

RI-LIO: Reflectivity Image Assisted Tightly-Coupled LiDAR-Inertial Odometry

Yanfeng Zhang^{ID}, Yunong Tian^{ID}, Wanguo Wang, Guodong Yang^{ID}, Zhishuo Li^{ID}, *Student Member, IEEE*, Fengshui Jing^{ID}, and Min Tan^{ID}

Abstract—In this letter, we propose RI-LIO, a new reflectivity image assisted tightly-coupled LiDAR-inertial odometry (LIO) framework that introduces additional reflectivity texture information to efficiently reduce the drift of geometric-only methods. To achieve this, we construct an iterated extended Kalman filter framework by blending the point-to-plane geometric measurement and the reflectivity image measurement. Specifically, the geometric measurement is defined as the distance from the raw point of a new scan to its nearest neighbor plane in the global incremental kd-tree map. The searched nearest neighbor point is used to render a sparse reflectivity image after LiDAR motion distortion information is given by its corresponding raw point. Then, the reflectivity measurement is built to align the sparse reflectivity image with the dense reflectivity image of the current scan by minimizing the photometric errors directly. In addition, based on the mechanism of high-resolution LiDAR, a corrected spherical projection model is proposed to project spatial points into the image frame. Finally, extensive experiments are conducted in structured, unstructured and challenging open field scenarios. The results demonstrate that the proposed method outperforms existing geometric-only methods in terms of robustness and accuracy, especially in the rotation direction.

Index Terms—Localization, mapping, projection model, reflectivity image, SLAM.

I. INTRODUCTION

LOCALIZATION is a fundamental capability of autonomous mobile robots, such as Unmanned Ground Vehicles (UGVs) and Unmanned Aerial Vehicles (UAVs). In recent

Manuscript received 27 September 2022; accepted 27 January 2023. Date of publication 9 February 2023; date of current version 16 February 2023. This letter was recommended for publication by Associate Editor S. Scherer and Editor S. Behnke upon evaluation of the reviewers' comments. This work was supported in part by the National Natural Science Foundation of China under Grants 61873267 and 62206275 and in part by the Special Funds for the Development of Science and Technology in Zhongshan under Grant 2020AG023. (*Corresponding author: Guodong Yang*)

Yanfeng Zhang, Yunong Tian, Guodong Yang, Zhishuo Li, Fengshui Jing, and Min Tan are with the Engineering Laboratory for Intelligent Industrial Vision, Chinese Academy of Sciences, Beijing 100190, China, also with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: zhangyanfeng2020@ia.ac.cn; yunong.tian@ia.ac.cn; guodong.yang@ia.ac.cn; lizhishuo2019@ia.ac.cn; fengshui.jing@ia.ac.cn; min.tan@ia.ac.cn).

Wanguo Wang is with the State Grid Intelligence Technology Co., Ltd., Jinan 250101, China (e-mail: wangwanguo03@163.com).

This letter has supplementary downloadable material available at <https://doi.org/10.1109/LRA.2023.3243528>, provided by the authors.

Digital Object Identifier 10.1109/LRA.2023.3243528

years, visual-inertial odometry (VIO) using RGB textures [1], [2] and LiDAR-inertial odometry (LIO) using geometric features [3], [4], [5], [6], [7] have been widely deployed to improve localization performance, especially when the Global Positioning System (GPS) cannot provide reliable localization results.

With the development of filtering techniques and factor map optimization, multiple sensors can be easily integrated into a single framework to provide effective pose estimation in single-sensor degraded scenarios [8], [9], [10], [11], [12], [13]. Several works that consider LIO and VIO as subsystems for data fusion are proposed, such as R3LIVE [14], LVI-SAM [15], and FAST-LIVO [16]. They exploit textures in environments by adding cameras and are still limited by the light-sensitivity and massive calculation of VIO algorithms. Unlike the passive exposure of cameras, reflectivity images measured actively by the LiDAR are virtually independent of illumination and have great potential for mapping object textures.

To generate high-resolution reflectivity images, high line rotating LiDAR is required. With the maturity of vertical cavity surface emitting laser (VCSEL) technology, we can pack many lasers into a dense array, which is similar to a camera light-sensitive array. In addition, some LiDARs (e.g., Ouster LiDAR) correct for reflectivity at the hardware level, making it independent of the measured distance and angle. Therefore, we tend to integrate reflectivity images into the existing LiDAR framework.

However, fusing reflectivity image also brings new challenges for pose estimation: 1) LiDAR produces a shutter effect through the mechanical rotation of the laser array, with reflectivity imaging time of up to 100 ms, causing a significant rolling shutter effect. 2) The imaging surface of a LiDAR is a toroidal sphere, which leads to a non-similar transformation between the LiDAR frame and the image frame. The non-similar transformation distorts features in a view, which causes some feature-based visual SLAM (e.g., ORB-SLAM [1], VINS-Mono [2]) methods inapplicable. 3) For cameras we can use a uniform parametric model (e.g., Brown-Conrady model [17]) to correct lens distortion, while motion distortion of a LiDAR scan is affected by the vehicle state and needs to be corrected for each pixel position individually.

To address the above challenges, we propose RI-LIO, a framework that tightly couples Inertial Measurement Unit (IMU), reflectivity residual, and geometric residual via an Iterated Extended Kalman Filter (IEKF) for pose estimation. The framework of the system is shown in Fig. 1. Instead of correcting image distortion to track feature points, RI-LIO projects the global map points into the image frame to minimize the photometric errors directly. IMU is used to provide the LiDAR motion distortion

System Overview

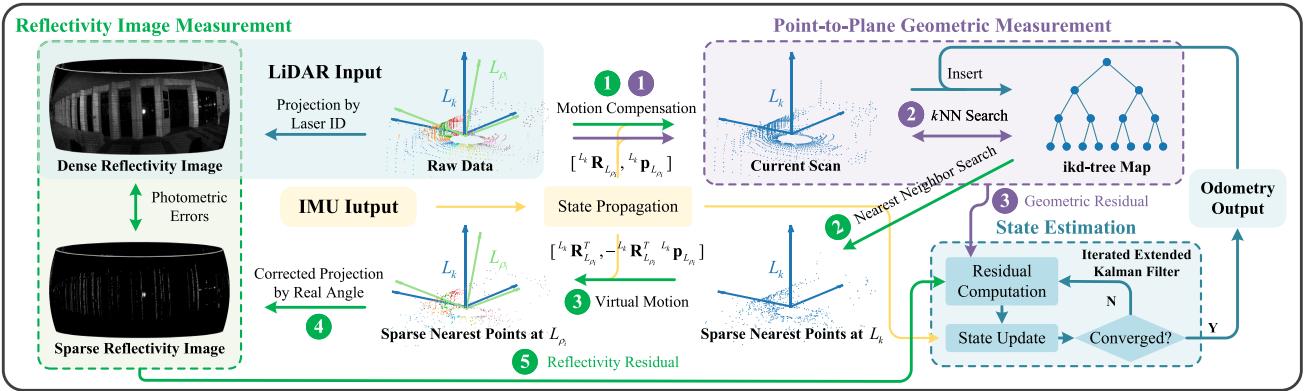


Fig. 1. System overview of RI-LIO. The system mainly contains reflectivity residual computation (green arrows and markers, see Section III-C), geometric residual computation (purple arrows and markers, see Section III-D), and state estimation (see Section III-E). IMU provides the LiDAR motion distortion information and the prior of state estimation. The different coloured points indicate that they are sampled at different LiDAR frames L_{ρ_i} at different time t_{ρ_i} .

information and the state estimation prior. In summary, our contributions are as follows:

- A compact LIO framework for fusing the point-to-plane geometric measurement and the reflectivity image measurement efficiently is presented to achieve robust, fast, and accurate localization.
- A corrected LiDAR projection model is proposed. It can mathematically generate reflectivity images from real LiDAR points without empty holes.
- A new reflectivity image measurement model is proposed. It optimizes the direct-sparse photometric errors between the reflectivity images generated from the global incremental kd-tree (ikd-tree) map and a new scan.
- Verification of our implemented system on structured, unstructured and challenging open field scenarios. The results show that our method significantly improves the rotation accuracy compared to geometric-only methods and computes in real time.

II. RELATED WORKS

A. Geometric-Only LiDAR SLAM

To ensure real-time performance, LiDAR SLAM methods typically extract edge and plane features to obtain geometric features. Specifically, LOAM [5] proposed by Zhang et al. firstly aligns two consecutive scans with feature points extracted based on the local smoothness and then registers multiple local maps to the global map. However, LOAM does not embed additional sensors such as IMU into the optimization process. For accuracy and robustness, some recent works tightly couple IMU preintegrations as a priori and maximize the posterior estimation of multiple scans (keyframes), such as LILI-OM [7] (sliding window optimization) and LIO-SAM [6] (factor graph smoothing).

Alternatively, without feature extraction, FAST-LIO2 [3] proposed by Xu et al. directly registers the raw points to the ikd-tree map and achieves global consistency. Our work builds on FAST-LIO2, inheriting the idea of direct methods. The reflectivity photometric errors are calculated directly from the global ikd-tree map and a new scan without image feature extraction.

Thus, it can simply strengthen the constraints of pose estimation, especially when there are no prominent geometric features.

B. Reflectivity-Fusion LiDAR SLAM

Several recent works have attempted to incorporate reflectivity information into the current LIO framework. In the loop closure detection section, Wang et al. presented a novel global descriptor termed intensity scan context (ISC) [18], which calibrates the intensity to reflectivity for structure re-identification. It achieves a better recall rate and recall precision than existing geometric-only methods. Shan et al. imposed the ORB feature descriptor and DBoW to a high-resolution intensity image for fast place recognition and used RANSAC-PnP to align the matched visual feature points in the image frame [19]. This work shows the potential of applying visual SLAM techniques to LiDAR intensity images, but it does not analyse the LiDAR imaging model, so barrel distortion and motion distortion have significant impacts on its relocalization accuracy.

We use reflectivity image for pose estimation directly rather than for feature extraction and matching. Similar to our work is the intensity-SLAM [20] proposed by Wang et al. It constructs the reflectivity distribution map with reference to octomap and then minimizes both reflectivity cost and geometric cost for pose estimation. However, intensity-SLAM maintains the distribution map and gradient of reflectivity in 3D space, which is very time-consuming. In contrast, our system projects sparse map points into the current reflectivity image frame and only needs to calculate the 2D image gradient so that it can transform the reflectivity residual to the state space analytically and efficiently.

III. METHODOLOGY

A. Notation

We denote the set of points as \mathcal{P} . Let $\mathcal{I}[\cdot]$ and $\mathcal{M}[\cdot]$ denote the reflectivity values in the image and the global map, respectively. $G(\cdot)$, $I(\cdot)$, $L(\cdot)$, and $C(\cdot)$ represent the value of (\cdot) expressed in global frame, IMU frame, LiDAR frame and image frame, respectively. The manifold ${}^b\mathbf{R}_a \in SO(3)$ and ${}^b\mathbf{t}_a \in \mathbb{R}^3$ indicate the attitude and position of the a frame w.r.t b frame. The operators on the manifold are “ \boxplus ” and “ \boxminus ”. Every LiDAR scan

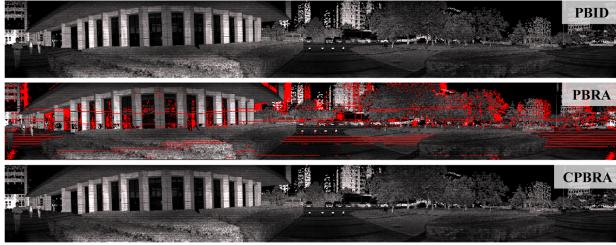


Fig. 2. The reflectivity images generated by different projection methods. The red pixels represent the empty holes generated by different methods when projecting the same LiDAR points of a scan into the image frame.

at time t_k is assembled by points at their respective sampling time ρ_i which is prior than t_k , so we denote L_k and L_{ρ_i} as the LiDAR frame at time t_k and ρ_i . For state estimation, we denote \mathbf{x} , $\hat{\mathbf{x}}$, and $\tilde{\mathbf{x}}$ represent the ground truth, predicted and error state value.

B. Reflectivity Image Generation

While LiDAR detects the range of each laser point, it also estimates the reflectivity value of illuminated surfaces from the intensity of echoes. Compared with the 3D point cloud, the 2D image view is a more compact representation of the reflectivity texture. In this section, we focus on spherical projection to generate reflectivity images. Specifically, when a 3D point cloud \mathcal{P} is received, we transform the spherical coordinates $[R, \theta, \varphi]$ of each point to image coordinates $[u, v]$ by a linear transformation, which can be expressed as

$$u = \frac{w}{2\pi}(\pi - \varphi), \quad v = \frac{h}{\theta^{fov}}(\theta^{up} - \theta) \quad (1)$$

where w, h are the width and height of the image, respectively, $\theta^{fov} = \theta^{up} - \theta^{down}$ is the vertical field-of-view of the LiDAR, and $\theta^{up}, \theta^{down}$ are the maximum and minimum elevation angles, respectively. However, when actually generating reflectivity images, there are two different methods for obtaining the image coordinates:

1) *Projection by Laser ID (PBID)* [21]: u and v are directly determined by encoder value $e = [1, w]$ and laser head ID $r = [1, h]$, i.e., $u = e$ and $v = r$. This means that the azimuth angle and the elevation angle are ideally divided equally according to the equation (1), i.e.,

$$\varphi_r^{ideal} = \pi - \frac{2\pi e}{w}, \quad \theta_r^{ideal} = \theta^{up} - \frac{r\theta^{fov}}{h} \quad (2)$$

where the subscript r means that the point is captured by the r -th laser head. This method ignores the real LiDAR point coordinates and generates images without empty holes, see Fig. 2. Therefore, when a new scan arrives, we generate a **dense reflectivity image** via PBID directly, see Fig. 1.

2) *Projection by Real Angle (PBRA)*: u and v are calculated by the real LiDAR point coordinates $L_{\rho_i} \mathbf{p} = (x, y, z)^T$. According to the transformation from Cartesian coordinates to spherical coordinates, we can obtain

$$\varphi_r^{real} = \arctan(y, x), \quad \theta_r^{real} = \arcsin\left(\frac{z}{R - n}\right) \quad (3)$$

where $R = \|L_{\rho_i} \mathbf{p}\|_2$ is the range and n is the length of the laser beam origin to its rotation center, as shown in Fig. 3. The image coordinates are then obtained by substituting equation

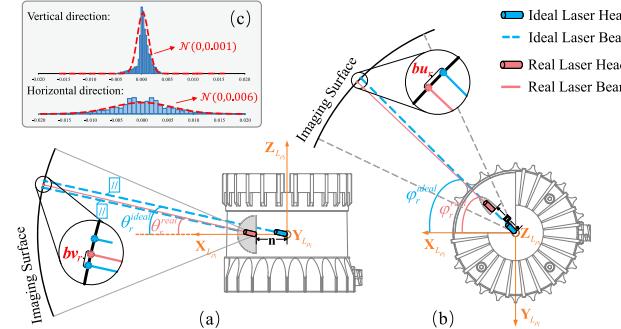


Fig. 3. The projection biases of the r -th laser head. The projection bias between PBID (ideal) and PBRA (real) in the vertical direction is shown in the side view (a), and the horizontal direction is shown in the top view (b). (c) shows the projection error distributions of CPBRA.

(3) into equation (1). This method reflects the real laser head distribution and projects the LiDAR points into the image frame mathematically. However, due to the uneven distribution of the real laser heads, there are many empty holes in the image generated by PBRA, see Fig. 2.

To unify the image frames and reduce empty holes, the pixel coordinates generated by PBRA should be corrected to PBID coordinates. Fig. 3(a) and (b) show the projection biases b_{vr} and b_{ur} of the r -th laser head between PBID and PBRA in the vertical and horizontal directions, respectively, which are expressed as

$$b_{ur} = \frac{w}{2\pi}(\varphi_r^{ideal} - \varphi_r^{real}), \quad b_{vr} = \frac{h}{\theta^{fov}}(\theta_r^{ideal} - \theta_r^{real}) \quad (4)$$

By combining equations (1)–(4), we propose the *Corrected Projection by RealAngle* (CPBRA). In this method, each point in the LiDAR frame $L_{\rho_i} \mathbf{p} = (x, y, z)^T$ is transformed via function $\Pi : \mathbb{R}^3 \mapsto \mathbb{R}^2$ to the PBID frame $C \mathbf{p} = (u, v)^T$, i.e.,

$$C \mathbf{p} = \Pi(L_{\rho_i} \mathbf{p}) = \begin{bmatrix} \frac{w}{2\pi}[\pi - \arctan(y, x)] + b_{vr} \\ \frac{h}{\theta^{fov}}[\theta^{up} - \arcsin(\frac{z}{R - n})] + b_{ur} \end{bmatrix} \quad (5)$$

Then, the points from the global map with no laser ID can be projected to generate a **sparse reflectivity image** via CPBRA.

To minimize the projection error, the following loss function is defined to obtain the b_{vr} and b_{ur} of each laser head:

$$E_{uv} = \sum_{r=1}^N \sum_{\mathbf{p} \in \mathcal{P}_r} C \mathbf{p}^{gt} - \Pi(L_{\rho_i} \mathbf{p}) \quad (6)$$

where $C \mathbf{p}^{gt}$ is the ground truth obtained by PBID, and \mathcal{P}_r represents the set of points captured by the r -th laser head.

With the gradient descent algorithm, the projection errors of CPBRA are reduced to within 0.02 pixel in both the vertical and horizontal directions. By analysing the projection error distributions as shown in Fig. 3(c), we approximately treat them as Gaussian distributions with zero mean in Section III-E2. In addition, the reflectivity image generated by CPBRA has no empty holes as PBID, see Fig. 2.

C. Reflectivity Image Measurement Model

In this section, the reflectivity image measurement is defined as the photometric errors between the dense and sparse reflectivity images (see the green dashed box in Fig. 1). The dense reflectivity image is generated directly from a new scan via

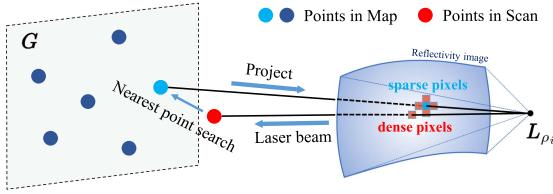


Fig. 4. The process of generating the sparse reflectivity image. The nearest neighbor point in the global ikd-tree map is searched to simulate being hit by a real laser beam and then projected to render the sparse pixel.

PBID. Because each input point corresponds to one pixel, the image is dense. While the sparse reflectivity image is rendered by the global map points, the generation process is as follows.

As shown in Fig. 4, similar to ray-casting [22], the laser beams are also rays, and they return when they encounter occlusion. Therefore, assume that the nearest neighbor (NN) point (light blue point) in the global ikd-tree map is captured by the same laser beam as the point of the current scan (red point); that is, the reflectivity of the NN point is used to render the image. Since there is still a gap (non-negligible error) between the NN point and the point of the current scan, we reproject the NN point into the image frame via CPBRA to obtain a more accurate pixel position. Because the current scan may have been downsampled or may have no NN point within a certain range, the rendered image is sparse.

Specifically, when a new scan arrives, the following steps are performed to calculate the reflectivity residual, with the serial numbers corresponding to the green markers in Fig. 1.

Step 1: Motion Compensation: First, we use IMU propagation to estimate the LiDAR motion at t_k , see Section III-E1. Then, we conduct the backward propagation proposed in [3] to obtain the motion compensation matrix $[{}^{L_k} \mathbf{R}_{L_{\rho_i}}, {}^{L_k} \mathbf{t}_{L_{\rho_i}}] \in SE(3)$ at t_{ρ_i} . Each point ${}^{L_{\rho_i}} \mathbf{p}$ is accumulated to the current scan L_k , i.e., ${}^{L_k} \mathbf{p} = {}^{L_k} \mathbf{R}_{L_{\rho_i}} {}^{L_{\rho_i}} \mathbf{p} + {}^{L_k} \mathbf{t}_{L_{\rho_i}}$.

Step 2: Nearest Neighbor Search: The scan point ${}^{L_k} \mathbf{p}$ will be further transformed to the world frame through the estimated pose $({}^G \hat{\mathbf{R}}_{L_k}, {}^G \hat{\mathbf{t}}_{L_k}) \in SE(3)$, i.e., ${}^G \mathbf{p} = {}^G \hat{\mathbf{R}}_{L_k} {}^{L_k} \mathbf{p} + {}^G \hat{\mathbf{t}}_{L_k}$. Then, its NN point ${}^G \mathbf{p}^N$ can be searched in the global ikd-tree map [23].

Step 3: Virtual Motion: For projection, the NN point ${}^G \mathbf{p}^N$ needs to be transformed to the frame of each sampling moment t_{ρ_i} of the LiDAR, which we define as *Virtual Motion*, i.e., ${}^{L_{\rho_i}} \mathbf{p}^N = {}^{L_k} \mathbf{R}_{L_{\rho_i}}^T ({}^G \hat{\mathbf{R}}_{L_k}^T ({}^G \mathbf{p}^N - {}^G \hat{\mathbf{t}}_{L_k}) - {}^{L_k} \mathbf{t}_{L_{\rho_i}})$.

Step 4: Corrected Projection by Real Angle: Then, the point ${}^{L_{\rho_i}} \mathbf{p}^N$ can be projected to generate a sparse reflectivity image \mathcal{I}_s via CPBRA presented in Section III-B, i.e., ${}^C \mathbf{p} = \Pi({}^{L_{\rho_i}} \mathbf{p}^N)$. The corresponding pixel value is determined by the reflectivity value of the map point, i.e., $\mathcal{I}_s[\Pi({}^C \mathbf{p})] = \mathcal{M}[{}^G \mathbf{p}^N]$.

Step 5: Reflectivity Residual Computation: If the result of pose estimation is true, the photometric error between the dense reflectivity image and the sparse reflectivity image should be zero, so the reflectivity residual is defined as

$$\mathbf{z}_c = \mathcal{I}_d[\Pi({}^C \mathbf{p})] - \mathcal{M}[{}^G \mathbf{p}^N] \quad (7)$$

where $\mathcal{I}_d[\cdot]$ denotes the pixel values in the dense reflectivity image and is abbreviated as $\mathcal{I}[\cdot]$ in the remaining sections.

D. Point-to-Plane Geometric Measurement Model

We use the same scan-to-map method as FAST-LIO2 [3], which extracts plane elements directly in the global map and then treats the distance between each point of a new scan and its corresponding plane as the geometric residual (see the purple dashed box in Fig. 1).

Specifically, when a new scan arrives, the following steps are performed to calculate the geometric residual, with the serial numbers corresponding to the purple markers in Fig. 1.

Step 1: Motion Compensation. See Section III-C.

Step 2: K-Nearest Neighbor Search: See Section III-C.

However, here, the k -nearest neighbor (kNN) points ${}^G \mathbf{p}_k^N \in \mathcal{P}^N, k = [1, k]$ are searched in the global ikd-tree map.

Step 3: Geometric Residual Computation: The matrix composed of \mathcal{P}^N is decomposed by Singular Value Decomposition (SVD), where the normalized eigenvector corresponding to the smallest eigenvalue is the plane normal vector ${}^G \mathbf{u}$. The geometric residual is defined as

$$\mathbf{z}_c = {}^G \mathbf{u}^T ({}^G \mathbf{p} - {}^G \mathbf{p}_k^N) \quad (8)$$

E. State Estimation

1) State Definition and Propagation: We tightly couple the geometric residual and the reflectivity residual via the IEKF similar to FAST-LIO2 [3]. Assuming that LiDAR and IMU are rigidly attached together and the extrinsics $({}^I \mathbf{R}_L, {}^I \mathbf{t}_L) \in SE(3)$ has been calibrated, the state variables are defined as

$$\mathbf{x} \triangleq [{}^G \mathbf{R}_I^T {}^G \mathbf{v}^T {}^G \mathbf{g}^T \mathbf{b}_g^T \mathbf{b}_a^T {}^G \mathbf{g}^T] \in SO(3) \times \mathbb{R}^{15}$$

where ${}^G \mathbf{R}_I$ and ${}^G \mathbf{t}_I$ denote the attitude and position of the IMU frame w.r.t global frame, ${}^G \mathbf{v}$ and ${}^G \mathbf{g}$ are the velocity and gravity vector in the global frame, respectively, and \mathbf{b}_g and \mathbf{b}_a are IMU biases.

Before updating the state using the measurement models, the state can be propagated forwards in time using the IMU linear acceleration and angular velocity. Based on the standard inertial kinematic model proposed in [3], the state estimation prior $\hat{\mathbf{x}}_k$ with covariance $\hat{\mathbf{P}}_k$ can be obtained.

2) The Reflectivity Residual With Noise: The measurement noise of the reflectivity residual consists of two sources: the LiDAR reflectivity pixel noise in $\mathcal{I}({}^C \mathbf{p}_j)$ and the projection model error in ${}^C \mathbf{p}_j$:

$$\mathcal{I}({}^C \mathbf{p}_j) = \mathcal{I}^{gt}({}^C \mathbf{p}_j) + n_{r_j}, n_{r_j} \sim \mathcal{N}(0, \Sigma_{n_{r_j}}) \quad (9)$$

$${}^C \mathbf{p}_j = {}^C \mathbf{p}_j^{gt} + \mathbf{n}_{\Pi_j}, \mathbf{n}_{\Pi_j} \sim \mathcal{N}(0, \Sigma_{\mathbf{n}_{\Pi_j}}) \quad (10)$$

where \mathcal{I}^{gt} and ${}^C \mathbf{p}_j^{gt}$ are the ground truth of \mathcal{I} and ${}^C \mathbf{p}_j$, respectively, and ${}^C \mathbf{p}_j$ is projected from ${}^G \mathbf{p}_j^N \in \mathcal{P}_C = \{\mathbf{p}_1, \dots, \mathbf{p}_{mc}\}$. For derivation, the reflectivity residual \mathbf{z}_c is redefined as the function form $h_c(\hat{\mathbf{x}}_k, \mathcal{I}, {}^C \mathbf{p}_j)$. Then, we obtain the first-order Taylor expansion of the true zero residual:

$$0 = h_c(\mathbf{x}_k, \mathcal{I}^{gt}, {}^C \mathbf{p}_j^{gt}) = h_c(\hat{\mathbf{x}}_k \boxplus \tilde{\mathbf{x}}_k, \mathcal{I}, {}^C \mathbf{p}_j) + \eta_j^c \quad (11)$$

$$\approx h_c(\hat{\mathbf{x}}_k, \mathcal{I}, {}^C \mathbf{p}_j) + \mathbf{H}_j^c \tilde{\mathbf{x}}_k + \eta_j^c, \eta_j^c \sim \mathcal{N}(0, \Sigma_{\eta_j^c})$$

$$\text{where } \mathbf{H}_j^c = \left. \frac{\partial h_c(\hat{\mathbf{x}}_k \boxplus \tilde{\mathbf{x}}_k, \mathcal{I}, {}^C \mathbf{p}_j)}{\partial \tilde{\mathbf{x}}_k} \right|_{\tilde{\mathbf{x}}_k=0} \quad (12)$$

$$= \left. \frac{\partial \mathcal{I}({}^C \mathbf{p}_j)}{\partial {}^C \mathbf{p}_j} \cdot \frac{\partial \Pi({}^{L_{\rho_j}} \mathbf{p}_j^N)}{\partial {}^{L_{\rho_j}} \mathbf{p}_j^N} \cdot \frac{\partial {}^{L_{\rho_j}} \mathbf{p}_j^N}{\partial \tilde{\mathbf{x}}_k} \right|_{\tilde{\mathbf{x}}_k=0}$$

$$\frac{\partial \Pi(L_{\rho_j} \mathbf{P}_j^N)}{\partial L_{\rho_j} \mathbf{P}_j^N} = \begin{bmatrix} \frac{w}{2\pi} & 0 \\ 0 & \frac{h}{f} \end{bmatrix} \begin{bmatrix} \frac{y}{L} & \frac{x}{L} & 0 \\ \frac{xz}{LR^2} & \frac{yz}{LR^2} & -\frac{L}{R^2} \end{bmatrix} \quad (13)$$

$$\frac{\partial L_{\rho_i} \mathbf{P}_j^N}{\partial \tilde{\mathbf{x}}_k} = ({}^I \mathbf{R}_L {}^{L_k} \mathbf{R}_{L_{\rho_i}})^T \left[-{}^G \hat{\mathbf{R}}_{I_k}^T \left[{}^G \hat{\mathbf{R}}_{I_k}^T {}^G \mathbf{P}_j^N \right] \times \mathbf{0}_{3 \times 12} \right] \quad (14)$$

$$\Sigma_{\eta_j^c} = \Sigma_{n_{r_j}} + \mathbf{F}_{\mathbf{n}_{\Pi_j}} \Sigma_{\mathbf{n}_{\Pi_j}} \mathbf{F}_{\mathbf{n}_{\Pi_j}}^T, \mathbf{F}_{\mathbf{n}_{\Pi_j}} = \frac{\partial h_c(\tilde{\mathbf{x}}_k, \mathcal{I}, {}^C \mathbf{P}_j)}{\partial {}^C \mathbf{P}_j} \quad (15)$$

where $\tilde{\mathbf{x}}_k = \mathbf{x}_k \boxminus \hat{\mathbf{x}}_k$, skew-symmetric matrix is noted as $[\cdot] \times$. $L = \sqrt{x^2 + y^2}$, and $\frac{\partial \mathcal{I}({}^C \mathbf{P}_j)}{\partial {}^C \mathbf{P}_j}$ is the image gradient that can be calculated directly from the neighbor pixels.

3) The Geometric Residual With Noise: The measurement noise of the point-to-plane geometric residual mainly comes from the LiDAR ranging and beam-directing noise \mathbf{n}_p when measuring the point ${}^{L_{\rho_i}} \mathbf{p}_i \in \mathcal{P}_S = \{\mathbf{p}_1, \dots, \mathbf{p}_{m_S}\}$, i.e,

$${}^{L_{\rho_i}} \mathbf{p}_i = {}^{L_{\rho_i}} \mathbf{p}_i^{gt} + \mathbf{n}_{p_i}, \mathbf{n}_{p_i} \sim \mathcal{N}(0, \Sigma_{\mathbf{n}_{p_i}}) \quad (16)$$

where ${}^{L_{\rho_i}} \mathbf{p}_i^{gt}$ is the ground truth of ${}^{L_{\rho_i}} \mathbf{p}_i$. For derivation, the geometric residual \mathbf{z}_S is redefined as the function form $h_S(\tilde{\mathbf{x}}_k, {}^{L_{\rho_i}} \mathbf{p}_i)$. The ground truths ${}^{L_{\rho_i}} \mathbf{p}_i^{gt}$ and \mathbf{x}_k lead to the true zero residual, and its first-order Taylor expansion is

$$0 = h_S(\mathbf{x}_k, {}^{L_{\rho_i}} \mathbf{p}_i^{gt}) = h_S(\tilde{\mathbf{x}}_k \boxplus \tilde{\mathbf{x}}_k, {}^{L_{\rho_i}} \mathbf{p}_i) + \eta_i^S \approx h_S(\tilde{\mathbf{x}}_k, {}^{L_{\rho_i}} \mathbf{p}_i) + \mathbf{H}_i^S \tilde{\mathbf{x}}_k + \eta_i^S, \eta_i^S \sim \mathcal{N}(0, \Sigma_{\eta_i^S}) \quad (17)$$

$$\text{where } \mathbf{H}_i^S = \frac{\partial h_S(\tilde{\mathbf{x}}_k \boxplus \tilde{\mathbf{x}}_k, {}^{L_{\rho_i}} \mathbf{p}_i)}{\partial \tilde{\mathbf{x}}_k} \Big|_{\tilde{\mathbf{x}}_k=0} = {}^G \mathbf{u}^T \left[-{}^G \hat{\mathbf{R}}_{I_k}^T [{}^I \mathbf{R}_L {}^{L_k} \mathbf{p}_i + {}^I \mathbf{p}_L] \times \mathbf{I}_{3 \times 3} \quad \mathbf{0}_{3 \times 12} \right] \quad (18)$$

$$\Sigma_{\eta_i^S} = \mathbf{F}_{\mathbf{n}_{p_i}} \Sigma_{\mathbf{n}_{p_i}} \mathbf{F}_{\mathbf{n}_{p_i}}^T, \mathbf{F}_{\mathbf{n}_{p_i}} = \frac{\partial h_S(\tilde{\mathbf{x}}_k, {}^{L_{\rho_i}} \mathbf{p}_i)}{\partial {}^{L_{\rho_i}} \mathbf{p}_i} \quad (19)$$

Then, we combine the prior distribution of \mathbf{x}_k with the geometric and reflectivity residual distributions to obtain the maximum a posteriori (MAP) estimation of $\tilde{\mathbf{x}}_k$:

$$\begin{aligned} \tilde{\mathbf{x}}_k^* = \underset{\tilde{\mathbf{x}}_k}{\operatorname{argmin}} & (\|\mathbf{x}_k \boxminus \hat{\mathbf{x}}_k\|_{\hat{\mathbf{P}}_k} + \sum_{j=1}^{m_C} \|\mathbf{z}_C + \mathbf{H}_j^C \tilde{\mathbf{x}}_k\|_{\Sigma_{\eta_j^C}} \\ & + \sum_{i=1}^{m_S} \|\mathbf{z}_S + \mathbf{H}_i^S \tilde{\mathbf{x}}_k\|_{\Sigma_{\eta_i^S}}) \end{aligned} \quad (20)$$

F. Global Map Management

Our global map inherits the ikd-tree map from FAST-LIO2 [3] to insert and delete points. Compared to the static kd-tree map, the ikd-tree map can dynamically balance the tree structure and downsample according to resolution [23]. During the initialization of the algorithm, we directly use the points of the first scan to construct the ikd-tree map. When the odometry outputs, the points of the current scan are transformed to the world frame and then incrementally inserted into the ikd-tree map together with their reflectivity values, see Fig. 1.

TABLE I
END-TO-END TRANSLATION ERRORS ON THE CAMPUS DATASET (METERS)

Method	Start and end outdoors		Start and end indoors	
	campus_1	campus_2	campus_3	campus_4
RI-LIO (DSR=66)	0.114	0.059	0.098	0.179
FAST-LIO2 (DSR=66)	8.177	Fail	0.099	0.184
VoxelMap (DSR=66)	0.076	0.241	0.050	0.148
A-LOAM (DSR=4)	0.114	0.155	0.175	1.602
LILI-OM (DSR=4)	1.741	0.957	2.896	3.289
LIO-SAM (DSR=4)	0.107	0.052	0.442	0.562

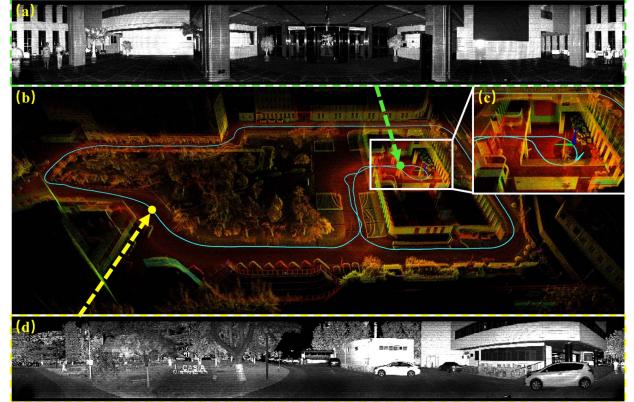


Fig. 5. Mapping results of RI-LIO using sequence *campus_3*. (b) shows the reconstructed point cloud rendered by reflectivity, and (c) shows the indoor details and the coincidence of the start and end coordinates. (a) and (d) show indoor and outdoor reflectivity images, respectively.

IV. EXPERIMENTS AND RESULTS

The existing public datasets do not provide the LiDAR intrinsics and high-resolution reflectivity images. In this section, we collected three private datasets (*campus_**, *road_**, and *field_**) to validate the proposed method.

The sensor suite for data collection includes an Ouster OS1-128 LiDAR with a built-in InvenSense ICM-20948 IMU, an onboard computer Jetson Xavier NX, and a D300 GNSS module for providing the outdoor ground truth with an accuracy of 10 mm + 1 ppm in RTK mode. These sensors are mounted on a self-driving logistics UGV and a 600 mm wheelbase quadrotor UAV.

To verify the effectiveness of introducing the reflectivity measurement and the direct idea, RI-LIO is compared with the direct LIO methods (i.e., FAST-LIO2 [3] and VoxelMap [4]) and the feature-based methods (i.e., A-LOAM, LILI-OM [7], and LIO-SAM [6]). VoxelMap refines the probability distribution of plane features in the voxel map based on FAST-LIO2. A-LOAM simplifies the optimization of LOAM [5].

All methods are implemented in C++ on the Robotics Operating System (ROS) and executed on a desktop computer equipped with an Intel i7-8700 CPU. To illustrate that direct methods do not rely on dense point features, we set the downsampling rate (DSR) of RI-LIO, FAST-LIO2, and VoxelMap to 66 (2000 input points per scan) for the sake of fairness. We set the DSR of the other methods to 4 (32700 input points per scan) to ensure that they can work properly. In addition, the common parameters involved in all algorithms are set to be consistent, and the loop closure is switched off. The results of all experiments are the mean values after five runs.

TABLE II
ABSOLUTE/RELATIVE TRANSLATION ERRORS ON THE ROAD DATASET (RMSE, METERS)

Sequence	Trajectory length (m)	Duration (s)	Average Speed (km/h)	RI-LIO (DSR=66)	RI-LIO (w/o ref.) (DSR=66)	RI-LIO (w/o cor.) (DSR=66)	FAST-LIO2 (DSR=66)	VoxelMap (DSR=66)	A-LOAM (DSR=4)	LILI-OM (DSR=4)	LIO-SAM (DSR=4)
road_1	1769.3	326.2	19.53	1.636 /0.089	3.734/0.087	2.460/ 0.085	4.039/0.090	13.610/0.094	8.274/0.097	11.292/0.203	11.037/0.142
road_2	2148.2	425.8	18.16	5.359 /2.580	5.969/2.565	8.552/ 2.537	5.806/2.577	24.541/2.546	12.395/2.885	19.550/5.527	16.998/4.743
road_3	2771.6	597.6	16.70	5.559 /2.672	7.691/2.690	10.370/2.722	6.863/2.732	10.113/ 2.633	13.274/4.878	16.123/4.291	15.176/4.867
road_4	2643.5	491.9	19.35	4.264 /3.038	5.174/3.496	8.762/3.038	4.970/3.037	12.558/ 2.963	12.437/4.740	11.793/4.766	7.055/7.979

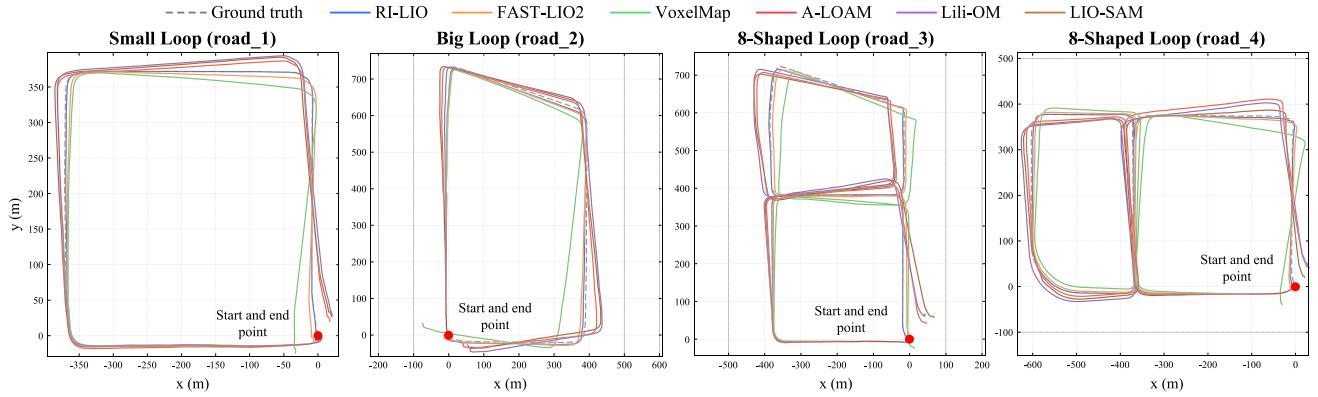


Fig. 6. Trajectory comparison between RI-LIO, FAST-LIO2, Voxelmap, A-LOAM, LILI-OM, LIO-SAM, and ground truth in the unstructured scenario experiments.

A. Structured Scenario Experiments

Structured scenario experiments were designed to show that our algorithm has comparable accuracy with geometrically enhanced (e.g., backend fusion and map refinement) algorithms in rich geometric feature scenarios. We collected the *campus_** dataset using an UGV on the campus of the Institute of Automation, Chinese Academy of Sciences. To further analyse the effect of scale on the algorithms, sequences *campus_1* and *campus_2* start and end outdoors, while sequences *campus_3* and *campus_4* start and end indoors.

Due to building blockage, the GPS does not converge to an acceptable accuracy throughout. Therefore, our acquisition trajectory starts and ends in the same place to assess the end-to-end error of the algorithms, see Table I. As the vehicle was at low speed, VoxelMap achieved the best results by using repeatedly scanned planes to build an accurate map of probability distributions. LIO-SAM also achieves the best results outdoors, but the feature extraction is not adaptive to indoor and outdoor scale changes, resulting in reduced accuracy in sequences *campus_3* and *campus_4*. Without any enhancement of geometric features and back-end optimization, our method errors differ from the best methods by only 0.01 m ~ 0.05 m. In contrast, FAST-LIO2, which has exactly the same geometric measurement as ours, shows failure and large drift in sequences *campus_1* and *campus_2*. Finally, we present the reconstructed map of sequence *campus_3* in Fig. 5, and it is clear that our calculated indoor starting and ending coordinates almost coincide in Fig. 5(b).

B. Unstructured Scenario Experiments

In this subsection, we show the capacity of our algorithm for precise localization on large-scale suburban roads. We collected the *road_** dataset on suburban roads located in Huairou District, Beijing. There are almost no buildings and only street trees on the



Fig. 7. The acquisition trajectory aligned satellite map for the *road_** dataset. The vehicle is manually controlled starting and ending at the same position.

sides of the selected roads. For more comprehensive evaluations, we collected several motion trajectories of different lengths and shapes, see Fig. 7.

The D300 GNSS module was in RTK mode throughout the experiments, so we treat its output positions as the ground truth. Table II shows the root-mean-square error (RMSE) results of each method under different motion trajectories. Since the extrinsics of LiDAR and GNSS were not calibrated, the absolute/relative translation errors (A/RTE) are calculated after trajectory alignment by evo [24]. To verify the reflectivity image measurement and the corrected projection model (CPBRA), we enable only geometric measurement (termed RI-LIO (w/o ref.)) and use the uncorrected projection model (PBRA) (termed RI-LIO (w/o cor.)), respectively.

RI-LIO has the best accuracy compared to all other algorithms in the ATE evaluation. Since RI-LIO (w/o ref.) is consistent with FAST-LIO2 in the geometry measurement, they perform

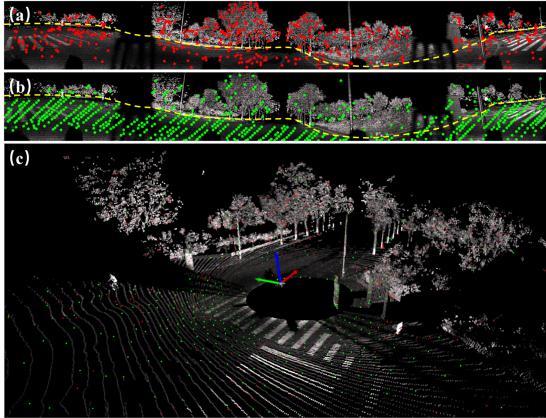


Fig. 8. The effective reflectivity points (red) and the effective geometric points (green) of a scan in the reflectivity image (a)(b) and the point cloud (c) using sequence *road_1*. The yellow dashed lines in (a) and (b) divide the plants above from the ground below.

at the same level. RI-LIO (w/o cor.) projects map point to an incorrect pixel position and cause an inaccurate reflectivity residual, so it will obviously reduce the performance. To explain the improvement of RI-LIO over RI-LIO (w/o ref.), we recorded the effective reflectivity points and geometric points of a scan, see Fig. 8. It is clear from the distribution of the effective reflectivity points that the proposed method can capture both geometric textures on the ground (e.g., lane lines and zebra lines) and non-geometric textures on the plants (e.g., cluttered leaves and grass). In contrast, the number of effective geometric points decreases sharply in the plants above ground, which results in underconstrained pose estimation in the horizontal direction.

Notably, RI-LIO has less accuracy than the best methods by approximately 0.05 m in the RTE evaluation. In contrast, VoxelMap, which has a large absolute translation error, has a much smaller relative translation error. It is possible that the increased uncertainty of the estimated normal vectors of planes causes the drift to be in a direction away from the ground truth. RI-LIO, on the other hand, does not have the best RTE performance, but based on the additional reflectivity texture constraints in the horizontal direction (Fig. 8), is able to correct the drift direction close to the ground truth, resulting in better global performance. To demonstrate this characteristic more visually, we align the first 20 seconds trajectories of each method and the ground truth to plot the entire trajectories, as shown in Fig. 6. It is clear that the trajectories estimated by RI-LIO are the closest to the ground truth and can finally approach or return to the starting points. This suggests that our method has the least global drift during motion compared to the other methods.

C. Challenging Open Field Experiments

To test the performance of our algorithm under extreme conditions, we use an UAV to fly close to the ground above a vacant field with no significant landmarks in Huairou District, Beijing. During the experiments, the D300 GNSS module was working in RTK mode throughout, so we take its output positions as the ground truth.

In this scenario, only the ground serves as a significant geometric feature, and none of the geometric enhancement

TABLE III
ABSOLUTE/RELATIVE TRANSLATION ERRORS ON THE FIELD DATASET (RMSE, METERS)

Sequence	Trajectory length (m)	Maximum tilt angle (deg)	RI-LIO (DSR=66)	FAST-LIO2 (DSR=66)
field_1	488.7	27.9	0.398/0.121	0.541/0.129
field_2	546.4	30.6	0.461/0.064	0.565/0.070
field_3	711.9	31.1	0.696/0.081	2.441/0.092

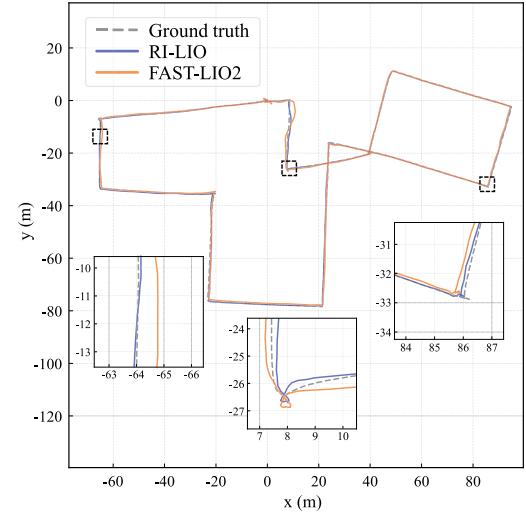


Fig. 9. Trajectory comparison between RI-LIO, FAST-LIO2, and ground truth using sequence *field_2*. The other methods are not demonstrated due to their failure in this scenario.

methods yield meaningful results. They all show a significant drift along the horizontal direction as they penetrate deeper into the hinterland, especially when the UAV performs yawing manoeuvres. After analysis, we find that the feature-based methods are almost only able to extract the ground plane features in this situation, which leads to underconstrained pose estimation in the horizontal direction. VoxelMap clusters the ground normal vectors, resulting in localized ground undulation textures being ignored. In contrast, FAST-LIO2 obtains some horizontally constrained components through local normal vector estimation and produces meaningful results.

In Table III, we test RI-LIO and FAST-LIO2 at different speeds and declination angles. RI-LIO achieves the best performance in all sequences and is more robust than FAST-LIO2 over long flight. This is because RI-LIO can add texture constraints on top of a small amount of geometric features in this scenario. Taking sequence *field_2* as an example, the average numbers of effective plane points and reflectivity points of RI-LIO are 267 and 408, respectively, and the drift is less than 0.04% (0.20 m end-to-end translation error over 546.4 m). In contrast, the average number of effective plane points of FAST-LIO2 is 264, and the drift is 0.06% (0.34 m end-to-end translation error over 546.4 m).

Fig. 9 shows the comparison of trajectories between RI-LIO, FAST-LIO2, and the ground truth. Without the constraints of ground textures, the trajectory of the geometric-only method FAST-LIO2 shows a significant drift, while RI-LIO makes its trajectory almost coincide with the ground truth by adding reflectivity textures. By merging the point cloud generated by

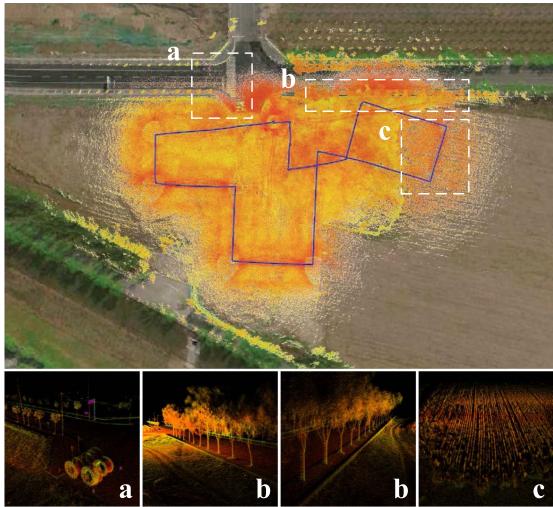


Fig. 10. Mapping results of RI-LIO using sequence *field_2*. The upper figure is the mapping result and trajectory embedded in the satellite map, and the lower figures are the point cloud details in the white dashed boxes.

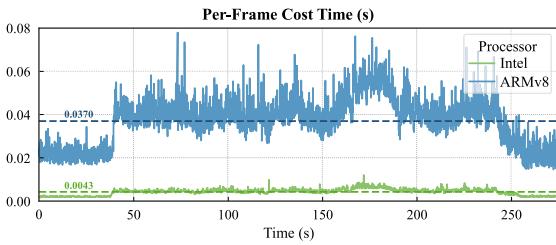


Fig. 11. Per-frame cost time of RI-LIO using Intel and ARMv8 processors.

RI-LIO into the satellite map, the accuracy of our algorithm in challenging scenarios is further examined, see Fig. 10.

D. Time Analysis

To test the real-time performance of our algorithm on desktop computers and small embedded computing platforms, we analyse the computational cost in the sequence *field_2*, see Fig. 11. The average processing time on Intel i7-8700 CPU@3.2 GHz and NVIDIA Carmel ARMv8@1.9 GHz are 0.0043s and 0.037s, respectively.

V. CONCLUSION

In this letter, we proposed RI-LIO, a reflectivity image assisted tightly-couple LiDAR-inertial odometry. Similar to the camera approaches, we build a corrected projection model to project the spatial points into the image frame. The sparse image rendered from the map is then directly aligned with the current reflectivity image to obtain the reflectivity residual. The IEKF framework tightly couples the traditional point-to-plane geometry residual with the reflectivity residual to obtain the optimized poses. Extensive quantitative and qualitative experiments show that the introduction of reflectivity images improves the accuracy and robustness of LIO. Furthermore, in the extreme case of LiDAR degradation and fast-moving UAVs, the method can show more

stable and accurate localization and mapping results. Future work will introduce back-end optimization and loop closure to improve the global consistency of localization.

REFERENCES

- [1] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.
- [2] T. Qin, P. Li, and S. Shen, "VINS-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, Aug. 2018.
- [3] 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.
- [4] 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.
- [5] J. Zhang and S. Singh, "LOAM: LiDAR odometry and mapping in real-time," in *Proc. Robot. Sci. Syst.*, vol. 2, 2014, pp. 1–9.
- [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.*, 2020, pp. 5135–5142.
- [7] K. Li, M. Li, and U. D. Hanebeck, "Towards high-performance solid-state-LiDAR-inertial odometry and mapping," *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 5167–5174, Jul. 2021.
- [8] S.-S. Huang, Z.-Y. Ma, T.-J. Mu, H. Fu, and S.-M. Hu, "LiDAR-monocular visual odometry using point and line features," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 1091–1097.
- [9] W. Shao, S. Vijayarangan, C. Li, and G. Kantor, "Stereo visual inertial LiDAR simultaneous localization and mapping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 370–377.
- [10] J. Zhang and S. Singh, "Laser-visual-inertial odometry and mapping with high robustness and low drift," *J. Field Robot.*, vol. 35, no. 8, pp. 1242–1264, 2018.
- [11] R. Voges and B. Wagner, "Interval-based visual-LiDAR sensor fusion," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 1304–1311, Apr. 2021.
- [12] X. Zuo, P. Geneva, W. Lee, Y. Liu, and G. Huang, "LIC-fusion: LiDAR-inertial-camera odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 5848–5854.
- [13] X. Zuo et al., "LIC-fusion 2.0: LiDAR-inertial-camera odometry with sliding-window plane-feature tracking," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 5112–5119.
- [14] J. Lin and F. Zhang, "R3live: A robust, real-time, RGB-colored, LiDAR-inertial-visual tightly-coupled state estimation and mapping package," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2022, pp. 10672–10678.
- [15] T. Shan, B. Englot, C. Ratti, and D. Rus, "LVI-SAM: Tightly-coupled LiDAR-visual-inertial odometry via smoothing and mapping," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 5692–5698.
- [16] C. Zheng, Q. Zhu, W. Xu, X. Liu, Q. Guo, and F. Zhang, "Fast-Livo: Fast and tightly-coupled sparse-direct LiDAR-inertial-visual odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Kyoto, Japan, 2022, pp. 4003–4009.
- [17] D. C. Brown, "Decentering distortion of lenses," *Photogrammetric Eng. Remote Sens.*, vol. 32, no. 3, pp. 444–462, 1966.
- [18] H. Wang, C. Wang, and L. Xie, "Intensity scan context: Coding intensity and geometry relations for loop closure detection," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 2095–2101.
- [19] T. Shan, B. Englot, F. Duarte, C. Ratti, and D. Rus, "Robust place recognition using an imaging LiDAR," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 5469–5475.
- [20] H. Wang, C. Wang, and L. Xie, "Intensity-SLAM: Intensity assisted localization and mapping for large scale environment," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 1715–1721, Apr. 2021.
- [21] T. Wu, H. Fu, B. Liu, H. Xue, R. Ren, and Z. Tu, "Detailed analysis on generating the range image for LiDAR point cloud processing," *Electronics*, vol. 10, no. 11, 2021, Art. no. 1224.
- [22] M. Levoy, "Display of surfaces from volume data," *IEEE Comput. Graph. Appl.*, vol. 8, no. 3, pp. 29–37, May 1988.
- [23] Y. Cai, W. Xu, and F. Zhang, "ikd-Tree: An incremental KD tree for robotic applications," 2021, doi: [10.48550/arXiv.2102.10808](https://arxiv.org/abs/2102.10808).
- [24] M. Grupp, "evo: Python package for the evaluation of odometry and SLAM," 2017. [Online]. Available: <https://github.com/MichaelGrupp/evo>