2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

CTA-LO: Accurate and Robust LiDAR Odometry Using Continuous-Time Adaptive Estimation

Yuezhang Lv, Yunzhou Zhang*, Xiaoyu Zhao, Wu Li, Jian Ning, Yang Jin [1]

***Abstract*— Accurate and robust LiDAR odometry is a crucial technology for robot localization. However, motion distortion and ranging error make it a bottleneck. Most existing methods are limited in accuracy and robustness because they simply compensate for motion distortion by constant velocity motion assumption without accurate model of ranging error. In this paper, we propose a high-precision and robust LiDAR odometry (LO), which utilizes continuous-time estimation to remove LiDAR distortion and builds the spot uncertainty model to quantify the ranging error. Generally, the number of variables in continuous-time estimation is several times higher than that in discrete-time ones, leading to insufficient constraints on the LiDAR odometry. To solve this problem, we propose a marginalization method to retain prior scans' constraints by exploiting the local support property of the B-spline. To further improve the odometry accuracy, we propose a residual adaptive weighting method and a probabilistic point cloud map based on the spot uncertainty model of LiDAR points. The experimental results show that our method outperforms state-of-the-art LiDAR odometry in accuracy and robustness.**

I. INTRODUCTION

Simultaneous Localization and Mapping (SLAM) is a crucial technology for robots to perceive and navigate in unknown environments. In recent years, LiDAR-based SLAM has been successfully applied to various robot platforms. However, the accuracy and robustness of this method face bottlenecks due to the motion distortion and ranging error of LiDAR. As an essential part of LiDAR-based SLAM, high-precision and robust odometry is a worthy topic.

LiDAR accumulates points within 100ms during motion, each point corresponding to one pose, which results in motion distortion. There are two methods for distortion compensation: discrete-time and continuous-time. Discrete-time odometry usually use the constant velocity motion assumption to compensate distortion [1]–[3]. But this assumption will fail quickly in aggressive motion scenes. Fast-lio2 [4] and Faster-LIO [5] use IMU to compensate distortion, but their methods rely on the accuracy of the IMU, which can lead to accuracy loss as the integration error increases. In continuous-time, Ct-icp [6] and ECTLO [7] utilize linear interpolation to parameterize the continuous-time trajectory to remove distortion. Other approaches are to parameterize the continuous-time trajectory with B-splines [8]–[11]. Compared with linear interpolation, B-spline is a higher-order interpolation method that can parameterize trajectory with higher complexity, as shown in Fig. 2.

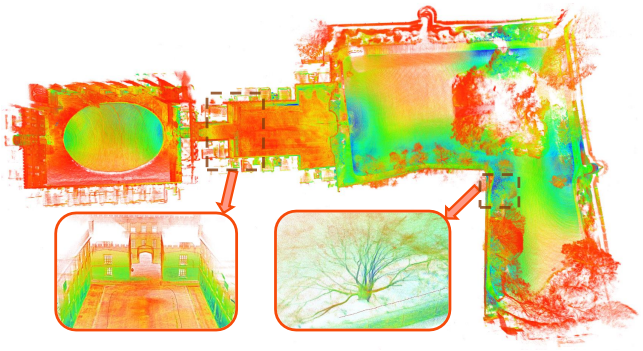

Fig. 1. CTA-LO generates detailed maps by estimating continuous-time trajectory with high accuracy and can recover details of buildings and trees well by B-spline interpolation. Colors indicate point intensities.

Nonetheless, B-spline introduces more optimization variables while improving accuracy. Only relying on the current scan's measurements will result in insufficient constraints, making it challenging to apply to LO. To solve this problem, we exploit the local support property of the B-spline to marginalize the control points of prior scans to retain the constraints of prior scans.

Existing LiDAR (Inertial) odometry [1]–[6] utilize point cloud to compose the local map and use point-to-edge or point-to-plane registration. The above algorithms have achieved good accuracy. However, they do not consider the noise of LiDAR point cloud, which leaves space for further improvement of their accuracy. In the field of geomatics, X. Chen [12] and D. Grant [13] improve the accuracy of mapping by modeling terrestrial laser scanner noise. Inspired by them, we perform spot modeling of LiDAR points to further enhance the accuracy of point cloud registration.

To this end, we propose CTA-LO, a high-precision and robustness LiDAR odometry that provides accurate localization and detailed mapping (Fig. 1). The main contributions of this paper are as follows:

- We propose a marginalization method to retain prior scans' constraints for solving the continuous-time insufficient constraints problem in LO, which improves the accuracy of compensating for the LiDAR distortion.
- We propose a residual adaptive weighting method and a probabilistic point cloud map based on the spot uncertainty model of LiDAR points, which further improves the accuracy of point cloud registration.
- The efficacy of our method is verified through extensive experiments on multiple datasets.

*The corresponding author of this paper
[1]Yuezhang Lv, Yunzhou Zhang, Xiaoyu Zhao, Wu Li, Jian Ning, Yang Jin are with College of Information Science and Engineering, Northeastern University, Shenyang 110819, China zhangyunzhou@mail.neu.edu.cn

This work was supported by National Natural Science Foundation of China (No. 61973066, 61471110) and Major Science and Technology Projects of Liaoning Province(No. 2021JH1/10400049).

II. RELATED WORK

In this section, we review the works on motion distortion and point cloud registration in LiDAR (Inertial) odometry.

A. Motion Compensation

Due to the characteristics of LiDAR continuous scanning, the point cloud will be distorted during the motion. LOAM [1], LeGo-LOAM [2], and F-LOAM [3] usually utilize a constant velocity motion model to compensate for the distortion of the current scan and then perform rigid registration. FAST-LIO2 [5] and LIO-SAM [14] use IMU to correct distortion, which rely on the correct pose prediction by IMU within 100ms. In the above methods, the point cloud rigid transformation remains constant during the optimization process, ignoring the characteristics of continuous scanning of LiDAR.

Unlike the rigid registration of point cloud, the non-rigid registration [10] based on continuous-time can correct the distortion while optimizing the pose. CT-ICP [6] and ECTLO [7] assume the pose at the beginning of the LiDAR scan is not equal to the pose at the end of the previous scan, then utilize linear interpolation within scan to compensate for LiDAR distortion. However, this assumption is inconsistent with the continuous scanning property of LiDAR, and linear interpolation does not effectively represent continuous-time trajectory. [8] and [9] use B-spline to interpolate keyframe poses to compensate for distortion in the local map. CLINS [10] and Elastic LiDAR Fusion [11] utilize B-spline to parameterize the trajectory and use the original measurements of LiDAR and IMU to estimate the current scan's control points. Compared with linear interpolation, B-spline is more consistent with the continuous scanning characteristics of LiDAR and can parameterize continuous-time trajectory with higher complexity. Their experiments show that the accuracy is improved compared with the discrete-time optimization. However, the above algorithms [10], [11] require additional sensors (IMU, wheels) to provide other constraints to avoid degradation, which is unsuitable for LiDAR odometry.

B. Point Cloud Registration

Point cloud registration based on ICP [15] is widely used in LiDAR odometry. LOAM [1] introduces feature extraction, which searches the k nearest points in the map and minimizes the distance of point-to-edge and point-to-plane in each iteration. Some subsequent works [2]–[6] inherit this framework. FAST-LIO2 [4] uses only plane features for point cloud registration and proposes ikd-Tree data structure, which achieves advanced results. Based on FAST-LIO2, Faster-LIO [5] proposes iVox, which further improves the speed of the k nearest neighbor search. The significant drawback of the above methods is that they ignore the noise of the LiDAR points.

SuMa [16] and SuMa++ [17] utilize the surfel map and update the confidence level of the existing surfel based on the angle and distance of the LiDAR measurements. The current scan is projected onto the image from spherical coordinates for data association, and the model view of the surfel map is rendered using OpenGL. But this approach relies on the GPU for rendering operations and loses 3D information due to the distance image. VoxelMap [18] utilizes voxels to represent the map, modeling the uncertainty caused by measurement noise and pose estimation error. Still, one drawback of voxel map is that they require good initial alignment [19].

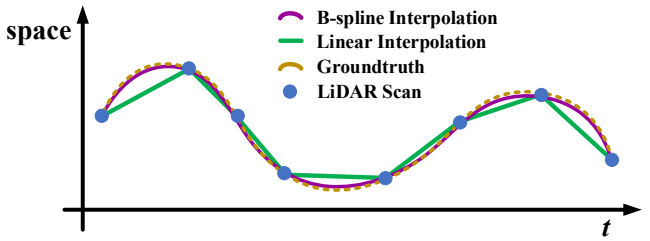

Fig. 2. Comparison of linear interpolation and B-spline interpolation. B-spline can parameterize motion trajectory of higher complexity and linear interpolation has obvious errors.

III. METHODOLOGY

In this section, we describe our LiDAR odometry method in detail. First, we model the uncertainty of the LiDAR points. Based on this, we propose a residual weighting formula and a probabilistic point cloud map (See III-B). Then, we compensate for the distortion using continuous-time and propose a marginalization strategy that preserves the prior scan constraints (See III-C). Fig. 3 shows an overview of our approach.

A. Preliminary

We use the rotation ${}^A_B\boldsymbol{R} \in SO(3)$ and translation ${}^A_B\boldsymbol{p} \in \mathbb{R}^3$ to denote the 6-DoF rigid transformation. Assume the LiDAR frame as $\{L\}$, the first LiDAR frame as the global frame $\{W\}$. $\mathrm{Exp}(\cdot)$ denotes the mapping of the Lie algebra to the Lie group, $\mathrm{Log}(\cdot)$ is its inverse operation.

B. Error ellipse model of LiDAR points

1) Spot error entropy model

Due to the divergence angle of the LiDAR beam, a spot is formed when the LiDAR beam project onto the object's surface [20]. The reflection point of the LiDAR may be located anywhere in the spot, so the shape of the spot reflects the uncertainty of the LiDAR point. The spot's geometry depends on the distance between LiDAR and the object surface and the incident angle relative to the LiDAR, as shown in Fig. 4. The incidence angle α can be defined by the following equation:

$$\alpha = cos^{-1}(\frac{\boldsymbol{P} \cdot \boldsymbol{N}}{|\boldsymbol{P}||\boldsymbol{N}|}) \tag{1}$$

where $\boldsymbol{P}$ is the beam direction and $\boldsymbol{N}$ is the plane normal vector.

When the beam is projected perpendicularly to surface, the incident angle α is 0. In such a case, the spot projected by the LiDAR onto the object's surface is circular, the spot diameter can be expressed as:

$$d = 2\rho \cdot tan(\beta/2) + D_0 \tag{2}$$

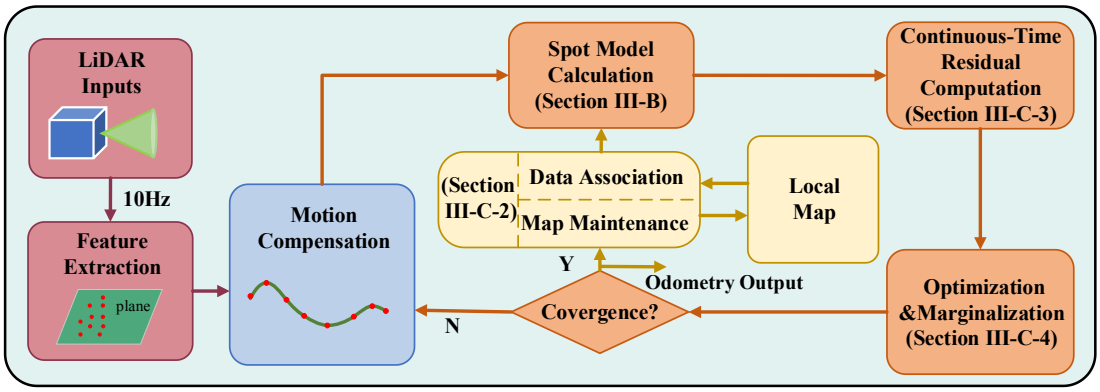

Fig. 3. Overview of our proposed CTA-LO approach. Once the scan is obtained, the feature points are associated with the local map and the spot probability model is calculated. The control points are estimated by continuous-time optimization with marginalization. After convergence, the uncertainty coefficients of the current scan are recovered and the current scan is added to the local map.

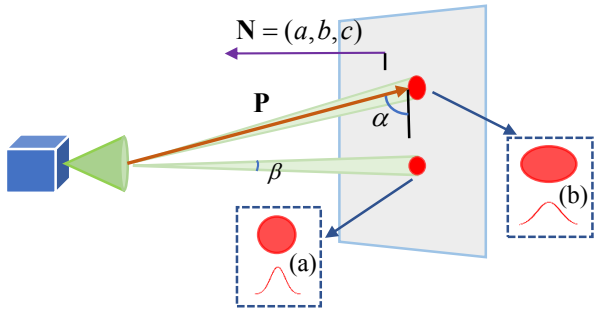

Fig. 4. Scanning geometry diagram. Beam divergence angle, plane normal vector, and beam direction are β, $\boldsymbol{N}$, and $\boldsymbol{P}$, respectively. (a) is the spot shape of vertically incident($\alpha = 0$). (b) is the spot shape at an incidence angle α.

where β denotes the beam divergence angle, D_0 denotes the diameter of the transmitting aperture, and ρ represents the distance from the LiDAR to the object surface.

When the incident angle α is not 0, the spot will appear as an ellipse, where the short axis of the ellipse is independent of the incident angle and can be expressed by d in equation (2). The long axis D can be described by the following equation:

$$D = \frac{2\rho \cdot tan(\beta/2) + D_0}{cos\alpha} \tag{3}$$

The area of the LiDAR spot can be calculated as:

$$S_\alpha = \pi \cdot \frac{1}{2}D \cdot \frac{1}{2}d = \frac{\pi d^2}{4cos\alpha} \tag{4}$$

Since the reflection point of the LiDAR is located at any position within the spot, the probability of the LiDAR position within the ellipse is equal. Suppose u is a point in the direction of the long axis (x-direction) of the spot ellipse and v is a point in the direction of the short axis (y-direction) of the spot ellipse. The probability density function of the LiDAR spot [21] is:

$$P(u,v) = \begin{cases} \frac{4}{\pi Dd} & , \frac{u^2}{D^2} + \frac{v^2}{d^2} \le 1 \\ 0 & , \frac{u^2}{D^2} + \frac{v^2}{d^2} > 1 \end{cases} \tag{5}$$

According to the probability density function of the LiDAR spot, the error entropy of the LiDAR spot [22] can be expressed by the following equation:

$$S = \frac{1}{2}e^{-\iint P(u,v)\ln P(u,v)dxdy} = \frac{\pi d^2}{8\cos\alpha} \tag{6}$$

2) Uncertainty of LiDAR Point

Error entropy is a function of the error standard deviation of measurement, which can quantify the uncertainty of measurement. Equation (6) indicates that the LiDAR noise is proportional to the square of the distance and inversely proportional to the cosine of the incident angle α. To improve the accuracy of point cloud registration, LiDAR points uncertainty formula based on the spot model is proposed below.

The effect of the incident angle on the LiDAR points noise is expressed by the coefficient $c(\alpha)$, defined as follows:

$$c(\alpha) = \cos\alpha, \alpha \in [0°, 90°] \tag{7}$$

Assuming that ρ_{min} is the minimum range of LiDAR ranging, and ρ_{max} is the maximum range of LiDAR ranging. The effect of scanning distance on the LiDAR points noise is expressed by the coefficient $c(\rho)$, which can be defined as follows:

$$c(\rho) = \frac{\rho_{max}^2 - \rho^2}{\rho_{max}^2 - \rho_{min}^2}, \rho \in [\rho_{\min}, \rho_{\max}] \tag{8}$$

Considering the effects of distance and incidence angle together, the uncertainty coefficient of the LiDAR points is:

$$c_p = c(\rho) \cdot c(\alpha) \tag{9}$$

Since the uncertainty coefficient represents the confidence level of each lidar point, we do not need to focus on the parameters of the lidar. We model the uncertainty coefficient of the point cloud for the following two purposes:

- During data association, the plane parameters are calculated according to the uncertainty of the map points (equation (14)).
- During optimization, the residual weights are adaptively adjusted according to the uncertainty of the LiDAR points in the current scan (equation (17)).

C. Continuous-Time Odometry

Since LiDAR is a non-instantaneous observation sensor, each point is observed from a unique pose. To reduce the motion distortion of the point cloud, we use B-spline to parameterize the 6-DoF trajectory, which is denoted as the 3D position $\boldsymbol{p}(t) \in \mathbb{R}^3$ and 3D rotation $\boldsymbol{R}(t) \in SO(3)$ with uniform B-spline, respectively. Defining t_{start} as the start time of the spline and t_{end} as the end time of the spline, we can parameterize the trajectory by control points and interpolate the pose by the timestamp $t \in [t_{start}, t_{end}]$. Conversely, the optimal control points can be derived from the LiDAR measurements.

1) B-spline Trajectory Representation

The B-spline is determined by the order k, $N+1$ control points $\{x_0, x_1, \ldots, x_N\}$, M node vectors $\{t_0, t_1, \ldots, t_M\}$, and $M = N + k + 1$. The B-spline is defined as follows:

$$x(t) = \sum_{i=0}^{N} B_{i,k}(t) x_i \tag{10}$$

where t is time variable, and $B_{i,k}$ can be given by the De Boer-Cox recurrence relation [23].

We use the cumulative B-spline method to represent the trajectory in continuous-time. To simplify the calculation, the B-spline equation for the position at time $t \in [t_i, t_{i+1})$ becomes:

$$\boldsymbol{p}(u) = \boldsymbol{p}_i + \sum_{j=1}^{k-1} \lambda_i(u) \cdot (\boldsymbol{p}_{i+j} - \boldsymbol{p}_{i+j-1}) \tag{11}$$

where $u = (t - t_i)/(t_{t+1} - t_i)$, $\boldsymbol{p}_i$ is the control points of $\boldsymbol{p}(u)$, and the coefficient $\lambda_i(u)$ is the i-th row vector of the matrix $\tilde{\boldsymbol{B}}_k$ that depends only on the order of the B-spline [24].

Extending equation (11) to the Lie group, the cumulative B-spline of SO(3) is defined as:

$$\boldsymbol{R}(u) = \boldsymbol{R}_i \cdot \prod_{j=1}^{k-1} \mathrm{Exp}\left(\lambda_i(u) \cdot \mathrm{Log}\left(\boldsymbol{R}_{i+j-1}^{-1} \cdot \boldsymbol{R}_{i+j}\right)\right) \tag{12}$$

where $\boldsymbol{R}_i$ is the control points of $\boldsymbol{R}(u)$.

2) Data Association and Probabilistic Map Maintenance

In terms of data association, to play the role of the spot model, we only extract plane feature points through the curvature of the LiDAR points [14]. To achieve data association between the current feature points and the local map, we transform the feature point of the current scan to the global frame using the continuous-time trajectory and search for the 10 nearest neighbor points in the local map. Suppose that each plane consists of 10 feature points $\boldsymbol{p}_i = [x_i, y_i, z_i]_{(i=1,\ldots,N)}$ and the uncertainty coefficient of each point is c_i. The plane equation is as follows:

$$ax + by + cz + d = 0, \boldsymbol{A}\bar{\boldsymbol{p}}_i = 0 \tag{13}$$

where $\bar{\boldsymbol{p}}_i$ is the homogeneous representation of $\boldsymbol{p}_i$ and $\boldsymbol{A} = [a, b, c, d]$ is the plane parameter. $\boldsymbol{A}$ can be obtained by solving the following linear least square problem:

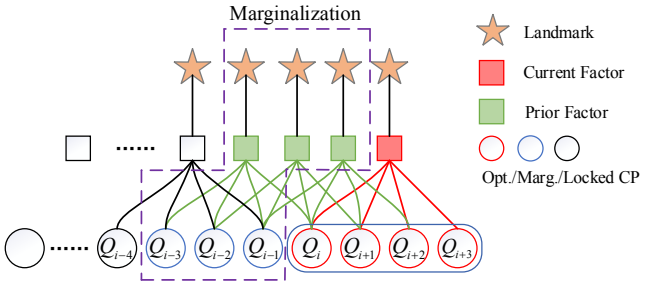

Fig. 5. Factor graph of our system. The trajectory of the current scan (red box) can be parameterized by the optimization control point (Opt. CP). The prior factors (green box) can also constrain the optimization control points. During optimization, we add the current and prior factors to the optimization and discard marginalization control points (Marg. CP) using the Schur complement.

$$\boldsymbol{A} = argmin \sum_{i=0}^{N} (c_i \boldsymbol{A} \boldsymbol{p_i})^2 \tag{14}$$

3) Residual Construction in Continuous-Time

We utilize the third-order uniform B-spline to parameterize the LiDAR trajectory. Assume $\{{}^L\boldsymbol{p}_j, j = 1, \ldots, m\}$ as the points under the k-th LiDAR frame L, ${}^W_L\boldsymbol{R}(t)$ and ${}^W_L\boldsymbol{p}(\boldsymbol{t})$ can be interpolated by the timestamp t of the feature points. After projecting to the global frame using the LiDAR pose ${}^W_L\boldsymbol{R}(t), {}^W_L\boldsymbol{p}(t)$, the feature point should lie exactly on a plane patch in the local map, the LiDAR optimization factor can be constructed as follows:

$$r_l = {}^W\boldsymbol{u}_j^T ({}^W_L\boldsymbol{R}(t) {}^L\boldsymbol{p}_j + {}^W_L\boldsymbol{p}(t)) + d = 0 \tag{15}$$

where ${}^W\boldsymbol{u}_j = [a, b, c]^T$ is the normal vector of the corresponding plane patch and can be calculated from Equation (14).

We slightly assume that the angular acceleration and the linear acceleration are zero, making the trajectory as smooth as possible. The smooth motion factor is as follows:

$$r_\alpha^R = {}^W_L\ddot{\boldsymbol{R}}(t) = 0, r_{acc}^p = {}^W_L\ddot{\boldsymbol{p}}(t) = 0 \tag{16}$$

We empirically give a small weight to the smooth motion factor to avoid the loss of accuracy.

4) Optimization with Marginalization

For the convenience of representation, we define the composite control point $\boldsymbol{Q}_i = \{\boldsymbol{R}_i, \boldsymbol{p}_i\}$ to represent the control points, including rotation and position control point. The accumulation time of the LiDAR scan is 0.1s, so we set the time interval of nodes to 0.1s.

Assume that the timestamp of the current scan lies in the $[u_i, u_{i+1})$ interval, we add a new control point $\boldsymbol{Q}_{i+3}$ to the B-spline when the current scan is obtained. From the factor graph, as shown in Fig. 5, the trajectory in the $[u_i, u_{i+1})$ interval is parameterized by the control points set $\boldsymbol{\Phi}_i = \{\boldsymbol{Q}_i, \boldsymbol{Q}_{i+1}, \boldsymbol{Q}_{i+2}, \boldsymbol{Q}_{i+3}\}$. The LiDAR factors in the $[u_{i-3}, u_i)$ interval can also constrain $\boldsymbol{\Phi}_i$ due to the local support of the B-spline. Suppose that the factors from the prior scans are added directly to the optimization, we will inevitably introduce new optimization variables $\boldsymbol{\Phi}_{\text{marg}} = \{\boldsymbol{Q}_{i-3}, \boldsymbol{Q}_{i-2}, \boldsymbol{Q}_{i-1}\}$. That will bring two problems, one is to

increase the number of optimization variables in the system, and the other is that $\mathbf{\Phi}_{\text{marg}}$ are the optimal control points for prior scans and do not need to be optimized. Therefore, we propose a marginalization method. Specifically, the factors in the $[u_{i-3}, u_i)$ interval are added to the optimization as prior constraints of $\mathbf{\Phi}_i$. Then, we use the Schur complement to retain prior constraints while discarding prior scans' control points.

Assume the optimization variable set for the current scan is $\mathcal{X} = (\mathbf{\Phi}_{\text{opt}} \cup \mathbf{\Phi}_{\text{marg}})$, where $\mathbf{\Phi}_{\text{opt}} = \mathbf{\Phi}_i$ is the control points set that parameterizes the trajectory of the current scan, and $\mathbf{\Phi}_{\text{marg}}$ is the control points set that is not irrelevant to the trajectory of the current scan that needs to be marginalized. The continuous-time trajectory can be obtained by solving the following nonlinear equations:

$$\mathcal{X} = \operatorname{argmin} \sum \rho\left(\|c_i r_{l_i}\|\right) + \left\|r_{\alpha}^{R}\right\| + \|r_{acc}^{p}\| \tag{17}$$

where r_{l_i} is the factor in $[u_{i-3}, u_{i+1})$, the Huber loss $\rho(\cdot)$ is used to reduce the effect of outliers and c_i can be calculated by equation (9). During optimization, the equation (17) can be linearized as:

$$\boldsymbol{H}\left(\mathbf{\Phi}_{\text{opt}} \cup \mathbf{\Phi}_{\text{marg}}\right) = \boldsymbol{b} \tag{18}$$

where $\boldsymbol{H}$ is the Hessian matrix of the optimization problem. The equation (18) can be written as a chunking matrix:

$$\begin{bmatrix} \boldsymbol{H}_{\text{oo}} & \boldsymbol{H}_{\text{om}} \\ \boldsymbol{H}_{\text{mo}} & \boldsymbol{H}_{\text{mm}} \end{bmatrix} \begin{bmatrix} \mathbf{\Phi}_{\text{opt}} \\ \mathbf{\Phi}_{\text{marg}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{b}_{\text{opt}} \\ \boldsymbol{b}_{\text{marg}} \end{bmatrix} \tag{19}$$

Using the Schur complement [25], equation (19) can be written as:

$$\left(\boldsymbol{H}_{\text{oo}} - \boldsymbol{H}_{\text{om}}\boldsymbol{H}_{\text{mm}}^{-1}\boldsymbol{H}_{\text{mo}}\right)\mathbf{\Phi}_{\text{opt}} = \boldsymbol{b}_{\text{opt}} - \boldsymbol{H}_{\text{om}}\boldsymbol{H}_{\text{mm}}^{-1}\boldsymbol{b}_{\text{marg}} \tag{20}$$

With marginalization, we can retain prior scans' constraints and discard optimization variables that are irrelevant to the current scan.

IV. EXPERIMENT

To validate the effectiveness of the proposed method, CTA-LO is evaluated using three public datasets (NCD [26], M2DGR [27], and the LiLi-OM dataset [28]). We first compare LiDAR odometry with state-of-the-art methods on NCD and M2DGR. Next, we perform ablation experiments to evaluate each module on M2DGR and NCD quantitatively and evaluate on the LiLi-OM dataset qualitatively. We finally discuss efficiency.

For a fair comparison, the algorithms involved in the experimental part are modified to better fit the dataset, we turn off the loop detection and only evaluate the odometry. All experiments are conducted on a computer with an AMD Ryzen7 5800H CPU and a 16G RAM PC.

A. Dataset

1) Ground robot scenarios: The M2DGR dataset is a large-scale dataset collected by a ground robot with RTK/ins as groundtruth. To evaluate the LiDAR odometry, we use 3 sequences of the *gate* and 8 sequences of the *road*.

2) High-frequency motion scenarios: The NCD contains five sequences, with scenes covering the entire campus, including buildings, outdoors, parks, and vegetation. On the *quad* sequence, the device is shaken at a walking speed of 1.5m/s, and the angular velocity of shaking reached 2.5rad/s on the *dynamic* sequence.

3) Small FoV LiDAR scenarios: The LiLi-OM dataset is collected by the Horizon LiDAR (FOV:81.7°×25.1°) carried by a ground robot. The scene covers building and lawn, where the LiDAR faces the wall several times.

B. Odometry Experiments

We compare with the state-of-the-art methods on M2DGR and NCD, including A-LOAM, LeGo-LOAM, and CT-ICP. We utilize absolute trajectory error (ATE) for accuracy comparison and perform a rigid transformation between the groundtruth and estimated trajectory before calculating ATE.

Table I shows that our approach achieves optimal results on most sequences. A-LOAM and LeGo-LOAM use discrete-time optimization and display extensive errors on the sequences with aggressive motions (*quad, dynamic, street07*), because the assumption of constant velocity motion would fail. However, our method is robust under aggressive motion. Although CT-ICP also utilizes continuous-time optimization, it considers continuous within-scan and discontinuous between scans, and uses linear interpolation within-scan to deal with motion distortion. In contrast, the B-spline used in this paper is a higher-order interpolation method with higher accuracy for trajectory fitting. Hence the accuracy of our approach is higher than CT-ICP on most sequences. Fig. 6 compares the trajectories of the general and aggressive motions. On the *street07* sequence, both our method and CT-ICP can perform robust localization under aggressive motion compared to the discrete-time method. However, the trajectory of CT-ICP displays obvious jaggedness, the trajectory of our method is smoother and closer to the groundtruth.

The *short, long, and park* sequences in NCD contain the woods. A-LOAM does nothing with the cluttered feature points (trees, grass), so the accuracy is poor. LeGo-LOAM clusters the point cloud to eliminate some cluttered points. CT-ICP searches the 20 nearest points, which reduces the effect of messy points to some extent but loses small features in the environment. Thanks to the spot uncertainty model proposed in this paper, our approach performs state-of-the-art on these three sequences. The trajectories on the *short* sequence is shown in Fig. 6.

C. Ablation Study

To verify the effectiveness of the marginalization in continuous-time optimization and the spot uncertainty model of the LiDAR points, we perform ablation experiments. In

TABLE I

ACCURACY (ATE) COMPARISON ON THE NCD AND M2DGR

	NCD (RMSE) [m]					M2DGR (RMSE) [m]		
Seq.(m)	short(1609)	long(3063)	quad(479)	dynamic(97.2)	park(696)	gate01(139)	gate02(289)	gate03(248)
A-LOAM	43.5806	84.0194	4.6221	3.3124	32.8425	0.2132	0.5048	0.2045
LeGo-LOAM	0.4104	1.1937	18.5707	13.6433	0.23340	0.4215	0.6623	0.1703
CT-ICP	0.5552	0.5761	**0.0981**	0.1426	0.1802	0.2231	0.3566	**0.1603**
ours	**0.3346**	**0.3807**	0.1195	**0.0802**	**0.1727**	**0.1410**	**0.3208**	0.1972
	M2DGR (RMSE) [m]							
Seq.(m)	street01(752)	street02(1485)	street03(424)	street04(840)	street05(421)	street06(480)	street07(1104)	street08(341)
A-LOAM	7.9923	5.2990	0.2563	4.8831	0.7442	0.5636	28.4630	4.7123
LeGo-LOAM	3.0846	3.5523	0.2938	0.8146	0.4039	0.5072	14.472	1.7235
CT-ICP	**0.2481**	2.6585	0.2439	**0.7996**	0.3558	0.4890	12.1098	0.2064
ours	0.4015	**2.6320**	**0.2263**	0.8188	**0.2059**	**0.2374**	**12.0754**	**0.17229**

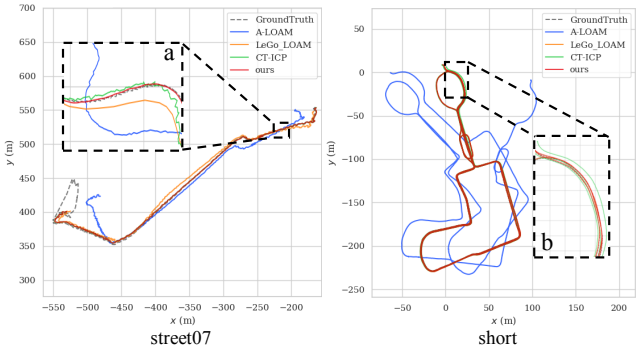

Fig. 6. Comparison on the aggressive motion (*street07*) and general motion (*short*).

addition, we compare the continuous-time to the constant velocity motion model to further verify the advantages of continuous-time in this paper. Table II shows the ATE for different combinations. Without marginalization, our approach fails quickly on these three sequences. Although the constant velocity motion model performs well on the easy sequences like *gate01*, a significant error occurs on the aggressive motion like the *street07*. In addition, the spot uncertainty model of LiDAR points improves odometry accuracy, especially in wooded environments like *short* sequence.

TABLE II

ATE[M] EVALUATION FOR ABLATION STUDY

Margin-alization	CT Module	Uncertainty Model	gate01	street07	short
✔	✗[1]	✗[2]	0.7542	27.5233	12.1421
✔	✗	✔	0.5866	23.6561	0.4730
✔	✔	✗	0.2968	12.4235	6.5371
✔	✔	✔	**0.1410**	**12.0754**	**0.3346**
✗	✔	✔	failed	failed	failed

[1] Using the constant velocity motion model.

[2] Without the spot uncertainty model of LiDAR points.

To better verify the effectiveness of the spot uncertainty model, we conduct experiments on a small FOV LiDAR. Since the LiLi-OM dataset has no groundtruth, we evaluate it in terms of map-building consistency. Fig. 7 shows the reconstruction result.

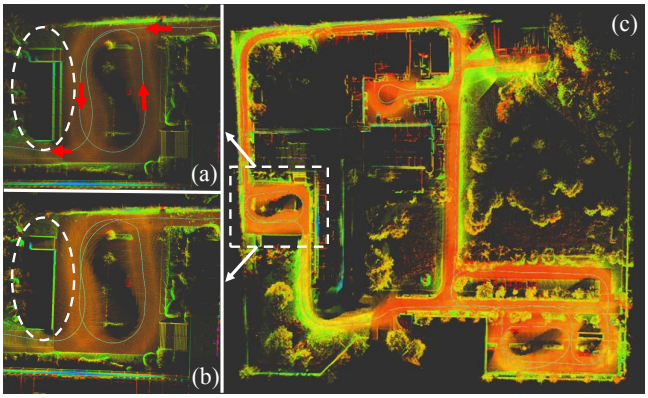

Fig. 7. (a) The point cloud shows significant ghosting without the spot uncertainty model. (b) The ghosting disappears completely after adding the spot uncertainty model. (c) The complete map in the LiLi-OM dataset.

D. Evaluation on Computational Efficiency

Time consumption of our system mainly focuses on optimization, where our optimization variables are four times more than discrete-time. We must iterate 2-5 times to ensure the correct data association during aggressive motion. Since the number of feature points extracted by different LiDAR varies, we can reach 70ms per scan on solid-state LiDAR (∼1500 feature points per scan).

V. CONCLUSIONS

In this paper, we propose an accurate and robust continuous-time LiDAR odometry. We utilize continuous-time trajectory to deal with LiDAR distortion. To avoid degradation, we propose a marginalization method that retains the constraints of prior scans. In addition, we propose a residuals adaptive weighting method and a probabilistic point cloud map based on the spot uncertainty model to further improve the accuracy. We compare CTA-LO with several open-source state-of-the-art LiDAR odometry, and experiments show that our method outperforms them in accuracy and robustness.

REFERENCES

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

[2] 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.

[3] 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.

[4] 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.

[5] C. Bai, T. Xiao, Y. Chen, H. Wang, F. Zhang, and X. Gao, "Faster-lio: Lightweight tightly coupled lidar-inertial odometry using parallel sparse incremental voxels," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4861–4868, 2022.

[6] 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.

[7] X. Zheng and J. Zhu, "Effective solid state lidar odometry using continuous-time filter registration," *arXiv preprint arXiv:2206.08517*, 2022.

[8] J. Quenzel and S. Behnke, "Real-time multi-adaptive-resolution-surfel 6d lidar odometry using continuous-time trajectory optimization," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 5499–5506.

[9] 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.

[10] J. Lv, K. Hu, J. Xu, Y. Liu, X. Ma, and X. Zuo, "Clins: Continuous-time trajectory estimation for lidar-inertial system," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 6657–6663.

[11] C. Park, P. Moghadam, J. L. Williams, S. Kim, S. Sridharan, and C. Fookes, "Elasticity meets continuous-time: Map-centric dense 3d lidar slam," *IEEE Transactions on Robotics*, vol. 38, no. 2, pp. 978–997, 2021.

[12] X. Chen, K. Yu, and H. Wu, "Determination of minimum detectable deformation of terrestrial laser scanning based on error entropy model," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 1, pp. 105–116, 2017.

[13] D. Grant, J. Bethel, and M. Crawford, "Point-to-plane registration of terrestrial laser scans," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 72, pp. 16–26, 2012.

[14] 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.

[15] 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. Spie, 1992, pp. 586–606.

[16] J. Behley and C. Stachniss, "Efficient surfel-based slam using 3d laser range data in urban environments." in *Robotics: Science and Systems*, vol. 2018, 2018, p. 59.

[17] X. Chen, A. Milioto, E. Palazzolo, P. Giguere, J. Behley, and C. Stachniss, "Suma++: Efficient lidar-based semantic slam," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 4530–4537.

[18] C. Yuan, W. Xu, X. Liu, X. Hong, and F. Zhang, "Efficient and probabilistic adaptive voxel mapping for accurate online lidar odometry," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 8518–8525, 2022.

[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] D. Lichti and S. J. Gordon, "Error propagation in directly georeferenced terrestrial laser scanner point clouds for cultural heritage recording, fig working week 2004, athens, greece, may 22–27, 2004 wsa2 modelling and visualization. scholar. google. com/url? sa= u&q= h ttp," *WSA2 Modelling and Visualization, Athens, Greece*, vol. 16, 2004.

[21] D. D. Lichti and S. Jamtsho, "Angular resolution of terrestrial laser scanners," *The photogrammetric record*, vol. 21, no. 114, pp. 141–160, 2006.

[22] Z. G. W. H. A. Q. CHEN Xijiang, HUA Xianghong, "Using error entropy to evaluate point cloud uncertainty influenced by spot," *Geomatics and Information Science of Wuhan University*, pp. 864–868, 2017.

[23] C. De Boor and C. De Boor, *A practical guide to splines*. springer-verlag New York, 1978, vol. 27.

[24] 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.

[25] G. Sibley, "Sliding window filters for slam," *University of Southern California, Tech. Rep*, 2006.

[26] M. Ramezani, Y. Wang, M. Camurri, D. Wisth, M. Mattamala, and M. Fallon, "The newer college dataset: Handheld lidar, inertial and vision with ground truth," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 4353–4360.

[27] J. Yin, A. Li, T. Li, W. Yu, and D. Zou, "M2dgr: A multi-sensor and multi-scenario slam dataset for ground robots," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2266–2273, 2021.

[28] 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.