iG-LIO: An Incremental GICP-based Tightly-coupled LiDAR-inertial Odometry

Zijie Chen, Yong Xu, *Member, IEEE*, Shenghai Yuan, and Lihua Xie *Fellow, IEEE*

***Abstract*—This work proposes an incremental Generalized Iterative Closest Point (GICP) based tightly-coupled LiDAR-inertial odometry (LIO), iG-LIO, which integrates the GICP constraints and inertial constraints into a unified estimation framework. iG-LIO uses a voxel-based surface covariance estimator to estimate the surface covariances of scans, and utilizes an incremental voxel map to represent the probabilistic models of surrounding environments. These methods successfully reduce the time consumption of the covariance estimation, nearest neighbor search, and map management. Extensive datasets collected from mechanical LiDARs and solid-state LiDARs are employed to evaluate the efficiency and accuracy of the proposed LIO. Even though iG-LIO keeps identical parameters across all datasets, the results show that it is more efficient than Faster-LIO while maintaining comparable accuracy with state-of-the-art LIO systems. The source code for iG-LIO has been open-sourced on GitHub: https://github.com/zijiechenrobotics/ig_lio.**

***Index Terms*—SLAM, sensor fusion, LiDAR-inertial odometry.**

I. Introduction

THE fusion of LiDAR and inertial sensors for odometry is widely used in autonomous navigation systems, especially in unknown environments without absolute measurements (e.g., GNSS). An efficient and accurate LiDAR-inertial odometry (LIO) is crucial for safe navigation [1], the front end of simultaneous localization and mapping (SLAM) [2], and large-scale mapping [3].

In recent years, LO/LIOs have improved the efficiency and accuracy via map management and registration metrics. For map management, LOAM [4] and its variant [5], [6] organize the spatial structure of the local map by kd-tree. Since the map contains thousands to millions of points, rebuilding the spatial structure becomes time-consuming when the local map updates. FastLIO2 [7] proposes a novel incremental kd-tree (ikd-tree) to represent the spatial pattern of the local map. The ikd-tree enables dynamic insertion and rebalancing, which saves the time of reconstructing the kd-tree. Nevertheless, the search complexity of the nearest points on a kd-tree is $O(m\log n)$, where n is the number of points in the local map, and m is their dimension. It is challenging to perform in real-time when dealing with a large number of laser points. Voxelization provides an alternative and efficient approach to organizing the spatial structure of point clouds. It takes $O(1)$ query time to associate the nearest neighbor voxel. The voxel-based methods [8]–[13] split the local map into the voxel structure and achieve significantly faster speed in registration. To prevent divergence in narrow environments, AdaLIO [14] imports an adaptive strategy in Faster-LIO [8] and wins first place in the Hilti SLAM Challenge 2023.

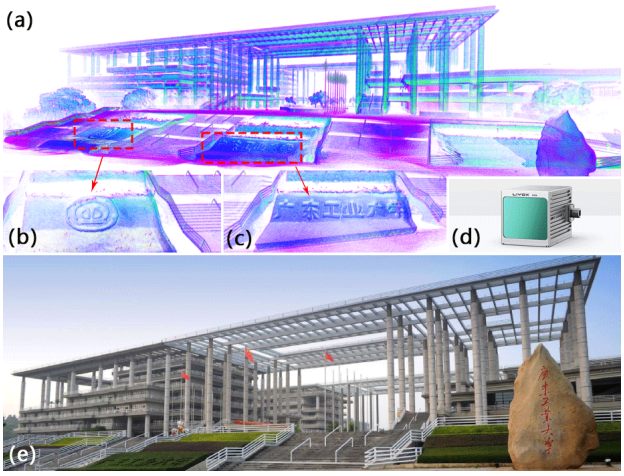

Fig. 1. The dense map (a) of the main gate of the Guangdong University of Technology (GDUT) (e) is reconstructed by iG-LIO with a handheld device (d). (b) and (c) show the reconstructed details of the map. The experimental video is available at https://youtu.be/zMktZdj4AAk.

The registration metric is one of the key components of LIO. Existing work includes the geometric features-based metrics [4]–[6], the dense surface representation-based metrics [15], [16], and the probabilistic distribution-based metrics [17], [18]. Generalized Iterative Closest Point (GICP) [19], which attaches a probabilistic model to Iterative Closest Point, is one of the widely used probabilistic distribution-based metrics. GICP estimates the surface covariance of each laser point and modifies its eigenvalues to achieve geometric feature-based metrics, including point-to-line, point-to-plane, and plane-to-plane. The surface covariance enables GICP to reduce the influence of incorrect correspondences and achieve accurate registration. VGICP [20] incorporates a voxel-based nearest neighbor search on GICP [19]. It is capable of processing 15,000 laser points at a rate of 30 Hz on the CPU. However,

Manuscript received: August, 13, 2023; Revised November, 5, 2023; Accepted December, 22, 2023.

This paper was recommended for publication by Editor Javier Civera upon evaluation of the Associate Editor and Reviewers' comments. This work was supported in part by the National Natural Science Foundation of China under Grant (62121004), and the Natural Science Foundation of Guangdong Province, China (2021B1515420008). (Corresponding author: Yong Xu.)

Zijie Chen and Yong Xu are with the Provincial Key Laboratory of Intelligent Decision and Cooperative Control, School of Automation, Guangdong University of Technology, Guangzhou 510006, China. zijiechenrobotics@foxmail.com, yxu@gdut.edu.cn.

Shenghai Yuan and Lihua Xie are with the School of Electrical and Electronic Engineering, Nanyang Technological University 63979, Singapore. {shyuan, elhxie}@ntu.edu.sg.

Digital Object Identifier (DOI): see top of this page.

integrating GICP into efficient and accurate LIO poses several challenges.

- Existing GICP-based LIOs [21]–[23] are not truly tightly-coupled with raw measurements. These LIOs are inadequate to maintain robustness in small field-of-view (FOV) LiDAR.
- The surface covariance estimation presented in the previous work [19], [20] is unsuitable for sparse and small FOV laser scans (e.g., solid-state LiDAR sampling at 100 Hz or operating indoors). The presence of far-apart laser points within these scans significantly affects the precision of the estimated surface covariance and the registration.
- Real-time registration using GICP, which relies on kd-tree for nearest neighbor search, becomes difficult when dealing with a dense scan.
- When GICP is extended to scan-to-map registration, the process of constructing a local map and estimating surface covariances becomes time-intensive.

Balancing efficiency in dense scans while preserving robustness in sparse and small FOV scans is challenging for the present GICP-based odometry [19]–[24]. This letter proposes iG-LIO, an incremental GICP-based tightly-coupled LiDAR-inertial odometry, to address the aforementioned challenges. Fig. 1 shows that iG-LIO is capable of constructing the fine structural details of the environment. The main contributions of this letter are summarized as follows.

- The GICP constraints are tightly-coupled with inertial measurement unit (IMU) constraints in a Maximum A Posteriori (MAP) estimation. The source code has been open-source on GitHub to benefit the community.
- A voxel-based surface covariance estimator (VSCE) is proposed to improve the efficiency and accuracy of the surface covariance estimation. Compared to the kd-tree based methods [19], [20], VSCE reduces processing time in dense scans (see Section III-A) while maintaining the accuracy of iG-LIO in sparse and small FOV scans (see Section III-B3, III-B5, III-C1).
- An incremental voxel map is designed to represent the probabilistic models of surrounding environments. Compared to non-incremental methods (e.g., DLIO [23]), it successfully reduces the time cost required for the nearest neighbor search and map management (see Section III-A).
- Extensive datasets collected from different FOV LiDARs are adopted to evaluate the efficiency and accuracy of the proposed iG-LIO. Even though iG-LIO keeps identical parameters across all datasets, the results show that it is more efficient than Faster-LIO and achieves competitive performance compared to state-of-the-art LIO systems.

The rest of this letter is structured as follows. Section II explains the implementation details of iG-LIO, and Section III presents the efficiency and accuracy of iG-LIO through extensive experiments. Finally, Section IV concludes this letter.

TABLE I
SOME IMPORTANT NOTATIONS

Notation	Explanation
$\mathrm{Exp}(\cdot)/\mathrm{Log}(\cdot)$	The association between Lie algebra and rotation [25].
$\mathbf{r}, \mathbf{J}, \mathbf{\Omega}$	The residual, Jacobian, and information matrix in optimization.
$\mathbf{R}_b^w, \mathbf{t}_b^w, \mathbf{v}_b^w$	The rotation, position, and velocity of the body (IMU) frame with respect to the world frame.
b_k	The IMU body frame at time k.
$\mathcal{M}[i].(\cdot)$	The element $(\cdot)$ of the i-th voxel in the voxel map $\mathcal{M}$.
$\mathcal{P}$	The point set of the LiDAR scan.
$\delta(\cdot)$	The error state of state $(\cdot)$.
$\check{(\cdot)}, \hat{(\cdot)}$	The prior and posterior estimation of state $(\cdot)$.
$(\cdot)^n$	The n-th update of state $(\cdot)$ in optimization, e.g., $\hat{\mathbf{R}}_{b_k}^{w,n}$ denotes the posterior rotation from the body frame at time k to the world frame after the n-th update.

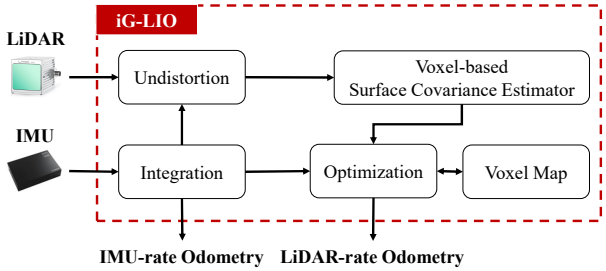

Fig. 2. The block diagram of iG-LIO, it receives LiDAR and IMU measurements and outputs IMU-rate odometry and LiDAR-rate odometry.

II. INCREMENTAL GICP-BASED TIGHTLY COUPLED LIDAR-INERTIAL ODOMETRY

A. System Overview

Table I shows the important notations in this letter. The sensing system state $\mathbf{x}$ is expressed as

$$\mathbf{x} \doteq \begin{bmatrix} \mathbf{R}_b^w & \mathbf{t}_b^w & \mathbf{v}_b^w & \mathbf{b}_\alpha & \mathbf{b}_g \end{bmatrix}, \tag{1}$$

where $\mathbf{b}_\alpha \in \mathbb{R}^3$ and $\mathbf{b}_g \in \mathbb{R}^3$ are the biases of a three-axis gyroscope and a three-axis accelerometer.

Fig. 2 shows a block diagram of the proposed odometry. Starting with an IMU integration, iG-LIO compensates for a motion distortion of the scan, generates a prior constraint, and outputs an IMU-rate odometry. Then, the voxel-based surface estimator samples the undistorted scan to estimate the surface covariance of each point. Next, the scan implements the nearest neighbor search in the voxel map via hash indexes to achieve GICP constraints. Both the prior and the GICP constraints are integrated into a MAP to estimate the sensing system state. Finally, the scan is incrementally inserted into the voxel map to update the distribution of each voxel grid.

The MAP estimation is formulated as

$$\min_{\mathbf{x}_k} \left\{ \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{M}} \left\| \mathbf{r}_{i,j}^{GICP} \right\|^2_{\mathbf{\Omega}_{i,j}^{GICP}} + \left\| \mathbf{r}_k^{prior} \right\|^2_{\mathbf{\Omega}_k^{prior}} \right\}, \tag{2}$$

where $\|\mathbf{r}\|^2_{\mathbf{\Omega}} = \mathbf{r}^\top \mathbf{\Omega} \mathbf{r}$. The residuals $\mathbf{r}^{GICP}$ and $\mathbf{r}^{prior}$ are computed from the GICP constraints (see Section II-D) and the prior constraints (see Section II-B). The matrices $\mathbf{\Omega}^{GICP}$ and $\mathbf{\Omega}^{prior}$ are the corresponding information matrices.

The MAP estimation (2) only maintains the state of a single moment, which can be regarded as tightly coupling IMU measurements with GICP registration. The tightly-coupled mode improves the robustness and accuracy of the state estimation in degenerate scenarios (e.g., LiDAR sampling at 100 Hz). Specifically, unobservable states in the MAP estimation are constrained by IMU measurements, instead of converging to a poor solution in the loosely-coupled mode (e.g., the IMU measurements provide an initial guess for pure GICP registration [24]).

B. IMU Constraints

The IMU integration predicts the prior estimation $\check{\mathbf{x}}$ and propagates the covariance of the error state $\delta\mathbf{x}$. Define a discretization time interval as $\Delta t \doteq t_{k+1} - t_k$, iG-LIO predicts the prior estimation $\check{\mathbf{x}}$ via midpoint method,

$$\begin{aligned}
\check{\mathbf{R}}^w_{b_{k+1}} &= \check{\mathbf{R}}^w_{b_k} \mathrm{Exp}(\bar{\boldsymbol{\omega}} \Delta t),\ \bar{\boldsymbol{\omega}} = \frac{\boldsymbol{\omega}_{m,k} + \boldsymbol{\omega}_{m,k+1}}{2} - \mathbf{b}_{g,k}, \\
\check{\mathbf{v}}^w_{b_{k+1}} &= \check{\mathbf{v}}^w_{b_k} + \bar{\boldsymbol{\alpha}} \Delta t,\ \check{\mathbf{t}}^w_{b_{k+1}} = \check{\mathbf{t}}^w_{b_k} + \check{\mathbf{v}}^w_{b_k} \Delta t + \frac{1}{2} \bar{\boldsymbol{\alpha}} \Delta t^2, \\
\bar{\boldsymbol{\alpha}} &= \frac{\mathbf{R}_k + \mathbf{R}_{k+1}}{2} - \mathbf{g}^w,\ \mathbf{R}_k = \check{\mathbf{R}}^w_{b_k} (\boldsymbol{\alpha}_{m,k} - \mathbf{b}_{\alpha,k}), \\
\mathbf{R}_{k+1} &= \check{\mathbf{R}}^w_{b_{k+1}} (\boldsymbol{\alpha}_{m,k+1} - \mathbf{b}_{\alpha,k}),
\end{aligned} \tag{3}$$

where $\boldsymbol{\omega}_m$ and $\boldsymbol{\alpha}_m$ are IMU raw measurements. The vector $\mathbf{g}^w$ is a constant gravity vector in the world frame.

To express the uncertainty of prior constraints in the MAP estimation (2), iG-LIO propagates the covariance of the error state $\delta\mathbf{x}$. This process is similar to the forward propagation in the iterated error state Kalman filter [26],

$$\delta\mathbf{x}_{k+1} = \mathbf{F}_k \delta\mathbf{x}_k + \mathbf{G}_k \mathbf{w}_k, \tag{4}$$

where

$$\begin{aligned}
\delta\mathbf{x}_k &= \begin{bmatrix} \delta\boldsymbol{\theta}_k^\top & \delta\mathbf{t}_k^\top & \delta\mathbf{v}_k^\top & \delta\mathbf{b}_{g,k}^\top & \delta\mathbf{b}_{\alpha,k}^\top \end{bmatrix}^\top, \\
\mathbf{w}_k &= \begin{bmatrix} \mathbf{n}_{\alpha,k}^\top & \mathbf{n}_{g,k}^\top & \mathbf{n}_{\alpha,k+1}^\top & \mathbf{n}_{g,k+1}^\top & \mathbf{n}_{b_\alpha}^\top & \mathbf{n}_{b_g}^\top \end{bmatrix}^\top.
\end{aligned}$$

The vectors $\mathbf{n}_\alpha$, $\mathbf{n}_g$, $\mathbf{n}_{b_\alpha}$, and $\mathbf{n}_{b_g}$ are white Gaussian noises of the acceleration measurement $\boldsymbol{\alpha}_m$, gyroscope measurement $\boldsymbol{\omega}_m$, acceleration bias $\mathbf{b}_\alpha$, and gyroscope bias $\mathbf{b}_g$, respectively. The transition matrices $\mathbf{F}_k$ and $\mathbf{G}_k$ are calculated based on the Appendices E in [27]. Then, the covariance $\check{\mathbf{P}}_{k+1}$ of the error state $\delta\mathbf{x}$ is computed iteratively by

$$\check{\mathbf{P}}_{k+1} = \mathbf{F}_k \check{\mathbf{P}}_k \mathbf{F}_k^\top + \mathbf{G}_k \mathbf{Q} \mathbf{G}_k^\top, \tag{5}$$

where $\mathbf{Q}$ is a covariance of the process noise.

iG-LIO integrates the IMU inputs until the next round of the LiDAR measurements arrives. The prior constraints in (2) are defined as

$$\mathbf{r}_k^{prior} \doteq \begin{bmatrix} \mathrm{Log}(\check{\mathbf{R}}^{w\ \top}_{b_k} \mathbf{R}^w_{b_k}) \\ \mathbf{t}^w_{b_k} - \check{\mathbf{t}}^w_{b_k} \\ \mathbf{v}^w_{b_k} - \check{\mathbf{v}}^w_{b_k} \\ \mathbf{b}_{\alpha,k} - \check{\mathbf{b}}_{\alpha,k} \\ \mathbf{b}_{g,k} - \check{\mathbf{b}}_{g,k} \end{bmatrix}, \mathbf{\Omega}_k^{prior} \doteq \check{\mathbf{P}}_k^{-1}. \tag{6}$$

C. Voxel Map

After each measurement update, iG-LIO incrementally adds the laser point to the voxel map and updates the probabilistic model of each voxel to represent the surrounding environment. This method improves the efficiency and accuracy as follows.

- The query time of the voxel-based nearest neighbor search is $O(1)$. Compared to the kd-tree-based method, this approach is computationally cheap and easily scalable for parallel processing [20].
- The voxel map enables iG-LIO to perform GICP in scan-to-map registration. It enhances the robustness and accuracy of state estimation compared to previous work using GICP in scan-to-scan registration [19], [20].

1) Map Construction: The voxel map characterizes each voxel $\mathcal{M}[i]$ by five elements: the number of points N, the sum of points $\mathbf{p}$, the sum of outer products $\mathbf{C}$, the mean of voxel $\boldsymbol{\mu}$, and the covariance of voxel $\mathbf{\Sigma}$. The structure of the voxel map is implemented via `std::unordered_map` in C++, and the hash index is calculated as [28],

$$\begin{aligned}
\mathbf{g} &= \begin{bmatrix} g_x & g_y & g_z \end{bmatrix}^\top \\
&= \begin{bmatrix} \mathrm{floor}(\frac{p_x}{s}) & \mathrm{floor}(\frac{p_y}{s}) & \mathrm{floor}(\frac{p_z}{s}) \end{bmatrix}^\top, \\
hash_index &= (g_x n_x) \mathbf{xor} (g_y n_y) \mathbf{xor} (g_z n_z) \mathbf{mod} \mathcal{M}_{max},
\end{aligned} \tag{7}$$

where p_x, p_y, and p_z are the position of the laser point $\mathbf{p} \in \mathbb{R}^3$. The function $\mathrm{floor}(x)$ computes the largest integer value not greater than x. The parameter s is a voxel resolution, and $\mathcal{M}_{max}$ is the maximum size in the voxel map. The values n_x, n_y, and n_z are large prime numbers (e.g., $n_x = 73856093, n_y = 83492791$, and $n_z = 471945$).

2) Map Update: The voxel map increment consists of two steps: update and delete. For the update step, the undistorted scan $\bar{\mathcal{P}}_k$ is transformed into the world frame via the posterior estimation $\hat{\mathbf{x}}_k$. The laser points in $\bar{\mathcal{P}}_k$ are subsequently stored into an active list $\mathcal{V}$ based on their hash indexes in (7). For each hash index within the active list $\mathcal{V}$, the number of laser points $\tilde{N}_i$, the sum of points $\tilde{\mathbf{p}}_i$, and the sum of outer products $\tilde{\mathbf{C}}_i$ are computed to update the voxel $\mathcal{M}[i]$,

$$\begin{aligned}
&\mathcal{M}[i].N = \mathcal{M}[i].N + \tilde{N}_i,\ \mathcal{M}[i].\mathbf{p} = \mathcal{M}[i].\mathbf{p} + \tilde{\mathbf{p}}_i, \\
&\mathcal{M}[i].\mathbf{C} = \mathcal{M}[i].\mathbf{C} + \tilde{\mathbf{C}}_i,\ \mathcal{M}[i].\boldsymbol{\mu} = \mathcal{M}[i].\mathbf{p} / \mathcal{M}[i].N, \\
&\mathcal{M}[i].\mathbf{\Sigma} = \frac{1}{\mathcal{M}[i].N + 1} (\mathcal{M}[i].\mathbf{C} - \mathcal{M}[i].\mathbf{p} \cdot \mathcal{M}[i].\boldsymbol{\mu}^\top).
\end{aligned} \tag{8}$$

To improve the symmetry of the metric [19], GICP modifies the covariance to approximate a local planar,

$$\mathcal{M}[i].\mathbf{\Sigma} = \mathbf{U} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \epsilon \end{bmatrix} \mathbf{V}^\top, \tag{9}$$

where the matrices $\mathbf{U}$ and $\mathbf{V}$ are the SVD decomposition results of $\mathcal{M}[i].\mathbf{\Sigma} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$. The scalar $\epsilon = 0.001$ denotes the covariance along the normal.

The goal of the delete step is to strike a balance between the storage space of the voxel map and the accuracy of the registration. Since the nearest neighbor search of the registration is only performed within a certain range of the

Algorithm 1: Updating and deleting in the voxel map

Input: Undistorted LiDAR scan $\bar{\mathcal{P}}_k$, current posterior estimation $\hat{\mathbf{x}}_k$, voxel map $\mathcal{M}$.
Output: voxel map $\mathcal{M}$.

Transform $\bar{\mathcal{P}}_k$ in world frame via $\hat{\mathbf{x}}_k$ and achieve $\bar{\mathcal{P}}_k^w$;
for *each point* $\mathbf{p}_i$ *in* $\bar{\mathcal{P}}_k^w$ **do**
  Compute $hash\_index$ of $\mathbf{p}_i$ via (7);
  $\mathcal{V} = \mathcal{V} \cup (idx = hash\_index, \boldsymbol{\mu} = \mathbf{p}_i)$;
end
Sort $\mathcal{V}$ via idx element;
for $i \in [0, ..., \text{size}(\mathcal{V})]$ **do**
  $\tilde{\boldsymbol{\mu}} = \mathbf{0}, \tilde{\mathbf{C}} = \mathbf{0}, \tilde{N} = 0$;
  for $j \in [i, ..., \text{size}(\mathcal{V})]$ **do**
    if $\mathcal{V}[i].idx == \mathcal{V}[j].idx$ **then**
      $\tilde{\boldsymbol{\mu}} = \tilde{\boldsymbol{\mu}} + \mathcal{V}[j].\boldsymbol{\mu}$, $\tilde{N} = \tilde{N} + 1$;
      $\tilde{\mathbf{C}} = \tilde{\mathbf{C}} + \mathcal{V}[j].\boldsymbol{\mu} \cdot \mathcal{V}[j].\boldsymbol{\mu}^\top$;
    else
      break;
    end
  end
  if *find* $\mathcal{V}[i].idx$ *in* $\mathcal{M}$ **then**
    Update voxel $\mathcal{M}[\mathcal{V}[i].idx]$ distribution via (8);
  else
    Create new voxel $\mathcal{M}_{new}(N = 0, \mathbf{p} = \mathbf{0}, \mathbf{C} = \mathbf{0}, \boldsymbol{\mu} = \mathbf{0}, \boldsymbol{\Sigma} = \mathbf{0})$;
    Update voxel $\mathcal{M}_{new}$ distribution via (8);
    $\mathcal{M} = \mathcal{M} \cup \mathcal{M}_{new}$;
    if $\text{size}(\mathcal{M}) > \mathcal{M}_{max}$ **then**
      Delete voxel via LRU cache strategy;
    end
  end
  $i = j$;
end

sensing system, the voxel map maintains a specific number of voxels via the LRU cache strategy [8].

The detailed process for updating and deleting in the voxel map can be found in the Algorithm 1.

D. Measurement Update

1) Voxel-based Surface Covariance Estimator (VSCE): The scan preparation in GICP registration involves three steps. A scan $\mathcal{P}_k$ undergoes a motion compensation to obtain an undistorted scan $\bar{\mathcal{P}}_k$. Subsequently, a voxel grid filter is applied to downsample the scan $\bar{\mathcal{P}}_k$ to $\bar{\mathcal{P}}'_k = \{\mathbf{p}_0, ..., \mathbf{p}_N\}$, which reduces sensor noise and ensures an adequate spatial pattern of the laser points. Finally, a set of covariances $\mathcal{C}_k = \{\boldsymbol{\Sigma}_0, ..., \boldsymbol{\Sigma}_N\}$ corresponding to the downsampled scan $\bar{\mathcal{P}}'_k$ is estimated using a kd-tree based neighbor search.

The voxel structure in the voxel grid filter allows VSCE to combine the downsampling and the surface covariance estimation. Specifically, VSCE estimates the covariance of each laser point based on the 26 neighboring voxels instead of the 20 closest points as in GICP [19]. This choice improves the registration accuracy in the sparse and small FOV scans (see Section III-C3).

2) GICP Constraints: The GICP constraints are modeled as distribution-to-distribution distances. During the nearest neighbor search, the point $\mathbf{p}_i$ is projected into the voxel map $\mathcal{M}$ via the hash index (7). The residual and the information matrix of the laser point p_i and voxel $\mathcal{M}[j]$ are defined as

$$\begin{aligned} \mathbf{r}_{i,j}^{GICP} &\doteq \mathcal{M}[j].\boldsymbol{\mu} - \mathbf{R}_{b_k}^w(\mathbf{R}_l^b \mathbf{p}_i + \mathbf{t}_l^b) - \mathbf{t}_{b_k}^w, \\ \boldsymbol{\Omega}_{i,j}^{GICP} &\doteq \sigma_{GICP}(\mathcal{M}[j].\boldsymbol{\Sigma} + \mathbf{R}_{b_k}^w \boldsymbol{\Sigma}_i \mathbf{R}_{b_k}^{w\top})^{-1}, \end{aligned} \tag{10}$$

where $\mathbf{R}_l^b$ and $\mathbf{t}_l^b$ are the extrinsic parameters between the LiDAR and the IMU. The matrix $\boldsymbol{\Sigma}_i$ represents the surface covariance of the point $\mathbf{p}_i$. The parameter σ_{GICP} denotes the weight of the GICP constraint.

In practice, iG-LIO modifies the surface covariance $\boldsymbol{\Sigma}_i$ to obtain the point-to-plane and plane-to-plane metrics. This modification is necessary since VSCE is unable to compute a stable surface covariance with fewer than five closest points. If the covariance cannot be estimated, iG-LIO sets the corresponding covariance in (10) to be $\boldsymbol{\Sigma}_i = \mathbf{0}$. This modification leads to a degradation from the plane-to-plane metric to the point-to-plane metric [19].

Additionally, iG-LIO employs a chi-squared test to remove outliers that could bring down the state estimation accuracy. The outliers include unsuitable local planar approximations and mismatched feature correspondences.

E. The iG-LIO Algorithm

This subsection aims to solve the MAP estimation presented in (2) and summarize the iG-LIO algorithm. The MAP estimation (2) is an unconstrained nonlinear least squares problem that can be solved by the Gauss-Newton method. The optimal $\delta\hat{\mathbf{x}}^n$ of the n-th iteration is computed as

$$\underbrace{\sum_i \mathbf{J}_i^\top \boldsymbol{\Omega}_i \mathbf{J}_i}_{\mathbf{H}^n} \delta\hat{\mathbf{x}}^n = \underbrace{\sum_i -\mathbf{J}_i^\top \boldsymbol{\Omega}_i \mathbf{r}_i}_{\mathbf{b}^n}, \tag{11}$$

where $\mathbf{J}_i$ is the Jacobian matrix of the residual $\mathbf{r}_i$ with respect to $\delta\hat{\mathbf{x}}^n$, and $\boldsymbol{\Omega}_i$ is the information matrix of the residual $\mathbf{r}_i$. In the case of the prior constraint, the Jacobian matrix is

$$\mathbf{J}_k^{prior} = \frac{\partial \mathbf{r}_k^{prior}}{\partial \delta\mathbf{x}} = \begin{bmatrix} \mathbf{J}_r^{-1}(\text{Log}(\check{\mathbf{R}}_{b_k}^{w\top} \mathbf{R}_{b_k}^w)) & \mathbf{0}_{3\times 12} \\ \mathbf{0}_{12\times 3} & \mathbf{I}_{12\times 12} \end{bmatrix}, \tag{12}$$

where $\mathbf{J}_r^{-1}(\cdot)$ is the inverse of the right Jacobian in [25]. For the GICP constraints, the Jacobian matrix is computed as

$$\mathbf{J}_{i,j}^{GICP} = \frac{\partial \mathbf{r}_{i,j}^{GICP}}{\partial \delta\mathbf{x}} = \begin{bmatrix} \mathbf{R}_{b_k}^w [\mathbf{R}_l^b \mathbf{p}_i + \mathbf{t}_l^b]_\times & -\mathbf{I} & \mathbf{0}_{3\times 9} \end{bmatrix}. \tag{13}$$

where $[\cdot]_\times$ is the skew operator in [27].

Assuming that the Guass-Newton method converges after $n \geq 0$ iterations, the measurement update observes the error state distribution $\delta\mathbf{x}_k \sim \mathcal{N}(\delta\hat{\mathbf{x}}^n, (\mathbf{H}^n)^{-1})$ in the tangent space of $\hat{\mathbf{x}}_k^n$ [29], [30]. However, the propagation in (5) requires the distribution $\delta\mathbf{x}_k \sim \mathcal{N}(\mathbf{0}, \hat{\mathbf{P}}_k)$ in the tangent space of $\hat{\mathbf{x}}_k^{n+1}$. To ensure the consistency of the covariance propagation, it is necessary to reset the covariance of the error state $\delta\mathbf{x}_k$ after the measurement update [27], [29]. The covariance of $\delta\mathbf{x}_k$ is

$$\hat{\mathbf{P}}_k = \mathbf{L}_k (\mathbf{H}^n)^{-1} \mathbf{L}_k^\top, \tag{14}$$

Algorithm 2: iG-LIO

Input: Last posterior estimation $\hat{\mathbf{x}}_k$ and covariance $\hat{\mathbf{P}}_k$, IMU measurements $\boldsymbol{\alpha}_{m,k\sim k+1}, \boldsymbol{\omega}_{m,k\sim k+1}$, LiDAR scan $\mathcal{P}_{k+1}$.

Output: Posterior estimation $\hat{\mathbf{x}}_{k+1}$ and covariance $\hat{P}_{k+1}$.

Integrate IMU measurements $\boldsymbol{\alpha}_{m,k\sim k+1}, \boldsymbol{\omega}_{m,k\sim k+1}$ to achieve prior estimation $\check{\mathbf{x}}_{k+1}$ and $\check{\mathbf{P}}_{k+1}$ via (3) (5);
Compensate scan $\mathcal{P}_{k+1}$ to obtain $\bar{\mathcal{P}}_{k+1}$;
Obtain $\bar{\mathcal{P}}'_{k+1}$ and $\mathcal{C}_{k+1} = \{\boldsymbol{\Sigma}_0, ..., \boldsymbol{\Sigma}_n\}$ via VSCE;
repeat
  for *each point* $\mathbf{p}_i$ *in* $\bar{\mathcal{P}}'_{k+1}$ **do**
    Compute $hash\_index$ of $\mathbf{p}_i$ via (7);
    if *find* $hash\_index$ *in* $\mathcal{M}$ **then**
      Add GICP association to the list $\mathcal{F}$;
    end
  end
  $\mathbf{H}^n = \mathbf{0}, \mathbf{b}^n = \mathbf{0}$;
  for *each element in* $\mathcal{F}$ **do**
    Compute $\mathbf{r}_{i,j}^{GICP}$ and $\boldsymbol{\Omega}_{i,j}^{GICP}$ via (10);
    if *pass the* χ^2 *test* **then**
      Compute Jacobian $\mathbf{J}_{i,j}^{GICP}$ via (13);
      $\mathbf{H}^n = \mathbf{H}^n + \mathbf{J}_{i,j}^{GICP^\top} \boldsymbol{\Omega}_{i,j}^{GICP} \mathbf{J}_{i,j}^{GICP}$;
      $\mathbf{b}^n = \mathbf{b}^n - \mathbf{J}_{i,j}^{GICP^\top} \boldsymbol{\Omega}_{i,j}^{GICP} \mathbf{r}_{i,j}^{GICP}$;
    end
  end
  Compute $\mathbf{r}_k^{prior}$, $\boldsymbol{\Omega}_k^{prior}$, and $\mathbf{J}_k^{prior}$ via (6) (12);
  $\mathbf{H}^n = \mathbf{H}^n + \mathbf{J}_k^{prior^\top} \boldsymbol{\Omega}_k^{prior} \mathbf{J}_k^{prior}$;
  $\mathbf{b}^n = \mathbf{b}^n - \mathbf{J}_k^{prior^\top} \boldsymbol{\Omega}_k^{prior} \mathbf{r}_k^{prior}$;
  Solve $\mathbf{H}^n \delta\hat{\mathbf{x}}^n = \mathbf{b}^n$;
  Update $\hat{\mathbf{x}}_{k+1}^n$ by $\delta\hat{\mathbf{x}}^n$ to obtain $\hat{\mathbf{x}}_{k+1}^{n+1}$, $n = n + 1$;
until $||\delta\hat{\mathbf{x}}^n|| < \epsilon$;
$\hat{\mathbf{x}}_{k+1} = \hat{\mathbf{x}}_{k+1}^n, \hat{\mathbf{P}}_{k+1} = \mathbf{L}_{k+1}(\mathbf{H}^n)^{-1}\mathbf{L}_{k+1}^\top$ via (15);
Update voxel map via Algorithm 1.

where the projection matrix $\mathbf{L}_k$ is computed following the derivation in [27],

$$\mathbf{L}_k = \begin{bmatrix} \mathbf{I} - \frac{1}{2}[\delta\hat{\boldsymbol{\theta}}^n]_\times & \mathbf{0}_{3\times 12} \\ \mathbf{0}_{12\times 3} & \mathbf{I}_{12\times 12} \end{bmatrix}. \tag{15}$$

Finally, the process of iG-LIO is summarized in Algorithm 2.

III. Experiments

The proposed framework is evaluated in six different datasets, including NCLT [31], the Newer College dataset (NCD) [32], ULHK [33], Botanic Garden (BG) [34], AVIA (from FastLIO2 [7] and r3live [35]), and self-collected GDUT. All sequences employ abbreviations due to space limitations, further details are provided in the GitHub page[1].

To demonstrate the efficiency and accuracy, iG-LIO is compared with state-of-the-art LIO systems: FastLIO2 [7], Faster-LIO [8], and DLIO [23]. Since DLIO assumes that the input scans are collected by a 360° mechanical LiDAR, it is not evaluated in AVIA and GDUT. VoxelMap [9] is another related work that retains the probability distribution in the voxel map. However, it is not included in the experiments because its open-source version removes the fusion of IMU.

[1]https://github.com/zijiechenrobotics/ig_lio

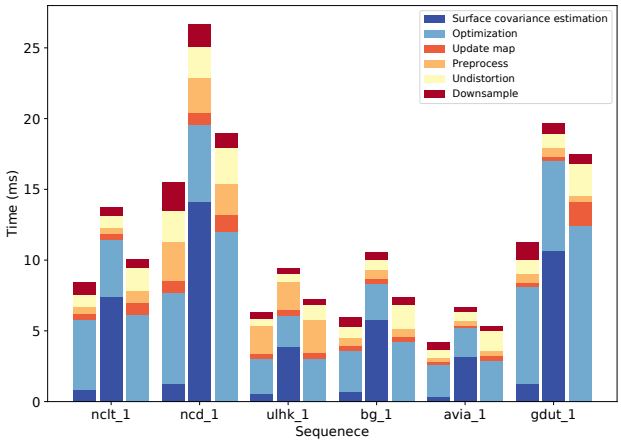

Fig. 3. The average runtime per step for iG-LIO (left), iG-LIO* (middle), and Faster-LIO (right).

Additionally, three ablation experiments are conducted to verify the efficiency of VSCE, and to compare the performance of three constraints: GICP, Normal Distributions Transform (NDT), and VGICP. All the experiments are executed on an Intel i7-10875H CPU (2.30 GHz × 16 cores) with 32 GB RAM using the robot operation system (ROS) in Ubuntu 18.04.

A. Efficiency

The efficiency of LIO is evaluated by the average runtime of each scan and the average number of effective feature points. In all sequences, Faster-LIO, FastLIO2, and DLIO maintain their default configurations. For iG-LIO, the voxel grid filter resolution and the voxel map resolution are set to be $0.5m$ for all sequences, allowing VSCE to estimate surface covariance via 26 neighboring $0.5m$ voxels. As shown in Table II, Faster-LIO exhibits slightly faster performance than iG-LIO in the 100 Hz datasets (avia_2 and avia_3). However, iG-LIO is $1.2 \sim 1.5$ times faster than Faster-LIO, $2.3 \sim 2.7$ times faster than FastLIO2, and $1.5 \sim 3.5$ times faster than DLIO in other datasets. Except for NCD datasets, DLIO takes more processing time than FastLIO2 since it rebuilds the submap of non-repeating environments.

Fig. 3 shows a significant difference in the optimization and undistortion between iG-LIO and Faster-LIO. While both of them employ `std::for_each` to execute `for` statements, iG-LIO utilizes `tbb::parallel_reduce` to parallelize the accumulation of Hessian matrices and residuals, thus achieving better efficiency. Regarding undistortion, iG-LIO enhances performance by removing redundant logic and reducing the copy operations of the point clouds in the memory.

B. Accuracy And Robustness

This section verifies the accuracy and robustness of the proposed LIO in various environments. Table III shows absolute

TABLE II
TIME EVALUATION (MS) AND AVERAGE EFFECTIVE FEATURE POINTS

Seq.	iG-LIO		iG-LIO*		Faster-LIO		FastLIO2		DLIO	
	Time(ms)	Feat.[1]/ Plane(%)[2]	Times(ms)	Feat. / Plane(%)	Time(ms)	Feat.	Time(ms)	Feat.	Time(ms)	Feat.
nclt_1	**8.524**	1639.29 / 37.27	13.716	1616.68 / 45.83	10.070	2100.42	22.536	1281.06	31.808	4703.86
nclt_2	**9.076**	1686.41 / 35.07	15.904	1693.65 / 43.63	10.883	2332.65	23.694	1363.17	32.504	4672.63
nclt_3	**8.382**	1520.05 / 39.52	17.717	1608.32 / 41.82	10.824	2373.81	22.698	1299.76	32.591	4616.82
nclt_4	**8.680**	1582.57 / 36.73	17.445	1605.68 / 40.51	10.116	2277.27	25.428	1228.30	32.031	4773.20
nclt_5	**8.031**	1299.67 / 39.63	13.995	1307.31 / 48.42	10.334	1876.65	20.355	1127.30	25.720	3394.01
ncd_1	**15.543**	6507.15 / 52.45	26.835	6567.61 / 66.31	18.932	6123.75	40.777	1989.40	47.384	10947.30
ncd_2	**15.788**	6847.09 / 59.00	27.359	6869.52 / 75.90	26.383	6979.60	43.929	1951.00	48.140	10493.10
ncd_3	**17.140**	6991.62 / 62.99	26.645	6984.83 / 79.95	22.069	6815.49	38.305	2692.77	31.445	9408.45
ncd_4	**17.770**	7545.71 / 67.81	29.416	7514.35 / 85.38	19.888	7264.91	40.748	3966.65	27.535	5646.77
ncd_5	**19.732**	7541.02 / 47.84	28.578	7560.64 / 60.41	24.858	6997.19	44.814	2273.57	36.468	12055.60
ulhk_1	**6.300**	1757.98 / 45.75	9.396	1775.00 / 57.52	7.230	2074.84	10.425	1533.36	11.145	7242.23
ulhk_2	**9.926**	2526.50 / 39.22	14.167	2540.40 / 50.26	10.168	3200.03	17.584	2475.83	22.565	10544.40
bg_1	**5.958**	2310.01 / 48.04	10.579	2346.46 / 61.51	7.367	2655.19	12.345	700.26	14.370	5535.90
bg_2	**6.151**	2406.73 / 49.46	10.908	2451.6 / 62.82	7.886	2748.74	12.723	726.06	15.168	5926.72
bg_1*[3]	**3.676**	1127.53 / 64.89	6.197	1138.60 / 78.14	4.120	1216.42	5.933	432.23	-[4]	-
bg_2*	**3.862**	1144.88 / 66.45	6.255	1157.84 / 79.63	4.138	1213.34	5.909	370.346	-	-
avia_1	**3.820**	947.06 / 66.39	6.120	948.12 / 80.90	4.433	1122.80	5.957	555.10	-	-
avia_2	0.869	145.03 / 48.65	1.250	143.91 / 65.08	**0.650**	155.04	0.901	80.49	-	-
avia_3	0.930	186.60 / 37.59	1.257	129.00 / 44.13	**0.690**	199.03	1.164	138.54	-	-
gdut_1	**4.390**	1778.01 / 26.79	7.370	1787.41 / 36.85	6.453	2221.89	9.645	1056.86	-	-

[1] "Feat." denotes the number of effective feature points used in optimization; [2] "Plane" denotes the average percentage of the plane-to-plane metrics employed in optimization; [3] "*" denotes the sequence tested with Livox avia; [4] "-" denotes the method did not participate in the sequence.

TABLE III
ABSOLUTE POSE ERROR (RMSE, METERS)

Seq.	iG-LIO	iG-LIO*	NDT-LIO	Faster-LIO	FastLIO2	DLIO	Dist.(m)
nclt_1	**1.673**	1.795	2.365	1.855	1.734	2.104	7.58
nclt_2	1.230	**1.209**	2.005	1.279	1.485	1.392	3.17
nclt_3	**1.558**	1.560	4.173	2.141	2.454	2.581	6.12
nclt_4	**1.496**	1.514	2.583	1.544	2.112	2.311	4.09
nclt_5	0.956	1.062	0.991	0.933	**0.890**	1.088	1.14
ncd_1	0.322	**0.317**	0.371	0.340	0.353	0.361	1.61
ncd_2	0.375	**0.361**	0.424	0.373	0.376	0.393	3.06
ncd_3	**0.099**	0.101	0.101	0.119	0.125	0.105	0.48
ncd_4	0.083	0.083	0.083	**0.079**	**0.079**	0.121	0.09
ncd_5	**0.125**	**0.125**	0.134	0.141	0.127	0.157	0.70
ulhk_1	**1.153**	1.271	1.338	1.272	1.196	2.005	0.60
ulhk_2	1.776	**1.770**	1.807	1.906	1.804	4.254	0.74
bg_1	**1.675**	1.758	1.704	1.697	1.911	1.798	0.76
bg_2	**1.606**	1.625	1.782	2.086	1.673	2.245	0.74
bg_1*[1]	3.324	**2.032**	4.651	9.825	37.625	-[2]	0.76
bg_2*	**2.891**	3.263	4.617	3.526	4.124	-	0.74

Note that the trajectories of BG sequences were evaluated with the origin alignment. The others were verified via the SE(3) alignment.
[1] "*" denotes the sequence tested with Livox avia.
[2] "-" denotes the method did not participate in the sequence.

TABLE IV
END TO END ERRORS (METERS)

Sequence	iG-LIO	iG-LIO*	NDT-LIO	Faster-LIO	FastLIO2	Dist.(km)
avia_1	<**0.1**	<**0.1**	<**0.1**	0.782	1.537	0.96
avia_2	<**0.1**	<**0.1**	<**0.1**	0.177	0.226	0.14
avia_3	<**0.1**	<**0.1**	<**0.1**	<**0.1**	<**0.1**	0.09
gdut_1	<**0.1**	<**0.1**	<**0.1**	<**0.1**	<**0.1**	0.27

pose errors (APE). Note that the results of DLIO in NCD may differ from those reported in [23], as the authors made a few changes before open-sourcing the code. To prevent FastLIO2 slippages at the start in NCD, the number of filter points is adjusted instead of skipping the first 100 poses. Therefore, the results of FastLIO2 are different from those in [23] but are consistent with the reported results in [2]. As no ground truth data is available in AVIA and GDUT, Table IV reports the end-to-end errors for the drift evaluation. Both AVIA and GDUT trajectories start and end at the same location.

1) Structural Environments: NCLT, NCD, GDUT, and AVIA (avia_2 and avia_3) are collected in structural environments. The NCLT and NCD are long-term datasets with 360° mechanical LiDAR (Velodyne HDL-32E for NCLT, Ouster OS1-64 for NCD) and IMU. The AVIA and GDUT are captured by handheld Livox avia. For NCLT, iG-LIO achieves comparable accuracy in APE (about $0.5\% \sim 1.1\%$) compared to FastLIO2, as shown in Table III. The performance of DLIO is decreased since the scans of NCD are significantly denser than NCLT. In terms of APE in NCD, iG-LIO demonstrates comparable accuracy with other algorithms but with a much faster speed. The avia_2 sequence is collected at a rate of 100 Hz, which presents a great challenge for robust registration due to the non-repetitive scanning. Both Faster-LIO and FastLIO2 exhibit a minor drift in the initial registration, while iG-LIO shows minimal drift and successfully returns to the starting position.

2) Aggressive Motion: For ncd_4, the operator manually flips the sensing system, and the maximum angular velocity reaches $183 deg/s$. Since the ground truth of NCD has nearly $3.0cm$ error in stationary [32], the APE of iG-LIO and FastLIO2 are considered to be identical in Table III. Instead of the Euler method used in FastLIO2, iG-LIO compensates for the motion distortions via the midpoint integration. Fig. 4(a) and (b) show that the mapping result of iG-LIO is clearer than that of FastLIO2.

3) Indoor Environments: In the avia_1 sequence, the operator employs Livox avia for large-scale indoor-outdoor map-

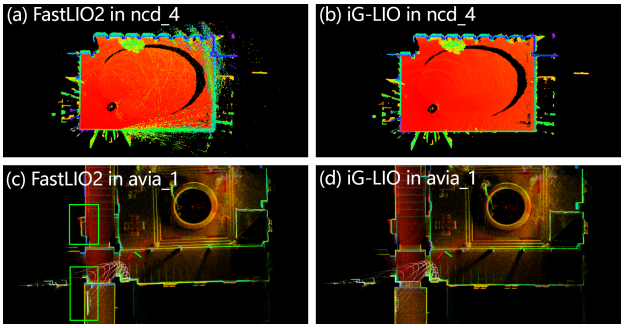

Fig. 4. (a) and (b) show the mapping results of FastLIO2 and iG-LIO in ncd_4. iG-LIO has less noise in rapid motion and dynamic environment. (c) and (d) show the mapping results of FastLIO2 and iG-LIO in avia_1. The green squares highlight the inconsistent map of FastLIO2.

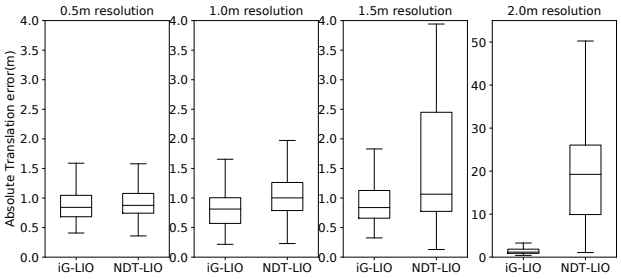

Fig. 5. The box plots show the translation errors at different resolutions. The translation error of NDT-LIO increases rapidly with larger resolution size. At a resolution of $2.0m$, the median error of iG-LIO and NDT-LIO are $1.16m$ and $19.25m$.

ping. As reported in Table III, both Faster-LIO and FastLIO2 suffer from drift in narrow and cramped environments due to the small FOV LiDAR. Fig. 4(c) illustrates that FastLIO2 generates a significantly inconsistent map when turning in the narrow space. In contrast, iG-LIO returns to the origin point and achieves a consistent map.

4) Dynamic Environments: This subsection exploits ULHK to analyze the robustness of the proposed LIO in dynamic environments. The ULHK is collected in highly urbanized scenes with Velodyne HDL-32E, where dense populations and a multitude of dynamic objects characterize the scenarios. Table III presents that the performance of DLIO deteriorates in dynamic scenarios. iG-LIO maintains accuracy as it adopts the chi-squared test to remove unsuitable local planar approximations and mismatched feature correspondences.

5) Unstructured Environments: BG sequences are collected with Velodyne VLP-16 and Livox avia in a large botanic garden. These sequences pose significant challenges for mapping and localization due to platform vibration, dense woods, and narrow paths. Table III reveals that both Faster-LIO and FastLIO2 exhibit substantial drift with Livox avia, because the limited FOV and the insufficient observation lead to the coupling of the estimated gravity direction with gyroscope bias during long-term mapping. In contrast, iG-LIO fixes the gravity direction and utilizes it to constrain the attitude estimation. Compared to the other LIOs, iG-LIO achieves the smallest APE variation in different FOV sensing systems, which demonstrates the adaptability and utility of VSCE and the local planar approximation in unstructured environments.

C. Ablation Study

1) Effectiveness of Voxel-based Surface Covariance Estimator (VSCE): To compare the effectiveness between VSCE and the kd-tree based method [19], [20], this study implements iG-LIO* using the iG-LIO framework. iG-LIO* estimates the surface covariance of each laser point via a range search in the kd-tree, where the search radius is set at twice the voxel filter resolution. According to Table II, the average percentage of the plane-to-plane metric in iG-LIO is 12% lower than in iG-LIO*. Despite this, iG-LIO delivers comparable accuracy to iG-LIO* while demonstrating significantly improved efficiency in Table II, Table III, and Table IV. As illustrated in Fig. 3, iG-LIO is more efficient than iG-LIO* in terms of the runtime for the surface covariance estimation, because the complexity of the hash index in the nearest neighbor search is $O(1)$ and it can be concurrently executed using libtbb.

2) Performance Comparison Between GICP Constraints and NDT Constraints: To obtain an incremental NDT-based LIO (NDT-LIO), this experiment sets the covariance in (10) to be zero, while keeping the covariance in (9) unchanged. Both iG-LIO and NDT-LIO set the resolution of the voxel grid filter and the voxel map to be $0.5m$. As demonstrated in Table III and Table IV, GICP constraints outperform NDT constraints.

To further validate the robustness of iG-LIO under varying resolutions, iG-LIO and NDT-LIO are evaluated with different resolutions in the nclt_5 sequence. As depicted in Fig. 5, the absolute translation error of both iG-LIO and NDT-LIO increases with larger resolution sizes. However, NDT-LIO's error exhibits a more rapid increase compared to iG-LIO. These results highlight the robustness of iG-LIO, even when operating with inappropriate voxel resolutions.

3) Accuracy Comparison Between GICP Constraints and VGICP Constraints: The incremental VGICP-based LIO (iVG-LIO) is built upon VGICP [20]. Specifically, iVG-LIO finds 20 closest laser points from a kd-tree and estimates the surface covariance. The voxel update (8) of iVG-LIO is replaced by

$$\begin{aligned} &\mathcal{M}[i].N = \mathcal{M}[i].N + \tilde{N}_i,\ \mathcal{M}[i].\mathbf{p} = \mathcal{M}[i].\mathbf{p} + \tilde{\mathbf{p}}_i, \\ &\mathcal{M}[i].\mathbf{C} = \mathcal{M}[i].\mathbf{C} + \tilde{\mathbf{C}}_i,\ \mathcal{M}[i].\boldsymbol{\mu} = \mathcal{M}[i].\mathbf{p}/\mathcal{M}[i].N, \\ &\mathcal{M}[i].\boldsymbol{\Sigma} = \mathcal{M}[i].\mathbf{C}/\mathcal{M}[i].N, \end{aligned} \tag{16}$$

where $\mathcal{M}[i].\mathbf{C}$ denotes the sum of covariances in the voxel $\mathcal{M}[i]$. The matrix $\tilde{\mathbf{C}}_i$ represents the sum of covariances of each hash index in the active list $\mathcal{V}$.

This study evaluates the accuracy of the constraints via avia_1 (operating indoor) and avia_2 (solid-state LiDAR sampling at 100 Hz). The identical parameters are set for iG-LIO and iVG-LIO in both sequences. Compared with the maps of iVG-LIO in Fig. 6, iG-LIO retains finer structural details of the environments. The results indicate that the surface covariance estimation in iVG-LIO is unsuitable for sparse scans, as the estimated covariance from the far-apart laser points is unable to formulate an appropriate probabilistic model of the local surface. Moreover, the voxel update in

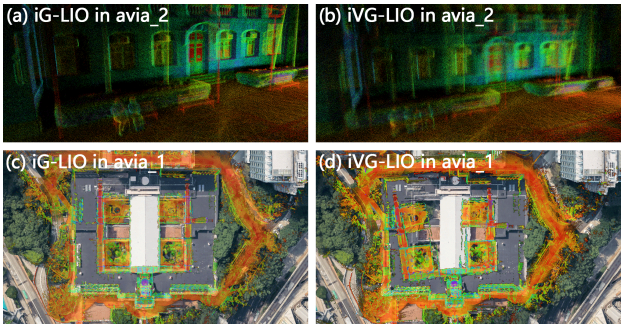

Fig. 6. The mapping results of iG-LIO (see (a) and (c)) and iVG-LIO (see (b) and (d)) in avia_2 and avia_1. iG-LIO retains finer structural details of the environments compared with the maps of iVG-LIO.

(16) accumulates the estimation errors into each voxel, which affects the registration accuracy.

IV. Conclusions

This letter presented iG-LIO, an open-source odometry that tightly-coupled GICP constraints and IMU constraints in a MAP estimation. iG-LIO used VSCE to estimate the distribution of scans, and utilized the incremental voxel map to represent the probabilistic models of surrounding environments. Extensive experiments and ablation studies verified the efficiency and accuracy of the proposed LIO, encompassing structural, aggressive motion, indoor, dynamic, and unstructured environments. Even though iG-LIO kept identical parameters across all sequences, it achieved nearly the same accuracy as the state-of-the-art LIOs but with faster speed.

References

[1] C. Cao, H. Zhu, Z. Ren, H. Choset, and J. Zhang, "Representation granularity enables time-efficient autonomous exploration in large, complex worlds," *Sci. Robot.*, vol. 8, no. 80, p. eadf0970, 2023.

[2] T.-M. Nguyen, D. Duberg, P. Jensfelt, S. Yuan, and L. Xie, "Slict: Multi-input multi-scale surfel-based lidar-inertial continuous-time odometry and mapping," *IEEE Robot. Autom. Lett.*, vol. 8, no. 4, pp. 2102–2109, Feb. 2023.

[3] X. Liu, Z. Liu, F. Kong, and F. Zhang, "Large-scale lidar consistent mapping using hierarchical lidar bundle adjustment," *IEEE Robot. Autom. Lett.*, vol. 8, no. 3, pp. 1523–1530, Jan. 2023.

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

[5] 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.*, 2020, pp. 5135–5142.

[6] 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. Automat.*, 2020, pp. 8899–8906.

[7] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "FAST-LIO2: Fast direct LiDAR-Inertial odometry," *IEEE Trans. Robot.*, vol. 38, no. 4, pp. 2053–2073, Aug. 2022.

[8] C. Bai, T. Xiao, Y. Chen, H. Wang, F. Zhang, and X. Gao, "Faster-LIO: Lightweight tightly coupled lidar-inertial odometry using parallel sparse incremental voxels," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 4861–4868, Feb. 2022.

[9] C. Yuan, W. Xu, X. Liu, X. Hong, and F. Zhang, "Efficient and probabilistic adaptive voxel mapping for accurate online LiDAR odometry," *IEEE Robot. Autom. Lett.*, vol. 7, no. 3, pp. 8518–8525, Jul. 2022.

[10] J. Liu, Y. Zhang, X. Zhao, and Z. He, "FR-LIO: Fast and robust lidar-inertial odometry by tightly-coupled iterated kalman smoother and robocentric voxels," 2023, arXiv:2302.04031.

[11] X. Ji, S. Yuan, P. Yin, and L. Xie, "LIO-GVM: an accurate, tightly-coupled lidar-inertial odometry with gaussian voxel map," 2023, arXiv:2306.17436.

[12] Z. Yuan, F. Lang, T. Xu, C. Zhao, and X. Yang, "Semi-elastic LiDAR-inertial odometry," 2023, arXiv:2307.07792.

[13] P. Dellenbach, J.-E. Deschaud, B. Jacquet, and F. Goulette, "Ct-icp: Real-time elastic lidar odometry with loop closure," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2022, pp. 5580–5586.

[14] H. Lim, D. Kim, B. Kim, and H. Myung, "AdaLIO: Robust adaptive LiDAR-inertial odometry in degenerate indoor environments," in *Proc. IEEE 20th Int. Conf. Ubiquitous Robots*, 2023, pp. 48–53.

[15] J. Behley and C. Stachniss, "Efficient surfel-based SLAM using 3D laser range data in urban environments," in *Proc. Robotics: Sci. Syst.*, 2018, pp. 9296–9306.

[16] X. Chen, A. Milioto, E. Palazzolo, P. Giguère, J. Behley, and C. Stachniss, "SuMa++: Efficient LiDAR-based semantic SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 4530–4537.

[17] M. Magnusson, A. Lilienthal, and T. Duckett, "Scan registration for autonomous mining vehicles using 3D-NDT," *J. Field Robot.*, vol. 24, no. 10, pp. 803–827, Oct. 2007.

[18] M. Yokozuka, K. Koide, S. Oishi, and A. Banno, "LiTAMIN2: Ultra light LiDAR-based SLAM using geometric approximation applied with KL-divergence," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 11 619–11 625.

[19] A. Segal, D. Haehnel, and S. Thrun, "Generalized-ICP," in *Robot.: Sci. Syst.*, vol. 2, no. 4, 2009, pp. 435–442.

[20] 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. Automat.*, 2021, pp. 11 054–11 059.

[21] B. Kim, C. Jung, D. H. Shim, and A. Agha–mohammadi, "Adaptive keyframe generation based lidar inertial odometry for complex underground environments," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2023, pp. 3332–3338.

[22] A. Reinke *et al.*, "Locus 2.0: Robust and computationally efficient lidar odometry for real-time 3d mapping," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 9043–9050, Jun. 2022.

[23] K. Chen, R. Nemiroff, and B. T. Lopez, "Direct lidar-inertial odometry: Lightweight lio with continuous-time motion correction," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2023, pp. 3983–3989.

[24] K. Chen, B. T. Lopez, A.-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, Jan 2022.

[25] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual–inertial odometry," *IEEE Trans. Robot.*, vol. 33, no. 1, pp. 1–21, Feb. 2017.

[26] 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.

[27] J. Solà, "Quaternion kinematics for the error-state Kalman filter," 2017, arXiv:1711.02508.

[28] M. Teschner, B. Heidelberger, M. Müller, D. Pomerantes, and M. H. Gross, "Optimized spatial hashing for collision detection of deformable objects." in *Proc. Vis., Model., Visual. Conf.*, vol. 3, 2003, pp. 47–54.

[29] D. He, W. Xu, and F. Zhang, "Kalman filters on differentiable manifolds," 2021, arXiv:2102.03804.

[30] B. Bell and F. Cathey, "The iterated kalman filter update as a gauss-newton method," *IEEE Trans. Automat. Contr.*, vol. 38, no. 2, pp. 294–297, Feb. 1993.

[31] 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, 2015.

[32] 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.*, 2020, pp. 4353–4360.

[33] W. Wen *et al.*, "UrbanLoco: A full sensor suite dataset for mapping and localization in urban scenes," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 2310–2316.

[34] Y. Liu *et al.*, "BotanicGarden: A high-quality and large-scale robot navigation dataset in challenging natural environments," 2023, arXiv:2306.14137.

[35] J. Lin and F. Zhang, "R3LIVE: A robust, real-time, RGB-colored, LiDAR-inertial-visual tightly-coupled state estimation and mapping package," in *Proc. Int. Conf. Robot. Automat.*, 2022, pp. 10 672–10 678.