IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Gaussian or Plane? Both: Semantic-Driven Voxel Representation for LiDAR–Inertial Odometry

Haiyang Wu, *Graduate Student Member, IEEE*, George Vosselman, and Ville Lehtola, *Member, IEEE*

***Abstract*—Accurate LiDAR-inertial odometry (LIO) highly depends on the geometric fidelity of the underlying environment representation. We explore the new and interesting research direction of integrating semantic segmentation models into metric odometry algorithms to enrich their representational capacity. Specifically, this letter proposes a semantic-driven hybrid voxel representation in which an off-the-shelf 3D segmentation network assigns every voxel to either a planar or nonplanar class, using planar and Gaussian representations, respectively. Consequently, a hybrid scan matching strategy is presented using class-specific residual models that are tailored to the distinct error statistics of each surface category. The scan matcher is embedded within an Iterated Extended Kalman Filter (IEKF) for odometry and mapping. We evaluate our method on diverse platforms and environments, and show improved localization accuracy across various indoor and outdoor scenarios, while maintaining real-time performance.**

***Index Terms*—SLAM, mapping, LiDAR–inertial odometry, semantic-driven voxel representation.**

I. Introduction

LIDAR-INERTIAL Odometry (LIO) systems have been widely applied in fields such as autonomous driving [1] and acquiring data for digital twins [2]. The core of such systems lies in (i) representing the environment map with high fidelity, including generalization to various scenes, (ii) using the representation in scan matching between incoming LiDAR scans and the existing map [3], and (iii) running the method in real-time.

An urban environment contains, i.a., built components and vegetation, sometimes in irregular patterns. These could be described with parametric features [4], [5]. However, we do not wish to make a-priori assumptions about the way the environment should be represented, as doing so would limit our method to specific scenarios rather than diverse environments. Hence, we exclude parametric feature-based models and look into dense representations. There are three main approaches: point-based, surface element (surfel)-based, and voxel-based techniques. Point clouds preserve raw LiDAR returns and rich geometry, which benefits odometry [6], [7], [8] but incur high storage cost and provide only geometric position information. Surfel-based maps model locally planar surface elements, offering a sparse yet structured representation for dense reconstruction and precise pose estimation [9]; adding semantics to each surfel further aids scene understanding [10] at extra computational expense. Voxel-based maps partition space into regular or adaptive grids and fit a plane in each cell, enabling efficient encoding, indexing, and updates [11], [12], [13]. However, on curved surfaces or in sparse scans, plane fitting requires very fine voxels that may lack points for reliable estimation.

In scan matching for relative pose estimation, most methods rely on a single environment representation [3], limiting adaptability across scene structures, although exceptions exist [14], which still operate without switching strategies based on scene characteristics. Mainstream existing approaches fall into two categories: (i) geometric residuals that minimize point-to-plane (or similar) distances through explicit correspondences [15]; and (ii) probabilistic residuals that maximize data likelihood under Gaussian models [16]. Using geometric residuals allows for high accuracy when scene primitives match their assumptions but lose robustness under noise or structural mismatch, whereas probabilistic methods are noise-tolerant yet exploit geometry less and impose weaker constraints in structured areas. As most LIO systems use only one of these residual types [11], [17], they struggle to maintain high fidelity and generalize to environments with diverse structures. Combining deterministic and probabilistic models offers a promising alternative, but poses challenges in ensuring scale-consistent residual fusion.

To obtain high map fidelity and wide generalization capacity, we propose a semantic-driven adaptive mapping strategy: an online semantic segmentation module assigns super-labels to voxels, which are then modeled as either planes or Gaussians to form a hybrid representation. A matching hybrid cost function fuses geometric and probabilistic residuals under a unified scale. Experiments across diverse platforms and scenes show superior accuracy, and the implementation will be released after publication.

The main contributions of this letter are as follows:

- *Semantic-driven hybrid voxel representation* is proposed, where each voxel is modeled using either a plane or a Gaussian distribution, based on its semantic label. To this end, the semantic segmentation process is run in parallel with the odometry thread, and super-labels are dynamically updated using a sliding window. Each voxel is then represented accordingly, forming a unified hybrid voxel map.

Received 16 July 2025; accepted 27 October 2025. Date of publication 13 November 2025; date of current version 19 November 2025. This article was recommended for publication by Associate Editor L. Heng and Editor S. Behnke upon evaluation of the reviewers' comments. This work was supported in part by EU under Horizon Europe RIA under Grant 101136006 (XTREME) and in part by China Scholarship Council CSC under Grant 202407300017. *(Corresponding author: Haiyang Wu.)*

The authors are with the Department of Earth Observation Science, Faculty of Geo-Information Science and Earth Observation (ITC), University of Twente, 7522 NB Enschede, The Netherlands (e-mail: haiyang.wu@utwente.nl; george.vosselman@utwente.nl; v.v.lehtola@utwente.nl).

Our code is available at https://github.com/haiyang2022/Hybrid-VoxelMap.

Digital Object Identifier 10.1109/LRA.2025.3632730

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

- *Scale-consistent scan matching* is developed for hybrid voxels. Specifically, we construct a cost function that combines geometric and probabilistic residuals. Closed-form covariances are derived for both residuals under a unified scale, enabling the incorporation of geometric constraints in structured regions and probabilistic robustness in unstructured ones.
- *Real-time performance* in cross-platform and cross-scenario comparative evaluation is demonstrated across UGV, UAV, and handheld platforms in indoor, outdoor, and challenging environments. Compared with state-of-the-art (SOTA) methods, the proposed framework achieves significant improvements in trajectory accuracy, demonstrating its effectiveness.

II. Related Work

A. Environment Map Representation

Point-cloud maps store raw or downsampled LiDAR scans for alignment. LOAM [6] extracts edge and planar features but is sensitive to sparsity and noise. FAST-LIO2 [8] improves robustness by matching downsampled scans to submaps via an incremental KD-tree, whereas DLIO [17] achieves similar goals through continuous-time optimization; nevertheless, point-cloud maps remain unstructured and memory-intensive. Surfel-based maps represent the environment with sparse yet structured surface elements: SuMa [9] register scans by associating points with surfels, and SuMa++ [10] augments each surfel with a semantic label and confidence score for dense semantic mapping alongside geometric estimation. Although surfel maps are compact and preserve surface continuity, they often incur high computational costs.

Voxel-based maps represent the environment by dividing space into 3D voxels, which can be used to organize, aggregate, or model local geometry. VoxelMap [11] models each voxel with a fitted plane and performs point-to-plane matching while accounting for LiDAR noise and plane fitting uncertainty. VoxelMap++ [12] improves compactness by introducing a 3-DoF plane representation and coplanar voxel merging, while C3P-VoxelMap [13] further reduces memory usage via geometry-aware merging. These methods demonstrate efficient voxel-based mapping, but their reliance on planar assumptions limits adaptability in complex environments. Current adaptive techniques focus on the scanning-originated data variations: Adaptive-LIO [18] addresses the range-dependent point density variation by using larger voxels farther and smaller voxels closer to the LIDAR and AS-LIO [19] adapts the sliding window against aggressive field-of-view variation caused e.g. by fast rotations. Instead, we focus on adapting LIO against environmental variations.

B. Residual Modeling in Scan Matching

Once the environment representation has been chosen, it is essential to formulate the related scan matching equations between incoming LiDAR scans and the map, along with defining a suitable cost function for pose estimation.

The first category of methods minimizes geometric residuals by establishing explicit correspondences. Many approaches [7], [8], [20], [21], including voxel-map-based frameworks [11], [12], [13], adopt point-to-plane Iterative Closest Point (ICP) [15] or its variants to formulate such residuals. Additionally, residuals may be reduced computationally by increasing the IEKF update rate to tighten the deep coupling. sr-LIO [22] reconstructs LiDAR scans into smaller scan segments with a higher scan rate, and then performs a more detailed motion undistortion on these smaller segments, thereby reducing the point-to-plane residuals. These methods are effective in structured, planar regions where geometric primitives can be reliably extracted.

In contrast, the second category employs probabilistic models that avoid explicit data associations. For example, LiTAMIN [23] adopts a probabilistic perspective by modeling point-to-distribution residuals, following the NDT framework [16], and thus converts the map representation from a purely geometric form to a probabilistic distribution. LiTAMIN2 [24] further introduces KL divergence for distribution-to-distribution matching, and LIO-GVM [25] applies a similar strategy within a voxelized framework. iG-LIO [26] tightly couples Generalized ICP [27] constraints with IMU in a unified framework. These approaches tend to be more robust in unstructured or noisy environments.

However, relying on a single representation type introduces limitations, as it requires prior knowledge of the environment's properties and assumes structural homogeneity, which rarely holds in real-world scenarios. To address these limitations of relying on a single representation or residual type, we propose a hybrid voxel modeling scheme that differentiates between planar and nonplanar regions and adapts the residual formulation accordingly. This design aims to integrate geometric precision with probabilistic robustness within a unified optimization framework.

III. Methodology

A. System Overview

The overall system workflow is illustrated in Fig. 1. First, the IMU measurements are used to de-skew the raw LiDAR scan. The combined LiDAR–IMU data is then provided as an observation input to the state estimator to compute the pose of the current frame, $\mathbf{x}_t$. This estimated pose is used to transform the de-skewed point cloud into the global coordinate frame.

The transformed point cloud is processed by a semantic segmentation network to obtain point-wise labels. Based on a predefined voxel size v_{size}, the semantic point cloud is voxelized, and each voxel is classified as planar or nonplanar.

Newly generated planar and nonplanar voxels are fused with co-located voxels in the historical voxel map, updating their statistics and semantic categories to better reflect recent observations. All voxels are stored in a hash-based voxel grid (conceptually equivalent to the leaf level of an octree), enabling $O(1)$ approximate retrieval and forming a unified Hybrid VoxelMap. This incrementally updated voxel map provides hybrid measurement functions for the next iteration and drives further state updates.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

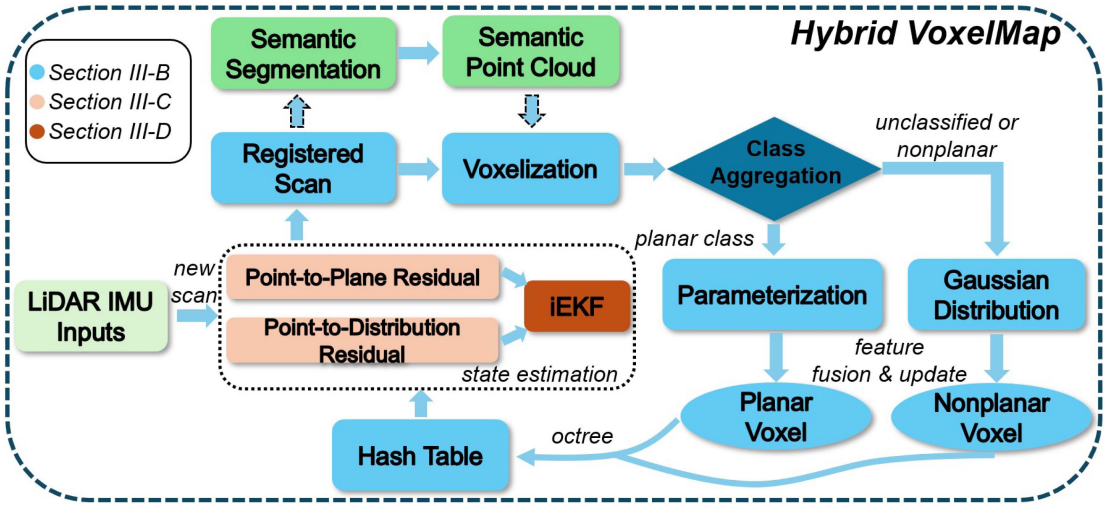

Fig. 1. System overview of Hybrid VoxelMap.

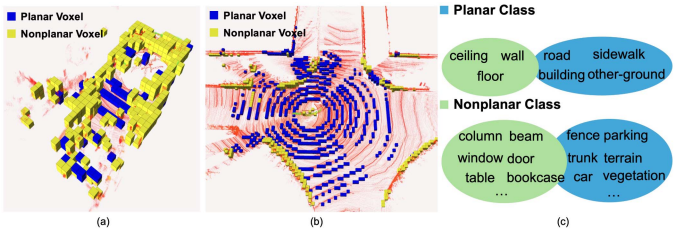

Fig. 2. Illustration of semantic-based voxel classification on a single LiDAR scan. (a) Indoor scene overlaid with historical point cloud (red). (b) Outdoor scene overlaid with historical point cloud (red). (c) Redefined semantic mapping table used for voxel classification.

B. Hybrid Voxel Representation

1) Semantic-Driven Adaptive Voxel Classification: We begin by introducing the hybrid voxel representation strategy used in our framework. For the k-th point-cloud frame $\mathcal{P}_k = \{\mathbf{p}_i\}_{i=1}^{N}$, a semantic segmentation network is invoked in an independent thread to assign point-wise labels l_i. As shown in Fig. 2(c), the original semantic labels are merged into two super-labels—$\mathcal{L}_p$ = planar and $\mathcal{L}_{np}$ = nonplanar—based on a predefined mapping table.

The labeled point cloud is then voxelized using a fixed voxel size S_v. For each voxel $\mathcal{V}$, if the number of contained points satisfies $|\mathcal{V}| > N_{\text{sem}}$, a majority vote is applied to determine its semantic label $L_{\mathcal{V}}$.

- *Planar representation (classified as planar voxel):* If $L_{\mathcal{V}} = \mathcal{L}_p$, a plane is fitted to the points in the voxel. When the smallest eigenvalue $\lambda_{\min}$ falls below a threshold τ_p, the planar representation provides rigid geometric constraints for state estimation.
- *Gaussian representation (classified as nonplanar voxel):* If $L_{\mathcal{V}} = \mathcal{L}_{np}$, or the plane-fitting fails, the voxel is treated as nonplanar. A Gaussian distribution is used to model the local point set and contributes probabilistic constraints.

This semantic-driven classification is refreshed at a low frequency by performing segmentation on one frame every W frames (default $W = 50$). For voxels with the same spatial index, new semantic labels overwrite the old ones, allowing the system to adapt dynamically to local environmental changes.

A pretrained RandLA-Net [28] is used in our implementation, but in principle any point-wise semantic segmentation network can be utilized, as the framework only requires point-level labels. Fig. 2(a) and (b) illustrates the voxel classification results produced by our semantic mapping on representative indoor and outdoor scenes.

2) Gaussian Representation: Each nonplanar voxel V_{np} is represented by a 3D Gaussian distribution that models the spatial scatter of its contained points. Given a point set $P = \{\mathbf{p}_i\}_{i=1}^{n}$, where $\mathbf{p}_i \in \mathbb{R}^3$, the mean $\boldsymbol{\mu} \in \mathbb{R}^3$ and covariance matrix $\Sigma \in \mathbb{R}^{3\times3}$ are computed as:

$$\boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{p}_i, \qquad \Sigma = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{p}_i - \boldsymbol{\mu})(\mathbf{p}_i - \boldsymbol{\mu})^{\top} \tag{1}$$

Thus, V_{np} is parameterized by the tuple $(\boldsymbol{\mu}, \Sigma)$.

3) Planar Representation: For a planar voxel V_{p}, the contained point set is assumed to satisfy the plane equation $ax + by + cz + d = 0$. The voxel mean and covariance are first computed using (1), and the covariance matrix is eigen-decomposed. The eigenvector corresponding to the smallest eigenvalue, $\mathbf{n} = (a, b, c)^{\top}$, is selected as the plane normal, and the intercept is computed as:

$$d = -\mathbf{n}^{\top}\boldsymbol{\mu} \tag{2}$$

Thus, V_{p} is parameterized by the tuple $(\mathbf{n}, d)$.

4) Map Update and Indexing: For each transformed point, the voxel index (i, j, k) is computed by discretization with size S_v. If the corresponding voxel already exists in the global map, its statistics are incrementally updated by

$$\begin{cases} n \leftarrow n + 1, \\ \mathbf{s} \leftarrow \mathbf{s} + \mathbf{p}, \\ \mathbf{S} \leftarrow \mathbf{S} + \mathbf{p}\mathbf{p}^{\top} \end{cases} \quad \Longrightarrow \quad \boldsymbol{\mu} = \mathbf{s}/n,\ \Sigma = \mathbf{S}/n - \boldsymbol{\mu}\boldsymbol{\mu}^{\top}, \tag{3}$$

where n is the sample count; $\mathbf{s} = \sum_{i=1}^{n}\mathbf{p}_i$ and $\mathbf{S} = \sum_{i=1}^{n}\mathbf{p}_i\mathbf{p}_i^{\top}$ are the first- and second-order sums; $\boldsymbol{\mu} \in \mathbb{R}^3$ and $\Sigma \in \mathbb{R}^{3\times3}$ are

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

the voxel mean and covariance (consistent with (1)); and $\mathbf{p} \in \mathbb{R}^3$ is the incoming point. Only the left updates are executed per new point. Otherwise a new voxel is created and inserted into the map. Gaussian voxels are parameterized by $(\boldsymbol{\mu}, \Sigma)$. Planar voxels additionally require eigen-decomposition of Σ to update the normal $\mathbf{n}$ and intercept $d = -\mathbf{n}^\top \boldsymbol{\mu}$. New or updated planar voxels are checked against their six neighbors for coplanarity; if satisfied, they are merged into clusters via a union–find set [12].

All voxels are stored in a hash grid, where indices (i, j, k) are mapped to hash keys. This ensures $O(1)$ average-time access and is functionally equivalent to managing the leaf layer of an octree, enabling efficient incremental map updates.

C. Hybrid Measurement Model

1) Point-to-Plane Match: We adopt the 3-DoF plane parameterization from VoxelMap++ [12] and define the measurement function for a planar voxel as $h(x)$, with residual:

$$r = -h(x) = -\frac{p_w^\top \omega + d}{\|\omega\|}, \tag{4}$$

where $x = (R,\ t)$ is the system state, $p_b \in \mathbb{R}^3$ is a point in the body frame, and $p_w = R\,p_b + t$ denotes its position in the world frame. The plane normal is defined as $\omega = (a,\ b,\ 1)$, and its normalization factor is $\|\omega\| = \sqrt{a^2 + b^2 + 1}$. Here we fix the third component to 1 (rather than using (a, b, c)) to remove the overall scale ambiguity and follow the dominant-axis parameterization in [12], switching the fixed component when a different axis is most aligned.

The residual Jacobian with respect to the state is given by:

$$H = \begin{bmatrix} ([p_b]_\times R^\top n)^\top & -n^\top \end{bmatrix}, \tag{5}$$

where $n = \omega/\|\omega\|$ is the unit normal of the fitted plane, and $[p_b]_\times \in \mathbb{R}^{3\times3}$ is the skew-symmetric matrix of the body-frame point p_b.

Residual variance modeling: We explicitly model residual uncertainty by combining two independent sources: the LiDAR measurement noise and the geometric uncertainty from local plane fitting.

The total residual variance is formulated as:

$$\sigma^2_{PL\_Voxel} = J_p\,\Sigma_L\,J_p^\top + J_{pl}\,\Sigma_{pl}\,J_{pl}^\top, \tag{6}$$

with:

$$J_p = \begin{bmatrix} \frac{\partial r}{\partial x}, & \frac{\partial r}{\partial y}, & \frac{\partial r}{\partial z} \end{bmatrix},$$
$$J_{pl} = \begin{bmatrix} \frac{\partial r}{\partial a}, & \frac{\partial r}{\partial b}, & \frac{\partial r}{\partial d} \end{bmatrix}. \tag{7}$$

where $\Sigma_L \in \mathbb{R}^{3\times3}$ is the LiDAR Cartesian covariance, and $\Sigma_{pl} \in \mathbb{R}^{3\times3}$ is the (a, b, d) covariance from local plane fitting—both computed as in [12].

2) Point-to-Distribution Match: A nonplanar voxel is modeled as a Gaussian distribution with mean μ and covariance Σ. To maintain consistency with point-to-plane residuals, we define the measurement function $h(x)$ as the scaled Mahalanobis distance, and the residual as:

$$r = -h(x) = -\sigma_{avg}\sqrt{(p_w - \mu)^\top \Sigma^{-1}(p_w - \mu)}, \tag{8}$$

where $p_w \in \mathbb{R}^3$ is the point transformed into the world frame, and $\sigma_{avg} = \sqrt{\text{trace}(\Sigma)/3}$ is a normalization factor that aligns the residual scale with that of point-to-plane distances. We adopt the trace average for efficiency, and note that highly anisotropic distributions, e.g. planes, are treated in the point-to-plane matching.

The corresponding residual Jacobian with respect to the state perturbation $\delta x = [\delta\theta^\top,\ \delta t^\top]^\top$ (with $\delta\theta \in \mathbb{R}^3$ the small-angle rotation error and $\delta t \in \mathbb{R}^3$ the translation error) is:

$$H = -\begin{bmatrix} \left(\frac{\partial h}{\partial \delta\theta}\right)^\top & \left(\frac{\partial h}{\partial \delta t}\right)^\top \end{bmatrix}, \tag{9}$$

with:

$$\frac{\partial h}{\partial \delta\theta} \approx \sigma_{avg}\,\frac{[p_b]_\times R^\top \Sigma^{-1}(p_w - \mu)}{\sqrt{E}},$$
$$\frac{\partial h}{\partial \delta t} = \sigma_{avg}\,\frac{\Sigma^{-1}(p_w - \mu)}{\sqrt{E}}, \tag{10}$$

where $E = (p_w - \mu)^\top \Sigma^{-1}(p_w - \mu)$, and $[p_b]_\times$ is defined above.

Residual variance modeling: To model uncertainty in the residual $h(x)$ of a nonplanar voxel, we account for both point measurement noise and voxel statistics.

The point covariance $\Sigma_{p_w} \in \mathbb{R}^{3\times3}$ is propagated from the local coordinate frame to the global frame using first-order error propagation, following the method in [11].

The voxel mean covariance Σ_μ is computed from the distribution of points within the voxel:

$$\Sigma_\mu = \frac{1}{n}\Sigma, \tag{11}$$

where n is the number of points in the voxel.

The Jacobians for point and mean contributions are:

$$J_{p_w} \approx \sigma_{avg}\frac{\Sigma^{-1}(p_w - \mu)}{\sqrt{(p_w - \mu)^\top\,\Sigma^{-1}\,(p_w - \mu)}},$$
$$J_\mu \approx -\,J_{p_w}. \tag{12}$$

The final residual variance becomes:

$$\sigma^2_{GD\_Voxel} = J_{p_w}\Sigma_{p_w}J_{p_w}^\top + J_\mu\Sigma_\mu J_\mu^\top, \tag{13}$$

where σ_{avg} is defined above and serves to align the residual scale with that of point-to-plane residuals.

D. Unified IESEKF Update

Given m_p point–plane and m_{np} point–distribution matches, we concatenate their residuals and Jacobians into $r \in \mathbb{R}^m$ and $H \in \mathbb{R}^{m\times6}$, with $m = m_p + m_{np}$. For each residual r_i we take the propagated scalar variance

$$\sigma_i^2 = \begin{cases} \sigma^2_{PL\_Voxel,i} & \text{(planar)}, \\ \sigma^2_{GD\_Voxel,i} & \text{(Gaussian)}, \end{cases} \tag{14}$$

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

TABLE I
COMPARISON OF ATE (RMSE, M) AND AVERAGE PER-FRAME COMPUTATION TIME (MS) ACROSS SEQUENCES AND METHODS (VALUES SHOWN AS ATE (TIME))

Seq.	Dist.(m)	FAST-LIO2	DLIO	sr-LIO	iG-LIO	PV-LIO	VoxelMap++	C3P-VoxelMap	Ours (GD)	Ours (full)
room01	27	0.407 (4)	0.166 (17)	0.801 (7)	0.403 (5)	0.397 (10)	0.163 (6)	0.159 (8)	**0.153** (10)	0.161 (10)
room02	45	0.323 (4)	0.205 (17)	0.329 (4)	0.306 (5)	0.317 (10)	0.157 (6)	0.135 (8)	0.134 (10)	**0.134** (10)
room03	71	0.430 (4)	0.189 (20)	0.440 (4)	0.434 (5)	0.427 (10)	**0.166** (7)	0.176 (8)	0.172 (11)	0.167 (11)
nya01	160	0.248 (8)	0.227 (20)	0.255 (8)	0.260 (6)	0.352 (29)	0.526 (12)	**0.209** (19)	0.231 (57)	0.239 (60)
nya02	249	0.247 (9)	0.237 (20)	0.239 (8)	0.238 (5)	0.255 (33)	1.383 (12)	0.212 (20)	0.209 (59)	**0.208** (60)
nya03	315	0.255 (8)	0.245 (20)	0.209 (9)	0.220 (5)	0.260 (33)	/	0.220 (20)	0.202 (60)	**0.200** (60)
Indoor Avg.	–	0.318	0.212	0.379	0.310	0.335	0.479	0.185	**0.184**	0.185
eee01	237	0.238 (11)	0.221 (20)	0.244 (11)	0.249 (5)	2.264 (59)	0.216 (24)	**0.201** (42)	0.208 (96)	0.207 (96)
eee02	171	0.251 (10)	0.218 (20)	0.212 (9)	0.219 (5)	1.001 (61)	2.926 (21)	0.215 (36)	0.183 (97)	**0.180** (95)
eee03	127	0.253 (10)	0.248 (20)	0.265 (9)	0.269 (5)	0.280 (58)	0.225 (20)	**0.214** (36)	0.250 (73)	0.239 (74)
gate01	139	0.175 (21)	0.134 (26)	0.177 (12)	0.139 (10)	0.139 (84)	0.121 (32)	0.150 (63)	0.118 (84)	**0.118** (84)
gate02	336	0.326 (18)	0.305 (26)	0.344 (13)	0.320 (9)	0.320 (77)	1.975 (33)	0.317 (52)	0.297 (79)	**0.283** (80)
gate03	296	0.213 (19)	0.115 (27)	0.242 (12)	0.195 (9)	0.207 (79)	0.342 (30)	0.114 (51)	0.113 (81)	**0.107** (82)
street05	420	0.368 (15)	1.018 (25)	0.271 (13)	**0.221** (8)	0.494 (65)	/	0.469 (35)	0.404 (72)	0.299 (73)
street06	479	0.392 (16)	0.377 (27)	**0.309** (13)	0.318 (9)	0.447 (68)	/	0.434 (37)	0.365 (84)	0.360 (78)
Outdoor Avg.	–	0.277	0.330	0.258	0.241	0.644	0.968	0.264	0.242	**0.224**

Note: / indicates failure due to severe drift or loss of tracking.

obtained from (6) or (13). To mitigate the influence of outliers, we incorporate robust weighting via the Huber kernel [29], yielding a weight $\alpha_i \in (0, 1]$ for each residual:

$$\begin{cases} w_i = \frac{\alpha_i}{\sigma_i^2}, \\ R = \mathrm{diag}\big(\sigma_1^2/\alpha_1, \ldots, \sigma_m^2/\alpha_m\big), \\ R^{-1} = \mathrm{diag}(w_1, \ldots, w_m). \end{cases} \tag{15}$$

With this diagonal R^{-1} the Iterated Error-State Extended Kalman Filter (IESEKF) update on the 6-DoF pose (R, t) is written explicitly as

$$\begin{cases} K = \big(H^\top R^{-1} H + P^{-1}\big)^{-1} H^\top R^{-1}, \\ x^+ = x^- + Kr, \\ P^+ = (I - KH)P. \end{cases} \tag{16}$$

Here P is the state covariance, R^{-1} encodes both propagated uncertainty and Huber robustness, I is the 6×6 identity matrix, and K yields the statistically optimal fusion of heterogeneous geometric and probabilistic constraints. Note that after robust weighting, R should be regarded as an effective observation covariance, reflecting both sensor noise statistics and the down-weighting effect of the robust kernel. Convergence is monitored via a sliding-window threshold on $\|x^+ - x^-\|$.

IV. EXPERIMENTS

In this section, we validate the proposed method through comprehensive experiments. Our algorithm is implemented based on VoxelMap++ [12], and RandLA-Net [28] is adopted as the semantic segmentation module. Specifically, we use the PyTorch implementation from [1] and employ the released pretrained models on S3DIS [30] and SemanticKITTI [31] for indoor and outdoor segmentation, respectively.

We evaluate the method across several representative scenarios, including two public datasets with structured indoor and outdoor environments (M2DGR [32] and NTU VIRAL [33]) and challenging unstructured campus scenes collected using our custom handheld device. Dataset details are summarized in Table II, and data acquisition platforms are shown in Fig. 3.

[1] https://github.com/liuxuexun/RandLA-Net-Pytorch-New

TABLE II
SUMMARY OF DATASETS USED IN OUR EXPERIMENTS

Dataset	Sequences	LiDAR	Platform
M2DGR	room 01–03, gate 01–03, street 05–06	Velodyne VLP-32	UGV
NTU VIRAL	nya 01–03, eee01–03	Ouster OS1-16	UAV
Ours	indoor01, outdoor01	Hesai XT-32	Handheld

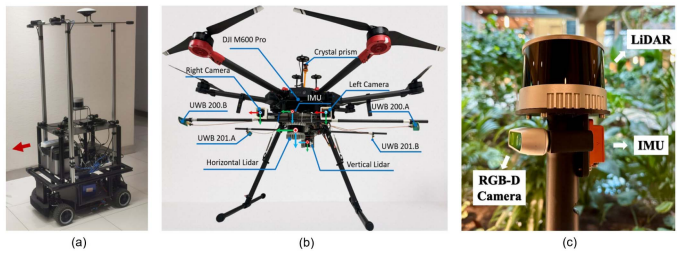

Fig. 3. Sensor platforms: (a) M2DGR [32], (b) NTU VIRAL [33], (c) Our handheld device.

We compare our method with SOTA algorithms, including point-cloud-based methods (FAST-LIO2 [8] and DLIO [17]) and voxel-map-based methods (sr-LIO [22], iG-LIO [26], PV-LIO[2] [11], VoxelMap++ [12], and C3P-VoxelMap [13]).

A. Experiments in Structured Indoor and Outdoor Environments

We first evaluate the proposed method on the M2DGR [32] and NTU VIRAL [33] datasets, which cover structured indoor and outdoor scenes. Absolute Trajectory Error (ATE) [34], reported as the root mean square error (RMSE), is used to quantify localization accuracy. The results are summarized in Table I.

Overall, the results validate the effectiveness of the proposed method. We denote the full version of our approach with semantic-driven hybrid voxel modeling as *Ours (full)*, and an ablation variant that models all voxels using Gaussian distributions representation as *Ours (GD)*.

[2] We use the third-party implementation from https://github.com/HViktorTsoi/PV-LIO, as the official VoxelMap code lacks IMU integration.

In structured indoor environments, both of our variants perform comparably to the strongest baseline, C3P-VoxelMap, showing minimal differences in accuracy. In structured outdoor environments, however, our method achieves notable improvements: the full method achieves a 7.1% reduction in RMSE relative to iG-LIO, and also achieves a 7.4% reduction relative to the GD variant. These results indicate that the proposed hybrid voxel representation effectively enhances localization accuracy in outdoor scenarios, while maintaining competitive performance indoors.

Next, we present detailed evaluations on each dataset to further analyze the performance across different environments.

1) M2DGR Dataset: The M2DGR dataset [32] was collected using a UGV platform equipped with a 32-line LiDAR and an IMU running at 150 Hz, as shown in Fig. 3(a). The evaluation includes three indoor sequences (room01–03), three outdoor campus-area sequences (gate01–03), and two street-scene sequences (street05–06). Ground-truth trajectories are provided by a laser tracker system for indoor scenes and an RTK/INS system for outdoor scenarios.

In the indoor sequences, our method achieves the best accuracy on two of the three sequences. However, *Ours (full)* performs comparably or slightly worse than *Ours (GD)*. This is mainly due to the limited generalization of the semantic segmentation model trained on S3DIS [30], as its scanning pattern and point density differ from M2DGR [32], leading to inaccurate semantic labels and suboptimal voxel classification. As illustrated in Fig. 2(a), only parts of the ground surface are correctly identified as planar in these indoor scenes.

In outdoor sequences, the performance differences between voxel-map-based and point-cloud-based methods are generally less pronounced. This can be attributed to the higher prevalence of nonplanar structures in these environments, which limits the effectiveness of voxel maps that rely solely on planar approximations. Notably, VoxelMap++ shows considerable variance across different sequences, suggesting that its plane-based voxelization strategy may lack robustness under such conditions. sr-LIO and iG-LIO achieve better performance on *street* sequences, which are weakly constrained, long straight-road environments, due to their targeted designs: iG-LIO stabilizes attitude by fixing the gravity direction, while SR-LIO suppresses drift through sweep reconstruction and segment-level undistortion. In the same scenarios, our ablation results show that *Ours (full)* achieves clear gains over *Ours (GD)*, confirming the effectiveness of the semantic-driven hybrid strategy in fusing planar and nonplanar cues.

2) NTU VIRAL Dataset: The NTU VIRAL dataset [33] was collected using a UAV platform equipped with a horizontally mounted Ouster-16 LiDAR and a 385 Hz IMU. The selected evaluation sequences include three outdoor campus sequences (eee01–03) and three indoor auditorium sequences (nya01–03). The ground truth was obtained using a 3D laser tracker.

Among the six tested sequences, *Ours (full)* achieves the best accuracy on three, demonstrating the adaptability of the proposed method to UAV-based platforms and low-resolution LiDAR sensors. However, on the nya01 and eee03 sequences, a noticeable degradation in performance is observed compared to the best-performing SOTA baselines. These two sequences exhibit higher average angular velocities (above 0.15 rd/s), which likely lead to increased accumulated errors in voxel statistics due to rapid motion.

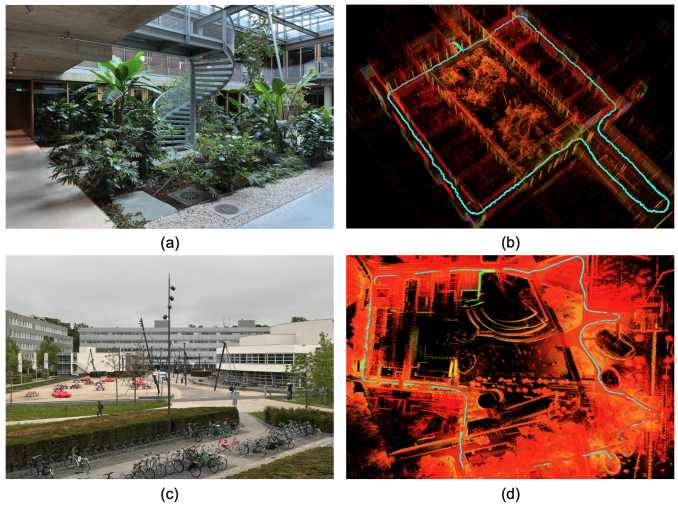

Fig. 4. Real and reconstructed views of representative environments. (a), (b) Indoor scene from a university building. (c), (d) Outdoor scene from a campus environment.

In five of the six sequences, *Ours (full)* consistently outperforms *Ours (GD)*, although the margin of improvement is relatively small. This modest gain is mainly attributed to the limited number of ground points captured by the UAV platform, which reduces the opportunity for the hybrid voxel strategy to impose additional constraints—particularly along the vertical axis, where LiDAR data tends to be more uncertain.

In terms of overall performance, voxel-map-based methods achieve comparable accuracy to point-cloud-based methods in this dataset. However, methods such as PV-LIO and VoxelMap++ exhibit large variations across sequences, reflecting their sensitivity to LiDAR resolution and environmental structure due to their reliance on plane fitting. While C3P-VoxelMap improves robustness through cross-voxel denoising, it still depends on a single type of geometric representation. In contrast, our proposed hybrid modeling approach enhances adaptability by combining both planar and probabilistic representations, offering improved consistency across varying scenarios.

B. Experiments in Challenging Scenarios

To further validate the adaptability of the proposed algorithm in complex environments and its applicability to handheld platforms, we collected a new dataset using a custom handheld device equipped with a 32-line LiDAR and a 400 Hz IMU. The data were recorded in representative indoor and outdoor campus environments. Fig. 4 shows real-world images alongside point cloud reconstructions generated by our method. The indoor scene in Fig. 4(a) features narrow hallways, large glass surfaces, and localized vegetation, while the outdoor scene in Fig. 4(c) includes dense vegetation, glass facades, and open areas, with a trajectory exceeding 1 km. These elements pose considerable

TABLE III
END-TO-END ERRORS IN OUR DATASETS (METERS)

Method	indoor01				outdoor01			
	x	y	z	total	x	y	z	total
FAST-LIO2	0.30	9.32	1.69	9.48	/	/	/	/
DLIO	0.23	0.68	1.07	1.29	15.69	21.45	21.59	34.24
PV-LIO	0.02	0.01	0.03	**0.04**	18.81	12.92	2.75	22.99
C3P-VoxelMap	10.61	13.64	3.21	17.58	27.54	29.91	3.33	40.79
Ours (GD)	0.02	0.03	0.04	0.06	11.41	18.75	3.72	22.26
Ours (full)	0.03	0.01	0.04	0.06	11.03	17.82	0.34	**20.96**
Distance (m)	189				1185			

Note: / indicates failure due to severe drift or loss of tracking.

challenges for localization: vegetation often violates the static scene assumption, narrow or open spaces reduce structural constraints, and glass surfaces yield weak or noisy LiDAR returns, or have multi-path effects (when the glass acts as a mirror). Robust localization and mapping under such conditions is therefore highly demanding.

All sequences start and end at the same location, allowing the end-to-end translational error to serve as a quantitative accuracy metric. Table III presents axis-wise and overall end-to-end translational errors for each evaluated sequence. VoxelMap++ is excluded from this table as it failed to produce valid results on all sequences. In the *indoor01* sequence, *Ours (full)* performs comparably to the best existing methods. Consistent with our earlier observations on the M2DGR [32] indoor sequences, the limited generalization of the semantic segmentation model results in no significant improvement of *Ours (full)* over *Ours (GD)*.

In contrast, in the more challenging *outdoor01* sequence, *Ours (full)* substantially outperforms all baselines in overall accuracy, particularly along the Z-axis. This improvement arises from our hybrid voxel modeling strategy, which effectively leverages semantic information to distinguish planar structures such as the ground. Vertical accuracy in this sequence is limited by LiDAR characteristics such as low vertical angular resolution and large incident angles of ground returns, which increase Z-axis uncertainty and cumulative drift. Our method improves vertical stability through semantic-aware planar constraints. Although noticeable errors remain in the X and Y directions, these are primarily attributed to large open areas and the presence of extensive glass facades, which offer sparse or invalid point returns, limiting localization reliability in horizontal directions.

Although C3P-VoxelMap performs well on public datasets, its accuracy drops significantly on both sequences. A potential reason is the presence of numerous nonplanar structures and glass surfaces, which introduce substantial noise. Performing planar fitting on the voxels containing such structures may yield erroneous parameters, thereby severely degrading localization performance.

Fig. 5 illustrates a representative scene from the *outdoor01* sequence, showing a glass building with clear holes and spurious reflections in the point cloud, caused by specular reflection on the glass surface. Nevertheless, our method succeeds in reconstructing structurally coherent and high-quality maps, further demonstrating its robustness in adverse sensing conditions.

Fig. 5. Image (top) and corresponding point cloud (bottom) near a reflective glass facade of the building in the *outdoor01* sequence.

TABLE IV
AVERAGE TOTAL MEMORY USAGE (MB)

Seq.	FAST-LIO2	DLIO	sr-LIO	iG-LIO	PV-LIO	VoxelMap++	C3P-VoxelMap	Ours (GD)	Ours (full*)	RandLA (RAM)	RandLA (GPU)
room	192	65	82	76	250	204	102	110	110	4634	1962
nya	201	48	71	112	810	201	154	199	199	4606	1962
eee	264	100	94	195	3409	678	428	379	401	4592	1645
gate	371	268	221	272	4473	939	974	323	500	4592	1759
street	447	365	302	500	8064	/	1691	831	820	4618	1738

*: with RandLA-Net running in parallel.

C. Resource Consumption

All experiments are conducted on a laptop with an Intel i5-11400H @ 2.7 GHz CPU, 32 GB RAM, and an NVIDIA RTX 3050Ti GPU. All methods use their default parameters; parameters for our approach are provided on the project page. A unified downsampling setting is applied for all methods to ensure fair comparison.

The average per-frame computation time across all benchmark sequences is summarized in Table I. Compared with other voxel-map-based approaches, our method incurs additional computational cost due to the more sophisticated hybrid modeling strategy. Nevertheless, it maintains real-time performance (< 100 ms) across all sequences. Our implementation runs RandLA-Net in a parallelized scheme, so there is no notable difference in resource consumption between *Ours (GD)* (without RandLA-Net) and *Ours (full)* (with RandLA-Net). RandLA-Net runtime is < 400 ms per scan, and it is run for every fifth scan.

Regarding memory usage shown in Table IV, the reported values represent the average memory consumption across all sequences of each scenario. We further include the RAM and GPU memory of RandLA-Net, which is required by our SLAM framework. Our method achieves significantly lower memory consumption than plane-fitting voxel-map-based baselines (in red), reaching levels comparable to point-cloud-based methods, as it avoids storing intermediate plane-fitting parameters for nonplanar voxels and supports larger voxel sizes.

D. Limitations

Our method balances accuracy and efficiency through a fixed voxel size, which ensures real-time performance but may overlook fine-scale planar structures in multi-scale environments. The framework further depends on semantic segmentation to distinguish planar from non-planar voxels; although mapping multiple semantic categories into two super-classes reduces sensitivity to fine-grained errors, misclassifications can still affect accuracy. Future work will address the limited generalization of the segmentation model and its computational overhead.

V. Conclusion

This letter addresses the challenge of high-fidelity environment representation for LIO, particularly its generalization across diverse scenes. We propose a semantic-driven hybrid voxel representation that integrates geometric and probabilistic modeling through dual residuals, enabling accurate and robust scan matching. The scan matcher is embedded in an IEKF framework, achieving real-time performance in both indoor and outdoor environments.

References

[1] A. Singandhupe and H. M. La, "A review of SLAM techniques and security in autonomous driving," in *Proc. 3rd IEEE Int. Conf. Robot. Comput.*, 2019, pp. 602–607.

[2] V. V. Lehtola et al., "Digital twin of a city: Review of technology serving city needs," *Int. J. Appl. Earth Obs. Geoinf.*, vol. 114, 2022, Art. no. 102915.

[3] D. Lee, M. Jung, W. Yang, and A. Kim, "LiDAR odometry survey: Recent advancements and remaining challenges," *Intell. Serv. Robot.*, vol. 17, no. 2, pp. 95–118, 2024.

[4] S. Karam, V. Lehtola, and G. Vosselman, "Simple loop closing for continuous 6DOF LiDAR&IMU graph SLAM with planar features for indoor environments," *ISPRS J. Photogramm. Remote Sens.*, vol. 181, pp. 413–426, 2021.

[5] L. Zhou, G. Huang, Y. Mao, J. Yu, S. Wang, and M. Kaess, "$\mathcal{PLC}$-LiSLAM: LiDAR SLAM with planes, lines, and cylinders," *IEEE Robot. Automat. Lett.*, vol. 7, no. 3, pp. 7163–7170, Jul. 2022.

[6] J. Zhang and S. Singh, "LOAM: LiDAR odometry and mapping in real-time," *Robot., Sci. Syst.*, vol. 2, no. 9, pp. 1–9, 2014.

[7] 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.

[8] 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.

[9] J. Behley and C. Stachniss, "Efficient surfel-based SLAM using 3D laser range data in urban environments," *Robot.: Sci. Syst.*, vol. 2018, 2018, Art. no. 59.

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

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

[12] C. Wu et al., "Voxelmap++: Mergeable voxel mapping method for online LiDAR (-inertial) odometry," *IEEE Robot. Automat. Lett.*, vol. 9, no. 1, pp. 427–434, Jan. 2024.

[13] X. Yang et al., "C^3 P-VoxelMap: Compact, cumulative and coalescible probabilistic voxel mapping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2024, pp. 7908–7915.

[14] J. Xie, Z. Wu, B. Wang, A. Xu, Y. Chen, and J. Li, "An unmanned aerial vehicle light detection and ranging simultaneous localisation and mapping algorithm based on factor graph optimisation for tunnel 3D mapping," *IET Radar Sonar Navig.*, vol. 18, no. 6, pp. 939–952, 2024.

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

[16] P. Biber and W. Straßer, "The normal distributions transform: A new approach to laser scan matching," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2003, vol. 3, pp. 2743–2748.

[17] 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.

[18] C. Zhao et al., "Adaptive-LIO: Enhancing robustness and precision through environmental adaptation in LiDAR inertial odometry," *IEEE Internet Things J.*, vol. 12, no. 9, pp. 12123–12136, May 2025.

[19] T. Zhang, X. Zhang, Z. Liao, X. Xia, and Y. Li, "AS-LIO: Spatial overlap guided adaptive sliding window LiDAR-inertial odometry for aggressive FOV variation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2024, pp. 10829–10836.

[20] J. Huang et al., "LA-LIO: Robust localizability-aware LiDAR-inertial odometry for challenging scenes," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2024, pp. 10145–10152.

[21] Z. Yuan, F. Lang, T. Xu, R. Ming, C. Zhao, and X. Yang, "Semi-elastic LiDAR-inertial odometry," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2025, pp. 9855–9861.

[22] Z. Yuan, F. Lang, T. Xu, and X. Yang, "SR-LIO: LiDAR-inertial odometry with sweep reconstruction," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2024, pp. 7862–7869.

[23] M. Yokozuka, K. Koide, S. Oishi, and A. Banno, "LiTAMIN: LiDAR-based tracking and mapping by stabilized ICP for geometry approximation with normal distributions," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 5143–5150.

[24] 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. 11619–11625.

[25] X. Ji, S. Yuan, P. Yin, and L. Xie, "LIO-GVM: An accurate, tightly-coupled LiDAR-inertial odometry with Gaussian voxel map," *IEEE Robot. Automat. Lett.*, vol. 9, no. 3, pp. 2200–2207, Mar. 2024.

[26] Z. Chen, Y. Xu, S. Yuan, and L. Xie, "iG-LIO: An incremental GICP-based tightly-coupled LiDAR-inertial odometry," *IEEE Robot. Automat. Lett.*, vol. 9, no. 2, pp. 1883–1890, Feb. 2024.

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

[28] Q. Hu et al., "RandLA-Net: Efficient semantic segmentation of large-scale point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11105–11114.

[29] P. J. Huber, "Robust estimation of a location parameter," *Ann. Math. Statist.*, vol. 35, no. 1, pp. 73–101, 1964.

[30] I. Armeni et al., "3D semantic parsing of large-scale indoor spaces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1534–1543.

[31] J. Behley et al., "SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 9296–9306.

[32] J. Yin, A. Li, T. Li, W. Yu, and D. Zou, "M2DGR: A multi-sensor and multi-scenario SLAM dataset for ground robots," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 2266–2273, Apr. 2022.

[33] T. -M. Nguyen, S. Yuan, M. Cao, Y. Lyu, T. H. Nguyen, and L. Xie, "NTU VIRAL: A visual-inertial-ranging-LiDAR dataset, from an aerial vehicle viewpoint," *Int. J. Robot. Res.*, vol. 41, no. 3, pp. 270–280, 2022.

[34] Z. Zhang and D. Scaramuzza, "A tutorial on quantitative trajectory evaluation for visual (-inertial) odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 7244–7251.