IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 29 November 2024, accepted 2 January 2025, date of publication 7 January 2025, date of current version 13 January 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3526626

RESEARCH ARTICLE

Inc-DLOM: Incremental Direct LiDAR Odometry and Mapping

KAIDUO FANG, RUI SONG, AND IVAN WANG-HEI HO, (Senior Member, IEEE)
Department of Electrical and Electronics Engineering (EEE), The Hong Kong Polytechnic University, Hong Kong

Corresponding author: Ivan Wang-Hei Ho (ivanwh.ho@polyu.edu.hk)

This work was supported in part by the Research Institute for Artificial Internet of Things (RIAIoT) under Project 1-CDJ5, and in part by the Department of Electrical and Electronic Engineering at The Hong Kong Polytechnic University under Project 4-ZZRM.

ABSTRACT Intelligent Vehicle (IV) research is gaining popularity due to the convergence of technological advancements and societal demands, which also leads to the fundamental demand for precise localization. However, the localization accuracy of most existing LiDAR Odometry methods is limited by the complex environment and high-frequency motion, leading to unsatisfactory performance. Moreover, the point cloud data generated by different LiDARs will possess different properties, such as spatial density, Field-of-View (FoV), perception distances, etc., which may have a great impact on LO methods, and makes the generalization of LO a noteworthy issue. To address these issues, we propose the method of Incremental Direct LiDAR Odometry and Mapping (Inc-DLOM). Our proposed Inc-DLOM has the following key contributions: 1) a voxel-to-voxel (V2V) scan matching scheme for scan-to-scan transform estimation; 2) the Incremental Voxel Mapping (IVM) method to incrementally update and maintain the historical mapping information; 3) the Incremental GICP solver to refine the global pose by IVM. To evaluate the performance in terms of accuracy and efficiency, extensive experiments have been conducted with both mechanical LiDAR and solid-state LiDAR on different robotic platforms, including public datasets and real robot data acquisition. The experimental results show that Inc-DLOM achieves better accuracy, efficiency, and generalizability than other comparison state-of-the-art LiDAR Odometry methods.

INDEX TERMS Autonomous driving, LiDAR odometry, SLAM, 3D point cloud.

I. INTRODUCTION

Intelligent vehicles, also referred to as autonomous vehicles are gaining popularity due to the convergence of technological advancements and societal demands [1], [2]. Precise localization and orientation estimation are fundamental objectives in autonomous navigation, which is of critical importance for the safety of intelligent vehicles [3], [11]. Due to the high accuracy and long-range perception, Light Detection And Ranging (LiDAR) has become one of the most important sensors for intelligent vehicles to perform different tasks [4], which raises the interest of LiDAR Odometry (LO) research.

In the past decade, the research topic of LiDAR Odometry (LO) has been extensively investigated, and the algorithms can be divided into registration-based methods and feature-based methods based on point cloud data pre-processing [5], [21], [26]. Registration-based methods mostly use the Iterative Closest Point (ICP) algorithm [12] and its variants to retrieve the pose from the registration results of consecutive point cloud pairs, while feature-based methods often get limited points with the hand-crafted extractor from the raw data for computing the pose [5], [21], [23]. Registration-based methods are often considered more accurate but too time-consuming to support real-time localization tasks [20]. On the other hand, feature-based methods are less robust in conditions where the feature extractor cannot get enough data to compute the pose [6], [7]. In addition, both types of LO methods rely on the storage of historical mapping data to perform additional refinements, which raises the challenges of memory and efficiency due to the mapping growth. Furthermore, the performance of most existing

The associate editor coordinating the review of this manuscript and approving it for publication was Laura Celentano.

LO methods is still to be enhanced, especially under the challenging scenarios, including textureless environments and high-frequency motions.

As a complement, solid-state LiDARs can offer different application scenarios than traditional mechanical LiDARs [8]. Unlike mechanical ones, which scan their surroundings through rotating structures, solid-state LiDARs generate point cloud data with a silicon chip [9], [10]. Due to different scanning principles, the point clouds generated from solid-state LiDARs have unique characteristics. For small-scale applications, such as Virtual Reality (VR) and indoor navigation, solid-state LiDARs are essential in the SLAM systems. However, most existing LO methods rely on data formats that are specific to mechanical LIDARs (such as ring, intensity, etc.), which makes them unavailable for solid-state LiDAR applications. In addition, different versions of mechanical LiDARs also have different characteristics, e.g., Field-of-View (FoV), number of channels, resolutions, etc., which are summarized in Table 1 [10], [34], [35], [36]. All of these mentioned properties introduce discrepancies in the point cloud data, and their implications for different LO methods can be significant, which makes the generalization of LO methods a key outstanding issue.

To address the issues above, we propose Incremental Direct LiDAR Odometry and Mapping (Inc-DLOM) in this paper. Inc-DLOM is a direct registration-based LiDAR Odometry (LO) method, which is fast, computationally efficient, and able to provide accurate pose estimation performance for different types of LiDAR in multiple working scenarios. The mapping result of Inc-DLOM is shown in Fig. 1. The main contributions of this paper are summarized as follows:

- As a general LO method, Inc-DLOM is designed without the details of the exact LiDAR type, which means the proposed LO framework is generalized for different robotic working scenarios, including mechanical LiDAR and solid-state LiDAR for different robotic platforms. The experimental results show that Inc-DLOM is more accurate and efficient when compared with different open-source LiDAR Ododmetry methods.
- A lightweight Generalized ICP (GICP) version [14], called Voxel-to-Voxel (V2V) GICP is proposed. The original point cloud data is transformed into a set of voxels, and the optimization object is also transformed into a "voxel-to-voxel" type, which will improve the computational efficiency without reducing the necessary information in the original data.
- We proposed an efficient mapping method called Incremental Voxel Mapping (IVM) with pre-computed voxels. The IVM implementation is based on spatial hashing, which enables incremental insertion and update of the historical mapping. The mapping construction is quite efficient since every insertion contains only one point cloud. With spatial hashing, the IVM can perform neighbor searching at the computation complexity of $\mathcal{O}(1)$. Then the global pose is refined by the Incremental GICP.

The paper is organized as follows: in Section II, we review the related works on the topic of LO; in Section III, we present the details of the proposed LO; in Section IV, the experimental results are shown and in Section V the conclusion and future works are discussed.

II. RELATED WORKS

The fundamental work in LO research is scan-matching, which addresses the problem of finding the relative pose (or transform) between two robot positions at which laser scans are taken. The most classical method for Scan-Matching is Iterative Closest Point (ICP) [12] proposed in 1992, which is a method that iteratively computes the SE(3) transform between two 3D point sets by associating the closest neighbors in different sets for each data point. In the past decades, ICP-based methods have been extensively investigated with various types of improvements [13], [14], [15], [16], [17], [18], such as 'point-to-plane' ICP [13] with a modified optimization objective. Generalized ICP (GICP) is a variant of 'plane to plane' ICP [14] that considers point data with probability density, making it one of the most popular ICP-based methods with good accuracy and robustness. Normal ICP (NICP) [15] is a powerful extension of GICP, this method extends the $\mathbb{R}^3$ point data to $\mathbb{R}^6$, where the extended three dimensions are normal vectors of local surfaces. GenZ-ICP [19] proposes a method that combines different error metrics within the ICP algorithm and introduces an adaptive weighting-based optimization strategy, which allows it to achieve good results in degenerate environments. Despite its robustness and accuracy in real-world problems, ICP-based methods are often considered too time-consuming and computationally expensive for real-time robotics localization. IMLS-SLAM [20] is an example with accurate estimation but over 1.2 s per frame processing time, which can not meet real-time localization requirements.

Feature-based LO methods are complementary to ICP-based methods, one of the typical examples of feature-based methods is Lidar Odometry and Mapping (LOAM) [21]. LOAM conducts a feature extractor based on the local curvature values of the points and performs transformation estimation between feature points instead of the full set of point cloud data. In the pose estimation of LOAM, different cost functions are defined for the edge, planar, and corner parts. Meanwhile, a historical point cloud map is maintained in the system for loop closure and back-end optimization to reduce the accumulating error from the scan-matching of consecutive laser scans. Several variants of LOAM-based methods have been investigated in the past decade, and most of them follow LOAM's idea of feature-based scan-matching. For example, LeGO-LOAM [22] optimized the implementation of LOAM to make it lightweight enough to support applications on the robotic platforms, F-LOAM [24] optimized the computational speed of LOAM, PVE-LIOM enhances the feature extration process by constructing the 2D scan image from LiDAR data [8], etc. ATI-CTLO utilizes the PCA-based technique to handle the LiDAR point

TABLE 1. Characteristics of different LiDARs.

Sensor	Type	Frequency	FoV	Horizontal Resolution	Vertical Resolution	Points per Second
Velodyne HDL-64E	mechanical	5-15Hz	$360^\circ \times 26.8^\circ$	0.08°	0.41°	2.2 M
Velodyne HDL-32E	mechanical	5-15Hz	$360^\circ \times 41.33^\circ$	0.08°	1.33°	695 k
Ouster OS-0 128	mechanical	10 Hz	$360^\circ \times 90^\circ$	0.175°	0.70°	2.6 M
Realsense L515	solid-state	30 Hz	$70^\circ \times 55^\circ$	0.07°	0.07°	23 M

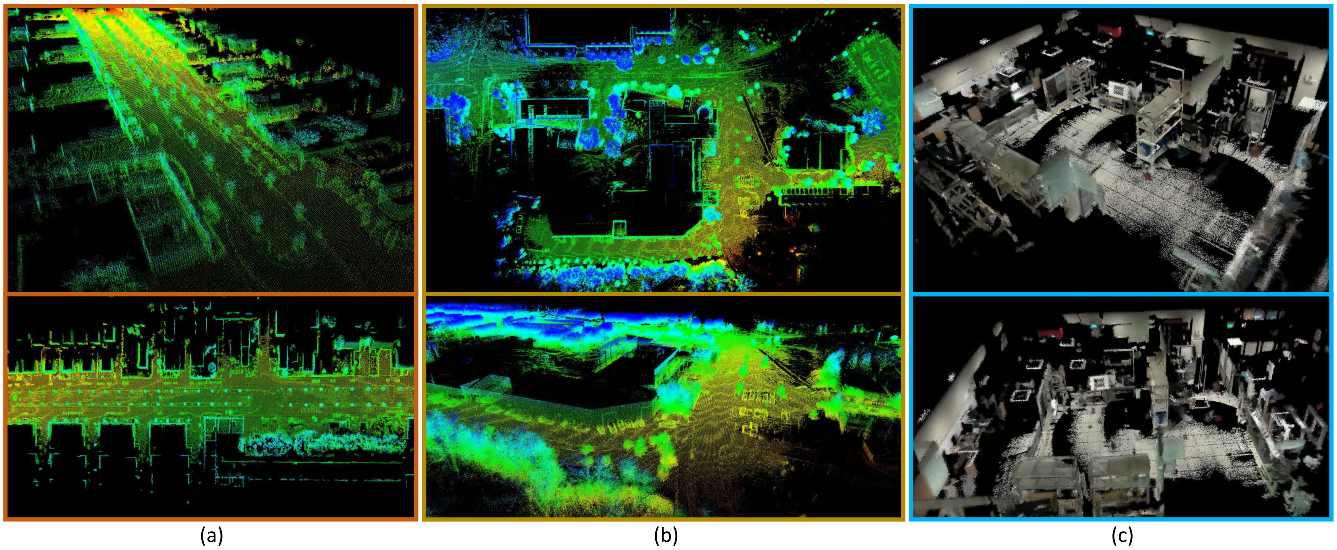


FIGURE 1. Inc-DLOM Mapping Result. The proposed Inc-DLOM can achieve satisfying mapping quality on different datasets: (a) KITTI Dataset collected by autonomous vehicle with Velodyne HDL-64E LiDAR. (b) NCLT Dataset collected by segway robot with Velodyne HDL-32E LiDAR. (c) SSL Dataset collected by Intel-Realsense L515 LiDAR. The point cloud mapping shows that Inc-DLOM is generalized for different LiDAR working scenarios.

clouds and flexibly adjust the temporal intervals according to the motion dynamics and environmental degeneracy [25]. However, feature-based methods are not as robust as direct methods when the feature extractor cannot extract enough accurate feature points, e.g., in textureless environments. In addition, these hand-crafted feature extractors are often computationally expensive.

Managing historical point cloud maps is essential for computational efficiency and accuracy. Existing methods often store and concatenate historical point cloud frames and access them with data structures such as KD-Tree or Octo-Tree to perform nearest-neighbor searches [27], [28]. Although special data structures can reduce computational complexity, they are no longer efficient as the point cloud mapping expands. This is because they usually require continuous data structure reconstruction during the matching process [21], which can make the scan-to-mapping module slow and memory-intensive. Therefore, the appropriate mapping management procedure for LO and the framework to work with it remains an outstanding issue.

In most public LiDAR localization datasets, researchers have focused more on data under autonomous driving scenarios, e.g., KITTI [36], Oxford RobotCar [29], MulRan [30], etc., where the robotic platforms are usually commercial vehicles with smooth motion and non-drastic acceleration. Concurrently, LiDAR is also used to localize robot platforms with vigorous rotations and abrupt acceleration changes, like the Segway robot of NCLT [34]. Most existing LO methods do not perform well in these challenging environments, mainly because most of them do not consider point cloud distortions within their frameworks and appropriately address such issues. The method proposed in [5] utilized modified Fast-VGICP [31] and pose graph optimization, achieved good performance for different autonomous driving scenarios. However, the performance for the high-frequency robotic platform is still worth for improvement. Moreover, solid-state LiDARs, such as the Intel Realsense L515, possess different scanning paradigms and produce data characteristics that diverge substantially from traditional mechanical LiDAR [10]. This discrepancy renders existing LO methods for mechanical LiDARs incompatible with solid-state LiDARs. Consequently, the generalization of LO approaches for different working scenarios and LiDAR types is worth investigating.

III. PROPOSED METHOD

In this section, the details of the proposed LO are presented. For the k-th point cloud input, noted as,

$$\mathscr{P}_k = \{\cdots, p_i, \cdots \mid \forall p_i \in \mathbb{R}^3\} \tag{1}$$

Inc-DLOM is estimating the global pose $\mathcal{T}_k$. $\forall \mathcal{T} \in \mathrm{SE}(3)$ has the general form of

$$\mathcal{T} = \begin{bmatrix} \mathcal{R} & t \\ 0 & 1 \end{bmatrix} \tag{2}$$

where $\mathcal{R} \in \mathrm{SO}(3)$ represents the rotation matrix, $t \in \mathbb{R}^3$ represents the translation vector. The point p can be transformed by $\mathcal{T}$ as,

$$\mathcal{T} \cdot p := \mathcal{R} \cdot p + t \tag{3}$$

These definitions will be adopted throughout the paper.

A. PREPROCESS

The point cloud will first be preprocessed, including the downsampling by Voxel Grid Filter and the point cloud deskew with the constant velocity model, shown in the light blue part in Fig. 2. The point cloud data is recorded continuously while the robotic platforms perform ego-motions. During point cloud data acquisition, severe accelerations and rotations will cause LiDAR point cloud distortions and lead to registration errors. In our proposed LO method, point cloud deskew with the constant velocity model is also included in the Preprocess part.

For $\mathscr{P}_k$, the relative motion is defined as (4), which is derived from $\mathcal{T}_{k-1}$ and $\mathcal{T}_k$.

$$\Delta \mathcal{T} = \mathcal{T}_{k-1}^{-1} \cdot \mathcal{T}_k \tag{4}$$

Then the motion velocity $\xi \in \mathfrak{se}(3)$ is derived from (4) as follows:

$$\xi = \frac{\mathrm{Log}(\Delta \mathcal{T})}{\Delta \tau} = \begin{bmatrix} \omega \\ \nu \end{bmatrix} \tag{5}$$

where $\Delta \tau$ is the relative time of single point cloud sensing, which is usually around 0.1 s for mechanical LiDAR. The operator $\mathrm{Log}(\cdot)$: $\mathrm{SE}(3) \to \mathbb{R}^6$ is the matrix logarithm for SE(3). ω and ν represent the lie algebra for the rotation and translation parts respectively.

For the typical mechanical LiDAR sensing paradigm, points in one point cloud are measured at different relative timestamps. For $p_i \in \mathscr{P}$, the data is measured with a relative timestamp $s_i \in [0, \Delta \tau)$ from the first measurement. Then $\forall p_i \in \mathscr{P}_k$ is deskewed as:

$$p_i = \mathrm{Exp}(s_i \cdot \xi) \cdot p_i \tag{6}$$

where the operator $\mathrm{Exp}(\cdot) : \mathbb{R}^6 \to \mathrm{SE}(3)$ is the matrix exponential for $\mathfrak{se}$(3).

B. SCAN-TO-SCAN (S2S) MATCHING

After the preprocess step, the k-th cloud data $\mathscr{P}_k$ is utilized to compute the relative transform with the previous cloud, which is denoted as $\mathcal{T}_{k-1}^k$. The process of Scan-to-Scan (S2S) matching consists two steps, voxelization and Voxel-to-Voxel GICP, for computing the voxels information and $\mathcal{T}_{k-1}^k$ respectively.

1) VOXELIZATION

$\mathscr{P}_k$ will be transformed from a point set into a voxel set $\mathcal{P}_k$, defined as (7).

$$\mathscr{P}_k \to \mathcal{P}_k = \{\cdots, \mathcal{V}_i, \cdots\} = \bigcup_{i=0}^{n} \mathcal{V}_i, \ \mathcal{V}_i = \{\mathcal{C}_i, p_i\} \tag{7}$$

In which, n is the size of $\mathcal{P}_k$, $\mathcal{V}_i$ is the i-th voxel in the point cloud data. $\mathcal{V}_i$ contains the local surface covariance $\mathcal{C}_i \in \mathbb{R}^{3\times3}$ and the coordinate $p_i \in \mathbb{R}^3$. It is worth noting that the number of voxels is much less than the original number of points since p_i that belongs to the subset of $\mathscr{P}_k$ is selected by Approximate Voxel Filter [37]. $\mathcal{C}_i$ is computed from nearby points searched by KD-Tree. In our implementation, NanoFLANN is used for improving the computational computation efficiency [32].

2) VOXEL-TO-VOXEL (V2V) GICP

Other than aligning the points correspondences mentioned in the original GICP paper [14], the key idea of V2V GICP is the registration of the target set $\mathcal{P}^{\mathcal{T}}$ and source set $\mathcal{P}^{\mathcal{S}}$, where the optimization problem is defined as (8). $\mathbb{F}(\cdot, \cdot)$ is the cost function taking two voxels as inputs, $\mathcal{V}_i^{\mathcal{S}} \in \mathcal{P}^{\mathcal{S}}$ and its closest neighboring voxel $\mathcal{V}_i^{\mathcal{T}} \in \mathcal{P}^{\mathcal{T}}$. Σ is the voxel-to-voxel covariance defined in (9), in which $\mathcal{C}_i^{\mathcal{T}}$ and $\mathcal{C}_i^{\mathcal{S}}$ are voxel covariances of $\mathcal{V}_i^{\mathcal{T}}$ and $\mathcal{V}_i^{\mathcal{S}}$ respectively.

$$\mathcal{T}^* = \underset{\mathcal{T} \in \mathrm{SE}(3)}{\mathrm{argmin}} \sum_i \mathbb{F}(\mathcal{V}_i^{\mathcal{T}}, \mathcal{V}_i^{\mathcal{S}}) = \underset{\mathcal{T} \in \mathrm{SE}(3)}{\mathrm{argmin}} \sum_i ||p_i^{\mathcal{T}} - \mathcal{T} \cdot p_i^{\mathcal{S}}||_{\Sigma^{-1}}^2 \tag{8}$$

$$\Sigma = \mathcal{C}_i^M + \mathcal{R} \cdot \mathcal{C}_i^S \cdot \mathcal{R}^T \tag{9}$$

C. SCAN-TO-MAPPING (S2M) MATCHING

The proper mapping construction and storage method is critical for LO's performance since the pose estimation often requires scan-to-mapping matching to eliminate the accumulated error. However, the usage of local point mapping in the direct LO is often considered as more memory and time-consuming than the one in feature-based methods due to the amount of data. To address this issue, we propose a novel scan-to-mapping scheme that is computationally efficient and can support lightweight, real-time mapping access.

1) GLOBAL VOXEL INFORMATION

As shown in Fig. 2, the voxel information (7) is recycled for the S2M scheme, which is transformed into the global frame to construct the historical mapping with the pose $\mathcal{T}_k$. The global voxel $\mathcal{V}_i^*$ is calculated as

$$\mathcal{V}_i^* = \mathcal{T}_k \cdot \mathcal{V}_i := \{\mathcal{R}_k \cdot \mathcal{C}_i \cdot \mathcal{R}_k^T, \ \mathcal{R}_k \cdot p_i + t_k\} = \{\mathcal{C}_i^*, \ p_i^*\} \tag{10}$$

2) INCREMENTAL VOXEL MAPPING

It is apparent that the size of historical mapping will grow as time passes for the LiDAR Odometry. Most keyframe-based LO methods build the keyframes based on self-motions. For general cases, the thresholds of rotation radians and translation distances are set for keyframe insertion to the

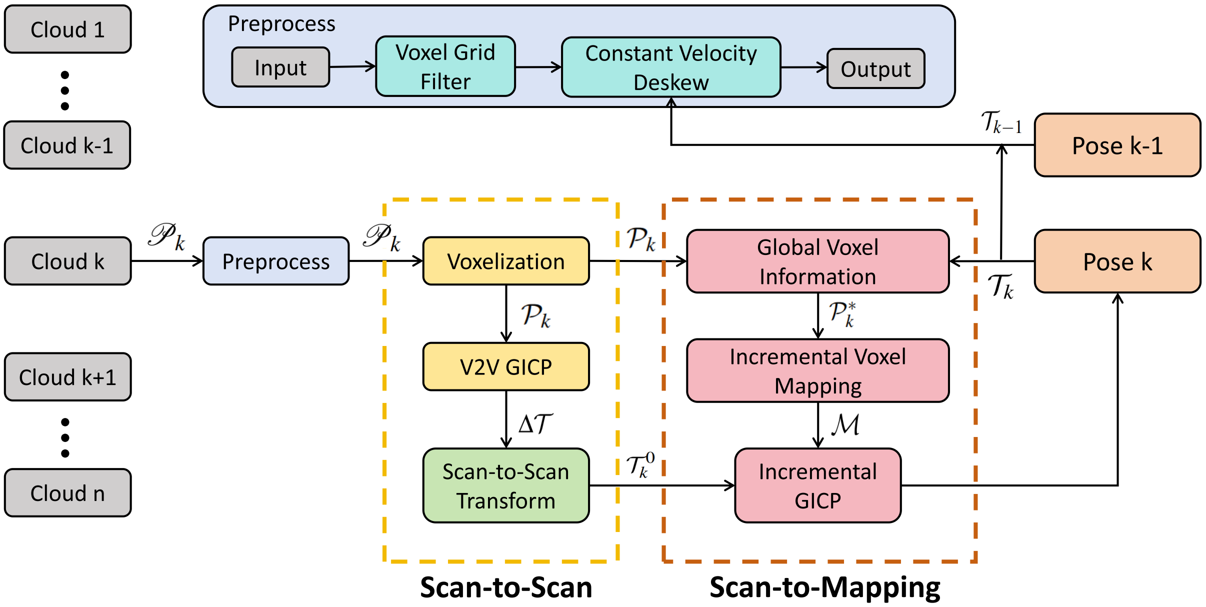


FIGURE 2. Inc-DLOM Framework. Inc-DLOM takes the point cloud $\mathscr{P}_k$ as input, and goes through the procedure of Preprocess. Then the deskewed and downsampled point cloud is utilized for S2S matching to compute the relative *S2S transformation (Scan-to-Scan Transform)* $\Delta\mathcal{T}$. In the procedure of S2M matching, $\mathcal{P}_k$ is generated by the *Voxelization* step, and taken by *V2V GICP*. The proposed *Incremental Voxel Mapping (IVM)* $\mathcal{M}$ will be updated by *Global Voxel Information* $\mathcal{P}_k^*$. The global pose $\mathcal{T}_k$ is computed by the proposed *Incremental GICP* method with the initial guess $\mathcal{T}_k^0$.

historical mapping. Even if efficient data structures like KD-Tree can be used for accessing the points, the operations are still time-consuming and computationally intensive, because keyframe insertion requires continuous data structure re-built. For example, LOAM-based methods are often not able to support real-time scan-to-mapping operation for every incoming point cloud data, even though the mapping cloud is processed with voxel grid downsampling. Besides, the self-motion thresholds are hard to set without extensive try-outs.

Incremental Voxel Mapping (IVM) is implemented based on the spatial hash table to insert and update the historical voxel mapping incrementally to address the mentioned issues. Voxels are stored in the hash table and every one of them is associated with the spatial key. To access one specific voxel, only p_i^* is needed to compute the spatial key as

$$\text{key}(\mathcal{V}_i^*) = \mathit{floor}(p_i^*/S_{IVM}) \tag{11}$$

where S_{IVM} is the pre-defined voxel size parameter of IVM. Since the hash mapping of the data is unique, $\mathcal{V}_i^*$ can be accessed with $\mathcal{O}(1)$ time complexity, which is essential for efficiency. The hash function is defined as

$$\begin{aligned} h(p_i^*) &= h(x,y,z) = ((x*a)\ xor\ (y*b)\ xor\ (z*c))\%d \\ &\quad a = 73856093, b = 471943, c = 83492791, \\ &\quad d = 10000000 \end{aligned} \tag{12}$$

where the input of $h(\cdot)$ is the 3-dim spatial voxel coordinate containing x, y, z, xor is exclusive OR, and $\%$ is the modulus operator. By the hash function in (12), every incoming set of $\mathcal{V}_i$ can be managed and stored in the hash table individually, which allows efficient access in the following steps.

In our proposed method, IVM is continuously inserting the incoming set of $\mathcal{V}_i^*$ to expand the historical mapping, which is denoted as

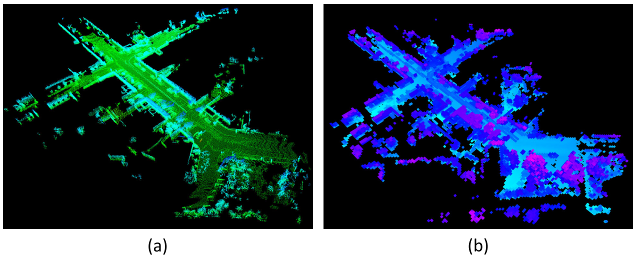
$$\mathcal{M} = \sum_{j=0}^{M} \bigcup_{i=0}^{n} \mathcal{V}_i^* = \bigcup \mathcal{V}_{\mathcal{M}}, \tag{13}$$

where M is the number of point cloud frames, and the sum operation means the concatenation of different voxel sets. While expanding $\mathcal{M}$, the voxels to be inserted are classified into two types, new voxels $\mathcal{V}_n^*$ and historical voxels $\mathcal{V}_h^*$, depending on whether key($\mathcal{V}^*$) is inserted into IVM before. All $\mathcal{V}_n^*$ will be inserted into $\mathcal{M}$ with the hash table process. $\mathcal{V}_h^*$ will be merged into the voxel with the same spatial key, which is denoted as $\mathcal{V}_{\mathcal{M}}$. The voxel merging operation is noted as $\mathcal{V}_r^* = \mathcal{V}_h^* \oplus \mathcal{V}_{\mathcal{M}}$, which is defined as the follows:

$$p_r^* = \frac{p_h^* + p_{\mathcal{M}}}{2} \tag{14}$$

$$\begin{aligned} \mathcal{C}_r^* = (&\mathcal{C}_{\mathcal{M}} + \mathcal{C}_h^* + (p_{\mathcal{M}} - p_r^*)(p_{\mathcal{M}} - p_r^*)^T \\ &+ (p_h^* - p_{\mathcal{M}})(p_h^* - p_{\mathcal{M}})^T)/2 \end{aligned} \tag{15}$$

The operation allows IVM to update $\mathcal{M}$ incrementally without storing the points data in voxels, which needs a large amount of memory usage as the mapping expands, as the traditional keyframe-based LO methods. Due to the parallelism of IVM, multi-processing can be utilized in our implementation to enhance computational efficiency. To illustrate the effect of the mapping procedure in


(a)

(b)

FIGURE 3. The visualization of IVM. (a) The plot of the point cloud mapping. (b) The plot of IVM on the mapping.

Inc-DLOM, Fig. 3 shows the visualization of IVM in the KITTI 00.

Fig. 3 (a) shows the mapping of the first 150 point clouds without processing, and (b) shows the IVM of the mapping, from which we can see that the IVM is ranging the exact distance as the original point cloud mapping. Table 2 shows the statistical comparison of the original point mapping and IVM, the voxel size of IVM is 1.50m, which gives rise to the best estimation result of Inc-DLOM in the KITTI Dataset [36]. Through the Incremental Voxel Mapping process, Inc-DLOM reduces the size of the mapping by hundreds of times, which is the reason why the scheme becomes extremely lightweight and fast. The IVM procedure proposed in this paper can preserve a huge amount of historical point cloud information with only limited memory costs. This property is essential for pose estimation since historical mapping will be used for Scan-to-Mapping pose refinement to eliminate accumulated drifting from Scan-to-Scan alignment, and IVM can maintain historical information to an extremely large range with reasonable memory costs.

TABLE 2. Statistical results of the IVM.

	Point Mapping	Inc-Voxel Mapping
Points / Voxels No.	2535005	15852
Memory Usage	29.01 MB	0.18 MB

3) INCREMENTAL GICP

Incremental GICP is the core part of the S2M matching scheme, which is described in Algorithm 1. The algorithm takes the historical mapping described in Subsection III-C2, voxel set described in Subsection III-B1, and the initial guess $\mathcal{T}_k^0 \in \mathrm{SE}(3)$ as inputs. $\mathcal{T}_k^0$ is computed from the previous pose and relative S2S transformation as (16).

$$\mathcal{T}_k^0 = \mathcal{T}_{k-1} \cdot \Delta\mathcal{T} \tag{16}$$

The difference between Incremental GICP and V2V GICP is mainly on lines 5-8, where getting $\mathcal{V}_\mathcal{M}$ by key($\mathcal{V}_i$) is the $\mathcal{O}(1)$ operation in IVM hash table instead of KD-Tree searching, and the cost function will be constructed if the voxel associated with key($\mathcal{V}_i$) can be searched in IVM.

The cost function of Incremental GICP is defined as (17), which is solved by Gauss-Newton iteratively.

$$\mathcal{T}_k = \underset{\mathcal{T}\in\mathrm{SE}(3)}{\operatorname{argmin}} \sum_i \mathbb{F}(\mathcal{V}_\mathcal{M}, \mathcal{V}_i) = \underset{\mathcal{T}\in\mathrm{SE}(3)}{\operatorname{argmin}} \sum_i ||p_\mathcal{M} - \mathcal{T}\cdot p_i||^2_{\Sigma^{-1}}$$
$$\Sigma = \mathcal{C}_\mathcal{M} + \mathcal{R}\cdot\mathcal{C}_i\cdot\mathcal{R}^T \tag{17}$$

A_i, b_i mentioned in line 9 of Algorithm 1 is the general format of the nonlinear least-square problem of $Ax = b$, where matrix L is the Cholesky square root of matrix Σ so that $\Sigma = L \cdot L^T$.

$$J_i = L^{-1} \cdot \left[[\mathcal{T}\cdot p_i]_\times,\ -I_{3\times3}\right] \tag{18}$$
$$A_i = J_i^T \cdot J_i,\ A_i \in \mathbb{R}^{6\times6} \tag{19}$$
$$b_i = -J_i^T \cdot L^{-1} \cdot (p_\mathcal{M} - \mathcal{T}\cdot p_i),\ b_i \in \mathbb{R}^{6\times1} \tag{20}$$

Incremental GICP allows the proposed LO to perform fast scan-to-mapping refinement, which improves the accuracy of the direct point cloud registration, while barely increasing the computation time. Table 3 shows the computational time cost for every procedure in Inc-DLOM for a single LiDAR point cloud frame measured in milliseconds (ms), here the voxel size in IVM is 1.0m, which is verified by the best accuracy test on the NCLT Dataset [34]. The results are tested by the first 4000 frames in **NCLT_1**, mentioned in Section IV. From Table 3, we can see that the procedure of S2M matching only increases the computation time by less than 1 ms, which is negligible for real-world applications.

Algorithm 1 Procudure of Incremental GICP

```
   Input: historical mapping $\mathcal{M}$, point cloud voxels $\mathcal{P}_k$,
          initial guess $\mathcal{T}_k^0$
   Output: global pose $\mathcal{T}_k$
1  $\mathcal{T}_k \leftarrow \mathcal{T}_k^0$;
2  while $\mathcal{T}_k$ not converged do
3      $A \leftarrow 0^{6\times6}$, $b \leftarrow 0^{6\times1}$;
4      for $i$ in $\{0, \cdots n\}$ do
5          $\mathcal{V}_\mathcal{M} \leftarrow$ getVoxel($\mathcal{M}$, key($\mathcal{V}_i$));
6          if $\mathcal{V}_\mathcal{M} == \varnothing$ then
7              continue
8          end
9          $A_i, b_i \leftarrow$cost($\mathcal{V}_\mathcal{M}$, $\mathcal{V}_i$);
10         $A \leftarrow A + A_i, b \leftarrow b + b_i$;
11     end
12     $\delta T \leftarrow$ GaussNewton($A$, $b$);
13     $\mathcal{T}_k \leftarrow \mathcal{T}_k \cdot \delta T$
14 end
15 return $\mathcal{T}_k$
```

IV. EXPERIMENTAL RESULTS

A. EXPERIMENTAL SETUP

To evaluate Inc-DLOM's performance and generalization, we firstly performed experiments on multiple public datasets with different working scenarios, the evaluation goes on both accuracy and efficiency. Starting with the KITTI

TABLE 3. Detailed time costs of subprocedures in Inc-DLOM.

	Preprocess		S2S Matching		S2M Matching	
Procedures	Downsample	Points Deskew	Covs Computing	Optimization	IVM Updates	Optimization
Time (ms)	1.554	6.538	4.128	0.870	0.003	0.872

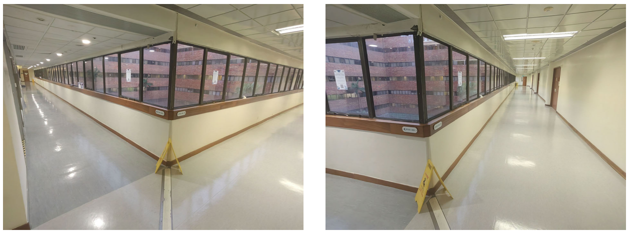


FIGURE 4. Data Collection Robotic Platform.

Odometry Benchmark [36], the classical dataset for the scenarios of autonomous driving localization. To test Inc-DLOM's performance on challenging scenarios, we evaluate the performance with the NCLT Dataset [34] and Newer College Dataset (NCD) [35], which are collected by a Segway robot and hand-held device of high-frequency self-motions. Fig. 6 shows the trajectories estimated by Inc-DLOM in different data sequences.

In addition, we also collected the custom dataset with our robotic platform, deployed with VLP-16 LiDAR, IMU, MaxTang FP650 NUC [42], shown in Fig. 4. The data acquisition scenario is in the corridor of PolyU buildings. The selected corridors lack identifying objects and are filled with plain walls, repetitive doors and windows, as shown in Fig. 6.

The chassis is a Magni Silver robot [33], a typical differential drive robot. The control is highly dependent on the feedback from wheel encoders, where the measurements are noisy and easy to interfere by slippery, which makes the robot containing uneven motions even with certain control inputs. Both the textureless environment and irregular high-frequency motions formed a challenging working scenario.

Besides, our proposed Inc-DLOM is tested with the data collected by Intel RealSense L515 [10], which has a different scanning scheme than traditional mechanical LiDARs. The performance on solid-state LiDAR is essential for the generalization evaluation. Finally, the computational efficiency is also evaluated.

For all aspects of the experiments, we compare our Inc-DLOM with different state-of-the-art open-source LO systems, including feature-based LO methods: Fast-LIO [38], LINS [39], LIOM [7], and direct LO method: HDL-SLAM [40] and Direct LiDAR Odometry (DLO) [26]. All of the experiments are conducted with a platform based on an AMD Ryzen 9 7940HS CPU with 4.00 GHz frequency.

B. AUTONOMOUS DRIVING SCENARIO

The evaluation is firstly measured based on the error metrics defined by the KITTI Odometry Benchmark as translational

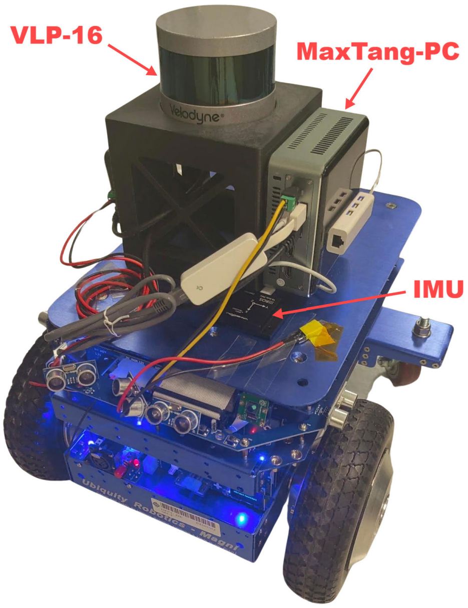


FIGURE 5. Data collection robotic platform.

drifting error in percentage and rotational drifting error in degree of 100 m for all possible subsequences of length (100,...,800) meters. Table 4 shows the evaluation results, the best results among the considered methods for each sequence are highlighted with black bold.

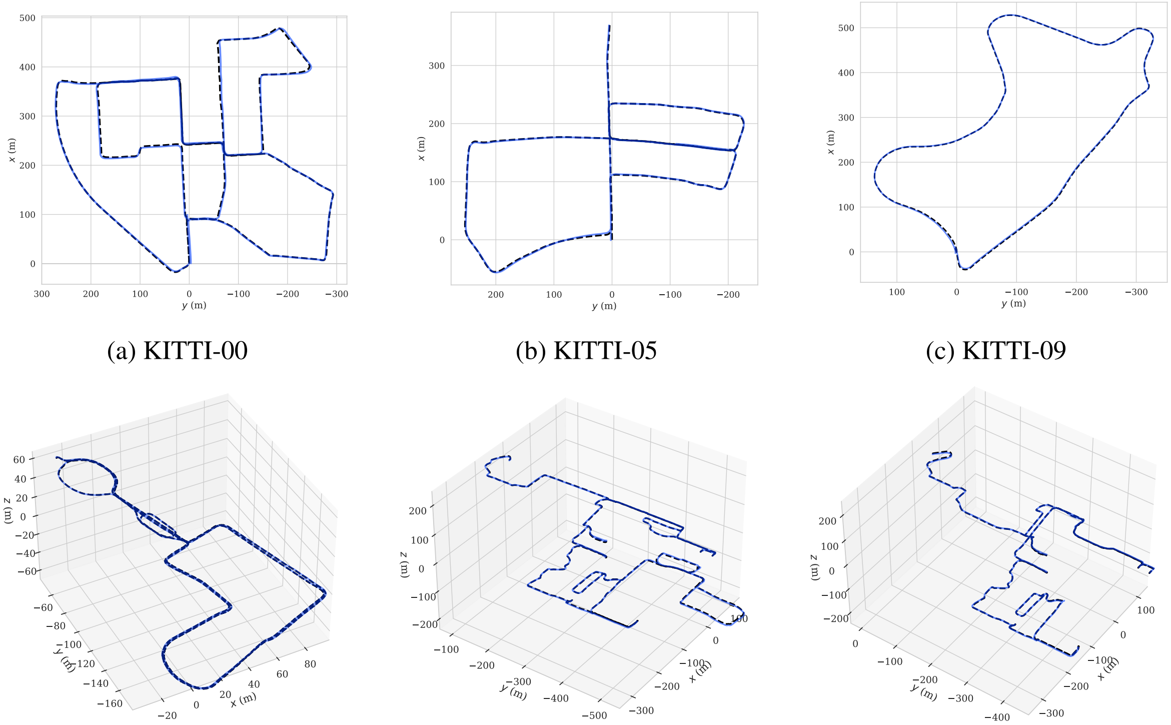
We also compared the Absolute Trajectory Error (ATE) [41], which is a more direct and intuitive error metric for localization performance evaluation. ATE is defined as (21),

$$ATE_i = \mathcal{T}_{est,i}^{-1} \cdot \mathcal{T}_{ref,i} \tag{21}$$

where $\mathcal{T}_{est,i}$ refers to the pose estimated and $\mathcal{T}_{ref,i}$ is the ground truth, and the root mean squared error (RMSE) is computed by (22).

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=0}^{N} ATE_i^2} \tag{22}$$

We compared the translational part of the RMSE of ATE, shown in Table 5, among which the best results are highlighted in red. From the evaluation, we can see that Inc-DLOM is able to achieve the best performance in almost all sequences both in the drifting and ATE evaluation. Although LIOM can get better performance on Sequence 07 in the ATE evaluation, it is not comparable with Inc-DLOM in both drifting and ATE in all other sequences. Overall, Inc-DLOM is outperforming all chosen state-of-the-art LO methods.



(a) KITTI-00 (b) KITTI-05 (c) KITTI-09

(d) NCD-1 (e) NCLT-2 (f) NCLT-3

FIGURE 6. **Inc-DLOM Trajectories on selected dataset sequences. The figure shows the estimated trajectories of dataset sequences, including multiple LiDAR types and different working scenarios. The plot of ground truths (dashed line) and estimated trajectories (blue) shows good estimation performance of our proposed method.**

TABLE 4. **Translational drifting error (%) / rotational drifting error (deg/100m) evaluation of KITTI dataset.**

	00	02	05	06	07	08	09	10
Inc-DLOM	**0.58 / 0.33**	**0.94 / 0.35**	**0.38 / 0.24**	**0.23 / 0.14**	**0.14 / 0.16**	**0.99 / 0.39**	**0.28 / 0.23**	**0.22 / 0.26**
Fast-LIO	3.24 / 1.35	1.84 / 0.55	0.51 / 0.26	1.34 / 0.48	0.46 / 0.29	2.68 / 1.07	0.36 / 0.34	0.51 / 2.07
LINS	8.46 / 3.45	10.08 / 3.04	7.64 / 3.56	1.12 / 0.86	4.29 / 2.12	8.75 / 3.53	11.49 / 3.60	9.35 / 3.27
HDL-SLAM	3.04 / 1.05	9.29 / 2.72	0.80 / 0.46	0.89 / 0.29	0.72 / 0.95	2.45 / 0.92	2.35 / 0.71	1.11 / 0.82
LIOM	9.75 / 2.37	18.85 / 5.08	9.75 / 2.37	17.24 / 6.33	0.20 / 0.48	21.69 / 9.06	10.96 / 6.54	4.75 / 2.60
DLO	2.37 / 0.53	6.36 / 0.94	1.50 / 0.43	0.48 / 0.22	0.95 / 0.49	4.96 / 1.51	3.12 / 0.46	4.06 / 0.74

TABLE 5. **RMSE of ATE (m) Evaluation of KITTI dataset.**

	00	02	05	06	07	08	09	10
Inc-DLOM	**3.85**	**7.21**	**2.51**	**1.43**	1.57	**7.91**	**2.32**	**1.45**
Fast-LIO	42.49	12.41	3.59	9.30	2.62	17.21	2.69	4.80
LINS	43.47	85.87	4.40	7.56	22.91	22.51	132.83	52.62
HDL-SLAM	22.44	98.26	4.96	5.68	4.41	19.26	21.15	5.58
LIOM	85.43	137.34	57.90	175.68	**1.18**	143.78	69.19	22.60
DLO	5.68	8.04	2.58	1.64	1.82	21.61	6.92	2.35

C. HIGH FREQUENCY MOTION SCENARIO

For robotic platforms with high-frequency self-motions, SLAM algorithms on autonomous driving scenarios may not be able to perform effectively, which makes it important to evaluate the SLAM performance in this aspect. The NCLT dataset is one of the typical datasets collected by a Segway robot with Velodyne HDL-32 LiDAR [34]. We have selected five sequences from the NCLT dataset, NCLT_1 (20130110), NCLT_2 (20120615), NCLT_3 (20120429), NCLT_4 (20120115), and NCLT_5 (20120511). The Newer College Dataset is collected by hand-held OS-128 LiDAR with long trajectories and high-frequency rotations [35], which is similar to the NCLT Dataset. We selected four sequences from the Newer College Dataset, NCD_1 (2020-03-10), NCD_2 (2021-11-30), NCD_3 (math-1), and NCD_4 (math-2). The evaluation of NCLT and NCD is shown in Table 6. From the table, we can see that our proposed Inc-DLOM outperforms all other benchmarks. LIOM and

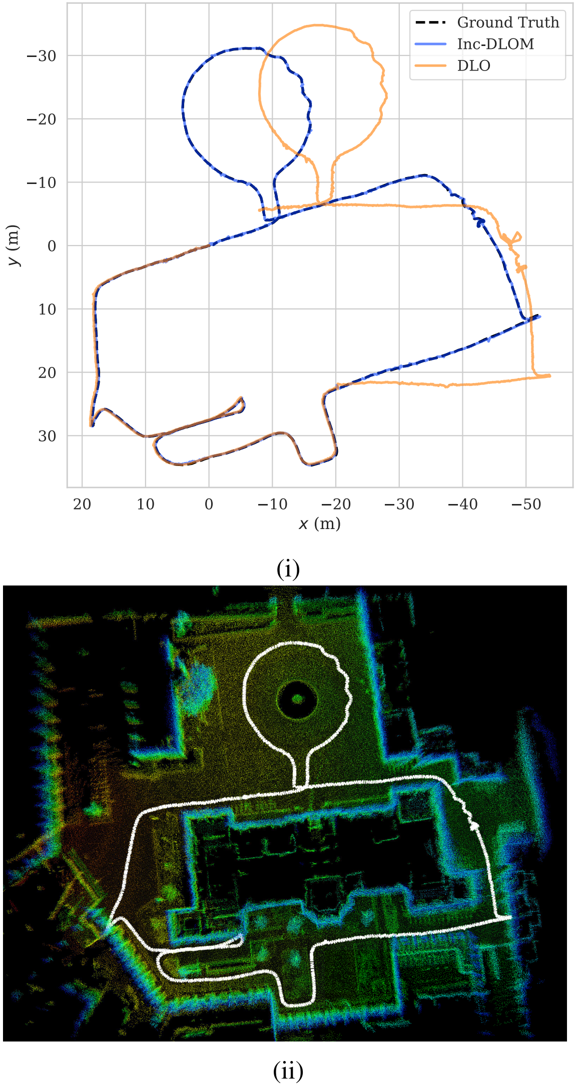


FIGURE 7. **Performance on NCD-Math-Hard. (i) Estimated trajectory (blue), comparing with the result from DLO (Orange). (ii) The mapping result from Inc-DLOM.**

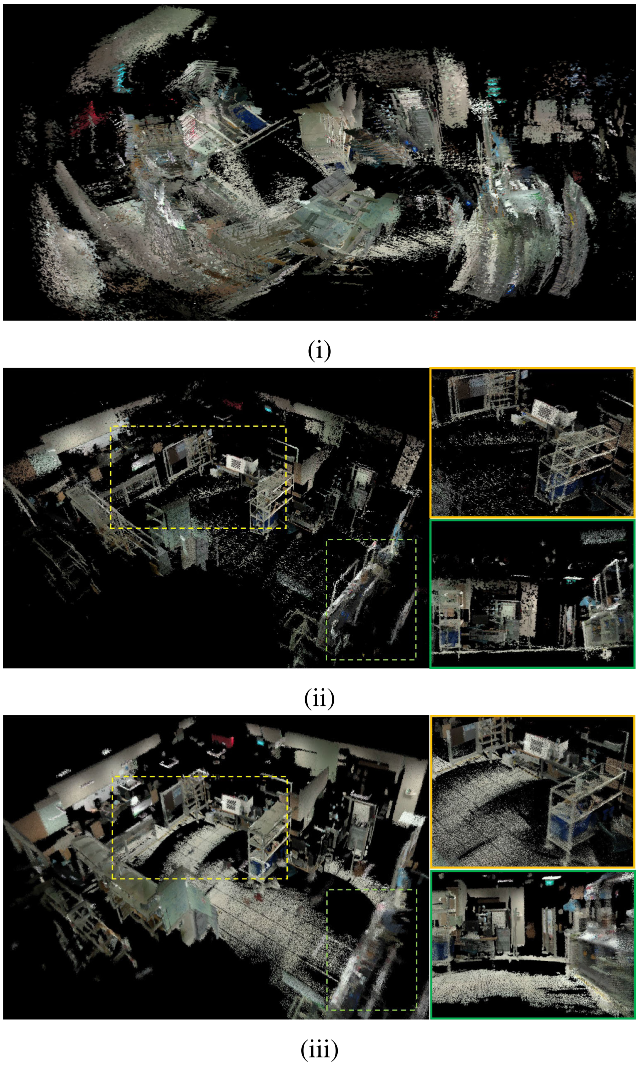


FIGURE 8. **RS-L515 Mapping Comparison. (i) HDL-SLAM failed to track the motion of a solid-state LiDAR. (ii) The mapping result from DLO is better than (i), but the zoomed views show many twisted objects mapping views. (iii) Inc-DLOM can achieve good mapping results.**

LINS are typical methods that fail on robotic platforms with high-frequency self-motions, while HDL-SLAM and DLO are direct LO methods for comparison.

In addition, we also evaluated our method on an extremely challenging dataset sequence, NCD-Math-Hard, which is full of rapid rotations in different locations and easy to lead tracking failure for SLAM algorithms. Fig. 7 (i) shows the estimated trajectory of Inc-DLOM (blue), in comparison with ground truth (dashed) and DLO result (orange), where the other selected methods failed. From the figure, it is obvious that DLO can track the motions in the beginning, however, the performance is affected when the LiDAR is conducting rapid rotations. On the contrary, our proposed Inc-DLOM can achieve robust performance even with vigorous motions and the trajectory is highly consistent with the ground truth. Fig. 7 (ii) shows the mapping result from Inc-DLOM.

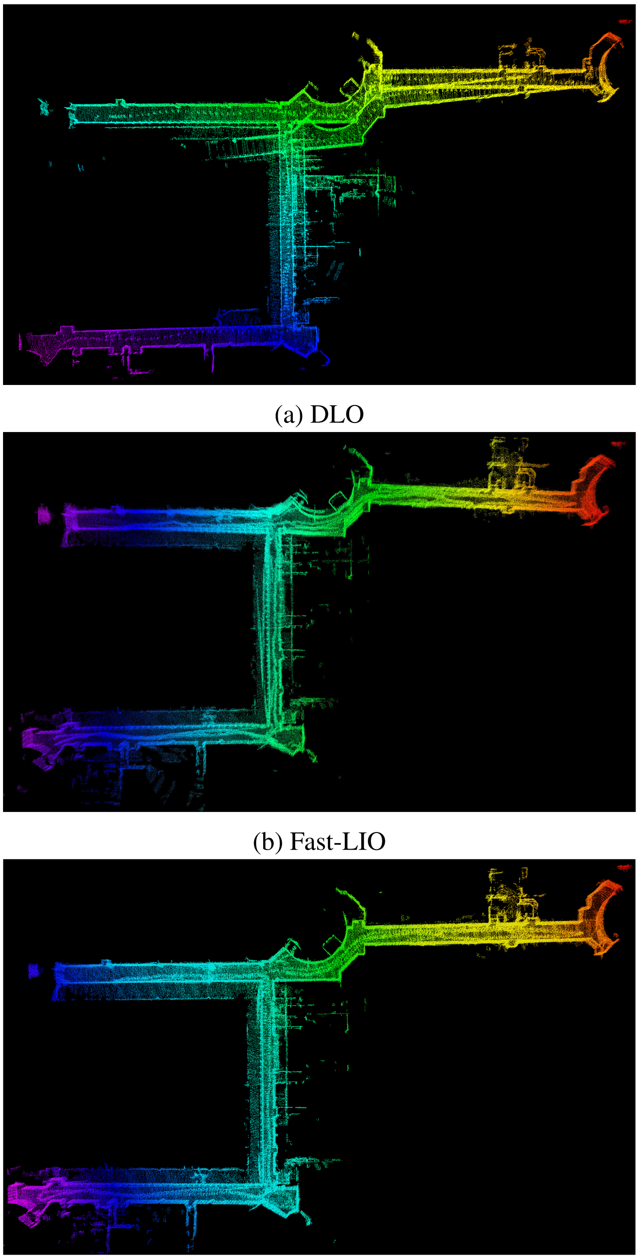
D. SOLID-STATE LIDAR SCENARIO

We also evaluate our Inc-DLOM on a dataset collected by an Intel Realsense L515 solid-state LiDAR (RS-L515) [10]. Compared with mechanical LiDARs, RS-L515 produces point clouds with a frequency of 30 Hz, smaller FoV, and higher spatial density as shown in Table 1. The data has different ranging distances and the points are not forming the spatial rings as mechanical LiDARs do. These kinds of unique information will make classical LO methods not applicable, e.g., Fast-LIO, LINS, and LIOM, since their feature extraction methods are highly dependent on the properties of mechanical LiDARs.

Therefore, we compare the mapping result of HDL-SLAM, DLO, and Inc-DLOM on the dataset as shown in Fig. 8. The mapping visualization of Inc-DLOM is shown in Fig. 8 (iii), in which two areas with distinctive features are zoomed. DLO can handle the tracking scenario of solid-state LiDAR,

TABLE 6. RMSE of ATE (m) evaluation of NCLT dataset and newer college dataset.

	NCLT_1	NCLT_2	NCLT_3	NCLT_4	NCLT_5	NCD_1	NCD_2	NCD_3	NCD_4
Inc-DLOM	**0.71**	**1.93**	**1.47**	**2.74**	**2.80**	**0.32**	**0.34**	**0.09**	**0.16**
Fast-LIO	0.87	2.35	1.61	2.74	3.13	1.28	1.06	0.31	0.49
LINS	28.30	151.22	fail	fail	fail	fail	fail	0.86	1.14
HDL-SLAM	28.93	38.48	82.29	fail	fail	2.57	1.64	0.97	2.47
LIOM	fail	fail	fail	fail	fail	fail	fail	0.48	2.08
DLO	1.56	2.37	7.26	11.84	4.72	0.43	0.96	0.17	0.31

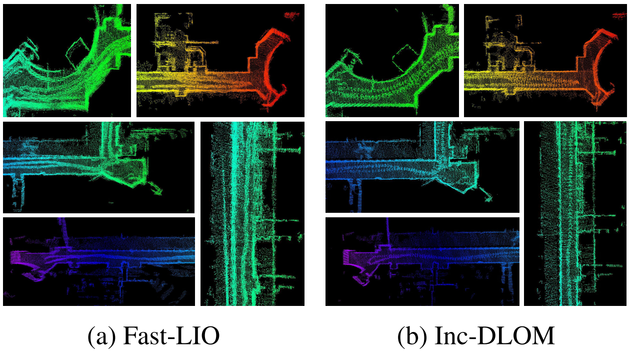


(a) DLO

(b) Fast-LIO

(c) Inc-DLOM

FIGURE 9. The point cloud mapping of different methods.



(a) Fast-LIO

(b) Inc-DLOM

FIGURE 10. Zoomed views of different parts of point cloud mapping.

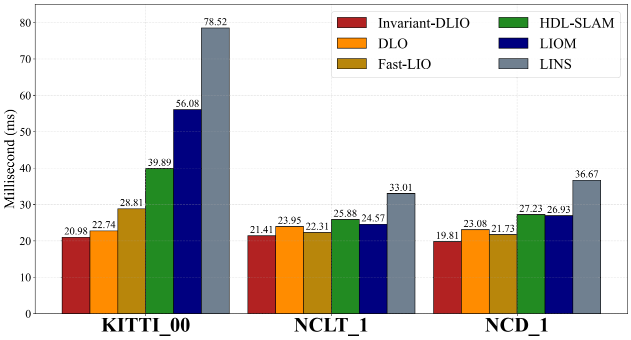


FIGURE 11. Computational Time per Frame (ms) Comparison.

but different zoomed views show that the mapping quality is not good. For point cloud mapping, shelves in the room are blurred and overlapped, and the view of the room also shows inconsistency. The mapping result of the proposed Inc-DLOM is much better as shown in Fig. 8 (iii), which demonstrates that it can successfully handle the general working scenarios on different robotics applications.

E. MAGNI ROBOT SCENARIO

As described in IV-A, we collected the dataset in a textureless environment with 6820 frames of point clouds, associated with IMU data of 200 Hz frequency. Only Fast-LIO and DLO can generate proper mapping performance for the corridor among all comparison methods. The point cloud mapping concatenated by the estimated poses is shown as an xy-plane view in Fig. 9. DLO can estimate the robot's motions properly in most parts of the trajectory but still generates inconsistency in the mapping. Both Fast-LIO and Inc-DLOM can provide proper estimation results, as their mappings show a proper view of the corridor. However, Fast-LIO's mapping consists of many blurred parts, which means the estimation error is preserved in the estimation. In comparison, our proposed Inc-DLOM can provide clearer and better point cloud mapping. The zoomed views of different parts of the point cloud are shown in Fig. 10.

F. COMPUTATIONAL EFFICIENCY

We evaluated the computational time per frame on KITTI_00, NCLT_1, and NCD_1 sequences in comparison with other methods. It is noteworthy that the various datasets employ different types of LiDAR sensors, resulting in significant variations in the size of the generated point cloud data. The figure illustrates that our proposed Inc-DLOM method achieves the fastest computational speed among all the selected methods. Furthermore, the computational time of Inc-DLOM remains relatively stable across different scenarios, in contrast to other methods such as HDL-SLAM, LIOM, and LINS, whose computation times increase significantly when processing larger datasets. Overall, the experimental results demonstrate that Inc-DLOM not only has the fastest processing speed but also consistently achieves over 50 Hz in a low-cost CPU environment. Its efficiency performance is notably more stable compared to other benchmarks.

V. CONCLUSION

We proposed a novel LiDAR Odometry framework, Incremental Direct LiDAR Odometry (Inc-DLOM) in this paper, which is accurate, computationally efficient, and generalized for multiple robotic platform working scenarios. Inc-DLOM contains point cloud deskew via a constant motion model, Voxel-to-Voxel GICP with the high computational efficiency of scan-to-scan registration method, and a novel scan-to-mapping refinement scheme based on Incremental Voxel Mapping (IVM). Different mapping operations including search, query, and access are within $\mathcal{O}(1)$ time complexity, allowing scan-to-mapping refinement scheme to avoid extensive time-consuming neighboring searches in the traditional point mapping. Eventually, the various components are more tightly integrated by the proposed Inc-DLOM framework. Evaluation of Inc-DLOM on multiple aspects demonstrates that it provides state-of-the-art localization for generalized robotic platforms with real-time performance. As future work, Inc-DLOM can be enhanced with additional sensor inputs, such as IMU, to implement a multiple-sensor fusion SLAM framework.

REFERENCES

[1] M. Haris and H. Nam, "Path planning optimization of smart vehicle with fast converging distance-dependent PSO algorithm," *IEEE Open J. Intell. Transp. Syst.*, vol. 5, pp. 726–739, 2024, doi: 10.1109/OJITS.2024.3486155.

[2] R. W. Wolcott and R. M. Eustice, "Visual localization within LiDAR maps for automated urban driving," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2014, pp. 176–183.

[3] M. Ji, W. Shi, Y. Cui, C. Liu, and Q. Chen, "Adaptive denoising-enhanced LiDAR odometry for degeneration resilience in diverse terrains," *IEEE Trans. Instrum. Meas.*, vol. 73, pp. 1–15, 2024.

[4] J. Yue, W. Wen, J. Han, and L.-T. Hsu, "3D point clouds data super resolution-aided LiDAR odometry for vehicular positioning in urban canyons," *IEEE Trans. Veh. Technol.*, vol. 70, no. 5, pp. 4098–4112, May 2021.

[5] J. Wang, M. Xu, G. Zhao, and Z. Chen, "Feature- and distribution-based LiDAR SLAM with generalized feature representation and heuristic nonlinear optimization," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–15, 2023.

[6] 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 *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Las Vegas, NV, USA, Oct. 2020, pp. 5135–5142.

[7] H. Ye, Y. Chen, and M. Liu, "Tightly coupled 3D LiDAR inertial odometry and mapping," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, Montreal, QC, Canada, May 2019, pp. 3144–3150.

[8] Y. Dong, L. Li, Y. Liu, S. Xu, and Z. Zuo, "PVE-LIOM: Pseudo-visual enhanced LiDAR-inertial odometry and mapping," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–13, 2023.

[9] K. Chang, J. Guo, Z. Wang, J. Cheng, J. Yu, G. Song, and J.-J. He, "SOA-integrated widely tunable laser array for all-solid LiDAR application," *IEEE Photon. Technol. Lett.*, vol. 36, no. 23, pp. 1381–1384, Dec. 1, 2024, doi: 10.1109/LPT.2024.3478811.

[10] H. Wang, C. Wang, and L. Xie, "Lightweight 3-D localization and mapping for solid-state LiDAR," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 1801–1807, Apr. 2021.

[11] X. Chen, I. Vizzo, T. Läbe, J. Behley, and C. Stachniss, "Range image-based LiDAR localization for autonomous vehicles," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Xi'an, China, May 2021, pp. 5802–5808, doi: 10.1109/ICRA48506.2021.9561335.

[12] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, Feb. 1992.

[13] Y. Chen and G. Medioni, "Object modelling by registration of multiple range images," *Image Vis. Comput.*, vol. 10, no. 3, pp. 145–155, 1992.

[14] A. Segal, D. Haehnel, and S. Thrun, "Generalized-ICP," *Robot. Sci., Syst.*, vol. 2, no. 4, p. 435, Jun. 2009.

[15] J. Serafin and G. Grisetti, "NICP: Dense normal based point cloud registration," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 742–749.

[16] S. Ramalingam and Y. Taguchi, "A theory of minimal 3D point to 3D plane registration and its generalization," *Int. J. Comput. Vis.*, vol. 102, nos. 1–3, pp. 73–90, 2013.

[17] S. D. Billings, E. M. Boctor, and R. H. Taylor, "Iterative most-likely point registration (IMLP): A robust algorithm for computing optimal shape alignment," *PLoS ONE*, vol. 10, no. 3, Mar. 2015, Art. no. e0117688.

[18] A. W. Fitzgibbon, "Robust registration of 2D and 3D point sets," *Image Vis. Comput.*, vol. 21, nos. 13–14, pp. 1145–1153, Dec. 2003.

[19] D. Lee, H. Lim, and S. Han, "GenZ-ICP: Generalizable and degeneracy-robust LiDAR odometry using an adaptive weighting," *IEEE Robot. Autom. Lett.*, vol. 10, no. 1, pp. 152–159, Jan. 2025, doi: 10.1109/LRA.2024.3498779.

[20] J.-E. Deschaud, "IMLS-SLAM: Scan-to-model matching based on 3D data," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 2480–2485.

[21] J. Zhang and S. Singh, "Low-drift and real-time LiDAR odometry and mapping," *Auto. Robots*, vol. 41, no. 2, pp. 401–416, Feb. 2017.

[22] T. Shan and B. Englot, "LeGO-LOAM: Lightweight and ground-optimized LiDAR odometry and mapping on variable terrain," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 4758–4765.

[23] P. Zhou, X. Guo, X. Pei, and C. Chen, "T-LOAM: Truncated least squares LiDAR-only odometry and mapping in real time," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022.

[24] H. Wang, C. Wang, C.-L. Chen, and L. Xie, "F-LOAM : Fast LiDAR odometry and mapping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2021, pp. 4390–4396.

[25] B. Zhou, J. Wu, Y. Pan, and C. Lu, "ATI-CTLO: Adaptive temporal interval-based continuous-time LiDAR-only odometry," *IEEE Robot. Autom. Lett.*, vol. 9, no. 12, pp. 11162–11169, Dec. 2024, doi: 10.1109/LRA.2024.3486233.

[26] K. Chen, B. T. Lopez, A. Agha-Mohammadi, and A. Mehta, "Direct LiDAR odometry: Fast localization with dense point clouds," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 2000–2007, Apr. 2022.

[27] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Commun. ACM*, vol. 18, no. 9, pp. 509–517, Sep. 1975.

[28] Y. Cheng and X. Tong, "Application of octree map in robotic environment reconstruction," in *Proc. IEEE 8th Annu. Int. Conf. CYBER Technol. Autom., Control, Intell. Syst. (CYBER)*, Tianjin, China, Jul. 2018, pp. 843–848.

[29] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000 km: The Oxford robotcar dataset," *Int. J. Robot. Res.*, vol. 36, no. 1, pp. 3–15, 2016.

[30] G. Kim, Y. S. Park, Y. Cho, J. Jeong, and A. Kim, "MulRan: Multimodal range dataset for urban place recognition," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Paris, France, May 2020, pp. 6246–6253.

[31] K. Koide, M. Yokozuka, S. Oishi, and A. Banno, "Voxelized GICP for fast and accurate 3D point cloud registration," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Xi'an, China, May 2021, pp. 11054–11059.

[32] J. L. Blanco and P. K. Rai. (2014). *NanoFLANN: AC Header-Only fork of FLANN A Library for Nearest Neighbor (NN) With Kd-Trees*. [Online]. Available: https://github.com/jlblancoc/nanoflann

[33] *Products Magni*. Ubiquity Robotics. Accessed: Jun. 12, 2024. [Online]. Available: https://www.ubiquityrobotics.com/products-magni/

[34] N. Carlevaris-Bianco, A. K. Ushani, and R. M. Eustice, "University of Michigan north campus long-term vision and LiDAR dataset," *Int. J. Robot. Res.*, vol. 35, no. 9, pp. 1023–1035, Aug. 2016.

[35] M. Ramezani, Y. Wang, M. Camurri, D. Wisth, M. Mattamala, and M. Fallon, "The newer college dataset: Handheld LiDAR, inertial and vision with ground truth," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Las Vegas, NV, USA, Oct. 2020, pp. 4353–4360.

[36] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.

[37] R. B. Rusu and S. Cousins, "3D is here: Point cloud library (PCL)," in *Proc. IEEE Int. Conf. Robot. Autom.*, Shanghai, China, May 2011, pp. 1–4, doi: 10.1109/ICRA.2011.5980567.

[38] W. Xu and F. Zhang, "FAST-LIO: A fast, robust LiDAR-inertial odometry package by tightly-coupled iterated Kalman filter," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 3317–3324, Apr. 2021.

[39] C. Qin, H. Ye, C. E. Pranata, J. Han, S. Zhang, and M. Liu, "LINS: A LiDAR-inertial state estimator for robust and efficient navigation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Paris, France, May 2020, pp. 8899–8906.

[40] K. Koide, J. Miura, and E. Menegatti, "A portable three-dimensional LiDAR-based system for long-term and wide-area people behavior measurement," *Int. J. Adv. Robotic Syst.*, vol. 16, no. 2, 2019, Art. no. 1729881419841532.

[41] M. Grupp. (2017). *Evo: Python Package for the Evaluation of Odometry and Slam*. [Online]. Available: https://github.com/MichaelGrupp/evo

[42] *MTN-FP650-Maxtang: Premier Supplier of A*. Accessed: Jul. 23, 2024. [Online]. Available: https://www.maxtangpc.com/show-106-194-1.html

KAIDUO FANG received the B.Eng. degree in vehicle engineering from Jilin University, Changchun, China, in 2018, and the M.S.E. degree in mechanical engineering and electrical and computer engineering from the University of Michigan, Ann Arbor, MI, USA, in 2021. He is currently pursuing the Ph.D. degree with the Department of Electrical and Electronic Engineering, The Hong Kong Polytechnic University. His research interests include autonomous driving, SLAM, and point cloud registration.

RUI SONG received the Ph.D. degree in instrument science and technology from Southeast University, in 2018. He is currently a Postdoctoral Researcher with the Department of Electrical and Electronic Engineering, The Hong Kong Polytechnic University. Before, he was an Assistant Research Fellow with the Institute of Robotics and Automatic Information Systems, Nankai University. In 2016, he was a Visiting Scholar with the University of Birmingham, U.K. His research interests include positioning and navigation for autonomous driving vehicles and mobile robots, SLAM, and multiple sensor fusion.

IVAN WANG-HEI HO (Senior Member, IEEE) received the B.Eng. and M.Phil. degrees in information engineering from The Chinese University of Hong Kong, Hong Kong, in 2004 and 2006, respectively, and the Ph.D. degree in electrical and electronic engineering from the Imperial College London, London, U.K., in 2010. In 2007, he spent a summer working at the IBM T. J. Watson Research Center, Hawthorne, NY, USA. After his Ph.D. graduation, he was with the System Engineering Initiative, Imperial College London, as a Postdoctoral Research Associate. He is currently an Associate Professor with the Department of Electrical and Electronic Engineering, The Hong Kong Polytechnic University, Hong Kong. In September 2010, he co-founded P2 Mobile Technologies Ltd. and served as the Chief Engineer. His research interests include wireless communications and networking, specifically in vehicular networks, intelligent transportation systems (ITS), and the Internet of Things (IoT). He primarily invented the MeshRanger series wireless mesh embedded system, which received the Silver Award in Best Ubiquitous Networking at Hong Kong ICT Awards 2012. His work on indoor positioning and IoT received a number of awards, including the Gold Medal at iENA 2019, the Gold Medal with the Organizers Choice Award at iCAN 2020, and the Gold Medal at the International Exhibition of Inventions Geneva, in 2021.