2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Onion-LO++: An Adaptive and Degradation-Resistant Continuous-Time LiDAR Odometry

Xiaolong Cheng, Ye Sun, Keke Geng, Tianxiao Ma, and Zhichao Liu

***Abstract*—In an era dominated by multi-sensor fusion, this paper explores the operational limits of LiDAR-only odometry. We introduce Onion-LO++, which is designed to overcome two practical limitations of Onion-LO: poor performance in geometrically degenerate environments and instability under high-motion conditions. In order to mitigate point cloud degradation, we propose a coarse-to-fine point cloud segmentation approach that extracts intensity and weak corner features from planar regions, while dynamically adjusting the downsampling rate based on the proportion of planar points to maximize geometric constraints. To handle high-motion scenarios, we integrate a continuous-time trajectory model into the backend optimization and introduce an adaptive onion factor that adjusts optimization parameters in real time. Extensive experiments on five challenging public datasets demonstrate that Onion-LO++ outperforms state-of-the-art methods and operates reliably across narrow spaces, degenerate scenes, high-speed motion, and high-altitude aerial mapping. We open-source the code on GitHub.[1]**

I. Introduction

LiDAR odometry is a core technology in the robotics field due to its exceptional depth-sensing ability and high-precision mapping [1]. In recent years, the application of LiDAR has rapidly extended beyond relatively stable ground vehicles, facilitated by the emergence of high-quality, lightweight LiDAR sensors [2]. Emerging platforms with higher mobility, such as unmanned aerial vehicles (UAVs) [3], quadruped robots [4], humanoid robots [5], and handheld mapping [6], are increasingly adopting LiDAR for perception and navigation. These emerging applications impose more stringent requirements on LiDAR odometry.

To address these challenges, existing works often incorporate IMU measurements to compensate for high-frequency aggressive motions and integrate visual cues to alleviate LiDAR point cloud degeneration. And numerous sensor-fusion systems have been developed, such as LiDAR-Inertial Odometry (LIO) [7], [3], [8], LiDAR-Visual Odometry (LVO) [9], [10], and LiDAR-Inertial-Visual Odometry (LIVO) [6], [11], [12]. However, LIO and LIVO systems are sensitive to IMU calibration errors and sensor noise. Visual input may also introduce 2D–3D projection errors and extra computation, degrading real-time performance. In addition, IMU data can be unavailable or hard to synchronize in some scenarios. Although pure LO exhibits lower stability than LIO under rapid motion, it performs well on most datasets including aerial platforms. More importantly, LO avoids extrinsic calibration and temporal synchronization with IMUs, making it more convenient for non-expert users. These factors motivate us to explore a robust LiDAR-only odometry for extreme scenarios.

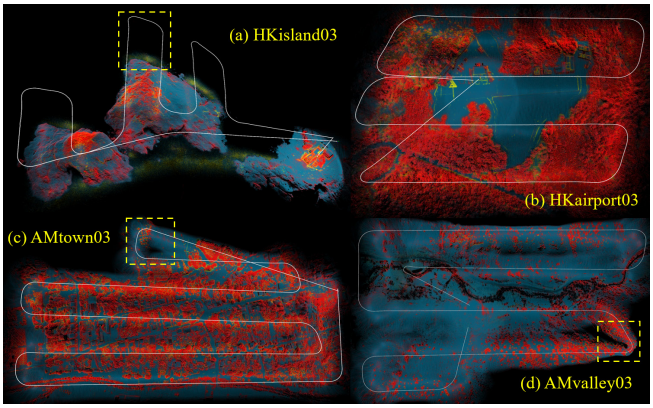

Fig. 1. The segmented point cloud maps of the MARS-LVIG dataset generated by our method. The blue points denote plane feature points, the yellow points denote intensity feature points, and the red points denote edge feature points. The yellow dashed boxes indicate potential failure zones.

This work further enhances the robustness of Onion-LO [13] while exploring the operational limits of LiDAR-only odometry. Onion-LO provides a unified odometry framework that supports easy configuration across different types of LiDAR sensors. However, its underlying design makes it prone to instability in extreme environments, limiting its performance under degenerate or high-dynamic scenarios. **1)** The constant velocity model [14] assumes that a robot maintains the same translational and rotational velocity as in the previous time step, providing a simple yet effective motion prior. However, under aggressive motions, inaccurate point cloud motion compensation can lead to instability, resulting in blurred or ghosted map reconstructions. To address this limitation, we employ a continuous-time model [15], which segments the point cloud and incorporates kinematic constraints to maintain

This work is supported in part by the National Natural Science Foundation under Grant no. 52272414 and 51905095, Jiangsu Graduate Innovative Research Program under Grant no. SJCX24_0072, National Key R&D Program of China 2023YFD2000303, and Yanetze River Delta Science and Technology Innovation Alliance Collaborative Research Project 2023C51GG1600. (Corresponding author: Keke Geng.)

All authors are with the Department of Mechanical Engineering, Southeast University, Nanjing, 201804, China (email: {230238059, 240240374, jsgengke, 230230347, 230248037}@seu.edu.cn).

[1]https://github.com/huashu996/Onion-LO++

robustness under aggressive motions. **2)** Onion-LO employs a multi-layer Onion-Ball structure to segment the point cloud by analyzing the eigenvalues of the covariance matrix [16] for each cell. Although computationally efficient, this segmentation approach struggles to extract informative features in planar regions. To address this limitation, we perform refined feature extraction on planar points, capturing both weak edge features and intensity features.

The goal of this work is to use LiDAR as a sole sensor to overcome point cloud degradation and high-dynamic scenarios. To investigate the operational limits of LiDAR-only odometry under extreme conditions. The main contributions of this work are categorized into the following four aspects:

- Building upon Onion-LO, we propose Onion-LO++, which primarily extends the algorithm's operational limits in scenarios with aggressive motion and LiDAR point cloud degradation.
- We introduce a coarse-to-fine point cloud segmentation strategy that extracts both intensity cues and weak edge features from planar regions, enhancing geometric constraints in degenerate environments.
- We introduce a continuous-time trajectory model for accurate pose estimation under aggressive high-frequency motion. The onion factor dynamically optimizes parameters in real time to handle challenging conditions.
- We validate Onion-LO++ on diverse public datasets, showing better robustness and accuracy than state-of-the-art LiDAR-only methods.

II. Related Work

With the advancement of LiDAR technology, a series of compact, lightweight, and high-resolution sensors has emerged, significantly expanding the range of LiDAR applications. Traditional LiDAR odometry frameworks [17], [18] are no longer suitable, as they heavily rely on specific scanning patterns. Lin et al [2] proposed the first odometry framework suitable for non-repetitive scanning LiDAR. However, it requires extensive parameter tuning depending on factors such as indoor or outdoor environments and motion speed. The KISS-ICP [14] further improved algorithm adaptability by leveraging a constant-velocity model and motion mode error to adjust iterative optimization thresholds, significantly enhancing the stability of (Iterative Closest Point) ICP. However, KISS-ICP cannot adaptively adjust key system parameters and can maintain stable performance only within a single scenario. To address this limitation, Onion-LO [13] introduces a multi-layer, onion-like structure that adaptively adjusts parameters based on the spatial characteristics of each LiDAR frame.

However, Onion-LO relies on a constant-velocity model and is therefore unsuitable for high-motion scenarios such as handheld mapping and UAV mapping. To enable LiDAR operation in high-motion scenarios, the University of Hong Kong proposed a series of LiDAR-inertial odometry frameworks [7], [3], [8]. However, the performance of Kalman filter-based tightly-coupled LiDAR–IMU frameworks heavily depends on IMU calibration and measurement noise. LiDAR points are captured as continuous streaming data, and researchers have attempted to address high-frequency motion issues directly from the LiDAR measurements. CT-ICP [19] introduces the concepts of intra-scan pose continuity and inter-scan pose discontinuity, which enhance robustness under high-frequency motion. However, its performance remains unreliable for handheld devices and aerial platforms. Zheng et al. [15] proposed Traj-LO, a continuous-time trajectory-based LiDAR odometry framework that leverages the intrinsic kinematic relationships within LiDAR point clouds to achieve performance comparable to tightly-coupled LiDAR–IMU odometry. Traj-LO performs linear interpolation within each segment of the point cloud, combined with global trajectory smoothness constraints, enabling stable operation in high-frequency motion scenarios such as handheld devices and UAVs. Nevertheless, Traj-LO relies on segmented processing of LiDAR points, rendering it more susceptible to degeneracy when deployed on solid-state LiDARs with a constrained field of view (FoV).

Point cloud degeneration poses a significant challenge for LiDAR-based odometry. The most straightforward approach proposed by researchers to address degeneration issues is integrating camera data. In FAST-LIVO [11], the system extracts and tracks key points from images, incorporating these visual features as additional geometric constraints into the IMU-based state estimation. This effectively mitigates trajectory drift caused by the accumulation of inertial measurement errors. FAST-LIVO2 [12] directly incorporates visual measurements into the optimization problem, significantly enhancing the system's robustness. However, due to the tightly-coupled design, the system may fail in dark environments where visual features become unreliable. LIVOX-CAM [10] is specifically designed for solid-state LiDARs and achieves coarse-to-fine iterative optimization by projecting visual features onto the point cloud. Its loosely-coupled framework allows stable operation even in dark environments. Wu et al. [20] propose a segment-based visual-aided LiDAR-inertial SLAM framework that introduces visual constraints within detected degenerate segments to enhance the robustness of LiDAR SLAM in geometrically degenerate environments. However, cameras typically operate at lower frame rates, which can result in motion blur and ghosting in high-speed scenarios. Li et al. address degeneration issues by combining geometric and intensity features, but the feature extraction method is only applicable to confined spaces. X-ICP [21] introduces constraints in point cloud registration to regulate pose updates along degenerate directions, preventing the optimization from diverging along weakly-constrained axes. This approach effectively mitigates drift in low-feature environments. Existing LiDAR-only methods struggle with degeneration, and it is nearly impossible to handle degenerate, high-dynamic scenes using LiDAR alone. This work explores the limits of LiDAR odometry.

III. MATH

In this section, we present the key components of the

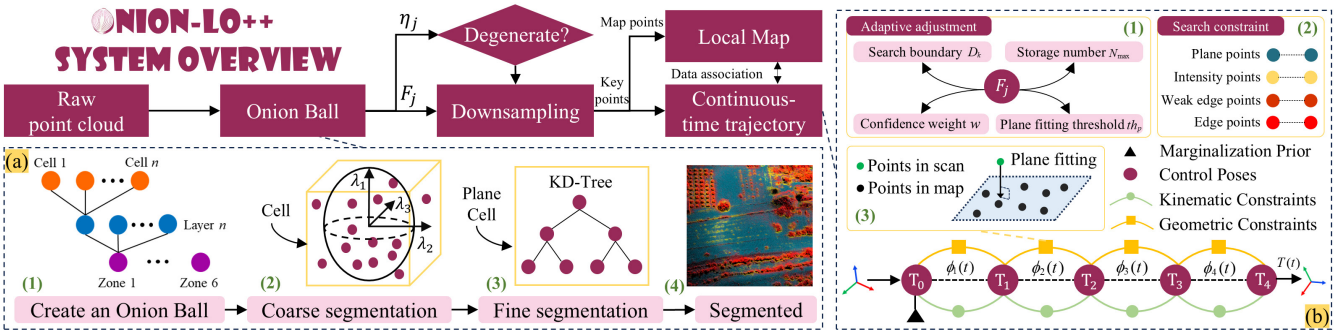

Fig. 2. The Onion-LO++ system overview. The F_j is onion factor used to adaptively adjust the iterative optimization parameters, the n_j is denotes the ratio of planar points and is used to determine whether the current scene is degenerate. (a) Onion-ball coarse-to-fine point cloud segmentation pipeline. (b) Continuous-time trajectory model pipeline.

Onion-LO++ system, which pipeline is illustrated in Fig. 2. The raw point cloud is first segmented using the onion-ball structure, and the onion factor F_j and the planar point ratio η_j of the current frame are computed. Then, the planar point ratio is used to determine whether the point cloud is degenerate and to select an appropriate downsampling strategy. Finally, the point cloud is partitioned into segments and optimized iteratively within the continuous-time trajectory model. The onion factor dynamically regulates the optimization process, resulting in robust and accurate pose estimation. All configuration parameters of the system are listed in Table I, the remaining parameters are adaptively calculated according to the current point cloud.

A. Onion Ball Processing

Onion-ball is a multi-layer parallel structure designed for efficient point cloud processing, enabling rapid analysis of spatial characteristics. Points are allocated into different cells according to their distance, elevation, and azimuth. Based on this representation, Onion-LO performs LiDAR odometry through hierarchical feature decomposition and layer-wise observation modeling, improving robustness to varying LiDAR types and scenes. For further implementation details, please refer to [13].

For j th frame point cloud P_j, the onion ball O_j is constructed by elevation θ_i, azimuth angles ϕ_i, and distance r_i of each point $p_i \in P_j$, shown in Fig. 2(a.1). The voxel size v_l of each layer can be expressed as

$$v_l = RD_l \sin(\delta), \tag{1}$$

where R is onion thickness, D_l denotes the layer index of the cell, δ is the onion resolutions. The volume of each cell can be expressed $v_l \times v_l \times v_l$. The total volume V_j of the current point cloud is estimated by aggregating the volumes of all cells with more than five points. The expected number of key points $N_{\exp}^{key}$ is defined as a configurable system parameter to guide the optimization process. Then, the onion factor F_j of the current frame can be expressed as

$$F_j = \sqrt[3]{V_j / N_{\exp}^{key}}, \tag{2}$$

which reflects the resolution of the current key points and serves as a guiding parameter for system adjustments.

Another function of the Onion-Ball structure is segmentation. Initially, a coarse segmentation is performed for each cell using its covariance matrix, categorizing the cells as planar or corner, as shown in Fig. 1(a.2). The segmentation resolution δ_s is chosen as the maximum of the LiDAR's horizontal and vertical resolutions (δ_h , δ_v). Empirically, this strategy improves the directional balance of point distribution inside each voxel. For layer L_j, the cell size is computed as

$$v_j^s = RD_j \sin(3\delta_s). \tag{3}$$

For each cell, the geometry covariance matrix $\sum_g \in \mathbb{R}^{3\times3}$ and geometry eigenvalues $\lambda_g^1, \lambda_g^2, \lambda_g^3$ can be calculated. The classification of each cell can be determined based on its eigenvalues, indicating whether it corresponds to a planar or edge feature. The classification criteria are as follows

$$\begin{cases} Planar & \lambda_g^1 > th_g \lambda_g^3 \;\&\&\; \lambda_g^2 > th_g \lambda_g^3 \\ Edge & else \end{cases}, \tag{4}$$

where th_g is the geometry judgment threshold, and is set to 10. Subsequently, the ratio η_j of planar points can be computed. The value of η_j is used to assess scene degeneration, as a higher proportion of planar points

TABLE I
ALL PARAMETERS OF THE PROPOSED APPROACH

	Parameter	Value
Onion Ball	Onion resolution δ	3°
	Onion thickness R	5 m
	Onion segmentation resolution δ_s	max (δ_h , δ_v)
Continuous-time trajectory	Expected number of key points $N_{\exp}^{key}$	3000
	Local map voxel size v_m	0.5 m
	Init interval Δt_{Init}	200 ms
	Seg interval Δt_k	20 ms
	Seg num K	3
Adaptively adjusted	Onion factor F_j	dynamically
	Downsampling voxel size v_j^1, v_j^2	
	Maximum voxel capacity $N_{\max}$	
	Maximum search distance D_k	
	Plane fitting threshold th_p, th_{np}	
	Residual confidence weight w_k^i	

provides limited geometric constraints and ambiguous local structures, making the point cloud more susceptible to degradation. When $\eta_j > 0.8$, the current point cloud contains a high proportion of planar points and is prone to degeneration. Therefore, fine segmentation is performed within planar cells to extract subtle features. As illustrated in Fig. 2(a.3), points in planar cells are organized into a KD-Tree. For each point, six nearest neighbors are retrieved to calculate the intensity eigenvalue λ_i^1 and the geometry eigenvalues $\lambda_g^1, \lambda_g^2, \lambda_g^3$. Based on the computed eigenvalues, finer features can be selected from planar points. The classification criteria are as follows

$$\begin{cases} Planar & \lambda_g^1 > 0.5 th_g \lambda_g^3 \,\&\&\, \lambda_g^2 > 0.5 th_g \lambda_g^3 \\ Intensity & \lambda_i^1 > th_i \\ Weak-Edge & else \end{cases}, \quad (5)$$

th_i is the intensity judgment threshold, and is set to 150. The segmented point cloud map is shown in Fig. 2(a.4), where planar points are colored in blue, edge points in red, and intensity feature points in yellow. After applying the onion-ball processing, the segmented point cloud P_j^s is obtained, and both the onion factor F_j and the planar-point ratio η_j are derived to guide the subsequent processing.

B. Downsampling

Voxel downsampling is regarded as an effective sampling strategy. However, many approaches [19], [14], [15] adopt a fixed voxel size, which limits the system's applicability in scale-varying scenarios. Therefore, similar to Onion-LO, the voxel size is adaptively adjusted using the onion factor. The voxel size of the first downsampling is set to $v_j^1 = F_j / 3$, and the sampled points are stored in the local map. The voxel size of the second downsampling is also set to $v_j^2 = F_j$, and the sampled points are used for iterative optimization. The adaptive downsampling voxel size ensures that our algorithm remains stable even in environments with abrupt scene changes.

An excessive proportion of planar points can cause system degeneration, as their limited geometric diversity leads to insufficient spatial constraints, resulting in rank-deficient optimization and unstable pose estimation. Therefore, when $\eta_j > 0.8$, to mitigate the impact of point cloud degeneration, the voxel size for planar points is increased to $3v_j^1$ and $3v_j^2$, while that for non-planar points is reduced to $v_j^1 / 3$ and $v_j^2 / 3$. This downsampling strategy preserves more geometric constraints in degenerate environments.

C. Local Map Management

The local map is managed using a voxel grid with a voxel size of v_m . The maximum number $N_{\max}$ of points stored per voxel is associated with the spatial resolution of the key points [10]. The $N_{\max}$ can be expressed as

$$N_{\max} = (v_m / \frac{F_j}{3})^3 . \quad (6)$$

This strategy allows the local map to adaptively use higher resolution in confined spaces and lower resolution in open areas. Since we extract point cloud intensity features, which are highly sensitive to changes in viewing angle and illumination, the local map retains only the most recent 50 frames of point clouds.

In summary, the local map is constructed using a voxel block with $v_m \times v_m \times v_m$. The maximum number $N_{\max}$ of points stored per voxel is regulated by the onion factor. The map size is limited, retaining only the most recent 50 frames of points. Before initiating the iterative optimization, points collected within Δt_{Init} are first accumulated to initialize the local map.

D. Continuous-Time Trajectory

The continuous-time trajectory using a piecewise linear representation incorporates geometric constraints, kinematic constraints, and marginalization conditions into the optimization, shown in Fig. 2(b), enabling more robust trajectory estimation under fast motion and highly dynamic scenarios. Each continuous-time trajectory $T(t)$ consists of K piecewise linear segments, where the k -th segment is represented by the function $\phi_k(t)$ with its start pose T_{k-1} and end pose T_k . The objective of the optimization is to estimate the control poses $\{T_k\}_{k=0}^K$ and obtain the continuous-time trajectory $T(t)$ by minimizing the weighted sum of geometric error, kinematic error, and marginalization error. The total residual can be expressed as

$$\{T_k^*\}_{k=0}^K = \arg\min_{T(t)} \left(E_g + E_v + E_m \right), \quad (7)$$

where E_g denotes geometric residual, E_v denotes kinematic residual, E_m denotes marginalization prior. Detailed derivations of the individual residual terms are provided in the referenced literature [15].

1) Geometric Constraints: Each linear segment has a duration of Δt_k and stores the points P_k within this time interval. For a point $p_k^i \in P_k$ within the k -th segment with timestamp t_k^i , its pose T_k^i can be computed using linear interpolation

$$T_k^i = \phi_k(t_k^i) = T_{k-1} \oplus \left[\left(\frac{t_k^i - t_{k-1}}{\Delta t_k} \right) \left(T_k \ominus T_{k-1} \right) \right], \quad (8)$$

where t_{k-1} is denotes the timestamp of the start of the k -th segment, the addition $\oplus$ and subtraction $\ominus$ of poses are defined in the $SE(3)$ space as $T \oplus \xi = T \cdot \exp(\xi)$ and $T_2 \ominus T_1 = \log(T_1^{-1} T_2)$, the $\xi \in \mathbb{R}^6$ is in $se(3)$ space, $exp : \mathbb{R}^6 \to SE(3)$ and $log : SE(3) \to \mathbb{R}^6$ are the mapping functions.

Each point $p_i \in P_k$, we explore the 7 closest voxels during nearest-neighbor search, subject to feature type consistency and a maximum search distance constraint D_k . The D_k is adjusted by the onion factor F_j , can be expressed as

$$D_k = 2F_j . \tag{9}$$

The search distance is adaptively adjusted according to the spatial scale. It is larger in large-scale environments and smaller in small-scale environments. Then, the plane is fitted using the six nearest neighbors. During plane fitting, smaller thresholds $th_p = 0.2F_j$ are applied to planar points to ensure accurate fitting, while larger thresholds $th_{np} = F_j$ are used for non-planar points to allow some tolerance. Combined with the previously described downsampling strategy, this plane threshold adjustment method enables the algorithm to remain stable in degenerate scenarios. The geometric error e_k^i of the point $p_k^i \in P_k$ is

$$e_{gk}^i = n_k^{i\mathrm{T}} \left(\phi_k \left(t_k^i \right) \cdot p_k^i - q_k^i \right), \tag{10}$$

where n_k^i denotes the normal vector of the fitted plane, q_k^i denotes the nearest neighbor of p_k^i . To mitigate the influence of mismatched points on the optimization, a residual confidence weight w_k^i is introduced and obtained by

$$w_k^i = \frac{\left(F_j / 2 \right)^2}{\left(F_j / 2 \right)^2 + \left\| e_{gk}^i \right\|^2} . \tag{11}$$

The geometric error of K trajectory is

$$E_g = \sum_{k=1}^{K} \sum_{i=1}^{N_k} w_k^i e_{gk}^{i\mathrm{T}} Q_r^{-1} e_{gk}^i , \tag{12}$$

where Q_r is the covariance matrix.

2) Kinematic Constraints: The trajectory smoothness constraint assumes that adjacent linear segments have the same velocities. The velocity of k -th segment is

$$\varpi_k = \left(T_k \ominus T_{k-1} \right) / \Delta t_k . \tag{13}$$

The pseudo-velocity $\check{\varpi}_k$ of the current segment is derived from the previous segment and can be expressed as

$$\check{\varpi}_k = \left(\check{T}_{k-1} \ominus \check{T}_{k-2} \right) / \Delta t_{k-1} . \tag{14}$$

The velocity difference between consecutive segments should be close to zero. Therefore, the kinematic error e_{vk} of k -th segment can be expressed as

$$e_{vk} = T_k \ominus T_{k-1} - \check{T}_{k-1} \ominus \check{T}_{k-2} . \tag{15}$$

The kinematic error of K trajectory is

$$E_v = \sum_{k=1}^{K} e_{vk}^{\mathrm{T}} Q_v^{-1} e_{vk} . \tag{16}$$

3) Marginalization: Marginalization priors are employed to preserve the constraints of states that fall outside the sliding temporal window, thereby preventing the loss of historical information and improving the global consistency of the trajectory estimation. The marginalization prior is represented in information form as

$$E_m = \frac{1}{2} \left(s_m \ominus \bar{s}_m \right)^{\mathrm{T}} H_m \left(s_m \ominus \bar{s}_m \right) + b_m^{\mathrm{T}} \left(s_m \ominus \bar{s}_m \right), \tag{17}$$

where $s_m \ominus \bar{s}_m$ denotes the state error in the tangent space, H_m is the information matrix, b_m is the corresponding linear term. H_m and b_m are obtained by marginalizing out the states outside the sliding window via the Schur complement [22].

Overall, the onion factor F_j is used to adaptively regulate multiple system parameters, such as segmentation resolution, downsampling voxel size, map storage strategy, and iterative search thresholds. The corresponding parameter settings are summarized in Table I, allowing the system to generalize across environments with different spatial scales.

IV. Experiment

In this section, we conducted all algorithmic experiments on a computer equipped with an Intel i7-14700K CPU, 64 GB RAM. We employ state-of-the-art LiDAR-only odometry methods, including KISS-ICP [14], Traj-LO [15], and Onion-LO [13], to conduct experiments on five public datasets. To better validate our algorithm, the datasets focus on highly dynamic motion scenarios and point cloud degradation scenarios. The system parameter values used in the experiments are listed in Table I.

A. Dataset

We evaluate Onion-LO++ against state-of-the-art methods on five publicly available and highly challenging datasets. The test scenarios span diverse and representative conditions, including unstructured outdoor environments, tunnels, corridors, UAV-based mapping, and handheld mapping. These settings involve significant variations in geometry, viewpoint, motion dynamics, and scene degeneracy, providing a rigorous assessment of algorithm robustness. To ensure fairness, an identical set of configuration parameters was applied across all sequences.

Botanic Garden Dataset [23] is a challenging benchmark for large-scale unstructured outdoor robot navigation. It provides stereo cameras, 3D LiDARs, and IMUs, supporting comprehensive evaluation of perception and odometry algorithms. Its complexity makes it ideal for testing SLAM performance and generalization in real outdoor scenarios.

GEODE Dataset [24] is a multi-LiDAR benchmark for SLAM in highly degenerate real environments. It includes challenging scenes like off-road areas, waterways, and tunnels with sparse, ambiguous features. We use its tunnel sequences from narrow-FOV solid-state LiDARs to test robustness against feature degradation.

HILTI 2021 SLAM Challenge Dataset [25] provides complex indoor environments such as offices, laboratories, and construction sites. The sequences used in this work are captured with the Livox MID-70 LiDAR, featuring point cloud degradation, textureless surfaces, and limited FOV, creating highly degenerate scenarios to evaluate the robustness of our method.

MARS-LVIG Dataset [26] is a large-scale UAV dataset for high-altitude SLAM. It covers diverse aerial scenes (airports, islands, towns, valleys) with speeds 3–12 m/s and altitudes 80–130 m. Providing RTK ground truth and multi-sensor data, it supports SLAM evaluation in high-altitude, dynamic, large-scale environments.

FAST-LIVO2 Dataset [12] is designed to evaluate the FAST-LIVO2 algorithm and includes LiDAR, IMU, and camera data. It contains numerous long-duration sequences

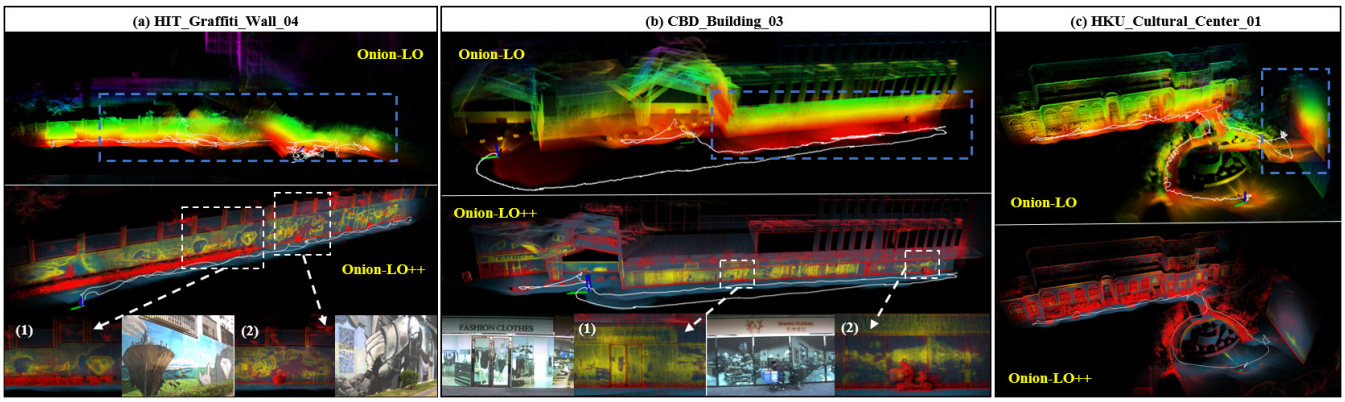

Fig. 3. Comparison of mapping results obtained by Onion-LO and Onion-LO++ on the FAST-LIVO2 Dataset. The blue dashed boxes represent degenerate regions, and the white dashed boxes indicate local zoomed-in areas, with the corresponding visible-light images provided for these regions.

with extreme point cloud degradation, making it ideal for testing algorithm performance under challenging conditions. Since the dataset does not provide ground truth, only qualitative analysis is conducted.

We perform qualitative analysis on the FAST-LIVO2 dataset and quantitative analysis on the Botanic Garden, GEODE, HILTI 2021, and MARS-LIVG datasets. For quantitative analysis, we use the evo toolkit [27] to obtain the Root Mean Square Error (RMSE) of Absolute Trajectory Error (ATE) on other datasets.

B. Quantitative Analysis of Public Datasets

1) Botanic Garden: This dataset is collected in unstructured outdoor environments and does not contain degenerate cases. Our method achieves comparable performance to Traj-LO, as both approaches employ iterative optimization based on a continuous-time trajectory model. The dataset involves ground robots traversing uneven terrain and experiencing intense vibrations, such as when ascending or descending ramps. These conditions violate the constant-velocity assumption, leading to a slight performance degradation of Onion-LO and KISS-ICP.

2) GEODE: In the Tunnel sequences, each segment exhibits varying degrees of degeneration. Specifically, in sequences Tunnel2, Tunnel3, and Tunnel5, the LiDAR is oriented directly toward the tunnel wall, resulting in short-term point cloud degradation. Our method demonstrates a clear advantage on this dataset, maintaining high accuracy even in degenerate sequences. Onion-LO, KISS-ICP, and Traj-LO exhibited varying degrees of drift across these sequences. This robustness stems from the Onion-Ball structure, which enables effective segmentation of the point cloud and the extraction of subtle features within planar regions. In addition, the real-time adjustment of the onion factor enables robust optimization even in extremely narrow spaces, such as along walls. The map constructed by our method is shown in Fig. 4. As shown in the figure, features can be successfully extracted even from degenerate wall surfaces.

3) HILTI 2021: This dataset includes both indoor and outdoor environments, exhibiting a wide range of structural variations. Such diversity poses challenges for methods that rely heavily on manual parameter tuning, often resulting in suboptimal performance for KISS-ICP. In contrast, Onion-LO can adaptively adjust its parameters, improving robustness. However, in severely degenerate RPG scenarios, Onion-LO suffers from significant drift due to prolonged degeneration. Our proposed method addresses these challenges by leveraging continuous-time trajectory modeling and the Onion-Ball structure for effective point cloud segmentation, achieving an ATE of 0.18 on the RPG sequence.

TABLE II
RMSE OF ATE (M) ON THE PUBLIC DATASET

Dataset	Sequence	**Ours**	Onion-LO	KISS-ICP	Traj-LO
Botanic Garden (Robot)	1005-00	2.95	3.40	4.70	2.41
	1005-01	2.01	2.41	2.82	2.04
	1005-07	3.52	5.48	14.52	3.39
	1006-01	2.47	6.13	9.04	3.46
	1008-03	3.18	4.18	7.41	3.41
	1018-00	0.61	1.02	0.88	0.47
	1018-13	1.10	1.12	1.31	1.11
GEODE (Handle)	Tunnel1	0.16	0.22	0.35	0.14
	Tunnel2	0.42	1.21	1.79	1.55
	Tunnel3	0.36	1.50	6.12	22.63
	Tunnel4	0.31	0.32	0.33	0.32
	Tunnel5	0.31	0.33	5.79	0.67
HILTI 2021 (Handle)	RPG	0.18	1.62	3.72	0.22
	Base1	0.15	0.17	6.85	0.31
	Base4	0.04	0.06	0.29	0.06
	Cons2	0.08	0.51	21.65	0.13
	Camp2	0.05	0.14	3.61	0.05
MARS-LIVG (UAV)	HKairport01	8.42	8.31	10.63	9.12
	HKairport02	16.78	18.02	18.16	17.95
	HKairport03	22.55	23.69	--	28.15
	HKisland01	0.36	0.63	1.021	0.35
	HKisland02	1.66	35.85	--	1.94
	HKisland03	2.42	--	--	4.37
	AMtown01	2.37	2.12	12.59	2.46
	AMtown02	2.53	2.97	--	4.30
	AMtown03	3.31	25.67	--	--
	AMvalley01	5.22	9.10	--	--
	AMvalley02	6.97	20.55	--	--
	AMvalley03	8.99	34.51	--	--

The orange values represent the best results in each sequence, '--' denotes that the algorithm fails in this sequence or the RMSE of ATE > 100.

4) MARS-LIVG: The MARS-LIVG dataset presents substantial challenges for LiDAR odometry due to its sparse point cloud density, highly turbulent motion, and frequent degeneracy. Currently, no LiDAR-only method is capable of operating stably on this dataset. KISS-ICP can only operate

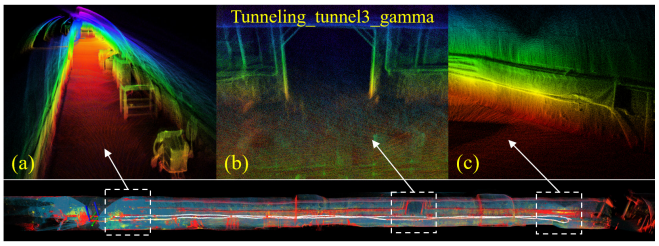

Fig. 4. The segmented point cloud map of Tunneling_tunnel3_gamma sequence. The blue pints denote plane feature points, the yellow points denote intensity feature points and the red points denote edge feature points. The (a), (b), and (c) are zoomed-in views of the map details, with (c) showing the degenerate point.

on the "01" sequences with relatively low flight speed, and it still suffers from large localization errors. Onion-LO adaptively adjusts its system parameters, leading to a substantial performance improvement in this scenario. However, in the more turbulent "03" sequence, the constant-velocity assumption no longer holds, resulting in unstable performance. In the HKisland03 sequence, due to the viewpoint being largely over the water surface, only a few valid points are captured. The abrupt turbulence at turning points further reduces the overlap between consecutive frames, leading to algorithm failure, shown in Fig. 1(a). Traj-LO employs a continuous-time model, making it suitable for high-speed motion scenarios. However, segment-wise processing of point clouds is more prone to drift in degenerate scenarios, as shown in Fig. 1(c). Although a full point cloud frame typically contains sufficient geometric features, each segment may not, leading to unstable registration. Similar to KISS-ICP, Traj-LO lacks the ability to adapt its parameters automatically, while its performance is still highly dependent on manually tuned parameters. In the AMvalley sequences, the UAV gradually approaches the mountain peak, shown in Fig. 1(d), causing drastic changes in the scanned space. The original system parameters fail to adapt to this situation, leading to algorithm failure. Our method is the only approach that remains robust without failure on this dataset, and some result maps are shown in Fig. 1. Through Onion-Ball point cloud segmentation, adaptive adjustment of the onion factor, and precise motion compensation enabled by the continuous-time model, our method achieves stable performance on this challenging dataset. Moreover, the Onion-Ball-based segmentation method remains effective even in very sparse point clouds.

C. Qualitative Analysis

To comprehensively assess the limitations of our algorithm, we conducted a qualitative evaluation on the FAST-LIVO2 dataset, which features prolonged and severe degenerate scenarios where only LiDAR–Inertial–Visual fusion methods can maintain stable operation. Remarkably, our algorithm remained stable on several long-term degenerate sequences, as illustrated in Fig. 3. Although quantitative ground-truth data are unavailable, performance can be qualitatively inferred from the closed-loop consistency between trajectory start and end points.

TABLE III
PROCESSING TIME PER LIDAR FRAME

Dataset	Onion time Mean (ms)		Total time Mean (ms)	
	Onion-LO	**Our**	Onion-LO	**Our**
Botanic Garden	2.19	3.85	28.42	29.92
GEODE	3.05	10.11	36.32	36.11
HILTI 2021	2.12	3.97	47.39	10.83
MARS-LIVG	2.37	5.11	58.83	45.23
FAST-LIVO2	3.36	9.02	27.68	38.91
Avg	2.62	6.41	39.73	32.20

In contrast, Onion-LO fails in all degenerate regions due to its reliance on coarse segmentation, which prevents effective feature extraction from planar surfaces. As shown in Fig. 3(1), our method successfully extracts both intensity and weak edge features from graffiti walls, providing additional geometric constraints during iterative optimization. These constraints substantially improve the robustness of point cloud registration under challenging degenerate conditions. However, the aforementioned sequences still contain some weak edge features. We observed in our experiments that in environments without weak corner features, using only LiDAR intensity is insufficient for stable performance.

D. Runtime Analysis

This section evaluates the runtime performance of our algorithm. The results, summarized in Table III, demonstrate that our method achieves real-time performance at 10 Hz across all tested datasets. The average processing time per frame is only 32.2 ms. In this experiment, fine segmentation was enabled for all sequences, which increased the processing time for the Onion-Ball structure compared to Onion-LO. In practical applications, fine segmentation only needs to be activated in potentially degenerate scenarios, thereby optimizing computational efficiency. Notably, even with fine segmentation enabled, our method maintains real-time operation on embedded platforms such as the NVIDIA Jetson series, highlighting its suitability for deployment in resource-constrained robotic systems.

V. CONCLUSION

Onion-LO++ builds upon Onion-LO and addresses its key limitations, enabling robust performance in both degenerate and highly dynamic scenarios. Leveraging the multi-layer Onion-Ball structure, we introduce a coarse-to-fine point cloud segmentation strategy that facilitates the extraction of intensity cues and weak edge features from planar regions. To improve estimation accuracy under aggressive motion, a continuous-time trajectory model is integrated into the iterative optimization. In addition, the onion factor adaptively adjusts system parameters according to the spatial characteristics of each frame, ensuring stable performance in rapidly changing environments. Notably, Onion-LO++ operates as a LiDAR-only framework, without relying on auxiliary sensors such as IMUs, while still maintaining strong robustness across diverse and challenging datasets. This sensor-independent design enhances its practicality and

engineering value for real-world deployments. In future work, we plan to incorporate visual information to further improve performance in extremely degenerate environments.

REFERENCES

[1] Y. Jia, S. Wang, S. Shao, Y. Wang, F. Zhang, and T. Wang, "M2UD: A Multi-model, Multi-scenario, Uneven-terrain Dataset for Ground Robot with Localization and Mapping Evaluation," arXiv preprint arXiv:2503.12387, 2025.

[2] Lin, J. R., and Zhang, F., "Loam_livox: A fast, robust, high-precision LiDAR odometry and mapping package for LiDARs of small FoV," 2020 IEEE International Conference on Robotics and Automation (ICRA), pp. 3126-3131, 2020.

[3] Xu, W., Cai, Y., He, D., Lin, J., and Zhang, F., "FAST-LIO2: Fast Direct LiDAR-Inertial Odometry," IEEE Transactions on Robotics, vol. 38, no. 4, pp. 2053-2073, 2022.

[4] Lajoie, P.-Y., and Beltrame, G., "Swarm-SLAM: Sparse Decentralized Collaborative Simultaneous Localization and Mapping Framework for Multi-Robot Systems," IEEE Robotics and Automation Letters, vol. 9, no. 1, pp. 475-482, 2024.

[5] Vedadi, A., Yousefi-Koma, A., Yazdankhah, P., and Mozayyan, A., "Comparative Evaluation of RGB-D SLAM Methods for Humanoid Robot Localization and Mapping," in 2023 11th RSI International Conference on Robotics and Mechatronics (ICRoM), pp. 807-812, 2023.

[6] Lin, J., and Zhang, F., "R (3)LIVE++: A Robust, Real-Time, Radiance Reconstruction Package With a Tightly-Coupled LiDAR-Inertial-Visual State Estimator," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 46, no. 12, pp. 11168-11185, Dec. 2024.

[7] Xu, W., and Zhang, F., "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.

[8] He, D., Chen, N., Kong, F., Yuan, C., and Zhang, F., "Point-LIO: Robust High-Bandwidth Light Detection and Ranging Inertial Odometry," Adv. Intell. Syst., pp. 2200459, 2023.

[9] Yuan, Z., Wang, Q., Cheng, K., Hao, T., and Yang, X., "SDV-LOAM: Semi-Direct Visual-LiDAR Odometry and Mapping," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PP, Mar. 29, 2023.

[10] Cheng, X., Geng, K., Liu, Z., Ma, T., and Sun, Y., "LIVOX-CAM: Adaptive Coarse-to-Fine Visual-assisted LiDAR Odometry for Solid-State LiDAR," IEEE Robotics and Automation Letters, pp. 1-8, 2025.

[11] Zheng, C., Xu, W., Zou, Z., Hua, T., Yuan, C., He, D., Zhou, B., Liu, Z., Lin, J., Zhu, F., Ren, Y., Wang, R., Meng, F., and Zhang, F., "FAST-LIVO2: Fast, Direct LiDAR–Inertial–Visual Odometry," IEEE Transactions on Robotics, vol. 41, pp. 326-346, 2025.

[12] Zheng, C., Xu, W., Liu, X., Guo, Q., and Zhang, F., "FAST-LIVO: Fast and Tightly-coupled Sparse-Direct LiDAR-Inertial-Visual Odometry," in 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems, Kyoto, Japan, pp. 4003-4009, 2022.

[13] Cheng, X., Geng, K., Liu, Z., Ma, T., and Sun, Y., "Onion-LO: Why Does LiDAR Odometry Fail Across Different LiDAR Types and Scenarios?," IEEE Robotics and Automation Letters, vol. 10, no. 10, pp. 10522-10529, 2025.

[14] Vizzo, I., Guadagnino, T., Mersch, B., Wiesmann, L., Behley, J., and Stachniss, C., "KISS-ICP: In Defense of Point-to-Point ICP – Simple, Accurate, and Robust Registration If Done the Right Way," IEEE Robotics and Automation Letters, vol. 8, no. 2, pp. 1029-1036, 2023.

[15] Zheng, X., and Zhu, J., "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.

[16] Cheng, X., Ma, T., Geng, K., Liu, Z., Wang, Z., and Yin, G., "SVM-LO: An Accurate, Robust, Real-time LiDAR Odometry with Segmentation Voxel Map for Autonomous Vehicles," in 2024 IEEE 27th International Conference on Intelligent Transportation Systems (ITSC), pp. 1870-1877, 2024.

[17] Zhang, J., and Singh, S., "Low-drift and real-time LiDAR odometry and mapping," Autonomous Robots, vol. 41, no. 2, pp. 401-416, 2016.

[18] Shan, T.X., and Englot, B., "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), pp. 4758-4765, 2018.

[19] Dellenbach, P., Deschaud, J.-E., Jacquet, B., and Goulette, F., "CT-ICP: Real-time Elastic LiDAR Odometry with Loop Closure," in 2022 International Conference on Robotics and Automation (ICRA), pp. 5580-5586, 2022.

[20] Wu, J., Cheng, X., Liu, F., Tang, X., and Gu, W., "Vision-Assisted LiDAR-Inertial SLAM Based on Multiple Segments Under Degenerate Environment," IEEE Transactions on Instrumentation and Measurement, vol. 74, pp. 1-13, 2025.

[21] Tuna, T., Nubert, J., Nava, Y., Khattak, S., and Hutter, M., "X-ICP: Localizability-Aware LiDAR Registration for Robust Localization in Extreme Environments," IEEE Transactions on Robotics, vol. 40, pp. 452-471, 2024.

[22] N. Demmel, D. Schubert, C. Sommer, D. Cremers and V. Usenko, "Square Root Marginalization for Sliding-Window Bundle Adjustment," in 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 2021, pp. 13240–13248, doi: 10.1109/ICCV48922.2021.01301.

[23] Liu, Y., Fu, Y., Qin, M., Xu, Y., Xu, B., Chen, F., Goossens, B., Sun, P.Z.H., Yu, H., Liu, C., Chen, L., Tao, W., and Zhao, H., "BotanicGarden: A High-Quality Dataset for Robot Navigation in Unstructured Natural Environments," IEEE Robotics and Automation Letters, vol. 9, no. 3, pp. 2798-2805, 2024.

[24] Chen, Z., Qi, Y., Feng, D., Zhuang, X., Chen, H., Hu, X., ... & Lu, P., "Heterogeneous LiDAR dataset for benchmarking robust localization in diverse degenerate scenarios," The International Journal of Robotics Research, vol. 02783649251344967, 2024.

[25] Helmberger, M., Morin, K., Berner, B., Kumar, N., Cioffi, G., and Scaramuzza, D., "The Hilti SLAM Challenge Dataset," IEEE Robotics and Automation Letters, vol. 7, no. 3, pp. 7518-7525, 2022.

[26] Li, H., Zou, Y., Chen, N., Lin, J., Liu, X., Xu, W., Zheng, C., Li, R., He, D., Kong, F., Cai, Y., Liu, Z., Zhou, S., Xue, K., and Zhang, F., "MARS-LVIG dataset: A multi-sensor aerial robots SLAM dataset for LiDAR-visual-inertial-GNSS fusion," The International Journal of Robotics Research, vol. 43, no. 8, pp. 1114-1127, 2024.

[27] Schubert, D., Goll, T., Demmel, N., Usenko, V., Stückler, J., and Cremers, D., "The TUM VI benchmark for evaluating visual-inertial odometry," in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018, pp. 1680-1687.