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

Cubemap-Based LiDAR-Inertial Odometry with Intensity Assistance

Yang Liu, Kazushige Yamamoto, Atsushi Matsui, Saburo Takahashi, Toshihisa Abe

***Abstract*— We present CUBE-LIO, a LiDAR-inertial odometry framework that leverages direct photometric constraints from LiDAR intensity to improve robustness in geometrically degenerate environments. At its core is an efficient cubemap projection that maps LiDAR intensity onto six cube faces, eliminating pole singularities and severe polar distortion. This yields a more uniform spatial sampling while avoiding the costly trigonometric operations typical of equirectangular mappings. Building on this representation, we introduce a semi-dense feature selection and direct optimization strategy based on intensity gradient magnitude. This strategy improves resilience to intensity noise and variations induced by range and incidence angle. Photometric constraints are jointly optimized with geometric measurements in a tightly coupled LIO pipeline. CUBE-LIO is sensor-agnostic and supports both spinning and solid-state LiDARs. Experiments on multiple public benchmarks demonstrate state-of-the-art accuracy and real-time performance, with particularly pronounced gains in scenes where the geometric structure is sparse or weak.**

I. INTRODUCTION

Driven by advances in 3D LiDAR technology and modern SLAM algorithms, LiDAR-based SLAM has become a core technology in robotics and automation. However, it still faces challenges in geometrically degenerate environments, such as tunnels or open spaces. Since LiDAR primarily captures geometric information, performance degrades when structural constraints are weak, often resulting in poor convergence and increased drift.

A common method to address this problem involves incorporating cameras to capture texture-rich information. Although fusing visual and geometric cues improves robustness, it increases hardware cost and system complexity, demanding precise calibration and synchronization. Moreover, cameras are sensitive to lighting conditions and degrade in low-light or high dynamic range environments.

An alternative approach leverages the intensity information provided by most LiDAR sensors. The intensity at each point, influenced by material properties, distance, and incidence angle, offers complementary non-geometric cues that can enhance localization robustness without requiring additional sensors. Recent methods (e.g., [2, 3]) project reflectivity or intensity measurements onto images using equirectangular projection, enabling joint optimization of geometric and photometric constraints. However, this projection introduces significant distortion near the poles and tends to be computationally inefficient.

To address these limitations, we propose a cubemap-based projection scheme for intensity-assisted LiDAR-inertial odometry. This model achieves faster computation than conventional equirectangular projections while reducing polar distortion. To mitigate intensity noise and variations due to measurement conditions such as distance and incidence angle, we introduce a semi-dense intensity gradient magnitude (IGM) optimization strategy. We select semi-dense, information-rich regions and perform photometric optimization based on IGM values that remain relatively consistent under different measurement conditions, thereby improving robustness in geometrically degenerate environments.

Fig. 1. CUBE-LIO achieves high accuracy on multiple geometrically degenerate public benchmarks.

The main contributions of this study are as follows:

- We propose a cubemap-based LiDAR-inertial odometry (LIO) framework that leverages LiDAR intensity. This cubemap projection effectively maps LiDAR intensity onto six faces of a cube, substantially reducing polar distortion and achieving high computational efficiency compared with equirectangular projection.
- We introduce semi-dense IGM-based feature selection and direct IGM-constancy optimization, leveraging LiDAR intensity cues to enhance localization robustness while maintaining resilience to intensity noise and measurement variations.
- Our intensity-assisted LIO framework is sensor-agnostic, supporting both spinning and solid-state LiDARs.

All authors are with the Panasonic Advanced Technology Development Co., Ltd., liu.yang@panasonic.com

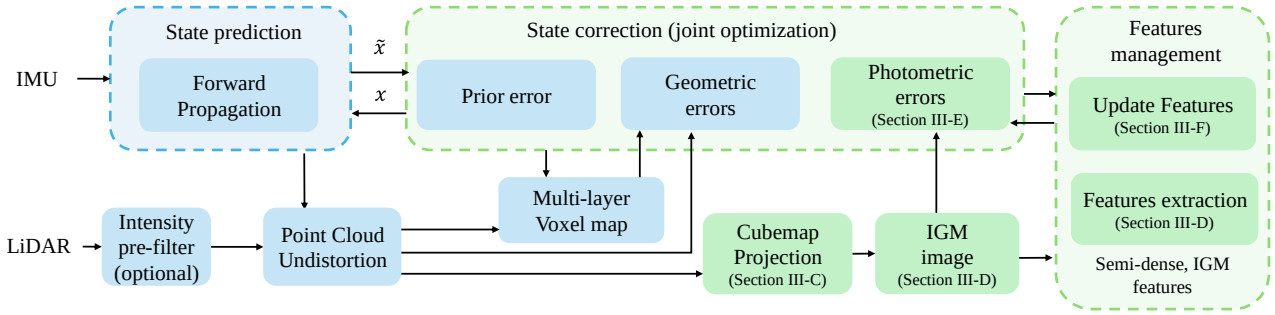


Fig. 2. Overview of the CUBE-LIO system. CUBE-LIO extends the iterated error-state Kalman filter (IESKF) framework [1] to jointly optimize both geometric and photometric errors from LiDAR measurements, with the photometric optimization (main contribution) highlighted in green.

II. RELATED WORK

A. Geometric-Only LiDAR(-Inertial) Odometry

LOAM [4], a seminal study on feature-based LiDAR odometry, utilizes edge and planar features for robust matching and nonlinear optimization, inspiring subsequent approaches such as LeGO-LOAM [5]. LeGO-LOAM enhances the efficiency and robustness of ground robots by introducing ground segmentation, whereas LIO-SAM [6] extends LOAM by adopting a factor graph framework [7] with IMU preintegration for tightly coupled LiDAR-inertial optimization. FAST-LIO [1] proposed a tightly coupled iterated error-state Kalman filter (IESKF) to fuse LiDAR feature points with IMU data. FAST-LIO2 [8] introduced an incremental k-d tree for efficient map management. VoxelMap [9] advanced map management using a voxel-based approach, offering faster processing speed than k-d tree structures. F3-SLAM [10] proposed a multi-layer 3D voxel map with adaptive feature selection, achieving high localization accuracy and computational efficiency.

Despite their success in structured environments, geometric-only methods have fundamental limitations in geometrically degenerate scenarios.

B. LiDAR-Visual-Inertial Odometry

Integrating visual sensors to address geometry-only LIO limitations in degenerate scenes has become a common approach for enhancing robustness. Recent LiDAR-Visual-Inertial Odometry (LIVO) methods, such as [11, 12], utilize direct methods or optical flow tracking to provide complementary visual constraints and mitigate the degradation of purely geometric information. Building upon these frameworks, FAST-LIVO2 [13] dynamically optimizes image patches and adjusts exposure times, achieving leading accuracy among current LIVO methods.

However, visual-LiDAR fusion increases system complexity and demands precise time synchronization and extrinsic calibration, whereas cameras are inherently sensitive to lighting variations.

C. Intensity-Assisted LiDAR-Inertial Odometry

Unlike camera imagery, LiDAR intensity is inherently temporally and spatially aligned with geometry and is insensitive to ambient illumination, enabling robust operation even in complete darkness.

Some studies have integrated intensity into ICP-like scan-to-map registration. Representative methods include [14, 15], which introduce explicit intensity residuals; [16], which retains purely geometric residuals while using intensity primarily for correspondence weighting; and [17, 18], which extract intensity edges/gradients as features. However, pipelines that rely on 3D intensity maps are constrained by map resolution, often smoothing fine-scale cues.

With the increasing density of LiDAR point clouds, recent approaches project intensity onto 2D images to effectively preserve fine-scale structures. Some methods [19, 20] extract salient points from intensity images using feature extraction frameworks (e.g., ORB and SuperPoint) and track them, in a manner similar to feature-based visual SLAM. However, in degenerate scenarios, salient point extraction may be unreliable. By contrast, RI-LIO [2] and COIN-LIO [3] adopt a direct approach similar to direct visual SLAM, projecting intensity values onto 2D images and jointly optimizing photometric and geometric errors. COIN-LIO selects geometrically complementary sparse patches for effective photometric optimization, demonstrating robustness in geometrically degenerate datasets. Recently, PG-LIO [21] incorporates intensity as a constraint factor into the factor graph framework.

These methods typically employ equirectangular projection to convert LiDAR data to images. However, this projection introduces significant distortion near the poles, tends to be computationally inefficient, and provides limited support for solid-state LiDARs.

Building upon the direct photometric optimization paradigm, our method introduces a cubemap-based projection to achieve efficient, low-distortion intensity imaging. Furthermore, we propose semi-dense IGM feature selection and direct IGM-constancy optimization, yielding features that are more robust and significantly enhancing odometry reliability in geometrically degenerate environments.

III. METHODOLOGY

A. Overall Framework

Our system (Fig. 2) adopts an IESKF framework as utilized in [1], for tightly coupled fusion of LiDAR and IMU

data, ensuring robust state estimation.

IESKF state estimation and geometric state corrections follow established methodology. For geometric feature management, we employ a multi-layer voxel grid structure [10]. Unlike previous geometric-only methods, our approach jointly optimizes geometric and photometric errors. As shown in Fig. 2, all LiDAR points are motion-compensated before photometric optimization, following [1]. For sensors like Ouster, intensity artifacts are pre-filtered using the method in [3] to ensure reliable optimization.

Given that the core contribution of this study lies in photometric optimization, the following sections focus on this aspect in detail.

B. Notation

We define the system state as

$$\mathbf{x} = \left[{}^{G}\mathbf{t}_B,\ {}^{G}\mathbf{R}_B,\ {}^{G}\mathbf{v},\ \mathbf{b}_\omega,\ \mathbf{b}_a\right]^T \tag{1}$$

where ${}^{G}\mathbf{t}_B$ and ${}^{G}\mathbf{R}_B$ represent translation and rotation from the body frame to the global frame, and ${}^{G}\mathbf{v}$ denotes velocity in the global frame. Specifically, ${}^{G}\mathbf{R}_B \in SO(3)$ and ${}^{G}\mathbf{t}_B,\ {}^{G}\mathbf{v} \in \mathbb{R}^3$. The terms $\mathbf{b}_\omega$ and $\mathbf{b}_a$ correspond to the gyroscope and accelerometer biases of the IMU. We assume the rigid-body transformations from LiDAR and IMU to the body frame are pre-calibrated and known. All sensor data are transformed into the body frame before processing. The variables $\mathbf{x}$, $\hat{\mathbf{x}}$, and $\tilde{\mathbf{x}}$ denote the true, predicted, and error states, respectively. The error state is the difference between the true and predicted states.

The measurement model $\mathrm{h}(\mathbf{x})$ can be linearized as follows: $\mathrm{h}(\mathbf{x}) = \mathrm{h}(\hat{\mathbf{x}} \boxplus \tilde{\mathbf{x}}) \approx \mathrm{h}(\hat{\mathbf{x}}) + \mathbf{H}\tilde{\mathbf{x}}$. where $\mathrm{h}(\hat{\mathbf{x}})$ is the residual evaluated at $\hat{\mathbf{x}}$ (denoted as $\mathbf{z}$), and $\mathbf{H}$ is the Jacobian of h. Once the photometric residual $\mathbf{z}_{\text{pho}}$ and its Jacobian $\mathbf{H}_{\text{pho}}$ are derived, geometric and photometric errors are jointly optimized in the maximum a posteriori estimation of $\tilde{\mathbf{x}}$

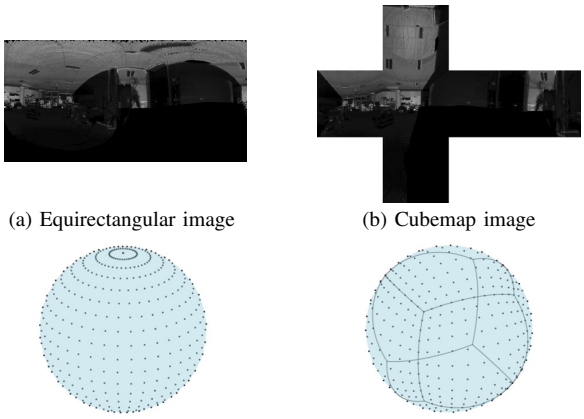
$$\begin{aligned}\arg\min_{\tilde{\mathbf{x}}}\Big(&\|\tilde{\mathbf{x}}\|_{\mathbf{P}} + \sum_{i=1} \|\mathbf{z}_{i,\text{pho}} + \mathbf{H}_{i,\text{pho}}\tilde{\mathbf{x}}\|_{\mathbf{\Sigma}_{\text{pho}}} \\ &+ \sum_{j=1} \|\mathbf{z}_{j,\text{geo}} + \mathbf{H}_{j,\text{geo}}\tilde{\mathbf{x}}\|_{\mathbf{\Sigma}_{\text{geo}}}\Big)\end{aligned} \tag{2}$$

C. Image Projection Model

Existing intensity-assisted LiDAR-inertial odometry methods typically employ equirectangular projection to convert LiDAR range/intensity data into 2D images. However, this representation presents three key limitations:

(i) Although the range/intensity image of a rotating LiDAR can be built via look-up table (LUT), photometric optimization still requires a large number of trigonometric reprojection computations, forming a persistent computational bottleneck.

(ii) Although adopting a LUT can improve the efficiency of equirectangular projection, it severely limits generality for irregular sampling patterns, such as those of solid-state and non-repetitive LiDARs, which often lack fixed elevation channels or uniform azimuth sampling.



(a) Equirectangular image (b) Cubemap image

(c) Sampling points for equirectangular (d) Sampling points for cubemap

Fig. 3. Comparison of projection models (note that (c) and (d) visualize sampling points on the unit sphere.)

(iii) The equirectangular projection introduces pronounced distortion in high-latitude (polar) regions, especially in hemispherical LiDARs, downward-facing setups, or multi-LiDAR configurations. Such distortion causes pixels at the poles to be severely stretched. Consequently, when computing image gradients via finite differences (e.g., exploring neighboring pixels), the effective spatial receptive field becomes extremely localized, which degrades the reliability of the gradient estimation. Furthermore, this stretching wastes significant computational resources in these oversampled regions, degrading feature scale and statistical uniformity.

To address these limitations, we introduce cube mapping from computer graphics [22] and propose a cubemap projection model for LIO. This efficiently maps 3D LiDAR points onto six cube faces, reducing polar distortion and computational overhead.

Cubemap projection (Fig. 3) (b) avoids severe pole distortions evident in equirectangular projection (a) (e.g., LED ceiling lights becoming substantially deformed). In Fig. 3 (c) and (d), where sampling points are visualized on a unit sphere, it can be observed that the cubemap projection (d) provides a more uniform distribution of sampling points during discretization compared to the equirectangular projection (c). This uniform sampling characteristic contributes to higher precision in feature extraction and optimization. Building on the cubemap projection model, we further develop methods for feature extraction and photometric error optimization.

Although low-distortion alternatives exist (e.g., HEALPix [23]), the cubemap is simpler and its six rectangular faces directly reuse OpenCV pipelines, offering clear implementation and integration advantages.

In this paper, we denote the cubemap projection by Π. This maps each 3D LiDAR point ${}^{B}\mathbf{p} = [x, y, z]^T$ to the augmented cubemap pixel coordinate $\bar{\mathbf{u}} = [\mathbf{u}, f]^T$. f is the face index and $\mathbf{u} = [u, v]^T$ is the 2D pixel coordinate on face f.

$$\bar{\mathbf{u}} = \Pi(\mathbf{p}) = \left[\left(1 + \frac{\mathbf{g}_u^{T\,B}\mathbf{p}}{\mathbf{g}_a^{T\,B}\mathbf{p}}\right)\frac{r}{2},\ \left(1 + \frac{\mathbf{g}_v^{T\,B}\mathbf{p}}{\mathbf{g}_a^{T\,B}\mathbf{p}}\right)\frac{r}{2},\ f\right]^T \tag{3}$$

where r denotes the pixel resolution of each cubemap face.

The face index f assumes values ranging from 0 to 5, corresponding to the six faces of the cube: $+X$, $-Y$, $-X$, $+Y$, $+Z$, and $-Z$, which can be determined based on the relative magnitudes of the coordinates x, y, and z.

$$f = \begin{cases} 0 & \text{if } |x| \geq |y|,\ |x| \geq |z|,\ x > 0 \quad (+X) \\ 1 & \text{if } |y| \geq |x|,\ |y| \geq |z|,\ y < 0 \quad (-Y) \\ 2 & \text{if } |x| \geq |y|,\ |x| \geq |z|,\ x < 0 \quad (-X) \\ 3 & \text{if } |y| \geq |x|,\ |y| \geq |z|,\ y > 0 \quad (+Y) \\ 4 & \text{if } |z| > |x|,\ |z| > |y|,\ z > 0 \quad (+Z) \\ 5 & \text{if } |z| > |x|,\ |z| > |y|,\ z < 0 \quad (-Z) \end{cases} \tag{4}$$

$\mathbf{g}_u, \mathbf{g}_v$ indicate the local horizontal and vertical directions on the face, respectively, and $\mathbf{g}_a$ aligns with the dominant axis of the face (the projection direction).

$$\mathbf{g}_u, \mathbf{g}_v, \mathbf{g}_a = \begin{cases} [0,-1,0]^T,\ [0,0,-1]^T,\ [+1,0,0]^T & f=0 \\ [-1,0,0]^T,\ [0,0,-1]^T,\ [0,-1,0]^T & f=1 \\ [0,+1,0]^T,\ [0,0,-1]^T,\ [-1,0,0]^T & f=2 \\ [+1,0,0]^T,\ [0,0,-1]^T,\ [0,+1,0]^T & f=3 \\ [0,-1,0]^T,\ [+1,0,0]^T,\ [0,0,+1]^T & f=4 \\ [0,-1,0]^T,\ [-1,0,0]^T,\ [0,0,-1]^T & f=5 \end{cases} \tag{5}$$

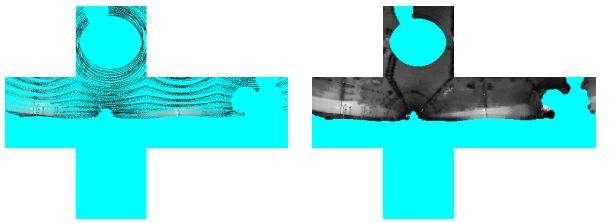
Although cubemap projection (3) provides low-distortion mapping and computational efficiency, it introduces discontinuities at seams between cube faces. This can adversely affect gradient-based optimization and related operations. To address this issue, we map pixels that fall outside a face to their corresponding positions on adjacent faces, ensuring continuity for interpolation and differentiation. Details are presented in Section III-E.

Direct point-wise projection typically results in sparse occupancy on cubemap faces due to the non-uniform sampling of LiDAR sensors. To produce continuous maps suitable for photometric error computation, we employ Inverse Distance Weighting (IDW) interpolation [24]. Figure 4 illustrates this effect on data from a Livox MID-360 in a tunnel, where cyan pixels indicate missing regions caused by sparse LiDAR sampling. With appropriate selection of r and interpolation radius, IDW effectively converts sparse LiDAR points into dense, continuous intensity cubemaps.

Building on the cubemap projection model, we generate intensity and range cubemaps, denoted as $\mathrm{I}(\cdot)$ and $\mathrm{R}(\cdot)$. Both cubemaps are integral to feature extraction.

D. Feature Extraction

To robustly extract tracking features from intensity cubemaps, we employ the intensity gradient magnitude (IGM).



(a) point-wise projection (b) IDW interpolation on cubemaps

Fig. 4. Effect of IDW interpolation on cubemaps.

Unlike raw intensity values, which vary with incidence angle and range, gradients emphasize local structural changes and are less sensitive to low-frequency measurement variations.

We compute the IGM using first-order Gaussian derivative filters [25], which intrinsically combine smoothing and differentiation, thereby regularizing the gradient estimation and yielding stable gradient responses even in the presence of noise.

Let G_x and G_y denote the horizontal and vertical Gaussian derivative responses, respectively. The IGM at pixel $\bar{\mathbf{u}}$ is defined as:

$$\mathrm{M}(\bar{\mathbf{u}}) = \sqrt{G_x(\bar{\mathbf{u}})^2 + G_y(\bar{\mathbf{u}})^2}. \tag{6}$$

Instead of uniform sampling or sparse feature detection, we adopt a semi-dense selection: we retain only pixels whose IGM exceeds a predefined threshold together with their immediate neighborhood. This focuses computation on information-rich regions, analogous to semi-dense pixel selection in visual odometry (VO) [26]. By concentrating the direct optimization on photometrically salient areas, we improve both robustness and accuracy.

For each selected pixel $\bar{\mathbf{u}}$, we recover the corresponding 3D point by applying the inverse cubemap projection $\Pi^{-1}(\bar{\mathbf{u}})$ with the range value of the pixel.

The inverse cubemap projection $\Pi^{-1}(\bar{\mathbf{u}})$ is defined as follows:

$$\Pi^{-1}(\bar{\mathbf{u}}) = \begin{cases} \left[1, -\frac{2u}{r}+1, -\frac{2v}{r}+1\right]^T & \text{if } f=0 \\ \left[-\frac{2u}{r}+1, -1, -\frac{2v}{r}+1\right]^T & \text{if } f=1 \\ \left[-1, \frac{2u}{r}-1, -\frac{2v}{r}+1\right]^T & \text{if } f=2 \\ \left[\frac{2u}{r}-1, 1, -\frac{2v}{r}+1\right]^T & \text{if } f=3 \\ \left[\frac{2v}{r}-1, -\frac{2u}{r}+1, 1\right]^T & \text{if } f=4 \\ \left[-\frac{2v}{r}+1, -\frac{2u}{r}+1, -1\right]^T & \text{if } f=5 \end{cases} \tag{7}$$

The 3D point in the body frame is then computed as follows:

$${}^{B}\mathbf{p} = \frac{\Pi^{-1}(\bar{\mathbf{u}})}{\|\Pi^{-1}(\bar{\mathbf{u}})\|}\mathrm{R}(\bar{\mathbf{u}}) \tag{8}$$

Finally, the 3D point in the global coordinate system is reconstructed as ${}^{G}\mathbf{p}$ using the transformations T and ${}^{B}\mathbf{p}$. The corresponding IGM value at $\mathbf{u}$ is denoted as $m = \mathrm{M}(\mathbf{u})$. The pair ${}^{G}\mathbf{p}$ and m represent the extracted feature points.

$${}^{G}\mathbf{p} = \mathrm{T}(\mathbf{x}, {}^{B}\mathbf{p}) = {}^{G}\mathbf{R}_B\, {}^{B}\mathbf{p} + {}^{G}\mathbf{t}_B \tag{9}$$

E. Photometric Optimization

We minimize the IGM-constancy residual, defined as the difference between the current-frame IGM at the projected pixel of the i-th tracked feature ${}^{G}\mathbf{p}_i$ and its reference value:

$$z_{i,\mathrm{pho}} = \mathrm{M}\big(\Pi(\mathrm{T}^{-1}(\mathbf{x}, {}^{G}\mathbf{p}_i))\big) - m_i \tag{10}$$

Here, $\mathrm{T}^{-1}(\cdot)$ denotes the transformation from the global frame to the body frame and is defined as

$${}^{B}\mathbf{p}_i = \mathrm{T}^{-1}(\mathbf{x}, {}^{G}\mathbf{p}_i) = {}^{G}\mathbf{R}_B^T\left({}^{G}\mathbf{p}_i - {}^{G}\mathbf{t}_B\right) \tag{11}$$

By applying the chain rule, the Jacobian of the photometric residual with respect to the state is given by

$$\mathbf{H}_{i,\text{pho}} = \frac{\partial \mathrm{M}(\bar{\mathbf{u}}_i)}{\partial \mathbf{u}_i} \cdot \frac{\partial \Pi({}^{B}\mathbf{p}_i)}{\partial {}^{B}\mathbf{p}_i} \cdot \frac{\partial \mathrm{T}^{-1}(\mathbf{x}, {}^{G}\mathbf{p}_i)}{\partial \mathbf{x}} \tag{12}$$

The Jacobian of the transformation $\frac{\partial \mathrm{T}^{-1}(\mathbf{x}, {}^{G}\mathbf{p}_i)}{\partial \mathbf{x}}$ is computed as

$$\frac{\partial \mathrm{T}^{-1}(\mathbf{x}, {}^{G}\mathbf{p}_i)}{\partial \mathbf{x}} = \begin{bmatrix} -{}^{G}\mathbf{R}_B^T & \left[{}^{G}\mathbf{R}_B^T({}^{G}\mathbf{p}_i - {}^{G}\mathbf{t}_B)\right]_\times & \mathbf{0} \end{bmatrix} \tag{13}$$

The Jacobian of the cubemap projection $\frac{\partial \Pi({}^{B}\mathbf{p}_i)}{\partial {}^{B}\mathbf{p}_i}$ is

$$\frac{\partial \Pi({}^{B}\mathbf{p}_i)}{\partial {}^{B}\mathbf{p}_i} = \frac{r}{2({\mathbf{g}_a^T}^{B}\mathbf{p}_i)^2} \begin{bmatrix} ({\mathbf{g}_a^T}^{B}\mathbf{p}_i)\mathbf{g}_u^T - ({\mathbf{g}_u^T}^{B}\mathbf{p}_i)\mathbf{g}_a^T \\ ({\mathbf{g}_a^T}^{B}\mathbf{p}_i)\mathbf{g}_v^T - ({\mathbf{g}_v^T}^{B}\mathbf{p}_i)\mathbf{g}_a^T \end{bmatrix} \tag{14}$$

The image gradient $\frac{\partial \mathrm{M}(\bar{\mathbf{u}}_i)}{\partial \mathbf{u}_i}$ is typically computed using a central finite-difference scheme. However, on the cubemap, finite differences may involve pixel offsets that cross face seams, resulting in sampling outside the current face. To address such cases, we define a seam-aware remapping operator $\mathcal{W}$ as follows:

$$\mathcal{W}(\bar{\mathbf{u}}) = \begin{cases} \bar{\mathbf{u}}, & \text{if } 0 \le u < r \ \wedge \ 0 \le v < r \\ \Pi\big(\Pi^{-1}(\bar{\mathbf{u}})\big), & \text{otherwise} \end{cases} \tag{15}$$

Using this operator, the central finite-difference gradient at $\bar{\mathbf{u}}_i$ is computed as follows:

$$\frac{\partial \mathrm{M}(\bar{\mathbf{u}}_i)}{\partial \mathbf{u}_i} = \begin{bmatrix} \frac{\mathrm{M}\left(\mathcal{W}(\bar{\mathbf{u}}_i + \boldsymbol{\delta}_u)\right) - \mathrm{M}\left(\mathcal{W}(\bar{\mathbf{u}}_i - \boldsymbol{\delta}_u)\right)}{2} \\ \frac{\mathrm{M}\left(\mathcal{W}(\bar{\mathbf{u}}_i + \boldsymbol{\delta}_v)\right) - \mathrm{M}\left(\mathcal{W}(\bar{\mathbf{u}}_i - \boldsymbol{\delta}_v)\right)}{2} \end{bmatrix} \tag{16}$$

where $\boldsymbol{\delta}_u = [1, 0, 0]^T$ and $\boldsymbol{\delta}_v = [0, 1, 0]^T$ denote pixel offsets in the u- and v-directions, respectively. The process of mapping pixels across face seams is visualized in Fig. 5. When the offset point crosses the face seam, $\mathcal{W}$ efficiently remaps it to a valid pixel location on the adjacent face.

This seam-aware strategy ensures differentiability throughout all regions of the cubemap, including across face seams. Consequently, features whose projections lie near face seams can still be robustly and accurately tracked, enhancing the stability and reliability of photometric optimization on the cubemap.

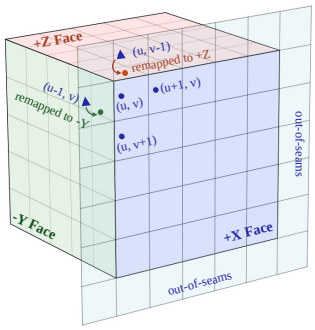


Fig. 5. Seam-aware remapping strategy for continuous gradient computation across face seams.

The final expression for $\mathbf{H}_{\text{pho}}$ is obtained by substituting (16), (14), and (13) into (12). In addition to the photometric residual z_{pho} in (10), the system state is estimated by jointly minimizing the photometric and geometric constraints as formulated in (2).

F. Update Features

After joint photometric and geometric optimization, semi-dense photometric features are refreshed to ensure subsequent computations. The update process comprises two steps: validation of existing features and discovery of new candidates. We project each historical feature onto the current cubemap. The following checks filter unreliable features:

- Mask check: Skip a feature if its location is already occupied by another retained feature to avoid redundancy.
- Range check: Discard features with invalid ranges (too far or too close).
- Occlusion check: Remove a feature if the range obtained by reprojection differs significantly from the cubemap range at that pixel; such a discrepancy usually indicates occlusion or scene dynamics.
- Photometric consistency: Reject features with a low IGM response or a large z_{pho}.

Features validated by all checks are updated with current IGM values and retained for the next frame. For uncovered regions, new features are selected using the method described in Section III-D.

IV. EXPERIMENTS

We comprehensively evaluate CUBE-LIO using public datasets with various LiDAR types. All experiments are conducted on an Intel i7-8700 workstation with 32GB RAM; GPU acceleration is not used.

A. ENWIDE Dataset Results

The ENWIDE dataset, collected using a handheld Ouster OS0-128 spinning LiDAR, is specifically designed to evaluate performance in geometrically degenerate environments, with each sequence containing severely challenging segments. We use the ENWIDE dataset to evaluate the performance of our method. As baselines, we compare against the state-of-the-art geometric-only method FAST-LIO2 [8]

TABLE I
ENWIDE DATASET: ABSOLUTE TRAJECTORY ERROR (RMSE, M)

Sequence	Length (m)	FAST-LIO2	RI-LIO	COIN-LIO	CUBE-LIO
TunnelS	252	×	×	**0.796**	0.884
TunnelD	180	×	×	0.485	**0.317**
IntersecS	279	0.400	×	0.461	**0.163**
IntersecD	388	×	×	1.757	**0.277**
RunwayS	334	×	×	1.184	**0.395**
RunwayD	357	×	×	3.491	**0.655**
FieldS	233	0.210	0.402	0.200	**0.187**
FieldD	288	6.973	16.572	0.993	**0.193**
KatzenseeS	243	0.472	×	0.502	**0.201**
KatzenseeD	177	0.861	×	0.678	**0.237**

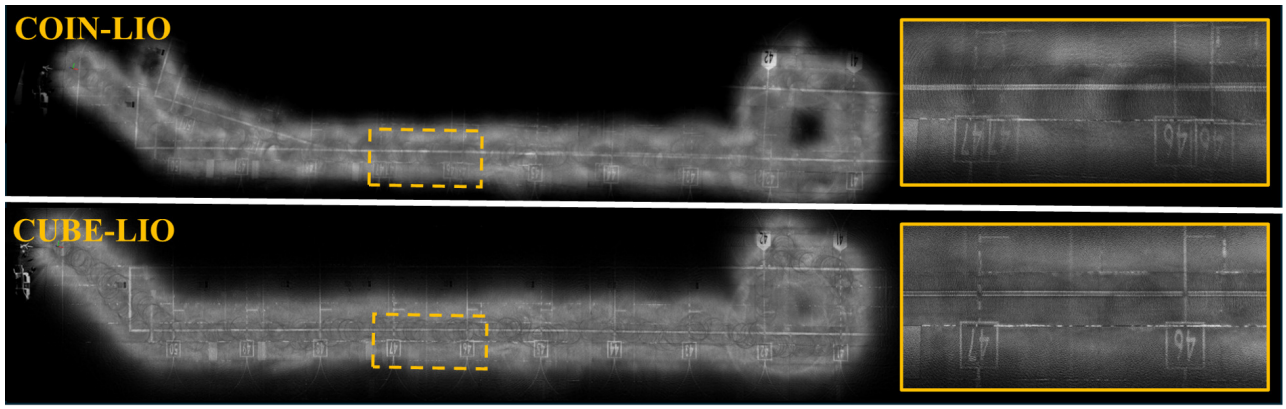


Fig. 6. RunwayD mapping (top COIN-LIO, bottom CUBE-LIO). Orange box: COIN-LIO ghosted ground digits from accumulated back-and-forth motion error; CUBE-LIO is sharp and artifact-free.

and the intensity-assisted methods RI-LIO [2] and COIN-LIO [3], using their provided default parameters. Performance is evaluated using Absolute Trajectory Error (ATE), and methods with an ATE exceeding 20 m are considered failures (indicated by "×").

Table I presents the ATE comparison. FAST-LIO2 fails in several scenarios. RI-LIO, although adopting photometric constraints, still diverges in most sequences. In contrast, COIN-LIO and CUBE-LIO achieve success in all scenarios by more effectively exploiting photometric information. Compared with COIN-LIO, CUBE-LIO achieves lower ATE in 9 out of 10 sequences.

Fig. 6 shows the mapping results generated by COIN-LIO and CUBE-LIO on the *RunwayD* sequence. In this sequence, aggressive LiDAR motion leads to substantial drift in the map produced by COIN-LIO. In contrast, CUBE-LIO avoids such drift and preserves clear and distinct ground digit patterns.

B. MARS-LVIG and FAST-LIVO2 Dataset Results

The MARS-LVIG dataset was collected using the DJI M300 RTK quadrotor equipped with a Livox Avia solid-state LiDAR and a camera. Ground-truth trajectories are provided by the RTK-GNSS system. For each representative scenario, we select one sequence for comparison. As the MARS-LVIG LiDAR is downward-facing, the point cloud covers the south pole in the body coordinate system.

RI-LIO and COIN-LIO rely on precomputed LUT, which are not compatible with the non-repetitive scanning pattern of Livox solid-state LiDARs. Therefore, we use FAST-LIO2 and FAST-LIVO2 [13] as baseline systems. FAST-LIVO2 is a state-of-the-art LIVO system that leverages visual information. For all FAST-LIVO2 experiments, we use the recommended parameters optimized for the MARS-LVIG dataset.

TABLE II

MARS-LVIG DATASET: ABSOLUTE TRAJECTORY ERROR (RMSE, M)

Sequence	Length (m)	FAST-LIO2	FAST-LIVO2	CUBE-LIO
HKairport03	1,828	0.785	2.163	**0.776**
HKairport_GNSS03	1,985	0.860	0.690	**0.322**
HKisland03	1,900	0.553	**0.535**	0.636
HKisland_GNSS03	1,955	0.624	0.455	**0.344**
AMtown03	3,099	2.314	2.214	**2.095**
AMvalley03	2,729	17.381	4.207	**2.355**
Featureless_GNSS02	2,810	12.075	2.533	**0.398**

Table II shows ATE results for MARS-LVIG sequences. *AMvalley03* and *Featureless_GNSS02* have strong geometric degeneracy. FAST-LIO2 drifts significantly in these challenging cases. FAST-LIVO2 improves accuracy by using visual information. CUBE-LIO achieves the most accurate trajectories in severe degeneracy. For other sequences, geometric features are more abundant. Performance gaps between methods become smaller in these cases. CUBE-LIO still delivers competitive or superior accuracy overall. This demonstrates robustness in both degenerate and structured environments.

To further demonstrate the mapping accuracy of CUBE-LIO under degenerate conditions, we use the FAST-LIVO2 dataset, collected using a handheld Livox Avia LiDAR and a camera. The *HIT_Graffiti_Wall* sequence is particularly challenging because the LiDAR observes a large planar wall, resulting in weak geometric constraints.

Since the FAST-LIVO2 dataset does not provide ground-truth trajectories, we present a qualitative comparison of mapping results in Fig. 7 for the *LiDAR-inertial_Graffiti_Wall* sequence, including FAST-LIVO2, FAST-LIO2, and CUBE-LIO.

Due to insufficient geometric information, FAST-LIO2 exhibits noticeable drift. FAST-LIVO2 leverages visual cues to generate high-precision colored maps. Notably, CUBE-LIO, using only LiDAR intensity, produces maps with minimal drift. Although CUBE-LIO does not provide colored maps, its patterns are delineated, and in certain regions, texture details surpass those of FAST-LIVO2. For example, when zooming in on regions A3 and C3, FAST-LIVO2 shows slight ghosting, whereas CUBE-LIO yields sharper and clearer textures. We attribute the advantage of CUBE-LIO over FAST-LIVO2 to the inherent temporal and spatial alignment of LiDAR intensity compared to the camera.

Fig. 7. Comparison of mapping results on the HIT Graffiti Wall sequence: *Top:* FAST-LIVO2, *Middle:* FAST-LIO2, *Bottom:* CUBE-LIO. CUBE-LIO provides sharper, less ghosted structures.

C. Ablation Study

To evaluate the contributions of the cubemap projection and IGM-constancy photometric optimization, we conduct ablation experiments on two challenging scenarios: *RunwayD* from ENWIDE and *Featureless_GNSS02* from MARS-LVIG. The former involves aggressive handheld motion with large viewpoint and range variation but no polar coverage; the latter is a quadrotor flight observing downward regions with stable attitude. Results are summarized in Table III.

The (cube, IGM) configuration represents the full CUBE-LIO system. The (equi, IGM) variant replaces the projection method with equirectangular projection, while keeping photometric optimization unchanged.

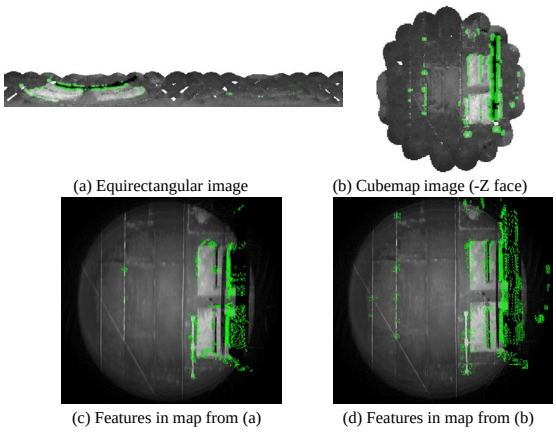
On *RunwayD*, the equirectangular variant performs adequately due to the limited vertical field of view of the Ouster OS0 sensor, which avoids the polar regions. However, in *Featureless_GNSS02*, the equirectangular variant suffers a severe drop in accuracy. Figure 8 details the reason for this degradation. In (a) and (b), green points denote selected semi-dense features on the equirectangular and cubemap intensity representations, respectively; (c) and (d) show the same features projected into the global map. Equirectangular stretching near the pole causes dense aggregation and hinders effective selection in those regions, whereas the cubemap yields a more uniform sampling across the field of view, enabling stable tracking. This validates the substantial advantage of cubemap projection when polar regions are present.

The "Time" column in Table III reports the average image generation/photometric optimization costs on the *RunwayD* sequence. Compared to the equirectangular variant, cubemap-based image projection and photometric optimization are faster by 38% and 43%, respectively. Additionally, the per-frame processing times of FAST-LIO2, COIN-LIO, and CUBE-LIO are 24.2 ms, 32.1 ms, and 29.5 ms, respectively. CUBE-LIO is only modestly slower than the geometric-only baseline while maintaining real-time performance.

To assess the efficacy of IGM-constancy optimization, we replace it with an intensity-constancy optimization (cube,

TABLE III
ABLATION STUDY: RMSE AND EXECUTION TIME

Proj.	Features	RunwayD (m)	Featureless GNSS02 (m)	Time (ms)
equi.	IGM	0.760	19.71	8.3/5.1
cube	intensity	1.734	0.720	–/–
cube	IGM	**0.655**	**0.398**	**5.1/2.9**



(a) Equirectangular image (b) Cubemap image (-Z face)

(c) Features in map from (a) (d) Features in map from (b)

Fig. 8. Comparison of feature distribution near the pole.

intensity), while keeping semi-dense feature selection still driven by IGM (ensuring identical feature locations). Across both *RunwayD* and *Featureless_GNSS02*, IGM-constancy outperforms intensity-constancy; the gain is especially pronounced on *RunwayD*, where rapid handheld motion induces substantial range and incidence-angle variation. These results confirm that IGM-constancy photometric optimization yields more stable residuals under dynamic observation conditions.

V. CONCLUSION

This study presents CUBE-LIO, an intensity-assisted LiDAR-inertial odometry system using a low-distortion, efficient cubemap projection for dense LiDAR image generation. The approach is sensor-agnostic, supporting both spinning and solid-state LiDARs. Building on this projection, we introduce an IGM-based feature selection and direct IGM-constancy optimization strategy, improving resilience to intensity noise and measurement variation. Extensive public-dataset experiments show consistent accuracy and robustness in both structured and geometrically degenerate environments. As high-density hemispherical LiDARs (e.g., Ouster Dome, RoboSense Airy, Hesai JT128) become more prevalent, we expect the advantages of cubemap projection to increase and will further refine the framework to advance robust, accurate LiDAR-inertial navigation.

REFERENCES

[1] W. Xu and F. Zhang, "Fast-lio: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3317–3324, 2021.

[2] Y. Zhang, Y. Tian, W. Wang, G. Yang, Z. Li, F. Jing, and M. Tan, "Rilio: Reflectivity image assisted tightly-coupled lidar-inertial odometry," *IEEE Robotics and Automation Letters*, vol. 8, no. 3, pp. 1802–1809, 2023.

[3] P. Pfreundschuh, H. Oleynikova, C. Cadena, R. Siegwart, and O. Andersson, "Coin-lio: Complementary intensity-augmented lidar inertial odometry," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 1730–1737.

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

[5] T. Shan and B. Englot, "Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 4758–4765.

[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 *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 5135–5142.

[7] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, "isam2: Incremental smoothing and mapping with fluid relinearization and incremental variable reordering," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 3281–3288.

[8] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "Fast-lio2: Fast direct lidar-inertial odometry," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2053–2073, 2022.

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

[10] Y. Liu, K. Yamamoto, A. Matsui, S. Takahashi, and T. Abe, "F3-slam: Flexible and fast features slam," *Transactions of the Society of Instrument and Control Engineers*, vol. 60, no. 12, pp. 631–637, 2024.

[11] 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 *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 4003–4009.

[12] J. Lin and F. Zhang, "R3live: A robust, real-time, rgb-colored, lidar-inertial-visual tightly-coupled state estimation and mapping package," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 10 672–10 678.

[13] C. Zheng, W. Xu, Z. Zou, T. Hua, C. Yuan, D. He, B. Zhou, Z. Liu, J. Lin, F. Zhu, Y. Ren, R. Wang, F. Meng, and F. Zhang, "Fast-livo2: Fast, direct lidar–inertial–visual odometry," *IEEE Transactions on Robotics*, vol. 41, pp. 326–346, 2025.

[14] H. Wang, C. Wang, and L. Xie, "Intensity-slam: Intensity assisted localization and mapping for large scale environment," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1715–1721, 2021.

[15] S. Li, B. Tian, X. Zhu, J. Gui, W. Yao, and G. Li, "Inten-loam: Intensity and temporal enhanced lidar odometry and mapping," *Remote Sensing*, vol. 15, no. 1, 2023.

[16] Y. S. Park, H. Jang, and A. Kim, "I-loam: Intensity enhanced lidar odometry and mapping," in *2020 17th International Conference on Ubiquitous Robots (UR)*, 2020, pp. 455–458.

[17] H. Li, B. Tian, H. Shen, and J. Lu, "An intensity-augmented lidar-inertial slam for solid-state lidars in degenerated environments," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–10, 2022.

[18] Z. Chen, H. Zhu, B. Yu, C. Jiang, C. Hua, X. Fu, and X. Kuang, "Ige-lio: Intensity gradient enhanced tightly coupled lidar-inertial odometry," *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1–11, 2024.

[19] T. Guadagnino, X. Chen, M. Sodano, J. Behley, G. Grisetti, and C. Stachniss, "Fast sparse lidar odometry using self-supervised feature selection on intensity images," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7597–7604, 2022.

[20] W. Du and G. Beltrame, "Real-time simultaneous localization and mapping with lidar intensity," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 4164–4170.

[21] A. Khedekar *et al.*, "PG-LIO: Photometric-geometric fusion for robust LiDAR-inertial odometry," *arXiv preprint arXiv:2506.18583*, 2025.

[22] N. Greene, "Environment mapping and other applications of world projections," *IEEE Computer Graphics and Applications*, vol. 6, no. 11, pp. 21–29, 1986.

[23] K. M. Gorski, E. Hivon, A. J. Banday, B. D. Wandelt, F. K. Hansen, M. Reinecke, and M. Bartelmann, "Healpix: A framework for high-resolution discretization and fast analysis of data distributed on the sphere," *The Astrophysical Journal*, vol. 622, no. 2, p. 759, 2005.

[24] D. Shepard, "A two-dimensional interpolation function for irregularly-spaced data," in *Proceedings of the 1968 23rd ACM national conference*, 1968, pp. 517–524.

[25] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, 1986.

[26] J. Engel, J. Stückler, and D. Cremers, "Large-scale direct slam with stereo cameras," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 1935–1942.