top of this page.

uniform B-spline representation, which deals with motion distortion, degeneracy and initialization failure in a continuous-time way without usage of IMU.
- Detailed joint optimization formulation and its corresponding analytical Jacobian to accelerate the process of NLS optimization.
- An adaptive knot spacing technique of non-uniform B-spline, which carries out dynamic adjustments considering the abundance of constraints and optimization speed, to balance the efficiency of spline representation and accuracy of our system.

II. RELATED WORK

A. LiDAR and LiDAR-inertial Odometry

LiDAR odometry could be divided into direct methods and feature-based methods according to the way of data association. Direct methods usually perform sensor pose optimization by leveraging raw measurements directly. Iterative closest point (ICP) [11], a classic direct method, carries out point-to-point registration by finding correspondence and optimization-based transformation calculation between two point clouds. Feature-based methods, firstly proposed in visual odometry, have been gaining popularity in LiDAR odometry by extracting and matching features from LiDAR scans. Zhang [2] proposed LOAM that extracts edge and plane features. LOAM motivated numerous works like Lego-LOAM [3], which extracted ground first and performs two 3-DOF decoupled estimations, and F-LOAM [4], which improved efficiency by cutting the scan-to-scan registration in LOAM. Lu [12] adopted an analytical decoupled scan matching method using plane feature. Recently, attention has been drawn back to the simple point-to-point ICP by KISS-ICP [13]. The work used the constant velocity model to process scan deskewing and dynamically adjusted the maximum distance for registration.

Our method is categorized into the direct method. We deploy a point-to-plane registration to construct geometric constraints with flexible weights and maximum searching distance inspired by KISS-ICP. However, the existing LiDAR odometry methods suffer from highly dynamic motion.

IMUs are integrated into LiDAR odometry system to address diverse weaknesses of LiDAR-only odometry. IMU measurements could be utilized to compensate the motion distortion of LiDAR scans, which practically leads to system failure facing sudden changes of orientation and acceleration. The straightforward approach [2] is to use accumulated IMU measurements to undistort LiDAR scans and provide an initial guess for registration when the constant velocity model does not work well. Another usage of IMU is to accelerate the speed of estimation convergence with a more reliable initial guess, even be treated as an odometry result when LiDAR degenerates. FAST-LIO [14] fused raw LiDAR points with IMU data in the framework of tightly coupled iterated extended Kalman filter. However, deployment of IMUs introduces extra errors to be eliminated. As mentioned in Point-LIO [1], the IMU measurements are saturated where it cannot reflect the actual motion. In this paper, we mainly focus on LiDAR-only odometry. To compensate the absence of IMU, we use marginalization factor to introduce a gauge prior and motion factor to constrain changes of pseudo-velocity in our method.

B. Continuous-time State Estimation

Classic state estimation of LiDAR odometry, e. g. Kalman filter [15] and factor graph [16] based, is usually processed once the LiDAR scans are received in a discrete-time way. A few of estimates at discrete time are created, approximation approaches shall be adopted on LiDAR points to match the states estimate's time. Another drawback of discrete-time estimation is the time desynchronization of multiple sensor fusion. Continuous-time estimation has two main classes: parametric approaches like splines and non-parametric approaches such as Gaussian processes.

In Gaussian process method, motions of robot are modeled using a Gaussian process, and states at flexible discrete time are optimized. CT-ICP [5] allowed discontinuity of trajectory between frames, and parameterized the intra-frame trajectory. Burnett [7] modeled Radar-inertial odometry with a Gaussian process motion prior. The advantages of Gaussian process are lighter computation cost than splines and more flexible states insertion comparing with discrete-time methods.

Parametric approaches tend to represent the trajectory using a set of temporal basis functions, of which B-spline is the most popular one. Kalibr [10] is the first work to use temporal basis functions on camera-IMU calibration task. Sommer [17] introduced recurrence relations for efficient time derivatives of cumulative b-splines on Lie groups, which is widely applied in visual odometry. CLIC [8] employed B-spline representation on LiDAR-inertial-camera SLAM and online calibration, however its efficiency is not satisfying. In conclusion, the upper bound of spline methods is higher and more consistent with dynamics in reality, while it suffers from higher computation cost than non-parametric approaches. Our method belongs to spline-based one. To achieve better overall performance of LiADR odometry, we used a non-uniform B-spline representation along with an adaptive knot spacing technique adjusting control poses.

III. PRELIMINARIES

A. Notations

For a quantity $\mathbf{x}$, $\check{\mathbf{x}}$ denotes a measurement of $\mathbf{x}$, and $\hat{\mathbf{x}}$ its estimate. For a matrix $\mathbf{M}$, $\mathbf{M}^\top$ denotes its transpose. For a transformation matrix $\mathbf{T} \in \mathrm{SE}(3)$, we have $\mathbf{T} = (\mathbf{R}, \mathbf{p})$, where $\mathbf{R} \in \mathrm{SO}(3), \mathbf{p} \in \mathbb{R}^3$ are the rotation and translation perspectively.

For the rotation group $\mathrm{SO}(3)$, its Lie algebra group $\mathfrak{so}(3)$ and the vector space $\mathbb{R}^3$, $(\cdot)^\wedge : \mathbb{R}^3 \to \mathfrak{so}(3)$ and $(\cdot)^\vee : \mathfrak{so}(3) \to \mathbb{R}^3$ denote mappings between Lie algebra and vector. $\mathrm{Exp}(\cdot) : \mathbb{R}^3 \to \mathrm{SO}(3)$ and its inverse $\mathrm{Log}(\cdot) : \mathrm{SO}(3) \to \mathbb{R}^3$ denote mappings between the rotation group and the vector space. We use right-plus $\oplus : \mathcal{M} \times \mathbb{R}^n \to \mathcal{M}$ and right-minus $\ominus : \mathcal{M} \times \mathcal{M} \to \mathbb{R}^n$ to denote the generalized operation on manifolds. We also use the right and left Jacobian J_r, J_l, and its inverse $J_{(\cdot)}^{-1}$ defined in [18].

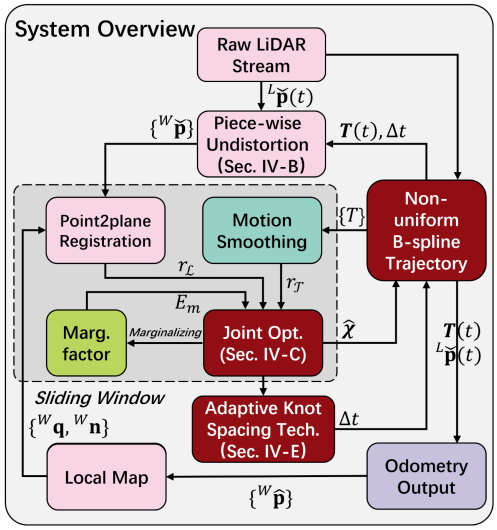

Fig. 2. System Overview of A&B-LO.

B. B-spline Trajectory Representation

B-spline is widely applied in computer vision and robotics due to its property of locality and C^{k-2} continuity for a B-spline of order k. In this work, a non-uniform cumulative B-spline is adopted to parameterize the continuous-time 6-DoF trajectory from LO. For the B-spline with order K and control points $\{\mathbf{p}_i\}_{i=1}^{K}$ at time $t \in [t_i, t_{i+1})$, the value of the spline is only depended on control points $\Phi_i \triangleq \{\mathbf{p}_i, ..., \mathbf{p}_{i+K-1}\}$. Considering computation cost [19] and compact form of the Jacobian, a split representation of two splines in $\mathbb{R}$ and $\mathrm{SO}(3)$ is presented as

$$\mathbf{R}(t) = \mathbf{R}_i \prod_{j=1}^{K-1} \mathrm{Exp}(\widetilde{\lambda}_j(t)\mathrm{Log}(\mathbf{R}_{i+j-1}^{-1}\mathbf{R}_{i+j})), \tag{1}$$

$$\mathbf{p}(t) = \mathbf{p}_i + \sum_{j=1}^{K-1} \lambda_j(t)\mathbf{p}_{i+j}, \tag{2}$$

$$\lambda_j(t) = \mathbf{M}_j^{(K)}\mathbf{u}(t), \widetilde{\lambda}_j(t) = \widetilde{\mathbf{M}}_j^{(K)}\mathbf{u}(t), \tag{3}$$

$$\mathbf{u}(t) = [1, u, ..., u^{K-1}]^\top, u = \frac{t - t_i}{t_{i+1} - t_i}. \tag{4}$$

$\mathbf{R}_i$ and $\mathbf{p}_i$ denote the rotation and translation control points at time t_i respectively. $\mathbf{u}(t), u$ are normalized temporal vector and unit in the time interval $[t_i, t_{i+1})$ respectively. $\mathbf{M}_j$ is the j^{th} row of $\mathbf{M}$. $\widetilde{\mathbf{M}}^{(K)}, \widetilde{\lambda_j}$ is the cumulative form of the blending matrix $\mathbf{M}^{(K)}, \lambda_j$. $\mathbf{M}^{(K)}$ is a constant matrix depending on order K for the uniform B-spline. For the non-uniform one, the time interval and spacing between neighboring knots is unequal. In this work, the Cubic B-spline with order 4 is adopted with the non-constant blending matrix $\mathbf{M}^{(4)}(t_i)$:

$$\mathbf{M}^{(4)} = \begin{bmatrix} m_{0,0} & 1 - m_{0,0} - m_{0,2} & m_{0,2} & 0 \\ -3m_{0,0} & 3m_{0,0} - m_{1,2} & 3m_{1,2} & 0 \\ 3m_{0,0} & -3m_{0,0} - m_{2,2} & 3m_{2,2} & 0 \\ -m_{0,0} & m_{0,0} - m_{3,2} - m_{3,3} & m_{3,2} & m_{3,3} \end{bmatrix}, \tag{5}$$

$$\widetilde{m}_{a,b} = \sum_{s=a}^{3} m_{s,b}. \tag{6}$$

$\widetilde{m}_{a,b}$ is the entry of the cumulative blending matrix $\widetilde{\mathbf{M}}^{(4)}(t_i)$ at row a, column b. $m_{a,b}$ is the entry of matrix $\mathbf{M}^{(4)}(t_i)$, which derived by knots $\{t_{i-2}, t_{t-1}, ..., t_{i+2}\}$, referred to [20] for details.

IV. METHODOLOGY

A. System Overview

The pipeline of our proposed A&B-LO is illustrated as Fig. 2. The LiDAR measurements are received as scans at a fixed frequency based on sensor configuration. The scans are buffered and sorted by their timestamp as precision under nanoseconds, and are also transformed into unified LiDAR coordinates for multi-LiDAR systems with extrinsic. The LiDAR segments with flexible knot spacing Δt are undistorted by ${}^W\check{\mathbf{p}}_t = \mathbf{R}(t){}^L\check{\mathbf{p}}_t + \mathbf{p}(t)$, where $\mathbf{T}(t) = (\mathbf{R}(t), \mathbf{p}(t))$ is sampled from the spline at time t (Section IV-B). Then undistorted segments are associated with the nearest plane under the world coordinates (generally the first LiDAR coordinates for odometry system) to formulate point-to-plane geometric constraints. With the combination of motion smoothing and marginalization constraints, the nonlinear least-square joint optimization is obtained for 6-DOF spline representation (Section IV-C). For flexible spline description, a rule to adjust the spacing of non-uniform control points is employed (Section IV-E). After Optimization, segments are transformed into the local map, while older segments are marginalized.

B. Piece-wise Preprocess

Existing works of LO suffered from in-frame motion distortion more or less. To some extent, the upper bound of LiDAR odometry estimation accuracy depends on whether the actual position of LiDAR points is recovered from LiDAR measurements with distortion. We treat LiDAR measurements as a stream based on its sampling time as accuracy of nanoseconds, then process the LiDAR segments $\{\mathbf{p}\}_i$ in the oldest time interval $[t_i, t_i + \Delta t)$ piece-by-piece, where Δt is a variable smaller than LiDAR sampling time. Rather than point-wise or frame-wise estimation, a balance between accuracy and computation efficiency is achieved: High frequency segment-wise method deskews the LiDAR points better compared to methods under a constant-velocity assumption or of the frame-wise linear interpolation; meanwhile, a large number of estimation process is saved compared to the point-wise one. New control poses are initialized by linear interpolation of the latest motion estimation. The segment points in the sliding window with size k are undistorted by ${}^W\check{\mathbf{p}}_t = \mathbf{R}(t){}^L\check{\mathbf{p}}_t + \mathbf{p}(t)$ according to the spline. Faster convergence of optimization is achieved by such a piece-wise preprocess technique.

C. LiDAR Odometry Joint Optimization based on Non-uniform B-spline Representation

Our proposed A&B-LO is formulated as the odometry estimation problem based on the Bayesian belief network model. We introduce the robot state with the continuous-time trajectory representation $\mathbf{T}_{t_0,t_n}$ in the time interval $[t_0, t_n)$, the spline $\mathbf{T}_i := \mathbf{T}_{t_0+(i-1)\Delta t, t_0+i\Delta t}$ in i-th estimation, the

LiDAR measurements $Z = \{z_m\}_{m=1,...,M}$, the local map $\mathcal{M}$ and the control input $U = \{u_n\}_{n=1,..N}$. The joint probability is given by

$$\begin{aligned} &p(\mathbf{T}, Z, \mathcal{M}, U) \\ \propto &p(\mathbf{T}_0) \prod_{n=1}^{N} p(\mathbf{T}_n|\mathbf{T}_{n-1}, u_n) \prod_{m=1}^{M} p(z_m|\mathbf{T}_{nm}, \mathcal{M}), \end{aligned} \tag{7}$$

where $p(\mathbf{T}_0)$ is the prior, $p(\mathbf{T}_n|\mathbf{T}_{n-1}, u_n)$ is the parameterized B-spline motion model, while $p(z_m|\mathbf{T}_{nm}, \mathcal{M})$ is the LiDAR measurement model. The motion and measurement models are assumed to follow the Gaussian model with zero-mean noise and covariance. The problem is converted into a non-linear least squares (NLS) formulation based on the maximum a posteriori (MAP):

$$\begin{aligned} &\arg\max_{\mathbf{T}} p(\mathbf{T}, Z, M, U) \\ =&\arg\min_{\mathbf{T}} \{\sum_{n=1}^{N} \|r_{\mathcal{T}}\|^2_{\Sigma_{\mathcal{T}}} + \sum_{m=1}^{M} \|r_{\mathcal{L}}\|^2_{\Sigma_{\mathcal{L}}}\}, \end{aligned} \tag{8}$$

where $\|r\|^2_{\Sigma} = r^\top \Sigma^{-1} r$ denotes the squared Mahalanobis distance, and $r_{\mathcal{T}}, r_{\mathcal{L}}$ denote the motion and LiDAR residuals respectively. To maintain real-time performance and memory occupancy under control, a sliding window with S segments is adopted, and an additional marginalization function is introduced to restore the marginalized information of the old segments. The odometry estimation is a reformulation by

$$\begin{aligned} \{T_j\}_{j=i-2}^{i+S+1} &= \arg\min_{\{T_j\}_{j=i-2}^{i+S+1}} E \\ &= \arg\min_{\{T_j\}} \left\{ \sum \|r_{\mathcal{T}}\|^2_{\Sigma_{\mathcal{T}}} + \sum \|r_{\mathcal{L}}\|^2_{\Sigma_{\mathcal{L}}} + E_m \right\}. \end{aligned} \tag{9}$$

Considering numerical stability and marginalization speed, we use a square root form [21] of the marginalization energy:

$$E_m = \frac{1}{2} \left\| r_m + J_m (T_0 - T_0^0) \right\|^2, \tag{10}$$

where T_0 is the marginalized control poses moving out of the window, T_0^0 is the linearization point that remains unchanged after updating r_m, J_m by $J_m = \widetilde{R}, r_m = \widetilde{r} - \widetilde{R}(T_0 - T_0^0)$. The first-estimate Jacobians are applied to prevent the destruction of null-spaces.

The formula can be organized as a weighted NLS problem f with the states $\hat{\mathcal{X}} = \{T_j\}_{j=i-2}^{i+S+1} = \{(R_j, p_j)\}_{j=i-2}^{i+S+1}$ to be estimated:

$$f = \sum \|r_{\mathcal{T}}\|^2_{\Sigma_{\mathcal{T}}} + \sum \|r_{\mathcal{L}}\|^2_{\Sigma_{\mathcal{L}}} + E_m \triangleq \left\| \hat{r}(\hat{\mathcal{X}}, \mathcal{Z}) \right\|^2_{\Sigma}, \tag{11}$$

where Σ denotes the weight matrix, $\hat{r}(\hat{\mathcal{X}})$ denotes the combined residual from three kinds of constraints. To minimize Equation (11), we update state $\hat{\mathcal{X}}$ by applying linearization at $\hat{\mathcal{X}^{it}}$ and solving for the perturbation σ^{it}:

$$f(\hat{\mathcal{X}}^{it} \oplus \sigma^{it}) = \left\| \hat{r}(\hat{\mathcal{X}}^{it} \oplus \sigma^{it}, \mathcal{Z}) \right\|^2_{\Sigma} \approx \left\| \mathbf{J}\sigma^{it} + \mathbf{r} \right\|^2_{\Sigma}, \tag{12}$$

$$\mathbf{J}^\top \Sigma \mathbf{J} \sigma^{it} = -\mathbf{J}^\top \Sigma \mathbf{r}, \tag{13}$$

$$\hat{\mathcal{X}}^{it+1} = \hat{\mathcal{X}}^{it} \oplus \sigma^j, \tag{14}$$

where it refers to the it^{th} iteration during single optimization, $\mathbf{r} = \hat{r}(\hat{\mathcal{X}}^{it}, \mathcal{Z})$, $\mathbf{J}$ is the Jacobian $\mathbf{J} = \left.\frac{\partial \hat{r}(\hat{\mathcal{X}}^{it} \oplus \sigma, \mathcal{Z})}{\partial \sigma}\right|_{\sigma \to 0}$ at the linearization point $\hat{\mathcal{X}}^j$. By introducing a relatively large marginalization prior J_{m0}, robust initialization along with efficient iteration will be achieved under the effect similar to the damped factor in the Levenberg-Marquardt (LM) optimization. It is noted that the LDLT decomposition of eigen library to solve the perturbation σ^{it}.

D. Residual and Jacobian of B-spline-related Factors

In this section, the analytical Jacobian of the LiDAR and motion factors are presented, which are crucial to the efficiency of the NLS opitimization.

1) LiDAR Factors: For any LiDAR point ${}^L\check{\mathbf{p}}_t$, that is compensated to the world coordinates ${}^W\check{\mathbf{p}}_t$ by the spline pose $\hat{\mathbf{T}}(t) = (\hat{\mathbf{R}}(t), \hat{\mathbf{p}}(t))$, its nearest neighbors of 5 points are searched to fit a plane with normal ${}^W\mathbf{n}$ by PCA. If the plane fitting succeeds, the point-to-plane registration residual is introduced:

$$r_{\mathcal{L}} = {}^W\mathbf{n}^\top \left(\hat{\mathbf{R}}(t) {}^L\check{\mathbf{p}} + \hat{\mathbf{p}}(t) - {}^W\mathbf{q} \right). \tag{15}$$

${}^W\mathbf{q}$ indicates the nearest neighboring point of ${}^W\check{\mathbf{p}}_t$ in the map. Inspired by KISS-ICP [13], an adaptive threshold technique for on-plane criterion τ and factor weight $\Sigma_{\mathcal{L}}$ is employed based on motion deviation $\Delta\mathbf{T}_i = \hat{\mathbf{T}}(t_i) \ominus \hat{\mathbf{T}}(t_{i-1})$, by which robust point-to-plane registration is achieved regardless of aggressive motion. For instance if the distance $\Delta\mathbf{d} = {}^W\check{\mathbf{p}}_t - {}^W\mathbf{q}$ exceeds the on-plane threshold τ_i, this factor is discarded. For more aggressive motion situations with larger $\Delta\mathbf{T}$, its weight should be constrained, inferring more uncertainty.

It is obvious that the LiDAR residual only relies on the control points $\hat{\mathcal{X}} = \{T_j\}_{j=i-2}^{i+S+2}$:

$$\frac{\partial r_{\mathcal{L}}}{\partial \hat{\mathcal{X}}} = \begin{bmatrix} \cdots & \frac{\partial r_{\mathcal{L}}}{\partial T_{i+j}} & \cdots \end{bmatrix}, j = 0, ..., K-1. \tag{16}$$

The chain rule is adopted to calculate the Jacobian w.r.t. $T_{i+j} = (R_{i+j}, p_{i+j})$:

$$\frac{\partial r_{\mathcal{L}}}{\partial R_{i+j}} = \frac{\partial r_{\mathcal{L}}}{\partial \hat{\mathbf{R}}(t)} \frac{\partial \hat{\mathbf{R}}(t)}{\partial R_{i+j}}, \frac{\partial r_{\mathcal{L}}}{\partial p_{i+j}} = \frac{\partial r_{\mathcal{L}}}{\partial \hat{\mathbf{p}}(t)} \frac{\partial \hat{\mathbf{p}}(t)}{\partial p_{i+j}}. \tag{17}$$

The Jacobians of $\frac{\partial r_{\mathcal{L}}}{\partial \hat{\mathbf{R}}(t)}, \frac{\partial r_{\mathcal{L}}}{\partial \hat{\mathbf{p}}(t)}, \frac{\partial \hat{\mathbf{p}}(t)}{\partial p_{i+j}}$ could be derived easily:

$$\frac{\partial r_{\mathcal{L}}}{\partial \hat{\mathbf{R}}(t)} = -{}^W\mathbf{n}^\top \hat{\mathbf{R}}(t) {}^L\check{\mathbf{p}}^\wedge, \frac{\partial r_{\mathcal{L}}}{\partial \hat{\mathbf{p}}(t)} = {}^W\mathbf{n}^\top, \frac{\partial \hat{\mathbf{p}}(t)}{\partial p_{i+j}} = \lambda_j \mathbf{I}_3. \tag{18}$$

To compute the Jacobians of $\frac{\partial \hat{\mathbf{R}}(t)}{\partial R_{i+j}}$, a recursive relation proposed by Sommer [17] is introduced:

$$\mathbf{d}_j = \mathrm{Log}(\mathbf{R}_{i+j-1}^{-1} \mathbf{R}_{i+j}), \mathbf{A}_j = \mathrm{Exp}(\widetilde{\lambda}_i \mathbf{d}_j), \tag{19}$$

$$\mathbf{P}_{K-1} = \mathbf{I}_3, \mathbf{P}_{j-1} = \mathbf{P}_j \mathbf{A}_j^\top. \tag{20}$$

TABLE I
ATE (M) AND RUNTIME (MS) RESULTS ON NTU VIRAL DATASET

Method	Sensor	eee			nya			sbs			rtp			tnp			spms			Runtime
		01	02	03	01	02	03	01	02	03	01	02	03	01	02	03	01	02	03	
CT-ICP	L_H	7.763	0.125	11.171	0.100	0.101	0.073	×	0.084	1.545	×	0.081	0.086	0.073	0.071	0.045	×	×	×	264.9
KISS-ICP	L_H	2.220	1.570	1.014	0.628	1.500	1.272	0.917	1.312	1.030	3.663	1.970	2.382	2.305	2.405	0.799	8.493	×	5.451	31.3
TRAJ-LO	L_H	0.055	0.039	0.035	0.047	0.052	0.050	0.048	0.039	0.039	0.050	0.058	0.057	0.505	0.607	0.101	0.121	×	0.103	117.4
Ours	L_H+L_V	0.046	0.030	0.030	0.040	0.044	0.041	0.037	**0.029**	**0.038**	0.047	0.056	0.054	**0.039**	**0.051**	**0.039**	0.060	0.078	0.063	53.1
CLIC	L_H+L_V+I	0.040	**0.021**	0.031	0.030	0.037	0.034	**0.033**	0.037	0.044	0.072	0.239	0.064	0.060	0.061	**0.053**	0.123	×	0.211	88.1
SLICT2	L_H+L_V+I	**0.025**	0.023	**0.024**	**0.021**	**0.023**	**0.021**	0.059	0.056	0.064	**0.025**	**0.032**	**0.028**	0.060	0.198	**0.053**	**0.041**	**0.070**	**0.042**	93.6
DLIO	L_H+L_V+I	0.109	0.126	0.154	0.213	0.251	0.234	0.197	0.201	0.215	0.330	0.257	0.517	0.112	0.138	0.084	×	×	×	18.3
IG-LIO	L_H+L_V+I	0.579	0.164	0.222	0.417	0.504	0.371	0.236	0.227	0.523	0.281	0.200	1.202	0.151	0.154	0.077	2.310	×	2.439	5.8
SR-LIO	L_H+L_V+I	0.190	0.149	0.201	0.198	0.180	0.152	0.204	0.211	0.207	0.180	0.117	0.194	0.129	0.092	0.070	×	×	×	55.3

The best results are in **bold**, the best results in the other category (LO vs LIO) are underlined. × indicates divergences.
L_H - Horizon OS1-16 LiDAR, L_V - Vertical OS1-16 LiDAR, I - IMU.

With the definition of the vector $\boldsymbol{\rho} = \mathrm{Log}(\hat{\mathbf{R}})$, the Jacobian of $\frac{\partial \hat{\mathbf{R}}(t)}{\partial R_{i+j}}$ is derived using the chain rule and **Theorem 5.6** from [17]:

$$\begin{aligned}\frac{\partial \hat{\mathbf{R}}(t)}{\partial R_{i+j}} &= \frac{\partial \hat{\mathbf{R}}(t)}{\partial \boldsymbol{\rho}} \left(\frac{\partial \boldsymbol{\rho}}{\partial \mathbf{d}_j} \frac{\partial \mathbf{d}_j}{\partial R_{i+j}} + \frac{\partial \boldsymbol{\rho}}{\partial \mathbf{d}_{j+1}} \frac{\partial \mathbf{d}_{j+1}}{\partial R_{i+j}} \right) \\ &= J_r(\boldsymbol{\rho}) \left(\frac{\partial \boldsymbol{\rho}}{\partial \mathbf{d}_j} J_r^{-1}(\mathbf{d}_j) + \frac{\partial \boldsymbol{\rho}}{\partial \mathbf{d}_{j+1}} J_r^{-1}(\mathbf{d}_{j+1}) \right) \\ &= \widetilde{\lambda}_j \mathbf{P}_j J_r(\widetilde{\lambda}_j \mathbf{d}_j) J_r^{-1}(\mathbf{d}_j) \\ &\quad - \widetilde{\lambda}_{j+1} \mathbf{P}_{j+1} J_r(\widetilde{\lambda}_{j+1} \mathbf{d}_{j+1}) J_r^{-1}(-\mathbf{d}_{j+1}). \end{aligned} \tag{21}$$

2) Motion Factors: For LIO systems, the ego motion could be constrained straightforwardly by maintaining the state of angular velocity, referenced acceleration, and biases measured by IMU. Kinematics is an intrinsic constraint that should be well-utilized. Therefore, the pseudo-motion factors based on control points are proposed in consideration of the convergence and robustness of our system:

$$r_{\mathcal{T}} = k_{i+j} \Delta T_{i+j} - k_{i+j-1} \Delta \hat{T}_{i+j-1}, \tag{22}$$

$$\Delta T_i = T_i \ominus T_{i-1} = \begin{bmatrix} R_i \ominus R_{i-1} \\ p_i - p_{i-1} \end{bmatrix}, \tag{23}$$

where $k_i = 1/(t_i - t_{i-1})$ indicates the time scale to balance the non-uniform time elapsed between control points. Noted that $\Delta \hat{T}$ represents the motion prior from the last optimization, which remains fixed during the optimization. The idea of control-point-based motion constraints rather than pose-based results from the local controllability property of B splines control points, which provides some kind of motion guidance against divergence caused by loss of tracking. At the same time, the computation cost of its Jacobians w.r.t T_{i+j}, T_{i+j-1} is cheap:

$$\frac{\partial r_{\mathcal{T}}}{\partial R_{i+j}} = \begin{bmatrix} J_r^{-1}(\mathbf{d}_j) & 0 \end{bmatrix}^\top, \qquad \frac{\partial r_{\mathcal{T}}}{\partial p_{i+j}} = \begin{bmatrix} 0 & \mathbf{1} \end{bmatrix}^\top, \tag{24}$$

$$\frac{\partial r_{\mathcal{T}}}{\partial R_{i+j-1}} = \begin{bmatrix} -J_l^{-1}(\mathbf{d}_j) & 0 \end{bmatrix}^\top, \frac{\partial r_{\mathcal{T}}}{\partial p_{i+j-1}} = \begin{bmatrix} 0 & -\mathbf{1} \end{bmatrix}^\top. \tag{25}$$

E. Adaptive Knot Spacing Technique

In this section, an adaptive knot spacing technique of splines is proposed to take advantage of the continuous-time optimization. Vandeportaele [22] adopted non-uniform B-spline to model trajectories for rolling shutter cameras facing with different types of handheld device motions. The IMU measurements and reprojection error are referred to determine the number of required control points per unit time. Inspired by [22], we proposed an adaptive knot spacing technique tailed for LiDAR odometry under diverse motions in aspects of geometric error, estimated motion and optimization process.

The knot temporal spacing $\Delta t_i = t_{i+1} - t_i \in [t_{min}, t_{max}]$ is initialized within configurable range, where t_{max} refers to the LiDAR sampling time in general. Then the knot spacing is halved or doubled $\Delta t_{i+1} \leftarrow 2\Delta t_i$ or $\Delta t_i/2$ from the perspective of the optimization process and its constraints, including the geometric and motion information. Note that the spacing is only changed once before each optimization; otherwise remains unchanged.

- STEP I: We check if the geometric point-to-plane constraints in the latest segment are sufficient. If the number of inlier $N_{inlier} < \tau_{p2p}$, the spacing is doubled for the purpose of offering more LiDAR factors to perform optimization;
- STEP II: The dynamic kinematic is considered by calculating the pseudo-acceleration $\widetilde{\alpha}_i = \Delta \mathbf{T}_{i+1}/\Delta t_i - \Delta \mathbf{T}_i/\Delta t_{i-1}$. The spacing is halved when $\|\widetilde{\alpha}\| > \tau_{hd}$, while doubled when $\|\widetilde{\alpha}\| < \tau_{ld}$;
- STEP III: The optimization process is taken into account. We halve the knot spacing for more accurate estimation if the maximum number of iterations is reached. Otherwise, we double the knot spacing for lower computation cost if the number of iterations $N_{it} < \tau_{it\_min}$.

The knot spacing remains unchanged if any of the adjustment conditions are not met. It is noted that the spacing has its range $\Delta t \in [t_{min}, t_{max}]$, and $\Delta t = t_{max}/2^a, t_{min} = t_{max}/2^b, a, b \in \mathbb{N}$ ensuring the numeric stability after long time operation.

V. EXPERIMENT

To demonstrate the effectiveness and efficiency of our A&B-LO, we implement our algorithm in C++ and conduct exhaustive experiments on datasets from various platforms, including the NTU VIRAL [23] , MCD [24], HILTI 2021 SLAM challenge [25] dataset and our dataset collected by

TABLE II
ATE (M) RESULTS ON MCD

Sequence	CT-ICP	TRAJ-LO	**Ours**	FAST-LIO2	CLIC	SLICT2
*_day_01	×	0.886	0.712	0.901	25.194	**0.498**
*_day_02	0.807	0.223	**0.167**	0.185	0.503	0.169
*_day_10	64.911	3.274	**0.591**	1.975	22.775	0.745
*_night_04	69.024	0.533	**0.326**	0.902	4.154	0.344
*_night_08	×	×	0.946	1.288	22.275	**0.817**
Sensor	L	L	L	L+I	L+I	L+I

The best results are in **bold**, the second best results are underlined. × indicates divergences.
L - Livox Mid70 LiDAR, I - IMU, * - MCD.

automated guided vehicle (AGV) under structured industrial environments. The experiments are carried out on a desktop computer running Ubuntu 20.04 with AMD Ryzen 5 3600 CPU @ 2.2 GHz and 16 GB memory. The relative knot temporal spacing hyperparameters are: initial knot spacing $\Delta t_0 = 25$ ms; $t_{min}, t_{max} = 12.5, 100$ ms; minimum number of geometric constraints $\tau_{p2p} = 10$; pseudo-acceleration threshold $\tau_{hd}, \tau_{ld} = 5.0, 1.0$ m/s^2; optimization iteration threshold $\tau_{it\_min}, \tau_{it\_max}$ = 5, 50. For general parameter tuning, t_{max} refers to the LiDAR scanning frequency (usually 10 Hz/100 ms), t_{min} is set to nearly or slightly below average runtime. τ_{hd}, τ_{ld} accords to the acceleration range during operation. $\tau_{p2p}, \tau_{it\_min}$ and τ_{it_max} could be fixed, or adjusted according to the distribution of iteration times.

A. Dataset

The NTU VIRAL dataset was captured by an uncrewed aerial vehicle (UAV) equipped with two LiDARs, two shutter cameras, multiple IMUs, and ultra-wideband (UWB) ranging units. Its sequences contain indoor and outdoor scenarios on the campus of Nanyang Technological University (NTU). The centimeter ground truth was measured from a Leica Nova MS60 MultiStation by tracking a crystal prism mounted on the top of the UAV. Multi-campus dataset (MCD) was collected at NTU from the all terrian vehicle (ATV) and at KTH and TUHH from a handheld setup (HHS) in the daytime and nighttime. Compared with NTU VIRAL, the average velocity and dynamics of MCD are relatively high. Both devices are equipped with the newer non-repetitive epicyclic LiDAR Livox Mid70 at 10 Hz and 9-axis IMU VN200 at 400 Hz, and classical cylindrical spinning OS1 LiDAR of 128 channels for ATV and 64 channels for HHS respectively. Hilti 2021 dataset covers diverse indoor and outdoor construction environments, including office, lab, basement, and parking. It was collected by a handheld platform with two LiDARs - 64 channels Ouster OS0 at 10Hz, 70°FoV Livox Mid70 at 10Hz and Analog Devices ADIS16445 IMU unit at 800 Hz. Millimeter ground truth was captured from prism tracking by the Hilti PLT 300 Total Station or optical tracking from the MoCap system.

B. Accuracy Test

For a thorough comparison of the performance of the LOs and LIOs, the following methods are selected: CT-ICP [5], KISS-ICP [13], Traj-LO [6], FAST-LIO2 [15], CLIC [8], SLICT2 [9], DLIO [26], IG-LIO [27], SR-LIO [28].

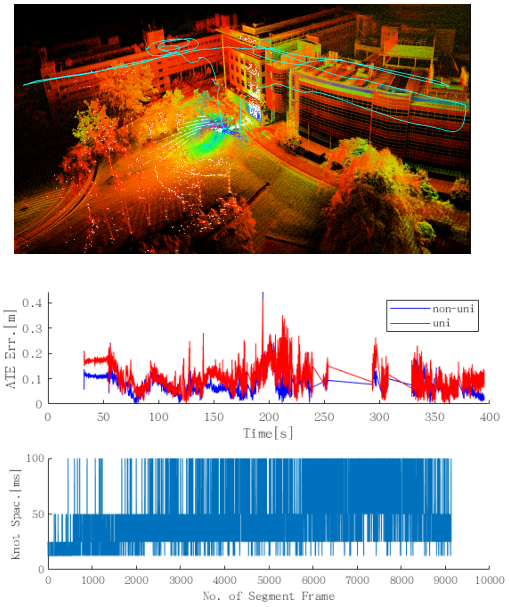

Fig. 3. Benchmark experiment on NTU VIRAL spms02. (a) Trajectory and point cloud mapping colored by intensity. (b) Top shows ATE comparison of our proposed non-uniform technique vs uniform. The ground truth is invalid from 250 to 300 s approximately. Bottom shows the dynamic knot spacing over segment frame.

TABLE III
ATE (M) RESULTS ON HILTI 2021 BENCHMARK

Sequence	CT-ICP	KISS-ICP	TRAJ-LO	**Ours**	FAST-LIO2	CLIC
RPG	0.188	0.187	0.170	**0.160**	0.182	0.394
Base	0.321	0.294	0.297	**0.120**	0.309	0.340
Base4	0.210	0.119	0.049	0.063	**0.033**	0.202
Lab	0.052	0.073	0.050	0.045	**0.035**	0.240
Cons2	0.087	0.835	0.063	**0.057**	0.066	0.327
Camp2	0.077	5.052	0.060	**0.058**	0.087	0.397
Sensor	L1	L1	L1+L2	L1+L2	L1+I	L1+I

The best results are in **bold**, the second best results are underlined.
L1 - OS0-64 LiDAR, L2 - Livox Mid70 LiDAR, I - IMU.

They are outstanding and widespread frameworks of the discrete-time (KISS-ICP, FAST-LIO2, IG-LIO, SR-LIO) and continuous-time (CT-ICP, Traj-LO, CLIC, SLICT2, DLIO) methods. KISS-ICP is a generic point-to-plane-based LO without parameter tuning, which inspires the adaptive point-to-plane registration part in this letter. FAST-LIO2 is an iterated Kalman filter-based LIO leveraging the raw LiDAR points. IG-LIO is tightly-coupled GICP-based LIO. SR-LIO performs the distortion correction by sweep reconstruction with higher frequency. For continuous-time, CT-ICP and Traj-LO are PLI-based LOs. CLIC first used B-spline to formulate the LiDAR-Inertial-Camera (LIC) odometry problem. SLICT2 is a B-spline-based LIO with the Build-Solve-Update process achieving quick convergence within few iterations. DLIO is a lightweight GICP-based LIO with continuous-time motion correction.

Our system is benchmarked against the SOTA LOs and LIOs mentioned above in terms of the root mean squared error (RMSE) of absolute trajectory error (ATE) computed by evo[1] and the average LiDAR-frame runtime. For the NTU VIRAL benchmark shown in Table I, our A&B-LO achieves

[1]https://github.com/MichaelGrupp/evo

TABLE IV
RUNTIME ANALYSIS (MS) OF A&B-LO

Sequence	t_{lid}	t_{seg}	MED	MAX	MIN	P95	P99
NTU_eee_02	48.9	19.6	19.2	58.8	1.2	41.2	56.3
MCD_night_08	28.5	11.4	11.5	31.3	0.6	18.0	25.2
Hilti_Cons2	22.2	8.98	8.0	22.4	1.1	15.7	21.5

The t_{lid} is the traditional per-lidar-frame processing time, calculated by $t_{\text{lid}} = \Sigma t/N$, where N denotes the total number of LiDAR frames. t_{seg} is the average processing time per segment, as the estimation frequency is non-constant for our system.

the best performance among LOs, even the lowest error in Sequence sbs02, sbs03, tnp01, tnp02, tnp03. Most of LIOs show competitive performance due to more reliable initial value for estimation obtained from IMU preintegration. DLIO, IG-LIO and SR-LIO, designed for long-term ground vehicle operation, perform slightly worse than spline-based methods. However, their efficiency advantage is evident. Our method outperforms those LIOs in several sequences which shows the potentials of continuous-time LO system. By comparison with PLI-based and Kalman-filter-based LOs, our non-uniform representation is proven valid and accurate. As shown in Fig. 3, we found situations of the ground truth missing (rtp01, rtp02, tnp01, spms01, spms02, spms03) to diverse extent. Official script $viral\_eval$[2] is adopted to exclude the missing part when calculating ATE. Generally, all the methods show reasonable performance with a centimeter level ATE under most sequences while scenarios (spms02) with aggressive motion and lack of features could be explored further.

To evaluate the performance of LOs and LIOs facing LiDARs with small FOV under long-time outdoor operation, we conduct experiments on the MCD benchmark presented in Table II with the livox LiDAR and IMU measurements only. Surprisingly, our A&B-LO gets the smallest ATE on most sequences. SLICT2 outperforms ours on some of the sequences with successive aggressive rotation where IMU preintegration takes effect. PLI-based LOs (CT-ICP and Traj-LO) are vulnerable to dynamic registration with a small FOV if the estimation frequency is set fixed. B-spline methods like SLICT2 and A&B-LO act better under such conditions, indicating the strength of B-spline representation.

We also conduct experiments on the Hilti 2021 benchmark to evaluate performance under low-speed indoor and outdoor construction environments. The ground truth provided online is sparse, about tens of seconds per pose. As shown in Table III, A&B-LO has the best overall performance with fusion of OS0-64 and livox, followed by FAST-LIO2 and Traj-LO. CLIC with livox and IMU has worse ATE due to perturbations by IMU caused by rough ground. With the help of LiDAR merging, our method could provide more accurate estimation and coarse 3D reconstruction on these construction sites.

C. Runtime Test

A runtime test is carried out to validate the real-time operation of A&B-LO illustrated in Fig. 4. Since the knot spacing of B-spline control points is tuned dynamically during operation, the total number of segment frame is larger than

[2]https://github.com/ntu-aris/viral_eval

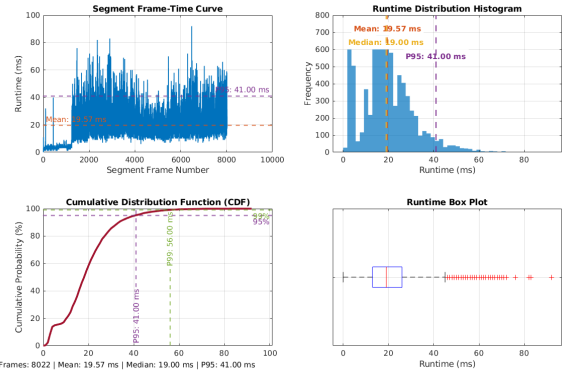

Fig. 4. Runtime curve/box plot and distribution histogram/CDF on NTU VIRAL eee02. Note that those sub-figures are plotted in terms of per-segment frame runtime.

the number of traditional LiDAR frame at 10Hz. For fair comparison, we analyze two types of average processing time t_{lid}, t_{seg}. The t_{lid} is the traditional per lidar-frame processing time in Table I, and t_{seg} is the average processing time per segment, as the estimation frequency is non-constant for our system. The spline-based methods (CLIC, SLICT2, Ours) suffered from lower running frequency, while our knot spacing technique improved it. We analyze the runtime of A&B-LO on across-dataset sequences shown in Table IV. Note that these per-frame stats are calculated in segment frame. Our t_{lid} on various sequences are all below 100ms meeting with real-time operation, and also show competitive runtime performance among the LO and LIO methods tested. Most of the runtime distribute in the range of 10ms to 25ms which demonstrates the meaningfulness of our proposed knot spacing adjustment technique. P95 of the tested sequences are under 50ms, and P99 under 100ms indicating almost every estimate is timely.

D. Ablation Study

To validate the effectiveness and efficiency of our proposed method, an ablation study is conducted on cross-dataset sequences to demonstrate the generality of our methods, which are treated as the challenging sequences from our benchmark experiments. We compare A&B-LO (Ours) with one without motion factors (Ours w/o MT), one without marginalization factors (Ours w/o MG), and our method with uniform B-spline representation (Ours w UNI) in terms of ATE of RMSE and the average runtime. For fair evaluation, we set the number of UNI control points manually, which is close to our non-UNI one. As shown in Table V, A&B-LO without the motion factors failed without motion constraints under aggressive rotation in spms02, indicating that the motion factor is irreplaceable for robust LiDAR-only odometry. In Fig. 5, we reconstruct the structured environments with high precision. Meanwhile, the weight tuning of the motion factor is essential. For low-speed and dynamic cases, appropriately small weight would increase the estimation accuracy. For highly dynamic cases, relatively big weight will ensure the robustness of systems. The ATE of A&B-LO without the marginalization factor rises significantly due to a bad initialization. The system w/o MG is easier to diverge during the initialization. The processing

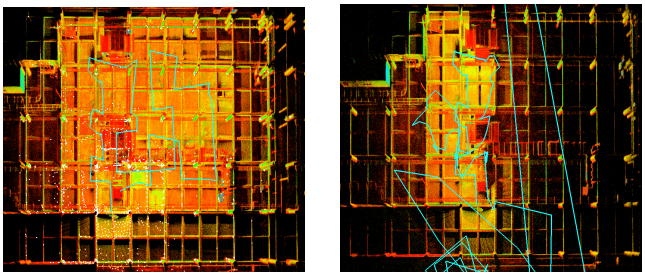

Fig. 5. Ablation odometry result under structured industrial environments captured by our ground vehicle. Left and right correspond to ours and that without the motion factor. High quality structure is reconstructed by ours, and degeneracy occurs in ours w/o MT.

TABLE V
ABLATION STUDY

Method	NTU_spms02			MCD_night08			Hilti_Cons2		
	ATE	Time	No.CP	ATE	Time	No.CP	ATE	Time	No.CP
Ours	0.078	63.2	9129	0.946	28.5	13560	0.057	22.2	12854
Ours w/o MT	×	×	×	×	×	×	0.230	21.8	-
Ours w/o MG	0.558	60.5	-	1.730	52.6	-	0.104	27.5	-
Ours w UNI	0.102	85.0	9125	1.437	40.6	13314	0.063	24.7	13300

× indicates divergences. - indicates untested. MT - motion factor, MG - marginalization factor, UNI - uniform B-spline. ATE - M, Time - ms. No. CP - Number of control points.

time is reduced by dropping of the marginalization update. As shown in Fig. 3, our method performs better in accuracy and efficiency compared to a uniform representation. With the help of the proposed knot spacing technique, A&B-LO estimates less frequently facing steady motion or quick convergence, and also when dealing with lack of LiDAR feature.

VI. CONCLUSIONS

In this letter, we introduced an efficient and robust continuous-time LiDAR odometry with non-uniform B-spline trajectory representation. With a combination of geometric, motion and marginalization constraints, our system provides SOTA estimation performance compared with the latest LOs and LIOs under divergent scenarios captured from all kinds of LiDARs. The analytical Jacobian is derived to achieve fast convergence. Both accuracy and efficiency of our A&B-LO are enhanced by adjusting the knot spacing of spline dynamically. To enhance system performance facing degeneracy, we will try to extend our system to a multi-sensor fusion one by taking advantage of the continuous-time method. The non-uniform B-spline representation could be used to evaluate and compensate trajectory planning of vehicle further.

REFERENCES

[1] D. He, W. Xu, N. Chen, F. Kong, C. Yuan, and F. Zhang, "Point-lio: Robust high-bandwidth light detection and ranging inertial odometry," *Advanced Intelligent Systems*, vol. 5, no. 7, p. 2200459, 2023.

[2] J. Zhang, S. Singh *et al.*, "Loam: Lidar odometry and mapping in real-time." in *Robotics: Science and systems*, vol. 2, no. 9. Berkeley, CA, 2014, pp. 1–9.

[3] T. Shan and B. Englot, "Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 4758–4765.

[4] H. Wang, C. Wang, C.-L. Chen, and L. Xie, "F-loam: Fast lidar odometry and mapping," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 4390–4396.

[5] P. Dellenbach, J.-E. Deschaud, B. Jacquet, and F. Goulette, "Ct-icp: Real-time elastic lidar odometry with loop closure," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 5580–5586.

[6] X. Zheng and J. Zhu, "Traj-lo: In defense of lidar-only odometry using an effective continuous-time trajectory," *IEEE Robotics and Automation Letters*, vol. 9, no. 2, pp. 1961–1968, 2024.

[7] K. Burnett, A. P. Schoellig, and T. D. Barfoot, "Continuous-time radar-inertial and lidar-inertial odometry using a gaussian process motion prior," *IEEE Transactions on Robotics*, 2024.

[8] J. Lv, X. Lang, J. Xu, M. Wang, Y. Liu, and X. Zuo, "Continuous-time fixed-lag smoothing for lidar-inertial-camera slam," *IEEE/ASME Transactions on Mechatronics*, vol. 28, no. 4, pp. 2259–2270, 2023.

[9] T.-M. Nguyen, X. Xu, T. Jin, Y. Yang, J. Li, S. Yuan, and L. Xie, "Eigen is all you need: Efficient lidar-inertial continuous-time odometry with internal association," *IEEE Robotics and Automation Letters*, 2024.

[10] 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*, 2012, pp. 2088–2095.

[11] P. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.

[12] Y. Lu, X. Du, C. Liu, and Q. Chen, "Circle fit matching: A fast analytical laser scan matching method for 2d laser scanners," *IEEE Transactions on Instrumentation and Measurement*, 2024.

[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 Robot. Autom. Lett.*, vol. 8, no. 2, pp. 1029–1036, 2023.

[14] 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.

[15] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "Fast-lio2: Fast direct lidar-inertial odometry," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2053–2073, 2022.

[16] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping," in *2020 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2020, pp. 5135–5142.

[17] C. Sommer, V. Usenko, D. Schubert, N. Demmel, and D. Cremers, "Efficient derivative computation for cumulative b-splines on lie groups," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 148–11 156.

[18] J. Sola, J. Deray, and D. Atchuthan, "A micro lie theory for state estimation in robotics," *arXiv preprint arXiv:1812.01537*, 2018.

[19] A. Haarbach, T. Birdal, and S. Ilic, "Survey of higher order rigid body motion interpolation methods for keyframe animation and continuous-time trajectory estimation," in *2018 International Conference on 3D Vision (3DV)*, 2018, pp. 381–389.

[20] K. Qin, "General matrix representations for b-splines," in *Proceedings Pacific Graphics '98. Sixth Pacific Conference on Computer Graphics and Applications (Cat. No.98EX208)*, 1998, pp. 37–43.

[21] N. Demmel, D. Schubert, C. Sommer, D. Cremers, and V. Usenko, "Square root marginalization for sliding-window bundle adjustment," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 13 260–13 268.

[22] B. Vandeportaele, P.-A. Gohard, M. Devy, and B. Coudrin, "Pose interpolation for rolling shutter cameras using non uniformly time-sampled b-splines," 01 2017, pp. 286–293.

[23] T.-M. Nguyen, S. Yuan, M. Cao, Y. Lyu, T. H. Nguyen, and L. Xie, "Ntu viral: A visual-inertial-ranging-lidar dataset, from an aerial vehicle viewpoint," *The International Journal of Robotics Research*, vol. 41, no. 3, pp. 270–280, 2022.

[24] T.-M. Nguyen, S. Yuan, T. H. Nguyen, P. Yin, H. Cao, L. Xie, M. Wozniak, P. Jensfelt, M. Thiel, J. Ziegenbein *et al.*, "Mcd: Diverse large-scale multi-campus dataset for robot perception," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 22 304–22 313.

[25] 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.

[26] K. Chen, R. Nemiroff, and B. T. Lopez, "Direct lidar-inertial odometry: Lightweight lio with continuous-time motion correction," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 3983–3989.

[27] Z. Chen, Y. Xu, S. Yuan, and L. Xie, "ig-lio: An incremental gicp-based tightly-coupled lidar-inertial odometry," *IEEE Robotics and Automation Letters*, vol. 9, no. 2, pp. 1883–1890, 2024.

[28] Z. Yuan, F. Lang, T. Xu, and X. Yang, "Sr-lio: Lidar-inertial odometry with sweep reconstruction," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024, pp. 7862–7869.